

ES1.5
L-14

U.M. IBRAGIMOV

DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALAR

DARSLIK



O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA
INNOVATSIYALAR VAZIRLIGI
BUXORO MUHANDISLIK-TEXNOLOGIYA INSTITUTI

U.M.Ibragimov

DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALAR

DARSLIK

BUXORO – 2024
«DURDONA» NASHRIYOTI

K-
ITI

19
7

32.973-018ya7
004.43+681.5(075)
I - 14

U.M.Ibragimov

Dasturlash tillari bazasida texnik ilovalar [Matn] : darslik /
U.M.Ibragimov – Buxoro: «Sadriddin Salim Buxoriy» Durdona nashriyoti, -
2023. – 452 b.

UO'K 004.43+681.5(075)
KBK 32.973-018ya7

TAQRIZCHILAR:

X.F. Djuraev

– Buxoro muhandislik-texnologiya instituti
"Texnologik jarayonlarni boshqarishning
axborot-komunikatsiya tizimlari" kafedrası
professori, t.f.d

Shafiyev T.R.

– Buxoro davlat universiteti "Axborot tizimlari
va raqamli texnologiyalar" kafedrası mudiri,
t.f.f.d. (PhD), dotsent

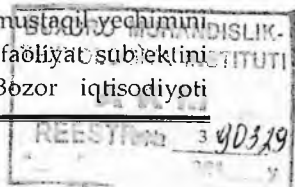
*O'zbekiston Respublikasi Oliy ta'lim, fan va innovatsiyalar vazirligi
2023 - yil 22 - dekabr "537" - sonli buyrug'iga asosan darslik sifatida tavsiya
etilgan. Ro'yxatga olish raqami 996-613*

KIRISH

O'zbekiston Respublikasi Prezidenti SH.M.Mirziyoyevning 2022 yil 28 yanvardagi "2022–2026 yillarga mo'ljallangan Yangi O'zbekistonning taraqqiyot strategiyasi to'g'risida"gi PF-60-son Farmoniga muvofiq, shuningdek, axborot- kommunikatsiya texnologiyalari sohasini yangi bosqichga olib chiqish bo'yicha ustuvor vazifalarni amalga oshirish maqsadida, 2022 yil 22 avgustdagi Prezidentimizning "2022–2023 yillarda axborot-kommunikatsiya texnologiyalari sohasini yangi bosqichga olib chiqish chora-tadbirlari to'g'risida"gi qarorining qabul qilinishi axborot-kommunikatsiya texnologiyalari sohasini yanada takomillashtirish, raqamli texnologiyalar sohasida masofaviy ta'lim shaklida kadrlar tayyorlash faoliyatini rivojlantirish orqali yiliga 6,5 ming nafardan ortiq yoshlarning axborot texnologiyalari yo'nalishida ta'lim olish imkoniyatlari yanada kengaydi.

Mazkur o'quv qo'llanma 60711900-Texnologik jarayonlarni boshqarishning axborot-kommunikatsiya tizimlari yo'nalishlari talabalari uchun mo'ljallangan bo'lib, bo'lajak mutaxassislarning dasturiy ilovalarni yaratish bo'yicha umumkasbiy tayyorgarligini oshirish maqsadida yaratilgan navbatdagi manbadir.

Mamlakatimiz mustaqillikka erishgandan so'ng fan-texnika va texnologiyalarning rivojlantirish, shuningdek ta'lim tizimini isloh qilish bo'yicha katta ahamiyatga ega bo'lgan molik ishlar amalga oshirilmoqda. Hukumatimiz tomonidan qabul qilingan "Ta'lim to'g'risidagi qonun" va "Kadrlar tayyorlash milliy dasturi" ning qabul qilinishi buning yorqin dalilidir. Hozirgi kunda ta'lim tizimida innovatsion ta'lim texnologiyalarini joriy etish, iqtisodiyotning turli sohalari uchun jahon ta'lim standartlariga javob beradigan kompetentli mutaxassislarni tayyorlash, shuningdek mavjud muammolarni mustaqil yechimini aniqlovchi va tegishli qarorlar qabul qiluvchi faoliyat sub'ektini tarbiyalash muhim ahamiyat kasb etadi. Bozor iqtisodiyoti



sharoitida faqat mustaqil fikrlash qobiliyatiga ega bo'lgan shaxsgina o'z muammolarini o'zi mustaqil hal qila oladi va jamiyatda o'z mavqeiga ega bo'ladi. Bunday qobiliyat esa ta'lim muassasalarida shakllantiriladi. Respublikamizda ishlab chiqarish korxonalarini modernizatsiyalash va zamonaviy texnologiyalar asosida jihozlash orqali xalqimizni yuqori sifatli, maxsulotlar va tovarlar bilan ta'minlashga erishish mumkin. Ushbu vazifalar texnologik jarayonlarni avtomatlashtirish va ishlab chiqarishni boshqarish sohalarida faoliyat olib borayotgan mutaxassislar oldida turgan muhim vazifalardan biridir. Albatta, bu vazifalarni bajarish uchun turli tuman texnologik jarayonlarni avtomatlashtirish, ayniqsa bugungi kunda informatsion-kommunikatsion texnologiyalarni qo'llash, avtomatlashtirilgan kichik va katta texnologik komplekslarni, kompyuter bilan boshqariluvchi jarayonlar faoliyat ko'rsatadigan korxonalarni yaratish kerak bo'ladi. Mashinasozlik, neft va gaz, kimyo sanoati va boshqa sohalarda texnologik jarayonlarni zamonaviy mikroprotessorli texnologiyalar va kompyuter dasturlari yordamida boshqarish hamda avtomatlashtirish komplekslari tadbiriq etilmoqda. Hozirgi zamon texnika taraqqiyoti darajasi, yangi takomillashtirilgan yuqori unumdorli avtomatlashtirilgan va yuqori aniqlikka ega bo'lgan kompyuter dasturlarini yaratishda fanning eng yangi yutuqlaridan foydalanishga asoslanadi. Bu borada Respublikamizning mavjud korxonalari uchun malakali mutaxassislarni tayyorlash ta'lim tizimida belgilangan ustuvor vazifalardan biridir. Shuning uchun, Respublikamizning ta'lim muassasalarida mavjud ta'lim yo'nalishlari va mutaxassisliklarning o'quv rejalarini, fan dasturlarini zamonaviy innovatsiyalar va xorij tajribalarini tahlil qilgan holda ishlab chiqish va ularni takomillashtirish yo'lga qo'yildi.

Mamlakatimizda ta'lim - tarbiya tizimini tubdan yangilash, uni zamon talablari darajasiga ko'tarish, vatanimizning kelajagi uchun jismonan va ruhan sog'lom avlodni tarbiyalash ishlari

davlat siyosatining asosiy yo'nalishiga aylandi. Bunda pedagoglarning vazifasi talabalarga doim yuqori bilim, ko'nikma, mahoratga, ilg'or axborot va pedagogik texnologiyalar asosida mashg'ulotlarni o'tish kerakligi davr talabidir. Shuni ta'kidlash joizki hozirgi kunda texnologik jarayonlarni boshqarishning axborot-kommunikatsiya tizimlari bakalavriat yo'nalashlarida texnologik jarayonlarni axborot-kommunikatsiya tizimlari orqali boshqarishni ta'minlovchi mutaxassislarni tayyorlashga xizmat qiladi.

“Dasturlash tillari bazasida texnik ilovalar” darsligini yozishdan maqsad talabalarda texnologik jarayonlarni avtomatlashtirish va boshqarishda dasturiy ta'minotning rolini anglab yetishi, dasturiy ta'minotning turlari, imkoniyatlari va uni yaratish bosqichlarini bilishi, texnologik jarayonlarni boshqarish va avtomatlashtirishda dasturiy ta'minotni qo'llash imkoniyatlari boshqarishda dasturiy ta'minotni rolini anglab yetishdan iborat.

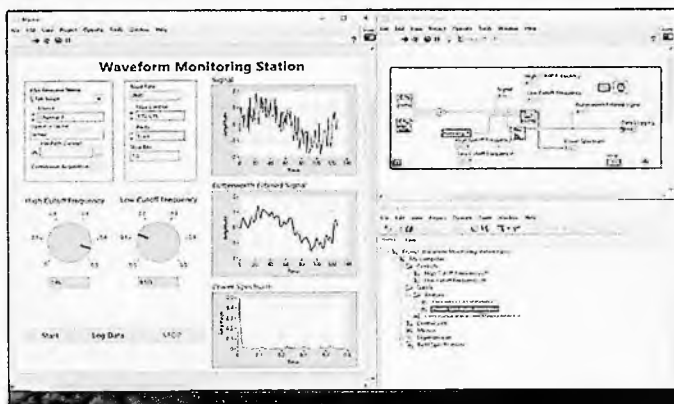
I MODUL. "DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALAR"

FANIGA KIRISH

§1.1. "Dasturlash tillari bazasida texnik ilovalar" fanining maqsadi va vazifalari. Labview instrumental vositasi va uning imkoniyatlari

LabVIEW – virtual laboratoriya uskunalari yaratuvchi dasturlash muhiti hisoblanib, siz ushbu muhit yordamida ilovalar ishlab chiqishingiz mumkin. Algoritmning barcha elementlarini grafik ko'rinishida tasvirlash oddiy dasturlash tillaridan farq qiladi. Chunki C, C++ yoki Java tillarida dastur matn ko'rinishida yaratiladi. Ushbu yaratuvchi muhit va ilovalardan foydalanish tadqiqotchilar uchun mo'ljallangan, ya'ni olimlar va muhandislar uchun dasturlash ular ishining ma'lum qismi hisoblanadi. Lekin bu nafaqat grafik dasturlash tiligina bo'lib qolmasdan undanda yuqori ahamiyatga ega.

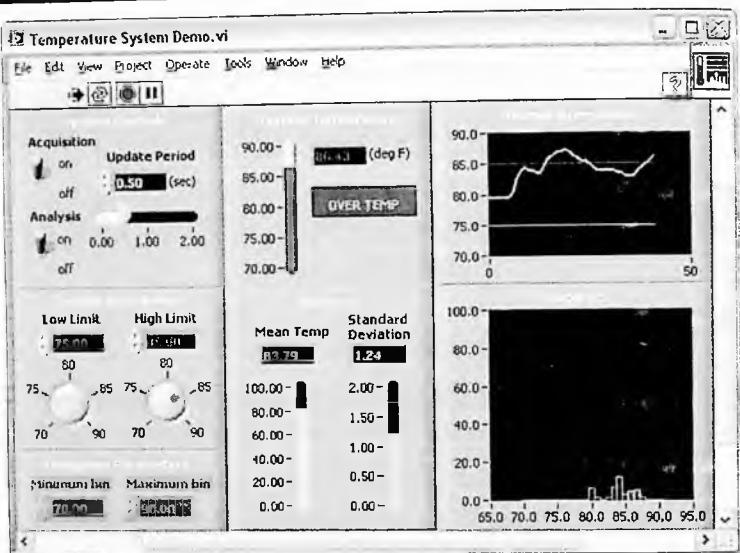
Kuchli grafik dasturlash tili ish unumdorligini yuzlab marotaba oshirish imkonini beradi. Oddiy dasturlash tillari yordamida tugallangan ilovani yaratish hafta yoki oylarni talab qilsa, Labview ni qo'llashda bir necha soat yetarli bo'ladi. Chunki bu paket natijalarni rasmiylashtirish, ma'lumotlarni tahlil qilish, turli o'lchashlarni dasturlash uchun maxsus yaratilgan. Chunki Labview oddiy grafik interfeysga ega va dasturlash uchun ham oddiy, jarayonlarni modellashtirishga g'oyalar taqdimotiga umumiy xarakterdagi ilovalarni tayyorlashga mos keladi va zamonaviy dasturlash asoslarini o'rganish uchun qulay hisoblanadi.



1.1.1-rasm. LabVIEW tizimida texnik ilovaning umumiy ko'rinishi

LabVIEW da yaratilgan o'lchash tizimi standart laboratoriya uskunalariga nisbatan katta ustunlikka ega. Chunki u zamonaviy dasturiy ta'minotning ko'p qirrali imkoniyatlaridan foydalanadi. Qo'yilgan masalani bajarish uchun LabVIEW boshqariladigan virtual uskunani tashkil qiladi. LabVIEW yordamida mavjud yondashuvga nisbatan kam xarajatlar hisobiga zarur virtual uskuna rusumini yaratish mumkin bo'ladi. Zaruriyat bo'lganda siz bir daqiqada tegishli o'zgartirishni kiritishingiz mumkin.

LabVIEW sizning masalangizni yechib beradigan dasturlash ishini osonlashtirish uchun yaratilgan. LabVIEW da maxsus virtual uskunalar kutubxonasi mavjud bo'lib, o'rnatiladigan apparatlar qiymatini kiritish yoki chiqarish (*data acquisition - DAQ*), umumiy foydalanish kanali bilan ishlash (*KOP, General Purposes Interface Bus - GPIB*), ketma-ket ulangan port yordamida qurilmani boshqarish, tahlil etish uchun dasturiy komponentlar, ma'lumotlarni taqdim qilish va saqlash, internet orqali bog'lanish imkonini beradi. Tahlil kutubxonasi signalni to'plash, qayta ishlash, turli xil filtrlar, statik ishlov berish, regressiv tahlil, chiziqli algebra va arifmetika massivlarini o'z ichiga oladi.



1.1.2-rasm. LabVIEW tizimida texnik ilovaning
foydalanuvchi interfeysi ko'rinishi

©LabVIEW dasturiy virtual uskuna deb ataladi. (*VU, virtual instrument*

– *VI*). Ammo u mavjud C yoki Basic dasturlash tillariga ham o'xshab ketadi.

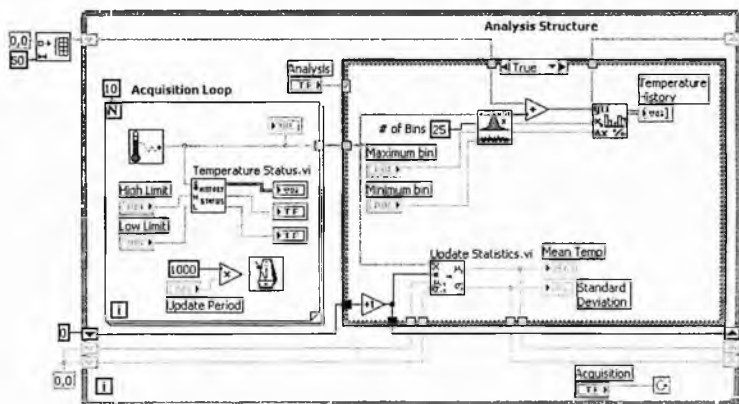
©Virtual uskuna uchta asosiy qismdan tashkil topgan:

VU ning yuzadagi paneli

Yuzadagi panel (*front panel*) – virtual uskunaning interfaol foydalanadigan interfeysi. Unda boshqarish dastagi, knopkalar, grafik indikatorlar va boshqarishning boshqa elementlari joylashgan bo'lib, bular foydalanuvchi tomonidan ma'lumotlarni kirituvchi moslama vazifasini bajaradi. Foydalanuvchi sichqoncha va klaviaturadan foydalanib ma'lumotlarni kiritadi hamda monitor ekranida dastur ishi natijalarini ko'radi:

VU ning diagrammalar bloki

Diagrammalar bloki (*block diagram*)-LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi. Diagramma-bloki real foydalanish mumkin bo'lgan ilova bo'lib hisoblanadi. Diagramma-bloki komponentlari ancha past darajali virtual uskunalar bo'lib hisoblanib, *LabVIEW* funktsiyasi joylashtirilgan dastur bajarilishini boshqaruvchi tuzilmalar va konstantalardan iborat. Yuzadagi panel ob'ektlari diagramma-blokida mos terminallar (*wires*) ko'rinishida taqdim qilingan va ma'lumotlar foydalanuvchidan dasturga yoki aksincha berilishi mumkin.



1.1.3-rasm. LabVIEW tizimida texnik ilovaning funksional qismi(diagramma-bloki) ko'rinishi.

Dasturlash kodidan diagramma-blokida dastur osti sifatida foydalanish uchun uning tasviri va ulanuvchi panelni aniqlash zarur bo'ladi, boshqa dasturlash kodida qo'llanilgan virtual uskuna dasturosti virtual uskuna deyiladi. (*VIL*, *Suble*) ikonka (*icon*) dasturlash kodining grafik taqdimoti hisoblanib, dasturlash kodida ob'ekt sifatida foydalanish mumkin. Ulanuvchi panel boshqa diagramma-blokidan ma'lumotlarni uzatuvchi mexanizm bo'lib hisoblanadi. Ulanuvchi panel dasturosti argumenti va

parametrlariga o'xshab virtual uskunaning kirish va chiqish ma'lumotlarini aniqlaydi.

Virtual priborlar ierarxik va modulli (modular) bo'ladi. Siz ulardan mustaqil ilova yoki virtual uskuna sifatida foydalanishingiz mumkin. Ushbu mantiqqa asosan *LabVIEW* modulli dasturlash kontseptsiyasi bo'lib hisoblanadi. Buning uchun siz oddiy masalalar qatoridan muhim amaliy masalani ajratasiz. Keyin har bir masalani yechish uchun virtual uskuna yaratasiz va dasturlash kodlarini uskunaning ancha yuqori darajali blok diagrammasida birlashtirib amaliy masalani butunligicha yechasiz. Modulli dasturlash texnologiyasi juda qulay bo'lib, bunda har bir virtual uskunaosti masalasi bilan alohida ishlash mumkin bo'ladi. Bundan tashqari past darajali virtual dastur osti bir necha ilovalarga xos masalalarni yechish imkonini beradi. Quyidagi jadvalda (1.1.1-jadval) *LabVIEW* ning asosiy terminlari va umumiy qabul qilingan ekvivalentlari an'anaviy dasturlash tillari uchun keltirilgan.

1.1.1-jadval. *LabVIEW* terminlari va ularni ekvivaletlari an'anaviy dasturlash tillari uchun.

LabVIEW	An'anaviy dasturlash tillari
Virtual uskuna(VU)	Dastur
Funktsiya	Funktsiya yoki metod
Virtual ichki uskuna(VIU)	Quyidastur yoki ob'ekt
Yuzadagi panel	Foydalanuvchi intrfeysi
Diagramma-bloki	Dastur kodi
G C,C++,Java,Basic va boshqalar	

Asosiy tayanch so'z va iboralar

Modul, dastur, texnika, tizim, virtual, uskuna, diagramma, blok, funksiya, kod, tillar, matn, muhit, ilova, grafik.

Takrorlash va mustaqil ishlash uchun savollar

1. Dasturlash tillar bazasida texnik ilovalar fani nima uchun kerak?
2. Texnik ilovalar yaratishda qanday dasturlash tizimlar mavjud?
3. LabVIEW nima?
4. Virtual uskuna nima?
5. Virtual uskunada yuzadagi paneli nima?
6. Virtual uskunada diagramma bloki nima?

§1.2. LabVIEW muhiti bilan tanishish

Oddiy til bilan aytganda yuzadagi panel (front panel) shunday oyna aytiladiki u orqali foydalanuvchi dastur orqali o'zaro harakatlanadi. Qachon siz virtual uskunani ishga tushursayiz yuzadagi panel ochiq bo'lishi kerak va unda siz bajarilayotgan dasturda ma'lumotlarni kiritish imkoni bo'ladi. Boshqa tomondan yuzadagi panel virtual uskuna bajarilishida natijalarni ko'rish mumkin.

Boshqarish elementlari va indikatorlar

Yuzadagi panel asosan boshqarish elementlari va indikatorlar to'plamidan iborat bo'ladi. Boshqarish elementlari ixtiyoriy o'lchash uskunasi mavjud bo'lgan odatiy boshqarish organlarini imitatsiyalaydi, masalan tugmalar va rejimlarga o'tkazgichlar. Boshqarish elementlari foydalanuvchiga ma'lumotlarni kiritish imkonini beradi. Ular virtual uskuna diagramma-blokiga ma'lumotlarni uzatishadi. Indikatorlar bajariladigan dastur natijalarini ya'ni chiqish ma'lumotlarini ko'rsatadi. Quyida keltirilgan o'xshatishlar yordamida virtual uskuna boshqarish elementlari va indikatorlarni ma'nosini tushunish osonroq bo'ladi.

Boshqarish elementi = foydalanuvchi kiritadigan ma'lumotlar =

ma'lumotlar manbaasi.

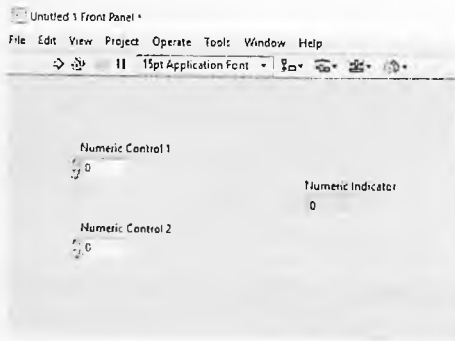
Indikatorlar = foydalanuvchiga chiqaradigan ma'lumotlar = ma'lumotlarni qabul qiluvchilar.

Bu tushunchalar to'liq o'zaro almashinmaydi, shuning uchun ularni farqini yaxshilab tushunib oling.

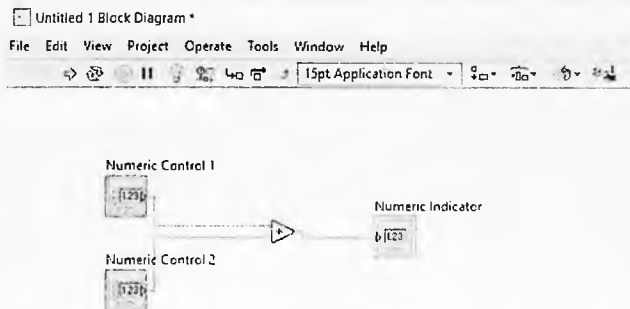
Siz boshqarish elementlari va indikatorlarni yuzadagi panelga olib o'tib joylashtirasiz. Bunda ularni birinchi boshqarish elementlari (*Controls*) panelining ichki palitralaridan tanlaysiz va kerakli joyga joylashtirasiz. Yuzadagi panelga ob'yekt paydo bo'lgandan keyin uning hajmi, formasi, holati, rangi va boshqa atributlarini oson o'zgartirish mumkin.

Diagrammalar-bloki

Diagramma-bloki oynasi *LabVIEW* virtual uskuna birlamchi grafik kodini saqlaydi. *LabVIEW* diagramma-bloki C yoki Basic singari odatiy dasturlash tillari matnli kodlari qatoriga o'xshaydi - bu xuddi real bajariladigan kod. Diagrammalar- blokini konstruktorlash aniq bir funktsiyalarni bajaruvchi ob'ektlarni o'zaro bog'lash orqali amalga oshiriladi. Bu paragrafda diagrammalar-bloki turli komponentlarini ko'rib o'tamiz: terminallar, bog'lamlar, o'tkazgichlar.



1.2.1-rasm. Ikkita sonni qo'shish amalini bajaruvchi ilova yuzadagi panelidagi boshqarish elementlari va indikatorlarini joylashuvi



1.2.2-rasm. Add.vi diagramma-bloki ma'lumotlar terminali, bog'lamlar va o'tkazgichlar (funktional bajariluvchi kod) ni saqlaydi

Siz boshqarish elementi yoki indikatorni yuzadagi panelga joylashtirsangiz *LabVIEW* diagrammalar-blokida avtomatik mos terminalni yaratadi. Qachonki siz diagrammalar-blokidan terminalni o'chirsangiz, *LabVIEW* uni yuzadagi paneldan avtomatik o'chiradi va aksincha: yuzadagi paneldan elementni o'chirganda uning terminali ham o'chiriladi. Boshqarish elementi terminali ramkasi yo'g'on chiziq bilan ajratilgan, indikatorlar terminali chegarasi esa mayin chiziq bilan belgilangan (1.2.3-rasm). Bu ikki tipdagi terminallarni adashtirmaslik muhim, chunki ular turlicha funktsiyalanadi (boshqarish elementi - kiritish, indikator-chiqarish, shuning uchun ularni o'zaro almashtirib bo'lmaydi).

Siz terminallarni diagrammalar-bloki kiritish va chiqarish porti sifatida yoki manbaa va qabul qiluvchi ma'lumotlar sifatida qarashingiz mumkin. *Numeric Control1* (1.2.3-rasm) boshqarish elementiga kiritiladigan ma'lumotlar yuzadagi paneldan chiqadi va *Numeric Control1* terminali orqali diagrammalar-blokiga keladi. So'ngra *Numeric Control1* terminalidagi ma'lumotlar o'tkazgich orqali *Qo'shish(Add)* funktsiyasi kirishiga keladi. Xuddi shunday ma'lumotlar summatorga ikkinchi boshqarish elementi terminalidan ham keladi. *Qo'shish* funktsiyasi hisoblashni bajargandan so'ng u o'zining chiqishida yangi

qiymatni yaratadi. Bu qiymat *Sonli indicator (Numeric)* terminaliga keladi va yuzadagi panelga chiqariladi, qaysiku u yerda foydalanuvchi uni ko'rishni mumkin.

Ma'lumotlar bog'lami – bu dasturning ixtiyoriy bajariluvchi elementining umumiy nomi. Bog'lamlar an'anaviy dasturlashdagi operatorlar, funktsiyalar va ichkidasturlarga o'xshaydi. Qo'shish (*Add*) va Ayirish (*Subtract*) bir bog'lam turini ifodalaydi. Bog'lamning boshqa turi tuzilish(structure) hisoblanib, an'anaviy dasturlash tillaridagi shart operatorlari yoki tsikllar singari siklik va shart bo'yicha kodni bajarishi mumkin. *LabVIEW* maxsus bog'lamlar turiga ham ega masalan Formula (*Formula Node*) bog'lami murakkab matematik formula va ifodalarni hisoblash uchun mo'ljallangan. Event Structure (*Hodisa tuzilishi*) bog'lam turi haqida gapirsak, u yuzadagi panel va diagrammalar-blokidagi harakatga javob qaytarish uchun mo'ljallangan.

Ma'lumotlarni o'tkazgichlar

LabVIEW virtual uskunasini *ma'lumotlarni o'tkazgichlar (wired)*, bog'lanadigan bog'lamlar va terminallar hisobidan yagona uskuna sifatida akslanadi. O'tkazgichlar terminal-manbaadan bir yoki bir necha terminal-qabul qiluvchiga ma'lumotlarni o'tkazish kanallari sifatida qo'llaniladi.

Agar siz o'tkazgichga birdan ko'p manbaa yoki birorta ham manbaani ulamoqchi bo'lmasangiz, *LabVIEW* sizning harakatlaringizni maquullamaydi va o'tkazuvchi *buzilgan (broken)* bo'ladi. Faqat bir ma'lumotlar manbaasi va bir necha qabul qiluvchi bo'lishi mumkin.

Ma'lumotlar oqimini dasturlash – oqim bo'yicha harakat.

LabVIEW matnli dasturlash tili hisoblanmas ekan, uning kodi qatorma qator bajarilmaydi. *LabVIEW* dasturini bajarilishini boshqarish tamoyili *ma'lumotlar oqimi (dataflow)* hisoblanadi.

Oddiy aytganda bog'lam kodi kirish terminallarining barchasiga ma'lumotlar kelgandan keyin bajariladi; ishi

yakunlangandan so'ng bog'lam ma'lumotlarni chiqish terminallariga uzatadi va ma'lumotlar manbaadan darhol boshqa qabul qiluvchi terminallarga boradi. Ma'lumotlar oqimi tamoyili matnli dasturlash tillaridagi *oqimni boshqarish (control flow)* usulidan tubdan farqlanadi. Matnli dasturlash tillarida buyruqlar ular yozilgan ketma-ketlikda bajariladi. Bu farqqa ko'nikish kerak. Shunday qilib, agar an'anaviy oqimi boshqarish buyruqlar yordamida boshqarilsa, ma'lumotlar oqimini qayta ishlash ma'lumotlarning o'zi tomonidan boshqariladi, ya'ni ma'lumotlarga bog'liq (data dependent).

IVU(Ichki virtual uskuna), ikonka va bog'lash paneli.

IVU – bu VU(*virtual uskuna*) bo'lib, uni siz boshqa VU dan chaqirasiz. Bunday ixtiyoriy virtual uskuna ishlashi mumkin. Aytaylik massivning o'rtacha qiymatini hisoblash VU sini yaratdiz: *Main.vi*. Siz uni mustaqil ishga tushurishingiz mumkin (*uskunalar paneli tugmalari yordamida*), unda u VU bo'ladi. Xuddi shunday siz uni boshqa dasturning ixtiyoriy nuqtasida chaqirishingiz mumkin va u darhol IVU bo'ladi.

Agar sizning virtual uskunangiz ichki virtual uskuna sifatida ishlasa, unda uning boshqarish elementlari va indikatorlari ularni chaqirgan virtual uskunadan ma'lumotlarni qabul qiladi va qaytaradi. VU ikonkasi (*Icon*) boshqa VU diagramma- blokida ichki uskuna sifatida qaraladi. Ikonka tasvir sifatida yoki kichik VU izohi sifatida akslanadi(yoki ikkalasi sifatida ham akslanishi mumkin).

Virtual uskuna *Bog'lanish paneli (connector) C* yoki Pascal dasturlash tili funktsiyalari parametrlari ro'yxati singari ma'noga ega bo'lib, bog'lanish paneli terminallari ichki uskuna ma'lumotlarini kiritish/chiqarish parametrlari sifatida harakatlanadi. Har bir terminal yuzadagi paneldagi o'zining boshqarish elementi yoki indikatoriga mos keladi. Ichki uskunani chaqirish paytida uning kirish parametrlari ulangan boshqarish

elementlariga nusxalanadi va ichki dastur bajariladi. Ichki uskuna tugaganda indikatorlar ma'lumoti chiqish parametrlari terminallariga nusxalanadi.

Har bir VU odatda yuzadagi panel va diagramma-bloking o'ng yuqori burchagida ikonka sifatida akslanadi.

VU bog'lanish paneli ikonka tagida yashiringan. Unga murojaat qilish uchun yuzadagi panel yordamchi menyusi ikonkasidagi "*Показать соединительную панель*" (Show Connector) opsiyasini tanlash orqali amalga oshiriladi. Agar siz bog'lanish panelini birinchi marotaba chaqirsangiz, *LabVIEW* 12 dona terminalli model panelini taklif etadi (ma'lumotlarni kiritish va chiqarish uchun olti modeldan).

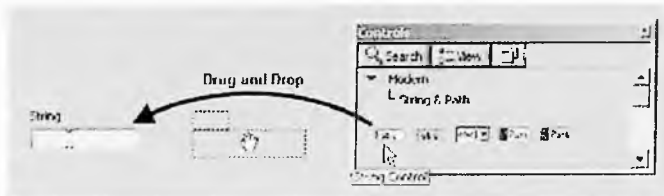
Takrorlash va mustaqil ishlash uchun savollar

1. Yuzadagi paneldan nima maqsadda foydalaniladi?
2. Boshqarish elementi va indikator orasidagi faqr nimada?
3. Diagrammalar blokidan nima maqsadda foydalaniladi?
4. Ma'lumotlar bog'lami nima?
5. Ma'lumotlarni o'tkazgichlari nima?

§1.3. Elementlarni bog'lash. Dasturni otladkalash metodikasi

Yuzadagi panelga ob'yektlarni joylatirish. Odatda dastur yaratish yuzadagi panelga boshqarish elementlari va indikatorlarni joylashtirishdan boshlanadi. Bu sizning kiruvchi ma'lumotlaringizni kiritish va VU dan chiqish ma'lumotlarini olish uchun kerak bo'ladi. Siz bu ishni amaliy yoki tajriba mashg'ulotlari topshiriqlarini bajarish mobaynida bajargan bo'lishingiz mumkin, shuning uchun biz ushbu metodikani yana bir marta eslatamiz. Agar kursorni *boshqarish elementlari* (*Controls*) palitrasiga etsangiz, 1.3.1-rasmda ko'rsatilganidek element nomini ko'rsatib turiladi. Agar qism palitra ustiga bossangiz unda uning tashkil etuvchi elementlari ko'rinadi.

Palitradan siz kerakli elementni tanlab, uni yuzadagi panelga joylashtirishingiz mumkin. 1.3.1-rasmga qarang.



1.3.1-rasm. Elementni palitradan olib, yuzadagi panelga joylashtirish

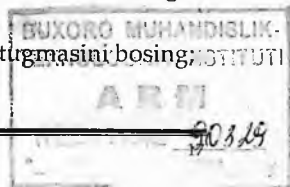
Endi yangi VU yarating va yuzadagi panelga sonli boshqarish elementini joylashtiramiz. Esda tuting yuzadagi panelga ob'yektni joylashtirish paytida, unga mos terminal diagrammalar-blokida paydo bo'ladi. Bir vaqtning o'zida yuzadagi panel va diagrammalar-bloki oynasini ko'rish uchun Oyna(Window) menyusidan Chap va O'ngga joylashtirish (*Tile Left and Right*) bandini tanlashingiz mumkin.

Boshqarish elementlari(*Controls*) palitrasini yuzadagi panelning ixtiyoriy bo'sh qismiga kontekst menyuni chaqirish (sichqonchanning o'ng tugmasini tanlab) orqali ko'rish mumkin.

Obyektlarni belgilash

Belgi (*Label*) – yuzadagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloki. U ob'yektni yaratayotganda standart qoidasi bo'yicha nomi bilan yuzadagi panel oynasida hosil bo'ladi (masalan: "*Numeric*", "*String*" va boshqalar). Agar belgi nomini o'zgartirmoqchi bo'lsangiz, ob'yektni joylashtirganingizdan so'ng darhol nomini klaviaturadan kiritishingiz mumkin. Belgi matnini o'zgartirganingizdan so'ng, kiritish jarayonini tugatish uchun quyida ko'rsatilgan harakatlardan ixtiyoriy birini amalga oshiring:

- Raqamli klaviaturadan qismidan <Enter> tugmasini bosib



• Instrumentlar panelidan ENTER tugmasini sichqoncha orqali tanlang;

• Yuzadagi panel yoki diagrammalar blogi bo'ylab yorliqdan tashqariga sichqonchani bosing;

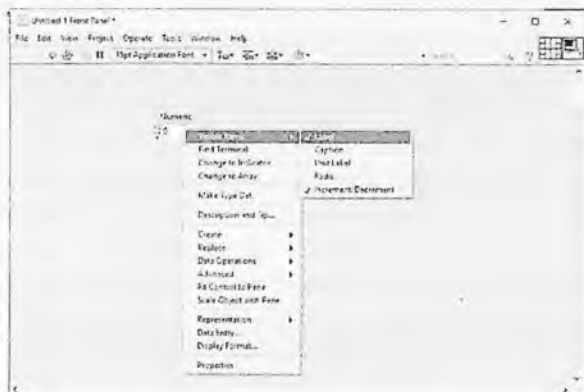
• < CTRL > + <Enter> yoki <CTRL> + <Return> tugmalarini bosing.

LabVIEW da 3 xil belgi turi mavjud: shaxsiy, o'tkazgichlar va erkin. Shaxsiy belgi alohida ob'yektlarga tegishli va u bilan birgalikda bo'ladi. Ular faqat shu ob'yektni bildiradi. Yuzadagi panelda boshqaruv elementi yoki indikator yaratish jarayonida darhol panelda uning belgisi-unikal(takrorlanmas) identifikator paydo bo'ladi. U yuzadagi panel ob'yekti uchun ham, uning terminali uchun ham bir xil. Erkin belgi biror bir alohida ob'yekt bilan bog'liq emas, u xoxishga ko'ra yaratilishi yoki o'chirilishi mumkin. O'tkazgichlar belgisi bu diagrammalar bloki izohlar muhitidir. Ularni faqat shu yerda qo'llash mumkin.

Shaxsiy yorlig'lar

Hozirgina yaratilgan ob'yektda belgi jimlik qoidasiga ko'ra ko'rinib turadi. Uni *Будимые элементы* >> *Метка (Visible items* >> *Label)* kontekst menyusi orqali yashirish mumkin. Siz shaxsiy belgilarni yashirishingiz mumkin, biroq bu belgilarni egalarisiz ko'chirish va o'chirish huquqingiz bo'lmaydi. Tuzilish va funksiyalarda ham belgilar mavjud, lekin bular *нефтверке* qoidasiga ko'ra yashiringan bo'ladi.

Belgi o'zida kodni hujjatlashtirish vositalaridan birini aks ettiradi. Obyektning belgisi uning funksional vazifasini ifodalashi lozim. Yuzadagi panel ob'yektlari uchun bu majburiy qoida, tuzilish uchun esa zaruriyatga qarab bajariladigan qoida hisoblanadi. VIU belgisini ham ko'rinadigan etish imkoniyati mavjud (amalda ular VIU nomlari hisoblanadi), lekin ularni tahrirlab bo'lmaydi.



1.3.2-rasm. Belgini ko'rinadigan etish uchun yordamchi menyuning
Visible Items >> Label opsiyasi

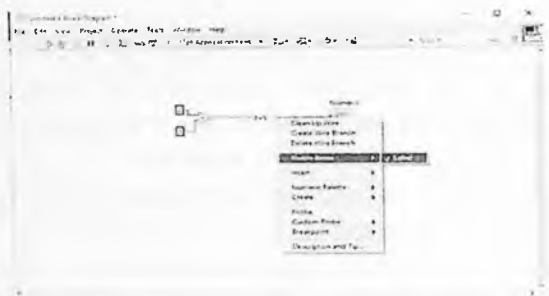
Sarlavhalar. Belgilardan tashqari, yuzadagi panel sarlavha (*caption*) ga ega bo'lishi mumkin.

Sarlavha – matn hisoblanib, u boshqaruvchi element yoki indikatorni izohlaydi. Boshqaruv elementi sarlavhasini yaratish uchun kontekst menyudan **Видимые элементы >>Заголовок** (Visible items>>Caption) elementini tanlang.



1.3.3-rasm. Sarlavhani ko'rinadigan etish uchun yordamchi menyuning
Visible Items >> Caption opsiyasi.

O'tkazuvchilar belgisi – diagrammalar blokini izohlash uchun juda qulay vositadir. Bu o'tkazuvchiga bog'langan matn bo'lib, boshqa joyga o'tkazilganda u bilan birgalikda ko'chib o'tadi. Ular nafaqat o'tkazgichlar qaysi terminal yoki bog'lamdan kelishi noaniq bo'lgan va qaysi manzilgan borishi noma'lum bo'lgan katta diagrammalar bloki uchun emas, balki to'g'ri VU qismlari uchun ham muhimdir. O'tkazuvchilar belgisini qo'shish uchun kontekst menyudan **Видимые элементы >> Метка (Visible items >> Label)** elementini tanlang.



1.3.4-rasm. O'tkazuvchiga belgi qo'shish

Erkin belgilarni yaratish

Erkin belgilar ob'yektga bog'liq emas. Siz uni xoxishingizga ko'ra yuzadagi panelda yaratishingiz, boshqa joyga ko'chirishingiz va undan foydalanishingiz mumkin. Ularni panellar va diagrammalar blokini nomlashda qo'llang. Erkin belgilarni yaratish va ixtiyoriy ko'rinadigan matnlarni tahrirlash uchun matn kiritish instrumentidan foydalaning.

Erkin belgilarni yaratish uchun **Инструменты (Tools)** panelidan matn kiritish instrumentini tanlab yaratishingiz mumkin. Chap tarafda matn kursori bilan mayda to'g'ri to'rtburchak hosil bo'ladi, kiruvchi ma'lumotlarni kiritish matni ko'rinishida qabul qilishga tayyor turadi. Matnni tering va uni yuqorida aytilgan to'rtta metoddan biri yordamida kiriting.

Erkin belgilar asosan diagrammalar blokini izohlash uchun mo'ljallangan.

Shrift, usul, o'lcham va ranglarni o'zgartirish

Siz *LabVIEW* da instrumentlar panelidan *Shrift (Font)* menyusi orqali matn parametrlarini o'zgartirishingiz mumkin. Siljitish instrumenti yordamida ob'yektni tanlang yoki matn kiritish yoki boshqarish instrumenti orqali matnni tanlab, so'ngra *Text Settings (Matn sozlovi)* menyusidan yangi parametrlarni o'zgartirish mumkin. O'zgartirish barcha tanlangan yoki belgilangan ob'ektlarga kuchga kiradi. Agar biror-bir ob'yekt tanlanmagan bo'lsa, kelgusida matnni ifodalash uchun *нефть* qoidaga ko'ra tanlangan shrift, siz o'zgartirgan parametrlardan iborat bo'ladi - *Text Settings (Matn sozlovi)* menyusi.

LabVIEW interfeysining turli qismlarida turlicha shriftlar ishlatiladi: *System*, *Application* va *Dialog*. Bu shriftlar har doim standart qoidaga ko'ra shriftlarni o'zgartiradi. Agar siz uni o'zgartirsangiz, bu barcha boshqarish elementlari uchun ta'sir qiladi.

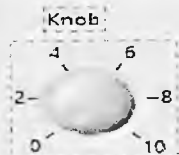
Application shrifti *boshqarish elementlari* palitrasi uchun ishlatiladi. *System* shrifti barcha menyularga ta'sir qiladi. *Labview* *Dialog* shriftlarini barcha dialog oynalarda ishlatadi.

Foydalanuvchi interfeysi agar u *VU* vazifasini bajaruvchi bilan bog'lanmagan bo'lsa, hech qanaqa o'ziga xoslikni ifodalamaydi. Real uskunani siz qachonki diagrammalar blokida *VIU* va tuzilishlar, funksiyalarni joylashtirsangiz yarata olasiz. Buning uchun *Boshqarish elementlari (Controls)* palitrasiga qanday kirgan bo'lsangiz, xuddi shu singari *Funksiyalar* palitrsiga kirish mumkin. So'ngra kerakli ob'yektni ichki palitradan tanlash va uni joylashtirish uchun diagrammalar blokida sichqonchani bosish kifoya. Diagrammalar blokida *Funksiyalar* palitrasining *Sonli* ichki palitrasidan *Qo'shish* funksiyasini o'tkazish misoli ko'rib o'tilgan edi.

Tahrirlash metodi

Panelda ob'yektni joylashtirganingizdan so'ng sizga uni bir joydan boshqa joyga ko'chirish, nusxalash, o'chirish va hokazo istaklar paydo bo'ladi. Quyida bu ishlarni qanday amalga oshirish ko'rsatilgan:

Obyektni belgilash. Ob'yektni bir joydan boshqa joyga ko'chirishdan oldin uni belgilashingiz lozim. Ob'yektni belgilash uchun siljitish instrumenti o'rnatilgan bo'lsa, sichqonchani tugmasini bosib. LabVIEW da ob'yekt belgilanganda punktirli chegara bilan uni o'rab oladi.



1.3.5-rasm Obyektni belgilash

Birdan ortiq ob'yektni belgilamoqchi bo'lsangiz <Shift> tugmasini bosib turib har bir ob'yekt ustiga sichqonchani bosib turib ularni belgilash mumkin. Xuddi shunday, ob'yektni belgilashni bekor qilmoqchi bo'lsangiz sichqonchani yana bir marta shu ob'yekt ustiga bosib.

Obyektlarni belgilashning yana boshqa yo'li ularni to'g'ri burchak ostida qamrab olishdir. Buni amalga oshirish uchun siljitish instrumenti ochiq sohaga sichqonchani bosib, uni tanlamoqchi bo'lgan ob'yektlar tanlanmaguncha diagonal bo'ylab harakatlantirish zarur. Agar sichqonchani qo'yib yuborsangiz to'g'ri burchak yo'qoladi va belgilangan ob'yektlar punktirli chiziq bilan o'rab olinadi.

Bir vaqtning o'zida ham yuzadagi panel, ham diagrammalar blokidagi ob'yektni belgilash mumkin emas.

Ob'yektni bir joydan boshqa joyga ko'chirish

Obyektlarni bir joydan boshqa joyga ko'chirish uni belgilash va siljitish yo'li bilan bajariladi. LabVIEW da <Shift> tugmasini bosib turgan holda ob'yektni gorizontal yoki vertikal yo'nalishlarda ko'chirish (ob'yektning dastlabki harakat yo'nalishiga qarab) mumkin.

Obyektni nusxalash.

LabVIEW da ob'yektni belgilaganingizdan so'ng siz uni *Правка (Edit)* menyusining *Копировать(Copy)* opsiyasi orqali nusxalab, yangi ob'yektni qayerga joylashtirmoqchi bo'lsangiz kursorni shu yerga o'rnatib *Вставить(Paste)* opsiyasini ishlatishingiz mumkin.

Obyektlarni o'chirish. Ob'yektlarni o'chirish uchun oldin ob'yektni belgilash, so'ngra *Правка(Edit)* menyusining *Удалить(Delete)* opsiyasini tanlang yoki <Delete> tugmasini bosish mumkin.

Ob'yekt o'lchamini o'zgartirish. Ko'plab ob'yektlar o'lchamini o'zgartirish mumkin. Ob'yektni ko'chirish instrumenti ob'yekt ustiga olib kelinsa, ob'yekt burchaklarida o'zgartirish joylari paydo bo'ladi. Siz uni o'zingiz xoxlagan hajmga kelgunicha shu burchaklardan foydalanib o'zgartirishingiz mumkin.

Ob'yektlarni bo'yash

LabVIEW da ko'plab ob'yektlarni rangini o'zgartirish mumkin. Ob'yekt yoki orqa plan rangini o'zgartirmoqchi bo'lsangiz, ularni sichqonchani o'ng tugmasi orqali bo'yash instrumenti rejimi tanlansa ranglar palitrasi paydo bo'ladi.

Sichqonchani tugmasini bosmasidan, kursorni palitra bo'ylab o'tkazing. Ob'yekt yoki orqa plan kursor o'tgan rangda bo'yaladi.

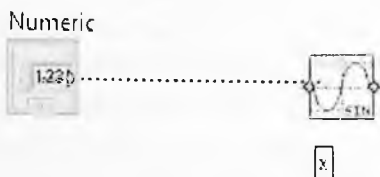
Obyektlarni tekislash va uning bo'linishi

Ba'zida VU bejirim ko'rinishi uchun displeyda ob'yektlarni tekislash va joylashtirish kerak bo'ladi. *LabVIEW* da mavjud funksiya bu jarayonni juda oddiy bajaradi. Ob'yektlarni chiziqli bo'ylab tekislash uchun oldin ularni belgilang, keyin esa *Выравнивание(Align)* menyusiga o'ting va tekislash usulini tanlang. *Распределение(Distribute)* ob'yektlarni bir-biridan bir xil masofada joylashishini ta'minlab beradi.

Elementlarni bog'lash

Ushbu paragrafda bilish shart bo'lgan elementlarni ulanish usullari keltirilgan.

Terminalarni ulash uchun ulash asbobi ("katushka") dan foydalanamiz. Ikki terminalni ulash uchun, birinchi terminal ustida uskuna sichqoncha bilan tanlanadi, ikkinchi terminalgacha sichqoncha etiladi va ustiga sichqoncha bilan bosiladi. Qaysi terminal birinchi tanlanishi farqi yo'q. Terminal ustida bog'lash kursori etilganda bog'lash hududi o'chib-yonishni boshlaydi. Sichqoncha bosilganda bu terminallar ulanishi boshlanadi.



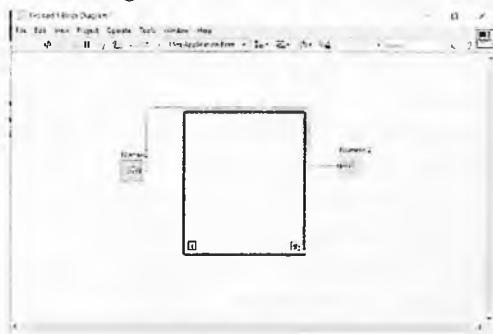
1.3.6-rasm. Sinus funksiyali chiqarish terminalini sonli terminallar bilan ulanishi

Birinchi terminalni ulashni boshlaganizdan, sichqonchani qimirlatdingizki *LabVIEW* blok-diagrammaga o'takazuvchini chizishni boshlaydi, xuddiki leska katushkadan ochilayotgandek. Bunda sizga sichqoncha tugmasini bosib turish talab etilmaydi.

Mavjud o'tkazuvchidan ulashni boshlash uchun, yuqorida keltirilganlarni bajarishingiz lozim: mavjud o'tkazuvchida boshlab yoki tugatishni katushka o'tkazuvchini tagida kelganda yonib o'chishni boshlaydi.

Barcha ob'yektlarning boshidan oxrigacha o'tkazuvchilarni tortish qiyinchilik tug'dirishi mumkin. Bu amalni osonlashtirish uchun *LabVIEW* da o'tkazuvchini avtomatik o'tkazish funksiyasi mavjud - minimal egilishlar soni bilan o'tkazuvchini joylashishini tanlash mumkin. Siz bu rejimni vaqtincha o'tkazuvchini yaratish boshlagandan so'ng <A> klavishini bosish orqali o'chirishingiz mumkin. <A> klavishasini qayta tanlash bu rejimni qayta yoqib yuboradi. Bundan tashqari siz o'tkazuvchini Clean Up Wire(o'tkazuvchini tozalash) yordamchi meysusi orqali optimizatsiyalash (qulay variant) mumkin.

Ayrim foydalanuvchilar o'tkazuvchini avtomatik joylashtirish, Word avtomatik formatlashdan ko'ra qulaymas deyishadi. Siz bu rejimni quyidagi menyu orqali o'chirishingiz mumkin: Tools >> Options >> Block Diagram >> «Enable automatic wire routing».



**1.3.7-rasm. O'tkazuvchini avtomatik joylashish
rejimida o'tkazgichlarning egilishi**

Avtomatik ulanish

Elementlarni ulashning yana bir usuli bu avtomatik ulashnisha sanaladi. Biror xil funktsiya tanlanganida elementlar boshqaruv palitrasida quyidagini sezishingiz mumkin. *LabVIEW* vaqtinchalik o'tkazuvchilar uchun ulanish mumkin bo'lgan joylarni ko'rsatadi. Agar siz Boshqaruv elementlar qatori bilan yoki ob'yektlar olib o'tsangiz *LabVIEW* 1.3.8-rasmda ko'rsatilganidek ob'yektlar ulanishini ko'rishingiz mumkin. Bu vaqtda sichqoncha tugmasini qo'yib yuborish ulanishini mustahkamlaydi.

Numeric



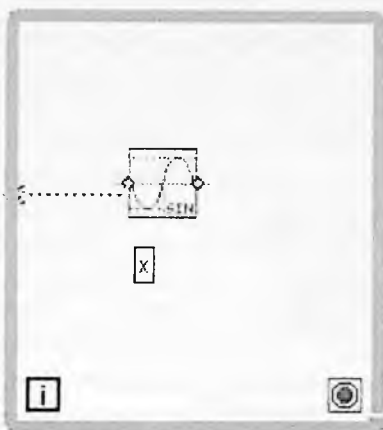
1.3.8-rasm. Avtomatik ulanish

Avtomatik ulanish ishlashi uchun, ob'yektlarni bir-biriga yaqin o'tkazish kerak. Agar sizga avtomatik ulashnisha noqulay yoki to'g'ri kelmasa, u holda siz o'z xoxlagan holda terminallarni ulashingiz mumkin.

Bazali operatsiyali ulanishlarni qo'llagan holda siz terminalni tuzilishlar tashqarisidan ham terminallar bilan ulashingiz mumkin. *LabVIEW* tuzilishlarda faqatgina o'tkazgich va tuzilish chegarasi kesishgan nuqtada "tunnel" yaratadi.

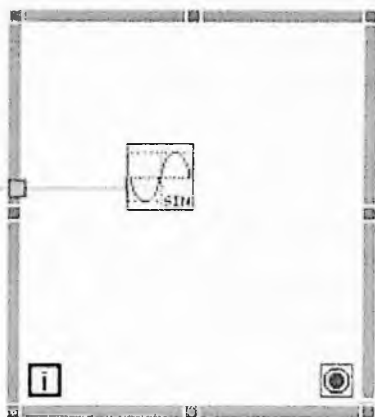
1.3.9-rasmda o'tkazgich chizilganda "tunnel" qanday ko'rinishga egaligi tasvirlangan bo'lsa 1.3.10-rasmda esa operatsiya tugatilgandan so'ng "tunnel" ko'rinishi tasvirlangan.

Numeric



1.3.9-rasm. O'tkazgich tuzilishdan tashqaridagi ob'yektdan ulangan holatgacha

Numeric



1.3.10-rasm. O'tkazgich tuzilish tashqarisidagi ob'yektdan tuzilish ichkarisiga ulangandan keyingi holat. Ma'lumotlar yuborilishi uchun "tunnel" hosil bo'lgan

Murakkab ob'yektlar ulanishi.

VIIU yoki standart uzalg'a elementlarni ulanish jarayoni vaqtida, ulanish uskunasini ikonkaga yaqinlashganda mavjud maslahatlar qatori va "mo'ylov" paydo bo'ladi. Ulanish jarayonida ahamiyat bersak agar o'tkazuvchilar "mo'ylovi" 1.3.11- rasmda



error out

ko'rsatilganidek VIIU ning ikonkasi atrofida paydo bo'lib, ma'lumotlar tipi stili ko'rinishida, o'tkazuvchining qalinligi va rangi taqdim etiladi. Hozir aytib o'tganimizdek "mo'ylov" avtomatik ulanishda ishlash qulay sanaladi.

1.3.11-rasm. Ulanish uskunasining yaqinlashuv paytida yordamchi va "mo'ylov" o'tkazgichining paydo bo'lishi

O'tkazgichlarning buzilishi

Elementlarni ulash jarayonida xatolikka yo'l qo'yilganda oddiy o'tkazgichlarning o'rniga qora punktli chiziq (Broken) buzilgan o'tkazgich paydo bo'ladi. Barcha xatoliklar tuzatilmaguncha VU ni ishlatish tugmasi buzilgan holatda turadi va dastur ishga tshmaydi.

Uzilgan o'tkazgichning o'rniga qizil xoj ("krest") tasvirlanishi yoki tasvirlanmasligini siz quyidagi sozlashlar orqali belgilashingiz mumkin: Tools >> Options >> Block Diagram >> "Show red Xs on broken wires".

Buzilgan o'tkazgichni bartaraf etish uchun siz uni belgilab olib, o'chirishingiz lozim. "Remove Broken Wires" opsiyasini "Edit" menyudan tanlangan holatda barcha buzilgan o'tkazgichlarni yo'qotish yaxshi variant hisoblanadi.

Buzilgan o'tkazgichlar kerakli axborotlarni saqlashi ham mumkin: buzilganlik - bu faqatgina uzilish emas balki noto'g'ri ulangan ma'lumotlar tipi hamdir. Bu holatda o'tkazgichlarni

o'chirishmaslik, balki tuzatish kerakdir. Buzilgan o'tkazgichlarni barchasini o'chirish buyrug'idan oqilona foydalanish kerak.

Agar sizga qaysi sababga ko'ra o'tkazgich buzilgani no'malum bo'lsa, dasturning ishga tushurush tugmasi murojlat eting yoki buzilgan o'tkazgichning yordamchi menyusini chaqiring va *"Xatoliklar ro'yxati (List Errors)"* operatsiyasini tanlang. Muamolar tasnifi oynasi tasvirlangan oyna paydo bo'ladi. Shu bilan birga yordam oynasida shu axborotlar ko'rsatiladi. O'tkazgichning buzilish sabarlari turlicha bo'lishi mumkin. Yordamchi menyuning quyidagi punktlari xatoliklarni tuzatishi mumkin: *"Delete Wire Branch"*, *"Create Wire Branch"*, *"Remove Loose Ends"*, *"Clean Up Wire"*, *"Change to Control"*, *"Change to Indicator"*, *"Enable Indexing at Source"* va *"Disable Indexing at Source"*. Faqat avval muammoni tushunish zarur va keyingina ushbu punktlardan birini tanlash zarur.

Elementlarni ulanishda maslahatlar

Quyidagi maslahatlar diagrammalar blokida elementlarni ulashni osonlashtiradi:

- sichqonchani bosmagan holatda siz ulanish yo'nalishini faqat 900 ga faqat bir marotaba o'zgartira olasiz;

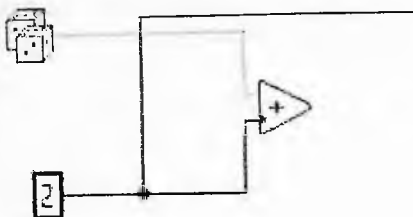
- o'tkazgich yo'nalishini o'zgartirish uchun shichqonchani chap tugmasini bosing va yo'naltiring;

- "Probel" klavishi yordamida terminaldan chiqayotgan o'tkazgich yo'nalishini o'zgartirish mumkin;

- Diagrammalar blokidagi o'tkazgichni ishini boshlash va tugatish uchun sichqonchani ikki marta chap tugmasini tanlash mumkin;

- O'zkazuvchilar kesishganida birinchi chizilgan o'tkazuvchida unchalik katta bo'lmagan bo'shliq paydo bo'ladi, xuddiki birinchi o'tkazuvchi ikkinchi chizilgan o'tkazuvchining tagida o'tgandek tuyuladi (1.3.12-rasm);

■ Tarqalgan o'tkazuvchilarni tarqalish nuqtasini LabVIEW qalin nuqta orqali tasvirlaydi. Buni quyidagi menyu orqali ko'rinmas holatga keltirish mumkin: Tools >> Options >> Block diagram >> "Show dots at wire junctions."



1.3.12-rasm. O'zkazuvchilar kesishganida birinchi chizilgan o'tkazuvchida unchalik katta bo'lmagan bo'shliq paydo bo'ladi

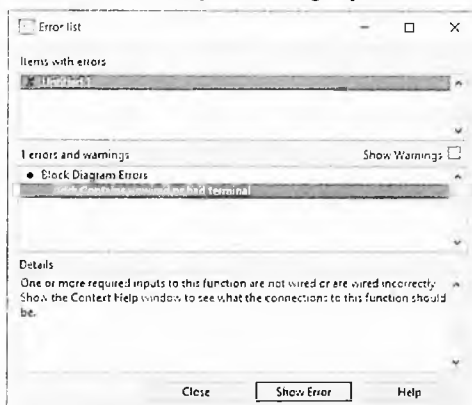
Nosoz VU da otladkalash

Siz qachondir bir /marotabadan xatolarsiz dastur tuzganmisiz? Yo'q bo'lsa kerak! LabVIEW sizga shaxsiy VU ishlab chiqishga yordam beradigan ko'plab o'rnatilgan nosozliklarni aniqlash instrumentlariga ega. Bu paragrafda qanday tarzda bu instrumentlardan maksimal foydalanish tavsiflangan.

Ishlamaydigan VU ni ishga tushirish mumkin emas. Ishga tushirish tugmasi VU da nosozliklar borligini ko'rsatuvchi singan strelka ko'rinishiga ega. Bunday holat VU ni yaratish jarayonida va VU ni tahrirlashda normal hisoblanib, toki siz diagrammalar blokidagi barcha terminallarni ulashni tugatmagunizcha ushbu holat o'zgarmaydi. VU ni ishlay olmaslik qobilyatini aniqlashtirish uchun sichqoncha bilan nosoz ishga tushurish tugmasini tanlash zarur.

Berilgan VU dagi mavjud barcha xatolar ko'rsatilgan xatolar ro'yxati (*Error List*) axborot oynasi paydo bo'ladi. Xoxishizga ko'ra siz boshqa ochilgan VU larga aloqador xatolar ro'yxatini oynaning yuqori qismidagi ro'yxatdan tanlash yo'li bilan ko'rishingiz

mumkin. Belgilangan xato haqida ko'proq axborot olish uchun uning ustiga sichqoncha bilan tanlang. VU dagi xatolik joylashgan joyni aniqlashtirish uchun ro'yxatdagi xatolik ustiga sichqonchani ikki marta bosning. *LabVIEW* mavjud xatolikni chaqirgan oynani mos ekranga chiqaradi va ob'yektni belgilaydi (1.3.13- rasm).



1.3.13-rasm.Xatoliklar ro'yxati oynasi

Ogohlantirish

Agarda sizga qo'shimcha yordam kerak bo'lsa dasturdagi otladkalash oynasida unda Xatolar ro'yxati oynasidan ogohlantirishni ko'rsatish (*Show Warning*) opsiyasini ishlatishiz mumkin. Ogohlantirishlar normal holat va ular dastur ishlash qobilyatini buzmaydi. Xuddi shunday birorta ham elementga ulanmagan boshqarish elementi terminali ham dastur ishlashiga ta'sir ko'rsatmaydi. Agar siz ogohlantirishni ko'rsatishni belgilab qo'ygan bo'lsangiz va sizda biror-bir ogohlantirish mavjud bo'lsa, u holda instrumentlar qatorida ogohlantirish (*Warning*) tugmasi paydo bo'ladi. Oynani ogohlantirishlar ro'yxati bilan ko'rish uchun (xatolar ro'yxati oynasida), shu tugmacha ustida sichqonchani bosning. Oynada ogohlantirishlar sabablari tushuntirilgan oyna qismi paydo bo'ladi.

Keng tarqalgan xatoliklar

Ayrim xatoliklar boshqa xatolarga nisbatan tez-tez takrorlanadi, shuning uchun keyinchalik siz ularning qaytarilishini oldini olishingiz uchun biz ularni ro'yxatga birlashtirishga qaror qildik. Agarda sizning VU da ishga tushurish tugmasi ishchi holatda bo'lmasa, unda quyida sanab o'tilgan nosozliklardan birini bartaraf etish muammoni hal etishda yordam berishi mumkin:

Kiritish funksiyasi talab etiladigan ma'lumotlar manbaasiga ulanmagan. VU har xil algoritmlarda testlashda ishlash vaqtida birorta ham ob'yekt diagrammalar blokida ulanmay qolmasligi kerak;

Diagramma blokida ma'lumotlar tipi to'g'ri kelmasligidan ishdan chiqqan nosoz o'tkazgich mavjud bo'ladi yoki bo'lmasa nimadir ostida yashirilganligi yoki shunchalik kichikligi tufayli ko'rish qiyin bo'lgan oxiri ulanamgan o'tkazgichlar mavjud bo'ladi;

VIU nosoz bo'lsa yoki siz bog'lash panelini diagramma blokiga joyshirtirganizdan so'ng qayta tahrirlagan bo'lsangiz shu holatga keladi. Yordamchi menyudagi Almashtirish opsiyasini yoki Aloqani VIU bilan yangilash (*Relink to SubVI*) tanlab, VIU ga qayta ulanish uchun ishlatishingiz mumkin.

Eng keng tarqalgan xatoliklar.

Sizda buzilgan, ko'rinmaydigan yoki bog'lam xususiyatlari yordamida qayta o'zgartirilganlik tufayli xatolik bo'lsa (bu haqda keyingi paragraflarda batafsil ko'rib o'taladi);

Siz tasodifan ikki boshqarish elementini bog'ladingiz yoki ikki boshqarish elementini bir indikatorga akslantirganiz sabab xatolik yuz beradi. Xatoliklar ro'yxati oynasida <<*Signal: has multiple sources*>> (bir necha ma'lumotlar manbaasi) yozuvi paydo bo'ladi. Odatda bunday xatoliklar bitta boshqarish elementini

akslantirish elementiga (*indikatorga*) almashtirish yo'li bilan to'g'irlanadi;

VU topilmadi yoki u sizda mavjud emas(sizga VU ni VIU quyi darajasiz yuborishgan bo'lsa) yoki siz uni VU ni sozlamay turib qayta nomlab chaqirmoqchi bo'lsangiz. Unda siz VIU ni qayta nomlasangiz xotiradagi barcha VU lar yangi nomni ko'rsatishga o'zgaradi. Siz bu o'zgarishlarni *LabVIEW* ni yopishdan oldin saqlashingiz kerak.

VU ni qadamba-qadam bajartirish

Dasturni otladkalash vaqtida ba'zan diagramma blokini tugunma-tugun bajartirish kerak bo'ladi. Ma'lumotlat tuguni o'zida VIU, funksiyalar, tuzilmalar, dastur kodi bilan tugunning o'zaro ta'siri, formulali tugunlar va xususiyat tugunlarini mujassamlashtiradi. Qadamba-qadam bajarishni boshlash uchun VU ni qadamba-qadam bajartirish tugmasining birini tanlab ishga tushirish zarur(ishga tushirish tugmasi o'rniga), undan so'ng vaqtinchali VU ni to'xtatish nuqtasini kiritib yoki sichqoncha bilan "*Pause*" tugmasini tanlash orqali to'xtatish mumkin. Dastur bajarilishini davom etish uchun "*Pause*" tugmasiga qayta murojaat eting. Siz VU lar bilan qadamba-qadam harakat vaqtida dastur bajarishni yoritishiz mumkin va shu tariqa ma'lumotlarni bir tugundan boshqasiga o'tish vaqtida kuzatishingiz mumkin.

Qadamba-qadam bajarish rejimida faol holatda bo'lgan xoxlangan uchta tugmani tanlab keyingi qadamga o'tish mumkin. Har bir tugma keyingi qadam bajarilish yo'lini belgilaydi:

Ichkariga qadam tugmasini tanlash (step Into) bizga VIU birinchi qadami yoki strukturalari bajarilishi sodir bo'lishini ta'minlaydi va undan keyin—"Pause" yuz beradi. Shuningdek *WINDOWS* da <<pastga strelkasi>> klavishini bosib bir vaqtning o'zida <CTRL> klavishini bosish mumkin;

Ichidan qadami (Step Over) tumasini tanlash. Undan so'ng VIU yoki tuzilishlar bajarilishi boshlanadi. (tsikl, ketma-ketlik va

hokazo), undan keyin "Pause" bo'ladi. Shu bilan birga WINDOWS da <<strelka o'ngga>> klavishini

<CTRL> klavishi bilan bir vaqtda bosish qulay;

Ichkaridan qadami(Step Out) tugamasini tanlash joriy diagramma bloki, tuzilish yoki VIU ni tugallanishini bajarish mumkin. Bu harakatdan so'ng "Pause" bo'ladi. Yoki bo'lmasa WINDOWS da <<strelka yuqoriga>> klavishini <CTRL> bilan bir vaqtda klavishlarini ishlatishingiz mumkin.

Dasturni bajarishda yoritish

Ba'zan sizning ma'lumotlaringiz qayerdaligini va ular bilan nima sodir bo'layotganini virtual kuzatish qulay. LabVIEW da siz VU diagramma blokida bajarilish animatsiyasini kuzatishingiz mumkin. Bu rejimni yoqish uchun instrumentlar tasmasidan yoritishni bajarish(Execution Highlightning) buyrug'ini ishlatish mumkin.

Ma'lumotlar bir tugundan boshqasiga harakati doirachali o'tkazgichlar harakati bo'ylab belgilanadi. E'tibor bering bu rejimda VU ancha sekin bajariladi. Yoritishni bajarish tugmasini sichqoncha bilan yana bir marotaba tanlash dasturni normal bajarilishini ta'minlaydi. 1.3.14-rasmda yoritish rejimida ishlatiladigan VU ning tashqi ko'rinishi ko'rsatilgan.



1.3.14-rasm. Yoritishni bajarish rejimida o'tkazgichlardan ma'lumotlar doirachalar ko'rinishida harakatlanadi, suzib chiquvchi satrlarda qiymatlar tasvirlanadi

Yoritish rejimi paytida ishlashda chiquvchi terminallarda avtomatik tarzda chiquvchi ma'lumotlar ko'rsatilgan.

Odatda yoritish rejimidan tugundan ma'lumotlar qanday o'tayotganini ko'rish uchun qadamba-qadam harakat rejimida undan keng foydalaniladi. Agar bu rejimlar birga qo'llanilsa relyefli bajarish tasvirlarida dasturosti belgilarida qanday VU lar ishlayotgani, qaysilari esa o'zini navbatini kutayotganini ko'rish mumkin.

Dasturni bajarishda to'xtatish nuqtasidan foydalanish

To'xtatish nuqtalari (*breakpoints*) VU ni zararlamaydi. Ular faqat otladkalash uchun ularni bajarilishini to'xtatadi. Pauzalar kerak qachonki siz tugun yoki dastur bajarilishi paytida o'tkazgich VU ga kelayotgan ma'lumotlarni tekshirishni xoxlasangiz. Dastur bajarilishini to'xtatish nuqtasiga yetganda u pauza tugmasini faollashtiradi. Shundan so'ng dasturni qadamba-qadam bajarishni o'tkazish mumkin, yuzadagi panel ob'yektlari kattaliklarini o'zgartirish yoki pauza hamda ishga tushirish tugmalarini tanlash yo'li bilan oddiygina dastur bajarilishini davom ettirish mumkin. To'xtatish nuqtasini o'rnatish uchun instrumentlar palitrasidan nazorat kiritish instrumenti bilan diagrammalar bloki ob'yektiga sichqonchani tanlah mumkin. Nuqtani bartaraf etishda ob'yekt ustiga sichqoncha bilan yana bir marotaba murojaat etish mumkin. Agarda tugmani tanlash to'xtatish nuqtasini o'rnatilishiga sabab bo'lsa, kursorning markaziy doirasi qora rangda bo'ladi. Tanlashdan so'ng yoki agarda siz to'xtash nuqtasida mavjud elementga kursorni etsangiz kursor markazi oq bo'ladi.

Joylashishiga qarab to'xtash nuqtasi o'zini har xil tutadi:

Agarda nuqta diagramma blokiga o'rnatilgan bo'lsa, unda diagramma bloki atrofida qizil chegara hosil bo'ladi va pauza diagramma bloki bajarilishi tugallangandan so'ng sodir bo'ladi;

Agarda nuqta ma'lumotlar tugunida o'rnatilgan bo'lsa, unda u qizil chegara bilan o'rab olinadi va pauza ma'lumotlar tuguni bajarilishidan oldin sodir bo'ladi;

Agarda nuqta o'tkazgichga o'rnatilgan bo'lsa, unda qizil doiracha paydo boladi, har qanday o'tkazgich qizil chegara bilan o'ralgan bo'ladi. Pauza o'tkazgich orqali ma'lumotlar o'tib bo'lgandan keyin sodir bo'ladi.

To'xtash nuqtasi tufayli VU bajarilishi to'xtasa diagramma bloki oynasi oldingi planga o'tadi va to'xtashni chaqirgan ob'yekt lipirlashni boshlaydi. To'xtash nuqtalari VU bilan birgalikda saqlanadi, lekin faqat bajarish jarayonida faollashadi. Bir nechta VU larda ko'p miqdorda to'xtash nuqtalarini boshqarishda "*To'xtash nuqtalri masteri (Breakpoint manager)*" xizmat qiladi. U to'xtash nuqtalarini vaqtinchalik o'chirib qo'yish yoki o'chirib tashlash hamda topishga ruxsat beradi. Nuqtani topish uchun ro'yxatdan unga sichqoncha bilan ikki marta murojaat etish zarur. Bunda diagramma bloki oynasi oldingi planga chiqadi va tanlangan nuqta elementi esa lipirlab turadi.

Gohida to'xtash nuqtasini o'chirib tashlamasdan vaqtinchalik nofaol holatga keltirib qo'yish foydali bo'ladi. Buni master oynasidan yoki dastur bajarilayotgan vaqtda yordamchi menyuu orqali amalga oshirish mumkin. Master oynasida to'xtatish nuqtasida kerakli elementlarni belgilang va ichi to'ldirilmagan doirachali tugmani tanlang. Ichi to'ldirilgan doirachalar tanlangan to'xtash nuqtalarini yana faollashtiradi. Qizil xoj tanlangan to'xtash nuqtalarini o'chirib tashlashini tushunish qiyin emas. Yordamchi menyuda diagramma blokining xoxlangan elementi dastur ishlayotgan vaqtda *Breakpoint* menyuoosti mavjud bo'lib, *Master* oynasiga o'tishga yoki sanab o'tilgan barcha ishlarni bajarishga ruxsat beradi.

Biz to'xtash nuqtali VU ni saqlashni maslahat bermaymiz, axir u faqat otladka intrumenti hisoblanadi. Bunday VU ni tasodifan saqlash mumkin va siz yoki sizning hamkasbingiz keyinchalik uni

ochsangiz o'ylanib qolasiz: Ishlab turgan loyiha birdan hech narsadan hech narsa to'xtab qoldi deb. Siz Edit >> Find and Replace menyusidan yoki tezkor murojaat klavishlari <Ctrl+F> (Windows) yordamida barcha o'rnatilgan to'xtash nuqtalarini tezda aniqlashingiz mumkin. Qidirish muloqot oynasida (Find) qidirishni tanlash zarur: (*Search for: Objects*) ob'yektlari (*Object: Others>> Breakpoint*) va ilova nusxasi va qidiqish qismini tanlang.

Takrorlash va mustaqil ishlash uchun savollar

1. LabVIEW da belgi nima va ularning turlari?
2. Sarlavhalarning maqsadi nima va ular belgilardan nimasi bilan farq qiladi?
3. Avtomatik ulanish haqida nima bilasiz?
4. Elementlarni bog'lashning necha xil usuli bor?
5. Elementlarni ulanishda qanday maslahatlar berasiz?
6. Nosoz VU lar qanday tuzatiladi?
7. VU ni qadaba-qadam bajartirish nima uchun kerak?
8. Dasturni bajarishda to'xtatish nuqtasidan nima maqsadda foydalaniladi?

I-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI

AMALIY MASHG'ULOT №1

LabVIEW muhiti bilan ishlashni boshlash, elementlar va diagrammalar blokidan foydalanish.

Mashg'ulotning maqsadi: *LabVIEW* muhitida ishlash texnologiyasini o'zlashtirish.

Taxminiy sonlarni generatsiyalaydigan va ularni ostsillografda grafik asklantiradigan oddiy VUni yaratish uchun:

1. LabVIEW dasturi ishga tushiriladi.

2. Ishga tushurish jarayonida LabVIEW ning **Getting Started** dialog oynasida **Новый VP (Blank VI)** opsiyasi tanlanadi. Blank VI punkti **New** oynasida joylashgan. Ekranda **Untitled 1** nomli yuzadagi panel paydo bo'ladi.

Boshqarish elementlari palitrasidan **Modern >> Graph** ichki palitrasiga o'tiladi.



1.4.1-rasm. Yuzadagi panelga grafiklarni aks etuvchi elementlarni joylashtirish

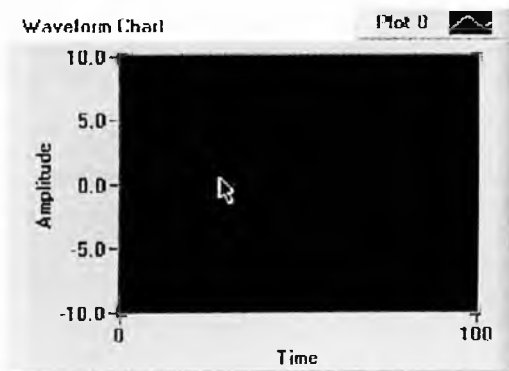
Graph ichki palitrasida **Waveform Chart** tanlanadi.

Ushbu elementni tanlab, yuzadagi panelga sichqoncha ko'rsatkichi keltirilganda, grafik indicator konturi ko'rinadi.



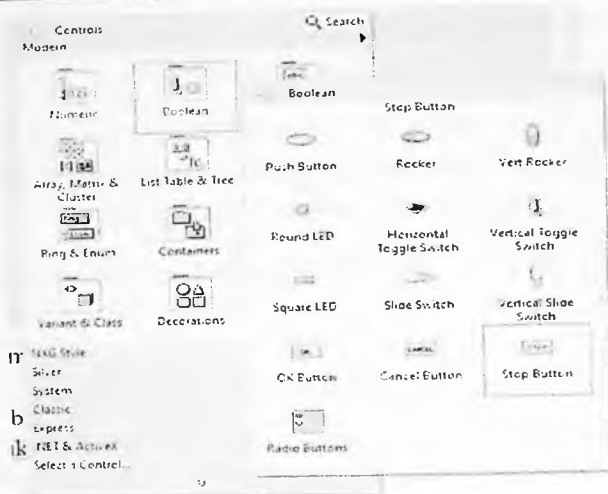
1.4.2-rasm. Yuzadagi panelda elementni xomaki joylashtirish pozitsiyasi

Yuzadagi panelning xohlagan nuqtasida kursorni o'rnatib, sichqonchaning chap tugmasi tanlansa, shu o'rinda grafik paydo bo'ladi.

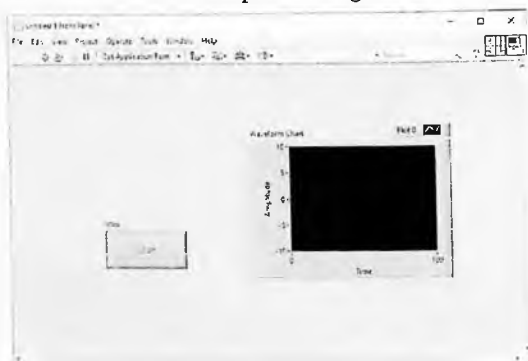


1.4.3-rasm. Waveform Chart elementi ko'rinishi

3. Modern palitrasiga qaytib, Boolean ichki palitrasiga o'tiladi. So'ngra Stop Button elementini tanlab, uni grafik indicator yonida joylashtirish lozim.



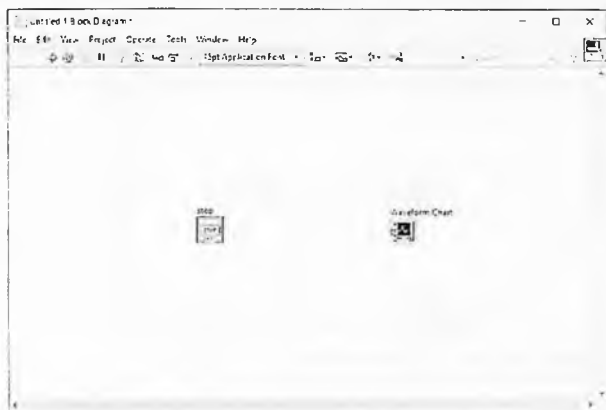
1.4.4-rasm. Boolean palitrasidagi elementlar



1.4.5-rasm. Yuzadagi panelning yakuniy ko'rinishi

4. Endi grafik masshtab $[-10 \ 10]$ dan $[0 \ 1]$ ga o'zgartiriladi. 10 qiymatini tanlab, unga sichqoncha tugmasi ikki marotaba chiqillatiladi. 1.0 qiymatni kiritiladi va oynaning yuqori qismida mavjud uskunalar panelida paydo bo'lgan kiritish tugmasi tanlanadi. Xuddi shunday -10 qiymati 0 bilan almashtiriladi.

5. Windows menyusidan Show diagram punkti tanlanib, diagram-blokiga o'tiladi. U yerda ikkita terminalni ko'rish kerak:



1.4.6-rasm. Diagramma blokida terminallarni ko'rinishi

6. Endi dastur segmenti takror bajarilishi uchun, terminallarni shart bo'yicha tsikl ichiga joylashtiriladi. Functions palitrasidagi Programming >> Structure ichki palitrasiga o'tib, While Loop tanlanadi.



1.4.7-rasm. Diagramma blokida While Loop tuzilmasini topish

Kursor ko'rinishini o'zgartiradi va kichik sikl ikonasiga aylanadi. Endi tugma va ostsillograf terminallarini qamrab olish kerak.



1.4.8-rasm. While Lopp elementi bilan terminallarni qamrab olish

Sichqoncha tugmasini qo'yib yuborilganda, shart bo'yicha tsikl shakllanadi. Ichida bo'sh o'rin bo'lishi uchun tsikl maydonini kattaroq qilish kerak.

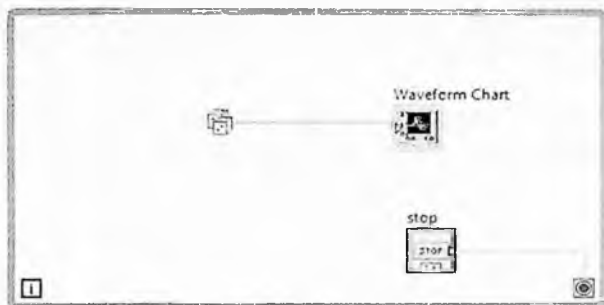


1.4.9-rasm. While Loop siklini diagramma blokida joylashtirish

7. Function palitrasiga o'tib, Programming >> Numeric ichki palitrasida Random number (0-1) funktsiyasi tanlanadi. Piktogrammani shart bo'yicha tsikl elementi ichiga joylashtiriladi.

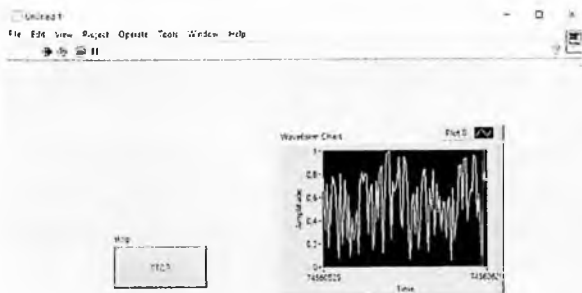
8. Uskunalar palitrasidan uskuna tanlanadi va quyidagi rasmda ko'rsatilganidek ob'ektlarni diagramma blokida joylashtiriladi

9. Uskunalar palitrasidan bog'lanish uskunasi ("katushka") tanlanadi. Random number (0-1) terminali sichqoncha bilan tanlanib, kursorni DBL terminaliga keltirilib, sichqonchanning chap tugmasi bosiladi.



1.4.10-rasm. Diagramma bloking yakuniy ko'rinishi

10. Dasturni ishga tushurish jarayonini yakunlash uchun, oldin yuzadagi panelga qaytish kerak. Ishga tushirish tugmasini tanlab, dastur ishga tushiriladi. Grafikda uzluksiz taxminiy sonlar ketma-ketligini ko'rish mumkin. Dastur ishini to'xtatish uchun Stop tugmasini tanlash lozim.



1.4.11-rasm. Ilovani ishga tushurilgandagi ko'rinishi

Amaliy mashg'ulotni bajarish tartibi

1. Nazariy ma'lumotlar bilan tanishib chiqadi. Belgilangan vazifalarni ketma-ket bajaradi va natijalarni tahlil qiladi.

2. Taxminiy sonlarni generatsiyalaydigan va ularni ostsillografda grafik asklantiradigan VUni yaratish texnologiyasini tushuntiradi.

AMALIY MASHG'ULOT №2

VIRTUAL USKUNALARNI OTLADKALASH VA XATOLARNI TUZATISH

Tahrirlash amaliyoti (Boshqarish elementlari)

Ishdan maqsad: Tahrirlash usullari bilan tanishish va ularni o'rganish. Boshqarish elementlari, indikatorlar va ularning imkoniyatlari

Esda tutish zarurki, digramma bloki oynasi faol bo'lganda faqat "*Funktsiya*" palitrasi ko'rinadi va aksincha yuzadagi panel faol bo'lganda esa boshqaruv elementi (*Controls*) paneli ko'rinadi.

LabVIEW da 4 turdagi oddiy boshqarish elementlari va indikatorlar mavjud: sonli (numeric), mantiqiy (*Boolean*), qatorli (*String*) va yo'l (*Path*). Bundan tashqari *LabVIEW* da massivlar, guruhlar, jadvallar, diagrammalar va grafiklar singari murakkab tiplar ham mavjud, lekin ularni biz keyingi paragraflarda ko'rib o'tamiz.

Sonli boshqarish elementlari va indikatorlar

Sonli boshqarish elementlari VU ga sonli qiymatlarni kiritish imkonini beradi. Sonli indikatorlar esa sonli qiymatlarni aks ettirish uchun xizmat qiladi. *LabVIEW* da turli ko'rinishdagi sonli ob'ektlar mavjud: aylana boshqarish qo'lchasi, yurgich, rezervuarlar, termometrlar va odatiy sonli maydonlar. Bu elementlar **Boshqarish elementlari (*controls*)** palitrasining **Modern >> Numeric** ichki palitrasida joylashgan. Bu palitradagi

elementlar boshqarish elementi yoki indikator sifatida xizmat qilishi mumkin, lekin odatda har bir ob'yekt uchun aniq tip belgilangan. Masalan, termometr indikator deyiladi, chunki termometr odatda shu maqsadda foydalaniladi. Aksincha tugma ob'yekti odatda yuzadagi panelda boshqarish elementi sifatida qo'llaniladi, chunki u asosan ma'lumotlar yoki buyruqni kiritish uchun xizmat qiladi.

Diagrammalar blokida sonli boshqarish elementlari va indikatorlarni terminallarining ko'rinishi- ma'lumotlarni aks ettirilishiga bog'liq. Turli ma'lumotlar tipi ularni xotirada turlicha saqlashni taqozo etadi, bu esa xotirani effektiv ishlatish imkonini beradi, chunki turli tipdagi sonli ma'lumotlar xotirada turlicha joy egallaydi. Bundan tashqari sonli ma'lumotlarni aks etish, sonlarni ishorasi bilan aks etish (manfiy sonlarni ham olishi mumkin) yoki ishorasiz (faqat nol va musbat sonlarni olish mumkin) ni ham bildiradi. Terminallar bir necha harflardan iborat belgilanishlarni saqlaydi, bu harflar zanjiri ulardagi sonlar tipini ifodalaydi, masalan, DBL ikkilangan aniqlikdagi kasr sonlarni belgisi hisoblanadi.



1.5.1-rasm. Boshqarish elementlarining (Controls) Modern >> Numeric ichki palitrasi elementlari

Quyidagi jadvalda LabVIEW da sonli ma'lumotlarni aks etish tiplari keltirilgan. Bu jadvalda sonli ma'lumotlarning xotirada egallaydigan hajmlari va sonli terminal sifatida tasvirlari ikonka ko'rinishda keltirilgan:

1.5.1-jadval.

Sonli ma'lumotlarning tiplari

Sonli ma'lumotlarning tiplari	Qisqartmasi	Terminal (ikonka)	Terminal	Hajmi (baytlarda)
Byte	I8			1
Unsigned Byte	U8			1
Word	I16			2
unsigned Word	U16			2
Long	I32			4
Unsigned long	U32			4
Quad	I64			8
Unsigned quad	U64			8
Single precision	SGL			4
Double precision	DBL			8
Extended precision	EXT			$10^a/12^b/16^c$
Complex single	CSG			8
Complex double	CDB			16
Complex extended	CXT			$20^a/24^b/32^c$
Fixed point	FXP			8 gacha ko'rsatiladi

Sonli elementlar va indikatorlarning tiplarini yordamchi menyuning Representation bo'limidan o'zgartirish mumkin.

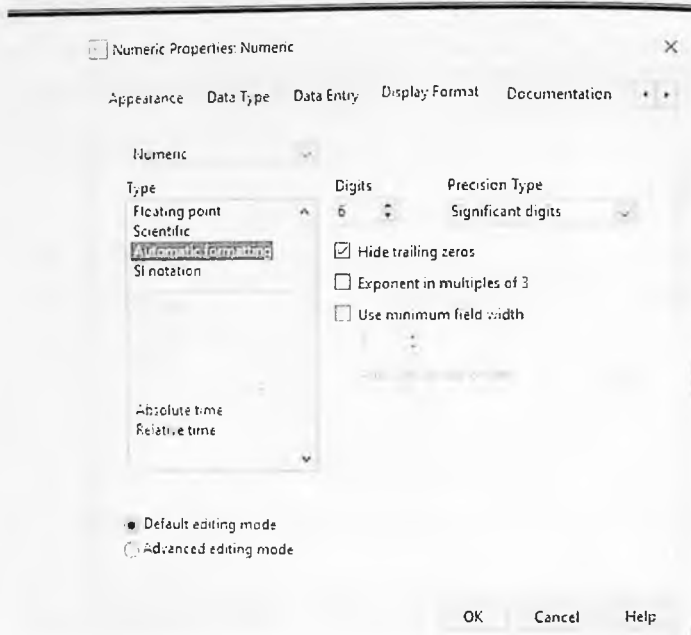


1.5.2-rasm. Sonli boshqarish elementi yoki indikatorning yordamchi menyusining Representation bo'limi ko'rinishi

Sonli indikatorlarning yordamchi menyusining **Adapt to Source** punkti birlamchi ma'lumotlarni avtomatik aks ettirish tipiga moslashtiriladi.

Format va aniqlik

LabVIEW sonli indikatorlarda ma'lumotlarni aks etish formatini tanlash imkonini beradi, ya'ni sonlarni, sana va vaqtni aks etish uchun kerak.



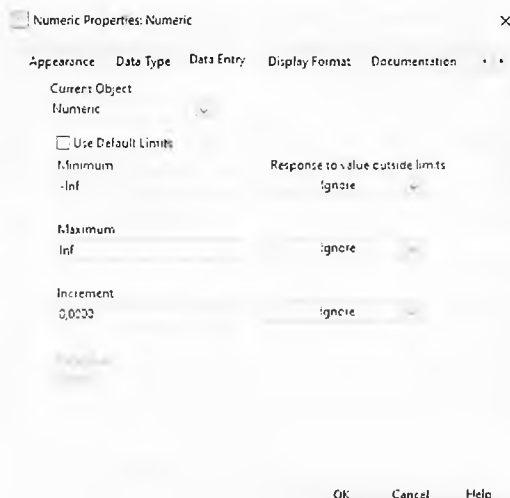
1.5.3-rasm. Elementning xususiyati oynasining Format bo'limi

Agar indikatorlar sonli qiymatlarni aks etish uchun sozlangan bo'lsa, turli aks etish tizimlaridan birini tanlash mumkin: suzuvchi vergulli, ilmiy, muhandislik yoki vaqtga nisbatan sekundlarda. Bundan tashqari verguldan keyingi aniqlikni (precision) ham tanlash imkoni mavjud (0 dan 20 gacha). Aniqlik faqat sonlarni ko'rsatishga ta'sir o'tkazadi, ichki aniqlik esa ma'lumotlarni xotirada akslanishiga bog'liq holatda saqlanadi.

Sonli diapazonlarni boshqarish

LabVIEW sonli qiymatlar va ularni oshirish diapazonlari harakatini aniqlash va sozlash imkonini beradi. Kerakli diapazonni tanlash ob'yektning yordamchi menyusining **Data Range** opsiyasi orqali amalga oshirilishi mumkin. Bu oynada standart bo'yicha parametrlarni qoldirish mumkin yoki sonni o'zgarish

diapazonlarini o'zgartirish mumkin, masalan maksimal va minimal qiymatlarni, kerakli diapazonni oshirish va kamaytirish qiymatlarini o'zgartirish. Agar qiymatlar ko'rsatilgan diapazondan chetga chiqqan taqdirda, ularni tsenariy bo'yicha harakatini ko'rsatish mumkin.



1.5.4-rasm. Xususiyatlar oynasining ma'lumotlar diapazoni bo'limi

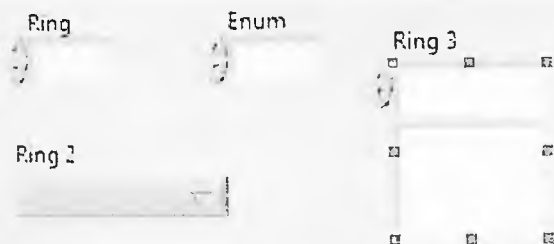
Tsenariylarda quyidagilarni tanlash mumkin:

✓ **Ignore** opsiyasi tanlansa, *LabVIEW* qiymatlarni o'zgartirmaydi va hech qanday harakatga keltirmaydi;

✓ **Coerce** opsiyasi tanlansa, agar qiymat minimumdan kichik bo'lsa, *LabVIEW* uni minimum qiymatga aylantiradi, agar qiymatlar maksimal qiymatdan katta bo'lsa, uni maksimal qiymatga aylantiradi.

Aylanma ro'yxatlar

Aylanma ro'yxatlar maxsus sonli ob'ektlar bo'lib, 16-bitli ishorasiz sonlarga mos qatorlar(matnlar), rasmlar va boshqalarni joylashtirish mumkin. Bu ob'ektlar Boshqarish elementlari (Controls) ning **Modern>>Ring&Enum** ichki palitrasida joylashgan. Ular asosan o'zaro istisno qiymatlarni tanlash uchun muhim hisoblanadi. Masalan: ishlash rejimlari, hisoblash funktsiyalari va hokazolar.



1.5.5-rasm. Aylanma ro'yxat elementlari.

Aylanma ro'yxatni shakllantirish uchun ob'ektning yordamchi menyusidan **Edit Items** punktini tanlash kerak.

Mantiqiy elementlar

Mantiqiy qiymatlar deganda, biz "on" va "off" qiymatlarni tushunishimiz kerak. Mantiiy qiymatlar ikki qiymatdan birini qabul qiladi: **ROST(True)** yoki **YOLG'ON (False)**. *LabVIEW* bizga turli ko'rinishdagi ko'plab o'chirib/yoquvchi elementlarni taklif etadi. Ular **Modern>>Boolean** va **Classic>>Classic Boolean** ichki palitralarida mavjud. Mantiiy elementlar boshqarish elementi yoki indikator bo'lishi mumkin (masalan: o'chirib/yoquvchi element boshqarish elementi sifatida xizmat etsa, svetodiodlar esa indikator sifatida xizmat qiladi).

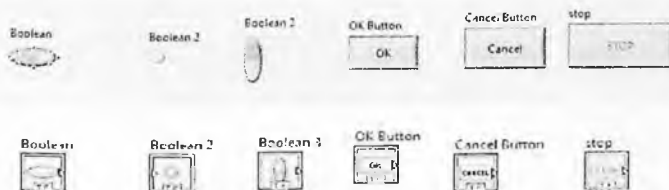


1.5.6-rasm. Modern>>Boolean ichki palitrasi elementlari

1.5.7-rasm. Classic>>Classic Boolean ichki palitrasi elementlari

Mantiqiy qiymatlar deganda, biz "on" va "off" qiymatlarni tushunishimiz kerak. Mantiqiy qiymatlar ikki qiymatdan birini qabul qiladi: ROST(True) yoki YOLG'ON (False). LabVIEW bizga turli ko'rinishdagi ko'plab o'chirib/yoquvchi elementlarni taklif etadi. Ular **Modern>>Boolean** va **Classic>>Classic Boolean** ichki palitralarida mavjud. Mantiqiy elementlar boshqarish elementi yoki indikator bo'lishi mumkin (masalan: o'chirib/yoquvchi element boshqarish elementi sifatida xizmat etsa, svetodiodlar esa indikator sifatida xizmat qiladi).

Mantiqiy terminallar diagrammalar blokida yashil rangda bo'yalgan bo'ladi va mos ravishda "TF" harflari bilan belgilangan.



1.5.7-rasm. Mantiqiy boshqarish elementlari va indikatorlarning ikonkalari(quyi qator) va ularning ko'rinishlari(yuqori qator)

Matnli ma'lumotlar

Matnli boshqarish elementlari va indikatorlar matnli ma'lumotlar bilan ishlashga mo'ljallangan. Matnli ma'lumotlar harakatlanadigan terminallar va o'tkazgichlar binafsha rangda bo'yalgan bo'ladi. Terminallar esa "abc" harflari bilan diagrammalar blokida belgilangan bo'ladi. Bu elementlarni siz **Modern>>String&Path** va **Classic>>Classic String&Path** ichki palitralaridan topishingiz mumkin.



1.5.8-rasm. Modern>>String&Path ichki palitrasi elementlari

1.5.9-rasm. Classic>>Classic String&Path ichki palitrasi elementlari

String&Path ichki palitrasi elementlari

Matnli elementlarning yordamchi menyusining quyidagi maxsus punktlaridan foydalanish mumkin:

- ✓ Enable Wrapping – agar matn elementi kengligidan uzun bo'lsa, uni keyingi qatorga o'tkazish;

- ✓ Visible Items>>Vertical Scrollbar – vertical siljitgichni ko'rsatish yoki yashirish;

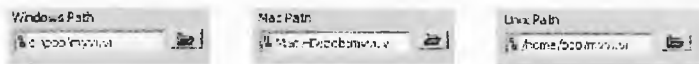
- ✓ Visible Items>>Horizontal Scrollbar – gorizontal siljitgichni ko'rsatish yoki yashirish;

- ✓ Limit to Single Line – bir necha qator kiritish rejimini o'chirish;

✓Update Value While Typing – diagramma blokida o'qiladigan qiymat foydalanuvchi har bir belgini kiritish paytida olinadi.

Faylga yo'l

LabVIEW da fayl va papkalarni manzilini ko'rsatish uchun maxsus boshqarish elementlari va indikatorlar mavjud:paths. Ularning terminallari diagrammalar blokida havorang-yashil rangda bo'yalgan bo'ladi.



1.5.10-rasm. Turli OT larda manzillarni ask etish elementlari.

Amaliy mashg'ulotni bajarish tartibi

1.Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajaring va natijalarni tahlil qiling.

2.Yaratilgan virtual jihozni namoyish eting va uning imkoniyatlarini tushuntirib bering.

AMALIY MASHG'ULOT №3

Dasturni otladkalash

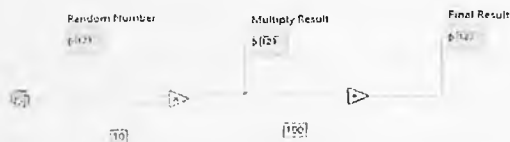
Ishdan maqsad: Dasturni sozlash usullari bilan tanishish va ularni o'rganish.

Ushbu amaliy mashg'ulotda siz VU da ishlamaydigan nosozliklarni topish va bu nosozliklarni bartaraf etishni, yaratilgan dasturni sozlash va uning ishlashini taminlashni o'rganasiz. Hamda dastur xatoliklarini topish boshqa instrumentlarni, shu jumladan yoritishni ishlatish paytida, qadamba-qadam harakatlashtirish rejimi va "sinovchi"larni (пробник) ishlatishni amaliyotda qo'llashni o'rganasiz.

1. Debug Exercise.vi VU ni oching.

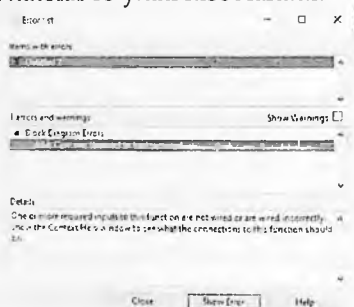
2. Diagrammalar blokida o'ting (1.6.1-rasm). Dasturni ishga tushurish ko'rsatgichi ishchi holatda emasligiga e'tibor bering.

Ishga tushurish tugmasi nima uchun bunday holatda ekanligini aniqlashingiz lozim va VU ni ishga tushurish tugmasini ishchi holatga keltirish lozim.



1.6.1-rasm. Tahrirlash uchun VU diagramma bloki

3. Sichqoncha bilan siniq yo'naltirgich(strelka) tugmasini tanlang. Paydo bo'lgan oyna bu "Xatolar ro'yxati" (1.6.2-rasm) VU dagi xatolar ro'yxati hisoblanadi.



1.6.2-rasm. VU ishlashiga yo'l qo'ymaydigan xatolar ro'yxati.

4. *Add: contains unwired or bad terminal* xatoligini belgilab oling.

5. Yo'qolgan ma'lumotlar o'tkazgichini yarating. Shundan keyin "Ishga tushirish" tugmasi ishlaydigan holatda bo'ladi. Agarda bunday bo'lmasa, nosoz o'tkazgichni o'chirish opsiyasini ishlating.

6. Yana yuzadagi panelga o'ting va VU ni bir necha marotaba ishga tushiring.

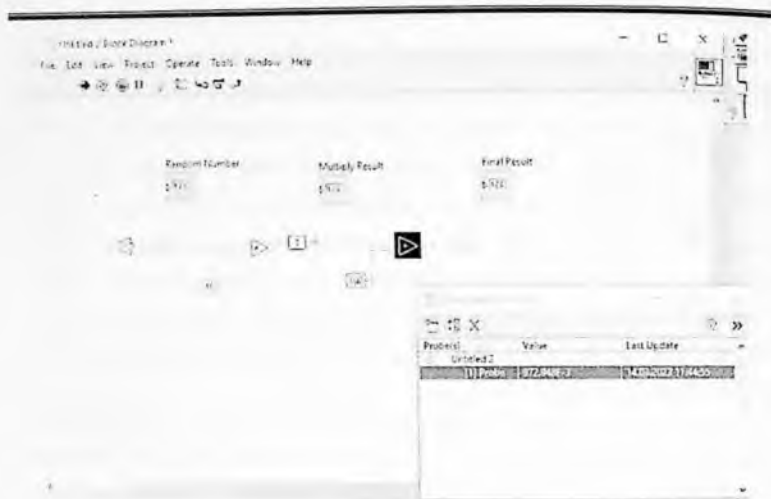
7. Yuza panelini (Oyna menyusida o'rnatish (title) buyrug'ini ishlatgan holda) va diagramma bloki va yuzadagi panelni shunday o'rnatishki, ularni bir vaqtda ko'rish imkoniyati bo'lsin.

8. Har gal ma'lumotlar tuguni bajarilishini xoxlaganingizda  ichidan qadamma-qadam tugmasini sichqoncha bilan tanlang (yoki  ichidan qadamma-qadam tugmasiga sichqoncha bilan tanlang (diagramma bloki tarkibini tugatishni bajarish uchun). Dastur qadamba-qadam bajarilishi hisobida yuzadagi panelida paydo bo'ladigan ma'lumotlarga e'tibor qarating. Avval tasodifiy sonlar generatsiyalanadi,

undan so'ng uni 10 ga ko'paytiriladi va nihoyat VU natijaga 100 ni qo'shadi. Diagramma bloki terminalida yangi ma'lumotlar kirganida yuzadagi panelda display elementlaridan har biri qanday yangilanishiga e'tibor qarating. "Pause" tugmasini tanlab, qadamba-qadam rejimidan chiqish va VU ishini tugallash mumkin.

9. Endi esa "sinovchilar"ni (пробник) ishga soling, ma'lumotlar o'tkazgichining istalgan segmentiga sichqonchani o'ng tugmasini tanlagan holda va "sinovchi"(пробник) opsiyasini tanlagan holda ularni o'rnatish mumkin.

10. Ma'lumotlarni qanday namoyish etishini, mos o'tkazgich orqali o'tishini 1.6.3-rasmda ko'rsatilgandek yana bir marta dasturni qadamlari bo'yicha bajaring va belgilang.



1.6.3-rasm. VU "Debugged Exercise.vi" diagramma blokini qadamba-qadam bajarish

11.Yoritish tugmasini tanlab, dasturni bajarishni yoritishni o'chiring.

12.Fayl menyusidan "Qanday saqlash"(Save As) opsiyasini tanlab o'zingizning papkangizni yaratib, uning tarkibiga VU ni boshqa nom bilan saqlang. Masalan: "Debugged Exercise.vi" deb nomlang.

13. Fayl menyusidan "Yopish"(Close) opsiyasini tanlab, VU ni yoping.

AMALIY MASHG'ULOTNI BAJARISH TARTIBI

1.Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajaring va natijalarni tahlil qiling.

2.Yaratilgan Debugged Exercise.vi virtual jihozini namoyish eting va uning imkoniyatlarini tushuntirib bering.

Tajriba mashg'uloti

**LabVIEW muhitidagi asosiy elementlar va diagrammalar
blokidan foydalanish. Hisoblashlarni bajarish.**

Ishdan maqsad: *LabVIEW* muhitida yuzadagi panelda boshqarish elementlari va indikatorlardan foydalanish. Diagrammalar blokida terminallardan foydalanish.

Ishni bajarish tartibi:

1. Yangi VU na oching va yuzadagi paneldan diagramma-blokiga o'ting.
2. Oynani hajmini shunday tanlangki ham yuzadagi panel ham diaramma- bloki ko'rinsin.
3. Boshqarish elementi paneli palitrasidan yuzadagi panelda sonli boshqarish elementi, qatorli(matnli) boshqarish elementi va mantiqiy indikatori joylashtiring. .

Raqamli boshqarish elementini yaratish uchun Controls >> Modern >> Numeric ichki palitrasidan Numeric Control ni tanlang. Xuddi shunday boshqa elementlarni ham yarating.



1.7.1-rasm. Yuzadagi panelda sonli boshqarish elementini joylashtirish

Numeric

0

String

OK Button

OK

1.7.2-rasm. Yuzadagi panelda sonli, matnli va mantiqiy boshqarish elementlarini joylashtirilgandagi ko'rinishi

4. Uskunlar panelidan (Tools) "*strelka*" uskunasi tanlang yuzadagi paneldan ixtiyoriy ob'yektni tanlab <Delete> tugmasini tanlang. Siz tanlangan ob'yektni shunday usulda o'chirishingiz mumkin. Qolgan ob'yektlarni ham shunday o'chiring va toza yuzadagi panelga ega bo'ling.

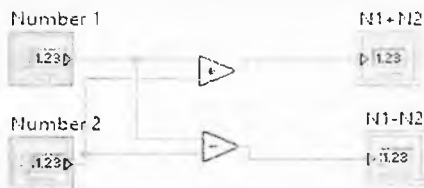
5. Yuzadagi panelga *Controls >> Modern >> Numeric* ichki palitrasidan sonli boshqarish elementini joylashtiring. Lekin sichqonchani chap tugmasini tanlamang, shunda siz boshqarish elementi tepasida kichik oynani ko'rasiz. Number 1 deb kiriting va uskunalar panelida *Ввод (Enter)* tugmasini tanlang. Xuddi shunday Number 2 boshqarish elementi hamda $N1+N2$ va $N1-N2$ sarlavhali indikatorlarini ham yarating.

 uskunasi yordamida Number 1 elementi *strelka*siga murojaat eting toki uning qiymati 4.00 bo'lmaguncha, Number 2 ning qiymati esa 3.00 bo'lsin.

6. Diagramma-blokiga o'ting. Diagramma-blokiga Function palitrasining Numeric ichki palitrasidan qo'shish (Add) funksiyasini joylashtiring. Shu ishni takrorlab Ayirish (Subtract) funksiyasini ham joylashtiring.

7. Qo'shish funktsiya yordamchi menyusini chaqiring va Видимые элементы >> Терминалы (Visible Items->Terminals) opsiyasini tanlang.

8. Siljitish uskunasi yordamida ba'zi terminallarni joyini o'zgartirishingiz mumkin. So'ngra bog'lash uskunasi yordamida terminallarni ulang. Bu uchun uskunalar palitrasidan kerakli uskunani tanlang. So'ngra bog'lash chizig'ini chizish uchun birinchi Number 1 terminalini tanlang, so'ng qo'shish funktsiyasi terminalini mos kirish qismini tanlang. Ekranda sarg'ish rangli uzluksiz chiziq paydo bo'ladi. Agar siz xatoga yo'l qo'yg'an bo'lsangiz qora punktir chiziq ekranda paydo bo'ladi.



1.7.3-rasm. Tajriba virtual uskunasi diagramma bloki

9. 8-punktdagi ishlarni ayirish amali uchun ham amalga oshiring.

10. VU ni ishga tushuring. N1+N2 indikator 7.00 qiymatini ko'rsatadi. N1- N2 indikator esa 1.00 qiymatini ko'rsatishi zarur. Siz kirish qiymatlarini o'zgartirib VU ni qaytib ishga tushurishingiz mumkin.

Bajarish uchun variantlar:

Variant №	Topshiriq	Variant №	Topshiriq	Variant №	Topshiriq
1	$N1+N2-N1$	6	$N1/N2*N1$	11	$N1-N2*N3$
2	$N1-N2+N1$	7	$N1*N2/N1$	12	$N1/N2*N3$
3	$N1+N2*N1$	8	$N1/N2+N1$	13	$N1+N2/N3$
4	$N1*N2-N1$	9	$N1/N2-N1$	14	$N1*N2*N3$

5	$N1*N2+N1$	10	$N1+N2*N3$	15	$N1/N2/N3$
---	------------	----	------------	----	------------

Tajriba mashg'uloti

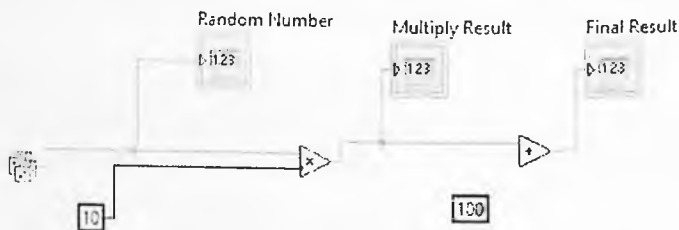
Virtual uskunalarni otladkalash va xatolarni tuzatish.

Ishdan maqsad: Ushbu mashg'ulotda ishlaymaydigan VU lardagi xatoliklarni topish va ularni bartaraf etishni ko'rib o'tamiz. So'ngra siz dasturni otladkalashning boshqa instrumentlarini ham qo'llashni ko'rib o'tamiz, masalan bajarish jarayonidagi yoritish, qadamba-qadam bajarish rejimlari va "Пробник" larni ishlatish.

Tajriba mashg'ulotini bajarish ketma-ketligi:

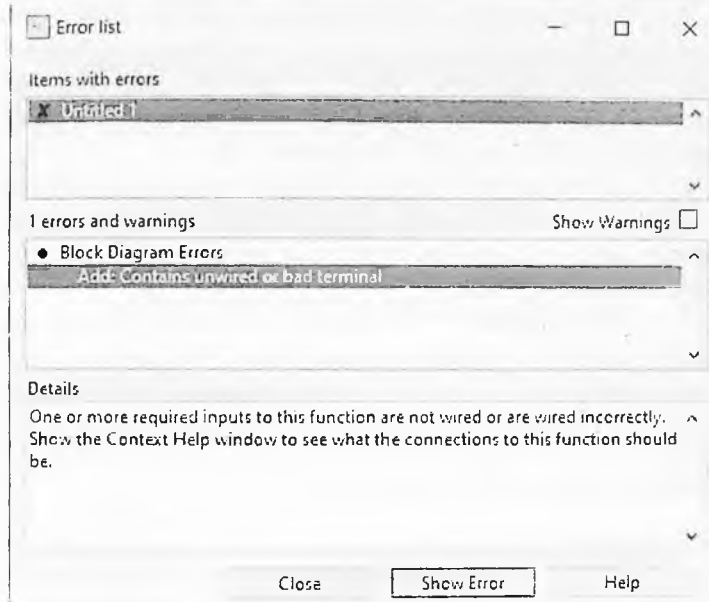
1. "Tajriba mashg'uloti o'qituvchingizdan" Debug Exercise.vi nomli VU olib oching.

2. VU ning Diagramma-blokiga o'ting (1.8.1-rasm). Dasturni ishga tushurish tugmasi ishchi holatda emasligiga e'tibor qarating. Siz nimaga ushbu tugma ishchi holatda emasligi va ushbu xatolikni tuzatib, VU na ishlash holatiga keltirishingiz zarur.



1.8.1-rasm. Tahrirlash uchun VU diagramma bloki.

3. Ishga tushurish buzilgan tugmasiga sichqoncha chap tugmasi bilan tanlang. "Xatolar ro'yxati" oynasida VU xatoliklarini ko'rsatuvchi oyna paydo bo'ladi (1.8.2-rasm).



1.8.2-rasm. VU ani ishga tushurishga xalaqit beruvchi xatoliklar ro'yxati.

4. "Add: contains unwired or bad terminal (Qo'shish funktsiyasi: kerakli terminal mavjud emas)" xatoligini tanlang. Xatolikka sichqoncha chap tugmasi bilan 2 marotaba murojaat eting yoki "Show Error" tugmasini tanlang. LabVIEW diagramma blokida aybdor funktsiyani ajratib ko'rsatadi va shu orqali sizga qidirib topishni osonlashtiradi.

5. Mos o'tkazgichni yarating. Shundan so'ng dasturni ishga tushurish tugmasi ishchi holatda bo'ladi. Agar ishga tushurish tugmasi ishchi holatga kelmasa o'tkazgichlarni o'chirib tashlash opsiyasidan foydalaning.

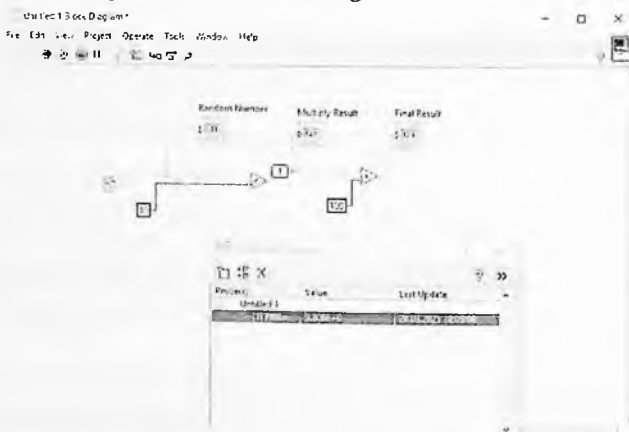
6. Qaytib yuzadagi panelga o'ting va VU ni bir necha marotaba ishga tushuring.

7. Yuzadagi panel va diagramma blokini yonma-yon joylashtiring, bunda ikkala oyna ko'rib turish imkoniyat mavjud bo'lsin. Yoritish tugmasini tanlab uni yoqing va VU ni qadamba-qadam rejimida mos tugmalarni diagramma bloki instrumentlar panelidan tanlab ishga tushuring.

8. Qadamba-qadam bajarish jarayonida yuzadagi paneldagi qiymatlarga e'tibor qarating. VU boshida tasodifiy sonni generatsiyalaydi, so'ngra uni 10,0 ga ko'paytiradi. Nihoyat, VU natijaga 100,0 ni qo'shadi.

9. Endi "Пробник" ni ishlatish vaqti keldi. Diagramma blokidagi ixtiyoriy o'tkazgich segmentiga sichqonchani o'ng tugmasini tanlang va menyudan "Пробник(Probe)" ni tanlang.

10. VU ni yana bir marotaba qadamba-qadam bajartiring va "Пробник" ma'lumotlariz o'tkazgichlardan qanday o'tishini aks etishini quyidagi 1.8.3-rasmda ko'ring.



1.8.3-rasm. Debugged Exercise.vi VU diagramma bloki.

11. Tuzatgan VU ni kompyuter xotirasida saqlang. Uni Debbuged Exercise.vi deb nomlang.

12. Fayl menyusidan Close opsiyasini tanlab VU ni yoping.

Mustqil bajarish uchun vazifa

"Tajriba mashg'uloti o'qituvchingizdan" Mistake VI.vi VU ni olib oching va undagi xatolarni toping hamda tuzating.

Izoh: Tekshirish uchun matematik ifoda:

$$\frac{a-b + \frac{c}{xab} + \frac{c^2-b^3-2}{x^3-d} + 9,23 + \frac{ab}{cx}}{\frac{g-t}{at} + b^4 - t(a-b)(c-d) - 25} - 100$$

I-MODUL BO'YICHA TEST SAVOLLARI

1. Yuzadagi panel an'anaviy dasturiy ta'minotda qanday tushuncha bilan mos keladi?

- A. Interfeys
- B. dastur kodi
- C. Terminallar
- D. Indikatorlar

2. Diagrammalar bloki oynasi (an'anaviy dasturiy ta'minotdagi kod bilan taqqoslaganda) o'zida nimani saqlaydi?

- A. grafik kodi
- B. boshqarish elementlarini
- C. Indikatorlarni
- D. Terminallarni

3. LabVIEW da dasturni bajarilishini boshqarish tamoyili nima hisoblanadi?

- A. ma'lumotlar oqimi (dataflow)
- B. G kod
- C. boshqarish elementlari
- D. Indikatorlar

4. Virtual uskuna nima?

- A. LabVIEW dasturi bo'lib, fizik o'lchash asbobi yoki instrumentini tashqi ko'rinishi va funktsiyasini modellashtiradi
- B. Ishlab chiquvchi xayolidagi mavjud uskuna
- C. Tekis aylanuvchan uskuna
- D. Sonlarni qayta ishlovchi uskuna

5. Virtual uskunada yuzadagi panel nimani bildiradi?

A.fizik uskuna paneli asosida modellashtirilgan virtual uskuna interfaol interfeysi

B.Virtual uskuna yuza qismida joylashgan panel

C.Hozirgi paytda boshqa oynalardan yuqorida bo'lgan oyna

D.Yuzadagi panel, diagramma-bloki, ikonka, VIU

6.Virtual uskunadagi diagrammalar bloki nima?

A.Dasturning kodini saqllovchi virtual uskuna oynasi

B.Hujjatdagi virtual uskunani sxematik ko'rinishini

C.O'tkazgichlar orqali ma'lumotlarni o'tish vaqt diagrammasi

D.Virtual uskuna interfeysi

7.LabVIEWdagi yuzadagi panel an'anaviy

dasturlashdagi.....ga mos keladi.

A.Foydalanuvchi interfeysi

B.Qism dastur

C.Dastur

D.Datur kodi

8.LabVIEWdagi VU an'anaviy dasturlashdagi.....ga mos keladi.

A.Dastur

B.Qism dastur

C.Foydalanuvchi interfeysi

D.Datur kodi

9.LabVIEWdagi VIU an'anaviy dasturlashdagi.....ga mos keladi.

A.Qism dastur

B.Dastur

C.Foydalanuvchi interfeysi

D.Datur kodi

10.LabVIEWdagi diagramma bloki an'anaviy dasturlashdagi.....ga mos keladi.

A.Datur kodi

B.Qism dastur

C.Dastur

D.Foydalanuvchi interfeysi

11.LabVIEW da yuzadagi panel nima?

A.virtual uskunaning interfaol foydalanadigan interfeysi

B.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

C.foydalanuvchi kiritadigan ma'lumotlar

D.foydalanuvchiga chiqariladigan ma'lumotlar

12.LabVIEW da diagrammalar-bloki nima?

A.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

B.virtual uskunaning interfaol foydalanadigan interfeysi

C.foydalanuvchi kiritadigan ma'lumotlar

D.foydalanuvchiga chiqariladigan ma'lumotlar

13.Yuzadagi panel asosan qanday elementlardan iborat bo'ladi?

A.boshqarish elementlari va indikatorlar

B.faqat boshqarish elementlari

C.faqat indikatorlar

D.dastur kodi

14.Virtual uskunalarda boshqarish elementi nima?

A.ma'lumotlar manbaasi

B.ma'lumotlarni qabul qiluvchilar

C.dastur kodi

D.plastik deformatsiyalash sinfi

15.Virtual uskunalarda indikator elementi nima?

A.ma'lumotlarni qabul qiluvchilar

B.ma'lumotlar manbaasi

C.dastur kodi

D.plastik deformatsiyalash sinfi

16.Virtual uskunaning asosiy to'rt komponentini toping?

A.Yuzadagi panel, diagramma-bloki, ikonka, bog'lash paneli.

B.Yuzadagi panel, diagramma-bloki, elementlar palitrasi, funktsiyalar palitrasi.

C.Instrumentlar palitrasi, elementlar palitrasi, funktsiyalar palitrasi, ikonka.

D.Yuzadagi panel, diagramma-bloki, ikonka, VIU

17.Virtual uskunada yuzadagi panel nimani bildiradi?

A.fizik uskuna paneli asosida modellastirilgan virtual uskuna interfaol interfeysi

B.Virtual uskuna yuza qismida joylashgan panel

C.Hozirgi paytda boshqa oynalardan yuqorida bo'lgan oyna

D.Yuzadagi panel, diagramma-bloki, ikonka, VIU

18.LabVIEWdagi ma'lumotlar bog'lami an'anaviy dasturlashdagiga mos keladi.

A.Operatorlar, funktsiyalar va qismdasturlar

B.Qism dastur

C.Foydalanuvchi interfeysi

D.Datur kodi

19.LabVIEWda to'g'ri hisoblangan jumlani ko'rsating?

A.Ma'lumotlar manbasi faqat bitta, qabul qiluvchi esa bir necha bo'lishi mumkin

B.Ma'lumotlar manbasi bir nechta, qabul qiluvchi esa faqat bitta bo'lishi mumkin

C.Boshqarish elementi terminali ramkasi mayin chiziq bilan ajratilgan bo'ladi

D.Indikatorlar terminali chegarasi yo'g'on chiziq bilan belgilangan bo'ladi

20.LabVIEW da "boshqarish elementi" nima?

A.foydalanuvchi kiritadigan ma'lumotlar

B.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

C.virtual uskunaning interfaol foydalanadigan interfeysi

D.foydalanuvchiga chiqariladigan ma'lumotlar

21.LabVIEW da "indikator" nima?

A.foydalanuvchiga chiqariladigan ma'lumotlar

B.foydalanuvchi kiritadigan ma'lumotlar

C.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

D.virtual uskunaning interfaol foydalanadigan interfeysi

22.Odatda diagrammalar-bloki nimalardan iborat bo'ladi?

A.terminallar, bog'lamlar, o'tkazgichlar

B.yuzadagi panel, bosqarish elementlari, indikatorlar

C.bosqarish elementlari, indikatorlar

D.bosqarish elementlari, indikatorlar, terminallar

23.LabVIEW da diagrammalar-blokida bir xil tipdagi boshqarish elementlari va indikatorlar nimasi bilan farqlanadi?

A.Terminallar ramkasi yog'on yoki mayinligi bilan

B.Terminallar ramkasi rangi bilan

C.Terminallar shakli bilan

D.Terminallardagi yozuv bilan

24.LabVIEW da ma'lumotlar bog'lami nima?

A.bu dasturning ixtiyoriy bajariluvchi elementining umumiy nomi

B.virtual uskunaning interfaol foydalanadigan interfeysi

C.foydalanuvchi kiritadigan ma'lumotlar

D.foydalanuvchiga chiqariladigan ma'lumotlar

25.LabVIEW da virtual uskunadagi diagrammalar blokida "Qo'shish" funktsiyasi joylashgan. Bu diagramma blokida qanday tipdagi ob'yekt turiga kiradi?

A.Bog'lamlar

B.Terminallar

C.O'tkazgichlar

D.Mahalliy o'zgaruvchilar

26.LabVIEW dasturini bajarilishini boshqarish tamoyili nima?

A.Ma'lumotlar oqimi

B.Funktsiyalarni ketma-ket bajarish

C.O'tkazgichlar

D.Bog'lamlar

27.LabVIEW da VIU(virtual ichki uskuna) nima?

A.bu VU(virtual uskuna) bo'lib, uni siz boshqa VU dan chaqirasiz

B.virtual uskunaning interfaol foydalanadigan interfeysi

C.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

D.foydalanuvchi kiritadigan ma'lumotlar

28.LabVIEW da "Belgi (Label)" nima?

A.yuzdagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloki

B.bu VU(virtual uskuna) bo'lib, uni siz boshqa VU dan chaqirasiz

C.virtual uskunaning interfaol foydalanadigan interfeysi

D.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

29.LabVIEW da necha xil belgi turi mavjud?

A.3

B.2

C.4

D.5

30.LabVIEW da belgilarning birinchi turi qanday nomlanadi?

A.Shaxsiy

B.O'tkazgichlar

C.Erkin

D.VIU(virtual ichki uskuna) belgilari

31.LabVIEW da belgilarning ikkinchi turi qanday nomlanadi?

A.O'tkazgichlar

B.Shaxsiy

C.Erkin

D.VIU(virtual ichki uskuna) belgilari

32.LabVIEW da belgilarning uchinchi turi qanday nomlanadi?

A.Erkin

B.Shaxsiy

C.O'tkazgichlar

D.VIU(virtual ichki uskuna) belgilari

33.LabVIEW da "Sarlavha" nima?

A.matn hisoblanib, u boshqaruvchi element yoki indikatorni izohlaydi

B.yuzadagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloki

C.virtual uskunaning interfaol foydalanadigan interfeysi

D.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

34.LabVIEW da "O'tkazuvchilar belgisi" nima?

A.diagrammalar blokini izohlash uchun juda qulay vositadir

B.matn hisoblanib, u boshqaruvchi element yoki indikatorni izohlaydi

C.yuzadagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloki

D.virtual uskunaning interfaol foydalanadigan interfeysi

35.LabVIEW da virtual uskunadagi diagrammalar blokida "Qo'shish" funksiyasi joylashgan. Bu diagramma blokida qanday tipdagi ob'yekt turiga kiradi?

A.Bog'lamlar

B.Terminallar

C.O'tkazgichlar

D.Mahalliy o'zgaruvchilar

II-MODUL. TUZILISHLAR YORDAMIDA DASTURLAR BAJARILISHINI BOSHQARISH

§2.1. sikllar tuzilmasining ikki turi

Tuzilish – ma'lumotlar tizimining muhim turi bo'lib, oddiy dasturlash tillari operatorlariga o'xshab virtual tillar operatorlariga o'xshab virtual uskunani bajarilishini boshqaradi.

Bu paragrafda biz *LabVIEW* ni tsikllar bo'yicha ikki asosiy tuzilishi: iteratsiya sonini qayd qilish va shart bo'yicha tsikllar tuzilishlarini ko'rib o'tamiz.

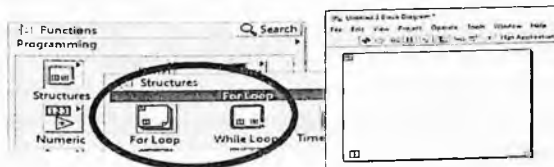
Tsikl tuzilishining ikki turi

LabVIEW jarayonni osonlashtirish uchun ikki asosiy tsikl tuzilmasini taklif qiladi: iteratsiya sonlarini qayd qilishi sikli va virtual uskunalarda takrorlanuvchi operatsiyalarni boshqarish uchun shart bo'yicha tsikl. Iteratsiya sonlarini qayd qilish sikli aniq bir necha bor bajarilsa, shart bo'yicha tsikl aniqlanadigan shartning haqiqiy yoki yolg'onini aniqlanguncha bajariladi.

Iteratsiya sonini qayd qilish sikli

Iteratsiya sonini qayd qilish sikli (for loop) bir qator operatsiyalarni uning ichki chegarasi (dastur osti) bajaradi. Bajariladigan operatsiyalar soni quyidagi usullardan biri orqali aniqlanadi:

- 1) Terminalga berilgan iteratsiya sonlari qiymati bo'yicha (count terminal);
- 2) Tsikl tarkibiga kirgan massivning o'lchami bo'yicha;
- 3) Tsikl muddatdan oldin to'xtatilishi mumkin;

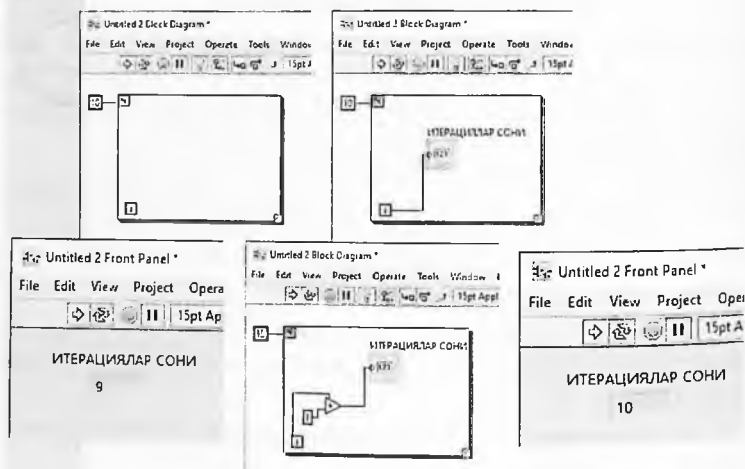


2.1.1-rasm. For Loop tuzilmasi ko'rinishi

Agar iteratsiyalar soni 0 ga teng bo'lsa, unda tsikl biror marotaba ham bajarilmaydi va uning chiqish chegaralarida standart holatda qiymatlar joylashgan bo'ladi.

Iteratsiyalar soni terminali (iteration terminal) tsiklni tugatilgan iteratsiyalar joriy sonini saqlaydi: 0-birinchi iteratsiyada, 1-ikkinchi iteratsiyada va hokazo, toki N-1 bo'lguncha. N-siz ko'rsatgan tsiklni bajarilish soni. Iteratsiya soni aniq bo'lgan tsikl an'anaviy dasturlashda quyidagi kodga ekvivalent:

For i=0 to N-1



2.1.2-rasm. Itertsiyalarni qayd etish sikllili qo'llanilishi

Shart bo'yicha tsikl (While loop) uning ichki chegarasida bajariladi va mantiqiy qiymatgacha (Boolean value) davom ettiriladi. Bunda shart terminalga ulangan tsikldan chiqish holati "Haqiqiy" (Ha, tsikl to'xtatilsin) ko'rinishga kelganda jarayon tugallanadi.

LabVIEW chiqish sharti terminalini har bir iteratsiya tugashi bo'yicha tekshiradi. Agar qiymat "yolg'on" (yo'q, tsikl to'xtatilmasin) bo'lsa keyingi iteratsiya bajariladi.

Shart bo'yicha tsiklning iteratsiya hisoblagich terminali (iteration terminal) aynan iteratsiya sonini qayd qilish siklidagi kabi amalga oshiriladi.

Shart bo'yicha tsikl quyidagi an'anaviy dasturlash tili kodiga ekvivalent bo'ladi:

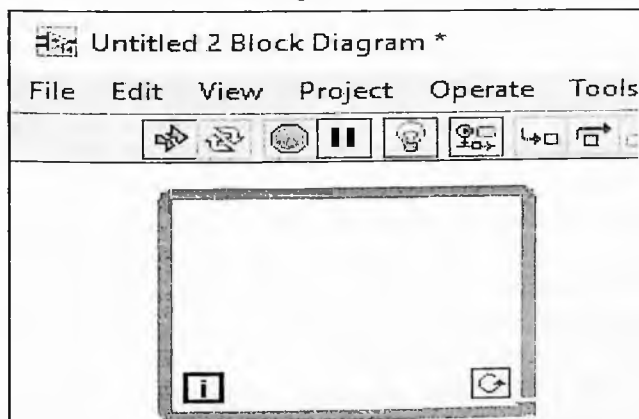
DO Execute subdigram

While condition is FALSE

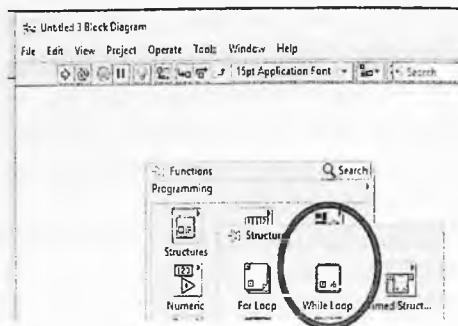
Siz tsikldan chiqish sharti terminali tekshish holatini o'zgartirishingiz ham mumkin. Bunda toki yolg'on (while false) bajarilishni toki haqiqiy (while true) ga o'zgartirish mumkin. Bunga erishish uchun shart terminaliga sichqoncha o'ng tugmasini tanlab, paydo bo'lgan yordamchi menyudan haqiqiy bo'lsa davom ettirish Продолжить, если Истина (Continue if True) opsiyasini tanlash zarur. Shart bo'yicha tsikl 2.1.3-rasmdagi kabi bo'ladi. Bunda tsikl quyidagi kodga ekvivalent bo'ladi:

Do Execute subdigram

While condition is TRUE



2.1.3-rasm. While Loop siklini Continue if True rejimidagi holati



2.1.4-rasm. While Loop sikli tuzilmasini diagramma blokida joylashtirish

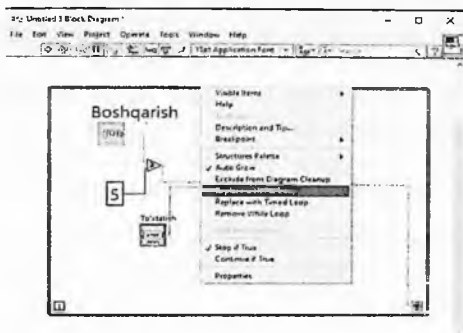
Agar ROST bo'lsa davom etish rejimidagi shart bo'yicha sikl terminalining ko'rinishi.

LabVIEW 7.0 versiyasidan boshlab diagramma blokiga joylashtirilgan tsikl avtomatik ravishda agar "Haqiqiy(True)" holatida bo'lsa, to'xtat rejimi o'rnatilgan bo'ladi. Oldingi versiyalarda shart bo'yicha tsikllarda avtomatik "Haqiqiy(True)" bo'lsa, davom ettirish rejimi o'rnatilgan edi. Misollarni bajarishda har doim tsikldan chiqish shartini va misol yozuviga mosligini tekshiring. Tsikl tuzilishini bir turdan boshqasiga almashtirish imkoniyati ham mavjud. Buning uchun tuzilishning yordamchi menyusidan tsiklning zarur turini boshqasiga "Almashtirish" (*Replace with*) topshirig'i beriladi.

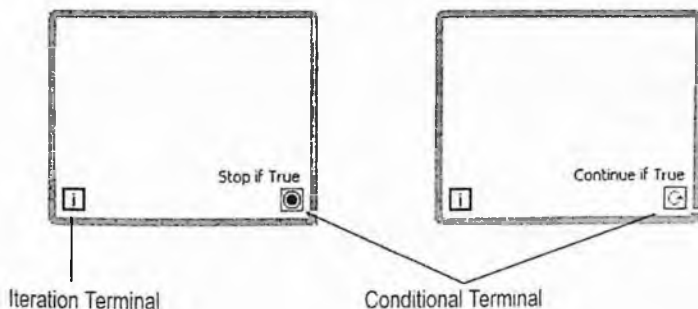
Iteratsiya sonini qayd qilish siklida muddatdan oldin to'xtatish imkoniyatidan foydalanish mumkin. Buning uchun tsikldan chiqish sharti terminalini qo'shimcha kiritish kerak bo'ladi. Bunda tsikl yordamchi menyusidan *conditional terminal* opsiyasini tanlash kerak. Iteratsiya hisoblagich terminali tashqi ko'rinishi o'zgarishi yuz beradi.

Tsikllar o'rtasidagi farqni belgilaymiz: shart bo'yicha tsikl doimo bir marta bo'lsada bajariladi, iteratsiya sonini qayd qilish sikli esa bir marotaba ham bajarilmasligi mumkin.

Tsikl tuzilishini bir turdan boshqasiga almashtirish imkoniyati ham mavjud. Buning uchun tuzilishning yordamchi menyusidan tsiklning zarur turini boshqasiga “Almashtirish” (*Replace with*) topshirig’i beriladi.



2.1.5-rasm. Tsiklni bir turidan ikkinchisiga almashtirish buyrug’i



2.1.6-rasm. Sikldan chiqish sharti terminaliga qo’shimcha kiritish.

Iteratsiya hisoblagich terminali tashqi ko’rinishi o’zgarishi yuz beradi.

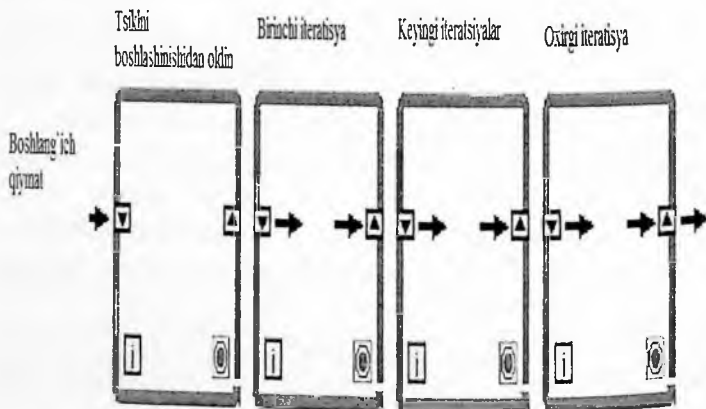
Tsikllar o’rtasidagi farqni belgilaymiz: shart bo’yich tsikl doimo bir marta bo’lsada bajariladi, iteratsiya sonini qayd qilish sikli esa bir marotaba ham bajarilmasligi mumkin.

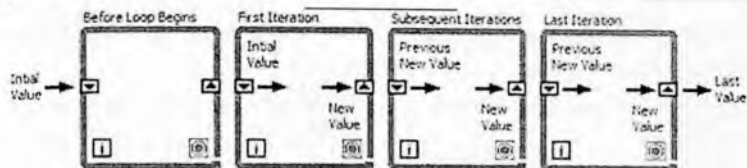
Takrorlash va mustaqil ishlash uchun savollar:

1. Tuzilishlar nima va ular LabVIEW da nima vazifani bajarishadi?
2. LabVIEW da tsikllarning necha xil turi mavjud?
3. Iteratsiya sonini qayd qilish sikli haqida nimani bilasiz?
4. Shart bo'yicha tsikl haqida nimani bilasiz?
5. Iteratsiya sonini qayd qilish sikli va shart bo'yicha tsikllarning faqrlari nimada?

§2.2. Siljuvchan registrlar

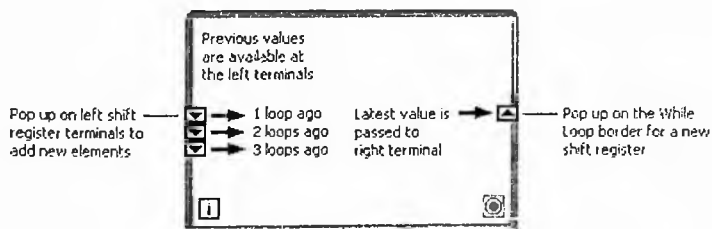
Siljuvchan registrlar shart bo'yicha tsiklda va aniq sonli iteratsiyali tsiklarda qo'llanilib, bir tsikl iteratsiyasidan keyingisiga qiymatni uzatish uchun ishlatiladigan aloxida o'zgaruvchi tipi hisoblanadi (2.2.1-rasm). Ular unikal va LabVIEW tizimida juda kerakli, ya'ni ayniqsa grafik dasturlash muhitida bu siljuvchi registrlar juda zarur. Siljuvchan registrlar tsiklning o'ng yoki chap chegarasida sichqonchani o'ng tugmasi yordamida paydo bo'ladigan yordamchi menyudan "Add Shift Register" buyrug'ini tanlash orqali yaratiladi.





2.2.1-rasm. Siljuvchan registrlar.

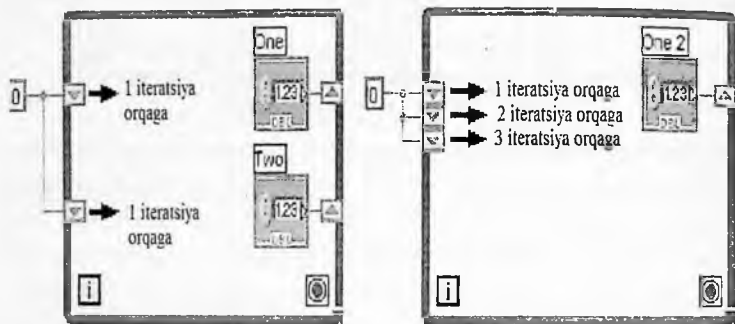
Siz siljuvchan registrni o'zgartirishiz mumkin va quyidagi 2.2.2-rasmda ko'rsatilganidek bir necha oldingi iteratsiyalardagi qiymatga murojaat etib bilasiz. Bu juda qulay imkoniyat turli iteratsiyalarda olingan ma'lumotlar qiymatini o'rtachasini olish imkonini beradi. Bu uchun siljuvchan registr chap terminalini kerakli iteratsiyalar sonigacha cho'zish zarur. Bundan tashqari yordamchi menyudan o'ng va chap tomon uchun bir donadan element ham qo'shib bo'ladi (*Add Element*).



2.2.2-rasm. Uch elementli shart bo'yicha tsiklda siljuvchan registrlar

Bir iteratsiyada ko'plab turli o'zgaruvchilarni saqlovchi turli siljuvchan registrlar sonini yaratish mumkin. Bu uchun tsikllar chegarasiga sichqonchani o'ng tugmasini tanlab yordamchi menyuni chiqarish va kerakli registrnlarni qo'shish mumkin. Chap terminal o'ng terminal bilan bir darajada bo'ladi, shuning uchun ularning birini siljitsangiz ikkinchisi ham avtomatik siljiydi. *Keng tarqalgan xatoliklarga yo'l qo'ymang – ko'p sonli siljuvchan registrlar xotirasidagi ko'p sonli o'zgaruvchilarni bir registr xotirasida joylashgan va oldingi iteratsiyalar natijasida*

olingan yagona o'zgaruvchi bilan adashtirmang. Quyidagi 2.2.3-rasmda ularning farqi keltirilgan:

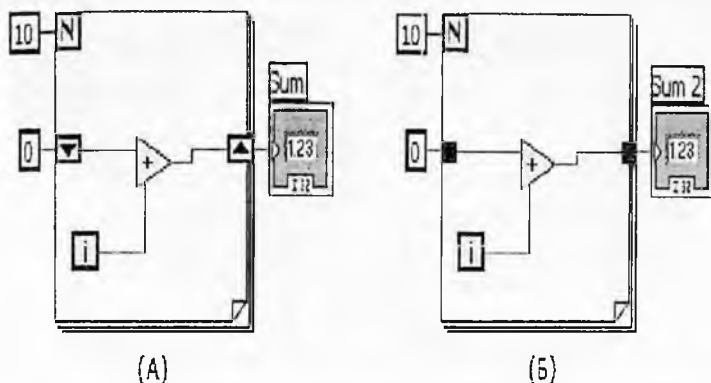


2.2.3-rasm. Ikki turli o'zgaruvchi (chapda) va o'zgaruvchning bir necha qiymati(o'ngda)

Siljuvchan registrlar nima uchun kerak?

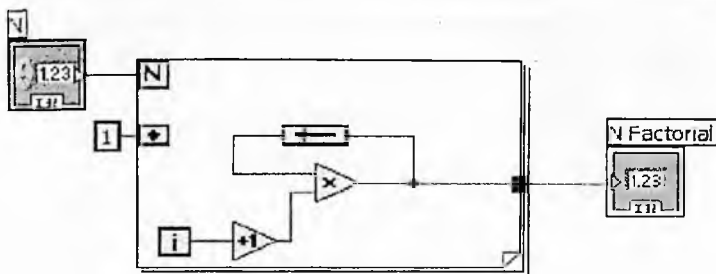
Quyida keltirilgan 2.2.4-rasmga qaraydigan bo'lsak (A) siklida qiymatlarni o'sish yig'indisi iteratsiyasi yaratilgan. Har bir iteratsiyada yangi summa qiymati siljuvchan registrlarda saqlanadi. Tsikl oxirida umumiy yig'indi indikatorga keladi. (B) siklida siljish registri yo'q, shuning uchun iteratsiyalar oralig'ida siz ma'lumotlarni saqlay olmaysiz. Buning o'rniga joriy ^{"i"} qiymatiga faqat nol qiymatini qo'shasiz va faqat oxirgi 9 qiymati tsikldan chiqadi.

Iteratsiyalar sikli ketma-ketligida olingan qiymatlarni o'rtachasini aniqlash uchun nima qilish kerak? Masalan: sizga temperaturani bir daqiqa tezligida o'lchash talab etilsin va so'ngra bu qiymatlarni bir soat oralig'ida o'rtacha qiymatini topish zarur bo'lsin. LabVIEW grafik tuzilishi jihatidan tsiklning bir iteratsiyasida olingan qiymatni keying iteratsiyaga siljuvchan registrlarni ishlatmasdan turib o'tkaza olmaysiz.



2.2.4-rasm. Siljuvchan registrlarni qo'llash. Siljuvchan registrlarni initsializatsiyalash

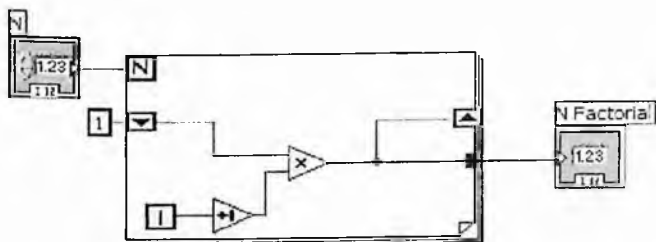
Ko'zda tutilmagan va mumkin bo'lgan xato tsiklar ishini oldini olish uchun har doim siljuvchan registrlarni ishlatishni maslahat beramiz. Aniq qiymatlarni registrga yozish uchun ya'ni siljuvchan regsitrlarni initsializatsiyalash uchun bu qiymatlarni va siljuvchan registr chap terminalini tsikldan tashqaridan o'tkazuchi orqali bog'lang (2.2.5-rasm). Agar siz siljuvchan registrlarni initsializatsiya qilmasangiz, dasturni birinchi marotaba ishga tushurish paytida registr ma'lumotlar tipi uchun standart holat uchun qiymat belgilanadi. Dasturni keyingi marotaba ishga tushurish jarayonida registr uskunani oldin ishga tushirilgan paytdagi qiymatlarini saqlaydi.

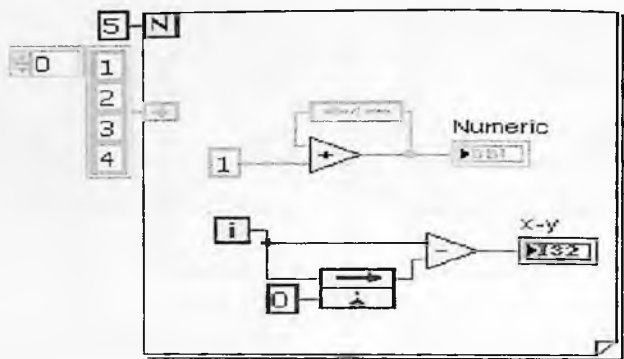


2.2.7-rasm. Teskari aloqa bog'lami

Bunday ulanish ma'lumotlar oqimi qoidalariga to'g'ri kelmaydi: (1) kod bo'limi toki barcha kirish qiymatlariga ega bo'lmasa bajarishni boshlamaydi, (2) esa kod bo'limi ishini tugatgandan so'nggina chiqish qiymatlari paydo bo'ladi. Ammo tsiklda biz chiqish va kirish orasida quyidagi 2.2.7-rasmida keltirganidek teskari aloqa uzelinesi qo'shish imkoniyatiga egamiz.

Odatda *LabVIEW* tsikl ichida kirish va chiqish terminallari o'rtasida bog'lanishni amalga oshirishda teskari aloqa uzeli joylashtiriladi. Bu imkoniyatni *Auto-insert Feed back Nodes in cycles* opsiyasini o'chirish orqali amalga oshiriladi. Bu buyruq *Options >> Block Diagram. Teskari aloqa uzelinesi qanday ishlashini tushunish uchun shuni eslab qolish kerakki bu xuddi siljuvchan registrlarning boshqacha shakli singari*. Quyidagi diagrammalar-bloklari ekvivalent.

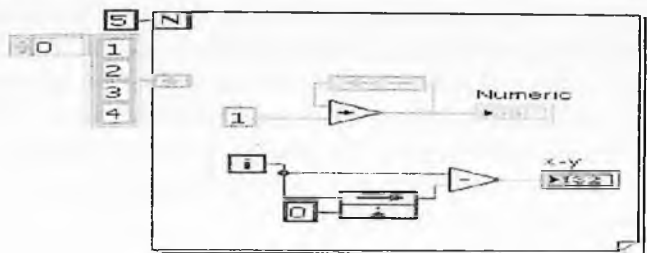




2.2.8-rasm. Siljuvchan registr va teskari aloqa uzelnini ko'rinishlari

Ular orasidagi strelka ma'lumotlar oqimi yo'nalishini ko'rsatadi. Uzelning kirishi – bu siljuvchan registrning o'ng elementi, chiqish–bu chap elementi hisoblanadi. Teskari aloqa uzelnini initsializatsiyalash uchun *Initializer Terminal* maxsus terminali mavjud bo'lib, uni *Initializer Terminal* yordamchi menyudagi opsiyasi orqali ko'rsatish yoki yashirish mumkin. *LabVIEW* siljuvchan registrnlarni osongina teskari aloqa uzellariga almashtirish mumkin: *Replace With* yordamchi menyudan foydalanilgan holda. Siljuvchan registrlar singari teskari aloqa bog'lami ham oldingi iteratsiyalardagi ma'lumotlarga murojaatni ta'minlashi mumkin.

Initsializatsiyalash paytida ishlatiladigan ma'lumotlar tipi qiymatini belgilash zarur: quyidagi 2.2.9-rasmda teskari aloqa yuqori bog'lami kasr sonlar bilan ishlaydi (double tipi), initsializatsiyalash paytida sonlar massivi beriladi. Oldingi qiymatlarni saqlash soni yordamchi menyuda sozlanadi: *Properties >> Configuratsion >> Delay*. Bu qiymat uzelda paydo bo'ladi, lekin siz initsializatsiyalanadigan ma'lumotlarni sonini o'zingiz belgilashingiz kerak.



2.2.9-rasm. Teskari aloqa bog'lamini initsializatsiyalash.

Teskari aloqa quyi uzeli joriy va oldingi iteratsiya qiymatlari orasidagi farqni hisoblash uchun ishlatiladi. Bu holatda u avtomatik joylashtirilmaydi uni Programming >> Structures palitrasidan olib joylashtirish kerak. Uning tashqi ko'rinishini ham sozlash muhim hisoblanadi: yo'nalishini o'zgaritirish va initsializatsiyalanadigan terminalni ichkariga kiritish.

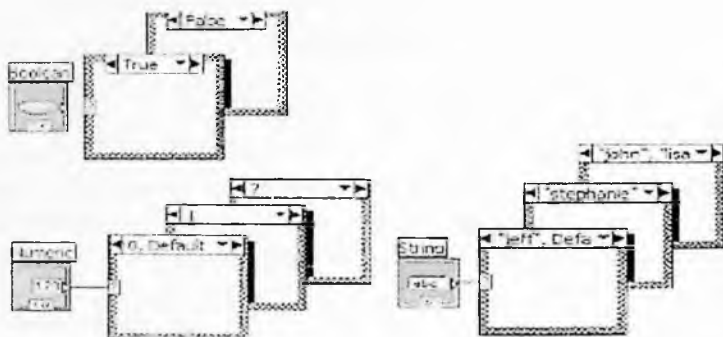
Takrorlash va mustaqil ishlash uchun savollar:

- 1.Siljувchan registrlar nima?
- 2.Siljувchan registrlardan nima maqsadda foydalaniladi?
- 3.Add shift registr va Add element buyruqlarining farqi?
- 4.Siljувchan registr larni initsializatsiyalash nima uchun kerak?
- 5.Teskari aloqa bog'lami nima va u nima uchun kerak?

§ 2.3. Variant tuzilmasi. Shart bo'yicha tsikl

Variant tuzilishi (*Case Structure*) shart mavjud bo'lgan matnni bajarish usuli bo'lib, ya'ni implikasiya operatoriga o'xshaydi(*if-then-else*). Bu tuzilmani siz Functions palitrasining Programming>>Structures ichki palitrasidan topishingiz mumkin. Quyidagi 2.3.1-rasmda keltirilgan variant tuzilmasida ikki yoki undan ko'p ichki dasturga ega yoki variantga ega. Sonli, matnli va mantiqiy qiymatlarga ko'ra siz taqdim etgan variant tuzilmasi selektorlaridagi variantlardan biri bajariladi. Agar variant tuzilishi

selektoriga mantiqiy qiymat ulangan bo'lsa, unda tuzilma ikkita mantiqiy qiymat variantiga ega bo'ladi: ROST va YOLG'ON. Agar selektorga sonli yoki qatorli ma'lumotlar ulangan bo'lsa, unda tuzilma "cheksiz" variantlarga ega bo'ladi. Boshlang'ich ikkita variant mavjud bo'ladi, lekin siz ularning sonini oshirishingiz mumkin. Quyidagi 2.3.1-rasmda ko'rsatilganidek bir variant uchun bir necha qiymatlarni ishlatish mumkin.



2.3.1-rasm. Turli variantlarga ega variant tuzilishi tipi

Bundan tashqari agar tuzilma selektori terminaliga beriladigan qiymat birorta variantga to'g'ri kelmasa siz doim Default variantini tanlashiz mumkin. Bu siz boshqa biron-bir variant haqida o'ylamay umumiy variantni ishlatish niyatida bo'lsangiz bu juda qulay yechim bo'ladi (ixtiyoriy hoat uchun tegishli).

Variant tuzilmasi tanlov terminaliga kelib tushadigan barcha mumkin bo'lgan qiymatlarni inobatga olish kerak, aks holda xatolik yuzaga keladi. "Default" varianti bu shuni amalga oshirishning eng oson usuli lekin yagona usuli emas. Ba'zi bir ma'lumotar tipi uchun variantlar soni cheklangan. Sonli variantlar uchun qiymatlar diapazonini berish mumkin: "3.." degani 3 dan boshlab tanlangan diapazonning maksimal qiymatigacha, "-2.2" degani -2 dan 2 gacha bo'lgan barcha qiymatlar. Variantlarni

qiymatlar diapazoni sifatida belgilash mumkin “..” (masalan, - 12.9). Bundan tashqari LabVIEW uch va undan ko'p butun sonlar ro'yxatini(vergul bilan) diapazon bilan almashtirishi mumkin. Diapazonarni matnlar uchun ham amalga oshirib bo'ladi, lekin natija unchalik aniq bo'lmaydi. Diagramma blokida variant tuzilmasini joylashtirganda birinchi marotaba u mantiqiy formada ifodalanadi. Tuzilmada sonli qiylatlarni ishlatish uchun selektroga sonli ma'lumotlar tipini berish kerak. Variant tuzilmasi ko'p sonli ichki diagrammalarga ega bo'lishi mumkin, lekin bir vaqtning o'zida siz faqat birtasini ko'rishiz mumkin.

Tuzilma yuqori qismi chap yoki o'ng yo'naltirgich selektoriga sichqoncha bilan tanlanganda mos ravishda keyingi va oldingi ichki diagrammalarni ko'rishingiz mumkin. Bundan tashqari tuzilmaning yuqori qismi selektoriga sichqonchani o'ng tugmasi bilan murojaat qilib, paydo bo'lgan menyudan barcha variantlarni ko'rishingiz hamda ulardan birini tanlashingiz mumkin. Variantlarga o'tishning boshqa usuli tuzilmaning chegarasiga sichqoncha o'ng tugmasi bilan murojaat qilish va “Show Case” opsiyasini tanlashdir. Agar siz selektorga kasr sonlarni bersangiz, LabVIEW bu sonni eng yaqin butun songacha (I32) yaxlitlaydi va ko'rsatilgan variantlar ichidan tanlovga o'tadi.

Siz variant tuzilmasi selektori terminalini chap chegaraning ixtiyoriy o'rnida joylashtirishingiz mumkin. Bu terminal doim ulangan bo'lishi kerak. Qachonki siz buni bajargan ekansiz, selektor avtomatik ravishda keladigan ma'lumotlar tipini qamrab oladi. Agar siz selektor chegarasiga keladigan ma'lumotlar tipini sonlidan mantiqiy tipga almashtirsangiz, 0 va 1 variantlari YOLG'ON va ROST variantlariga almashadi. Agar boshqa variantlar bo'lsa (2 dan N gacha) LabVIEW ularni o'chirmaydi, chunki mabodo siz ularni kelajakda o'zgartirishiz uchun saqlab qoladi. Shunday bo'lsa ham tuzilma ish boshlashidan oldin ortiqcha variantlarni o'chirib tashlash kerak. Selektor terminali qatorli ma'lumotlari tipi uchun qo'shtirnoqda keltiriladigan

qatrolar qiymatini aniqlashtirish kerak. Yagona istisno "Default" so'zi bo'lib, u qo'shtirnoqqa olinmaydi. "Default" varianti shuni bildiradiki, Default "Default" qiymatiga ega bo'ladi, standart holat bo'yicha variant emas.

Kiritish/chiqarish terminallarini ulash

Variantlar tuzilmasi barcha kirish terminallarida(kirish nuqtalari va selektorlar) ma'lumotlari barcha variantlar uchun murojaatli bo'ladi. Variantlar bilan ishlaganda albatta tuzilmaga kirish ma'lumotlarini yoki chiqish ma'lumotlarini ishlatish shart emas, lekin agar bir variantdan ma'lumotlar chiqarilsa unda barcha variantlardan ma'lumotlar chiqarilishi kerak. Bir variant tuzilmasidan tashqariga ma'lumotlarni chiqarganda barcha variantlarda o'sha joyning o'zida bo'yalmagan chiqish nuqtalari paydo bo'ladi. Barcha variantlar nuqtasi uchun ma'lumotlarni bermas ekansiz VU ni ishga tushurish belgisi "siniq" bo'lib qolaveradi. Shunga amin bo'lingki, mavjud chiqish nuqtasiga o'tkazgichni to'g'ridan-to'g'ri ulanganligiga amin bo'ling, aks holda siz ularning ko'p sonini yaratishingiz mumkin.

Sizda savol tug'ilishi mumkin: Har bir variant uchun chiqish ma'lumotlarini tuzilmasini ulash nima uchun kerak? Chunki variant tuzilmasi qaysi variant bajarilishidan qat'iy nazar keyingi uzelga ma'lumotlarni uzatish uchun ma'lumotlarni taqdim etishi lozim. LabVIEW kerakli qiymatni tanlashni majburlaydi, xuddi odatiy dasturlash tillari kabi mustaqil ravishda qiymatlarni kiritishni talab etmaydi.

Agar siz chiqish terminalida yordamchi menyudan "Use Default if Unwired" opsiyasini tanlasangiz, unda siz variantlar qiymatlarini ko'rsatmasangiz LabVIEW o'zi kerak deb hisoblangan qiymatlarni belgilaydi. Bu juda qulay bo'lishi mumkin, lekin bu funktsiyani ishlatishda ehtiyot bo'lish kerak, chunki xatoliklar bo'lmasigini oldini olish kerak.

Variantlarni qo'shish

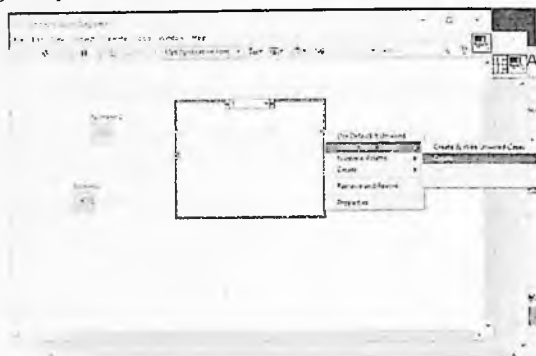
Agar siz tuzilma chegarasida sichqonchani o'ng tugmasini tanlasangiz, paydo bo'lgan menyu *"Add Case After"* va *"Add Case Before"* opsiyalarini taklif etadi. Siz *"Duplicate Case"* opsiyasini joriy variantni nusxalashlariz mumkin. *"Delete this Case"* opsiyasi yordamida joriy variantni o'chirish mumkin.



2.3.2-rasm. Variant tuzilmasida yangi variantlarni qo'shish

LabVIEW da dastur tuzish jarayonida ko'p holatlardan faqat bir holatda kirish qiymatlarini qayta ishlash talab etiladi, boshqa holatlarda ma'lumotlar qayta ishlanmasdan ya'ni o'zgartirishsiz uzatiladi. Bunda xuddi o'sha variant tuzilmasi kirish tunelini bir chiqish tuneliga ulashga majbur bo'lasiz. Yana shunday holatlar bo'ladiki, kop' sonli variantlar tuzilmasi orqali qo'shimcha o'tkazgich kerak bo'ladi. *LabVIEW* ning keyingi versiyalarida *"Linked Input tunnel"* yordamchi menyusi punkti orqali oson bajariladi. *Create & Wire Unwired Cases* va *Create* punktlari esa tanlangan chiqish tunelini ko'rsatilgan kirish tuneli bilan bog'laydi va variantning barcha yangi kadrlarida bu tunnellar bog'lanadi. Birinchi holatda *LabVIEW* tuzilmaning barcha kadrlarini ko'rib chiqadi va agar chiqish tunelida hech qanday ma'lumot kelmasa bog'lanishni yaratadi. Ikkinchi holatda mavjud kadrlarda bog'lanishlar yaratilmaydi. Quyidagi rasmda tunnellar orasida aloqa xuddi shunday bog'langan, shuning uchun chiqish tuneli

kvadrat shakli bo'yalmagan, bu esa ulanmagan kadrlarni mavjudligi haqida xabar beradi.



2.3.3-rasm. Variant tuzilmasida tunellardan foydalanish

Tunellarda bog'lanishlar amalga oshirilgandan so'ng tuzilma chegarasida kichik uchburchak paydo bo'ladi va menyuda Find va Clear punktlari murojaatli bo'ladi. Birinchi punkt topilgan bog'liq kirish tunelini faollashtirsa, ikkinchi punkt esa bog'lanishlarni uzmasdan turib tunellar o'rtasidagi aloqani o'chiradi.

Dialogli oynasi

LabVIEW siz bilan muloqot qilib biladimi? Hozirgi kunda boshqa dasturlar kabi foydalanuvchiga "OK" va "Cancel" singari xabar va bir necha tugmalardan iborat dialog oynasini chiqarish imkoniyati mavjud. Dialog funktsiyalarini Functions palitrasining Programming >> Dialog&User Interface ichki palitrasidan topish mumkin. Uch va undan ko'p tugmalar mavjud bo'lgan dialog oynalari mavjud. Har bir holatda siz dialog oynada chiqadigan xabarni va tugmalar sarlavhalarini ko'rsatishlariz mumkin. Ikki tugmali dialog oynasi tanlanganda tugmani ko'rsatuvchi mantiqiy qiymatni qaytaradi. Uch tugmali dialog oynasi tanlanganda tugmalar xalqa ro'yxat sifatida qiymat qaytaradi. Har bir holatda

dastur foydalanuvchi javobi berilgandan so'ng to'xtaydi. Bu dialog oynalari modal hisoblanadi: ular ochiq ekan siz *LabVIEW* ning boshqa oynasini faollashtirib bilmaysiz.

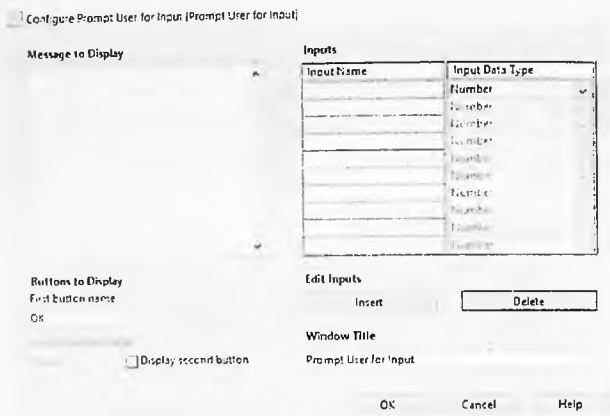
Keling misol ko'rib o'tadigan bo'lsak, quyidagi rasmda keltirganidek foydalanuvchi kritik operatsiya bajarilishini tasdiqladi.



2.3.4-rasm. Dialog oynalardan foydalanish.



2.3.5-rasm. Ekspress-dialog: xabar va tanlov oynalari.

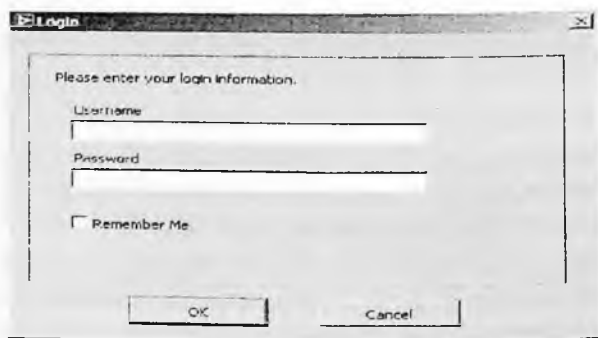


2.3.6-rasm. Sozlanadigan dialog oynalarini sozlash.

Ushbu rasmda ikki tugmali dialogni funktsiyasini ask etilishi ko'rsatib o'tilgan.

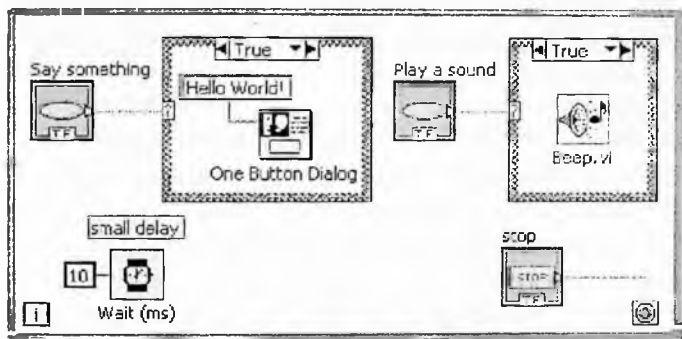
Ba'zan foydalanuvchidan masalan nom va parolni kiritish talab etiladi. Bu uchun Promt User EVU dan foydalaning.

Funktsiyani ishga tushurishda boshqarish elementlariga ega, sozlanadigan, matn xabarli va tugmalardan iborat dialog oynasi paydo bo'ladi.



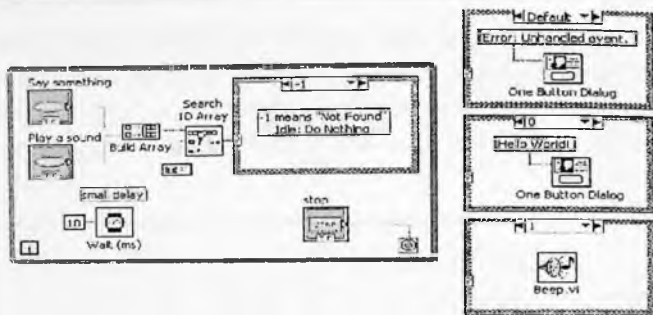
2.3.7-rasm. Foydalanuvchidan ma'lumotlarni kiritishni so'rash dialog oynasi.

Shunday qilib keyingi qadam: bir necha tugma bosilgan holatni qanday qayta ishlash mumkin? Buning oson yo'li: bir nechta variant strukturasini qo'shish. 2.3.8– rasmda ko'rsatilganidek oldin oddiy keyin esa murakkablashib boraveradi.



2.3.8 –rasm. 2 ta amal uchun 2 ta variant tuzilmasini tsiklda ishlatish

Bitta tsiklda bir necha variant bo'lgan holni nazariy jihatdan bilamiz: M ta amal uchun M ta variant strukturasini kerak bo'ladi tabiiy, ammo diagramma blokida joy yetarli bo'lmasa? "Yana bir variant tuzilmasini qo'shamiz" degan javob to'g'ri kelmaydi. Bu muammoni qanday yechimi bor? Bunday paytda 1 ta tuzilmada har bir amal uchun 1 tadan bir necha variant olish kerak. Biz mantiqiy tugmalar massivini shakllantirib, unda nolinch elementni topamiz. Ya'ni, bosilgan tugma nomerini aniqlaymiz va uni variant tuzilmasining tanlash terminaliga uzatamiz va bu bilan masalani yechgan bo'lamiz.



2.3.9-rasm. Tsikl ichida variant tuzilmasidan effektiv foydalanish.

Takrorlash va mustaqil ishlash uchun savollari

1. Variant tuzilmasi nima?
2. Variant tuzilmasida qanday turdagi ma'lumotlarni ulash mumkin?
3. Variant tuzilmasida kiritish/chiqarish terminallarini ulash haqida nimani bilasiz?
4. Variantlarni qo'shish va o'chirish qanday amalaga oshiriladi?
5. Variant tuzilmasida tunellarni avtomatik bog'lash nima uchun kerak?
6. Dialog oynalari qaysi holatda ishlatiladi?

§ 2.4. Massivlar

Massivlar nima?

Odatda skalyar tipdagi ma'lumotlar yagona qiymatga ega bo'ladi. *LabVIEW* da massiv (*array*) boshqa dasturlash tillaridagi kabi bir xil tipdagi elementlar to'plamini tashkil etadi. Massiv bir yoki bir nechta o'lchovli va birta o'lchamga 2^{31} tagacha element bo'lishi mumkin (bu albatta xotira hajmiga ham bog'liq).

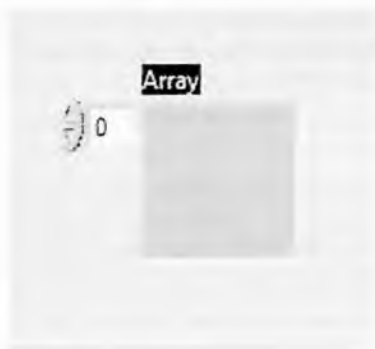
Massiv elementi bo'lib massiv, jadval va grafikadan tashqari har qanday tipdagi qiymatlar bo'lishi mumkin. Massivlarga

ularning indeksleri orqali murojaat qilinadi. Har bir elementning indeksi(index) 0 dan N-1 gacha oraliqdagi son bo'lishi mumkin (bu yerda N – massivdagi elementlar soni).

Bir o'lchovli massiv (1D) ga misol:

Massiv va klaster kabi murakkab tipdagi indikatorlarni yaratish uchun quyidagi amallarni bajarish lozim:

Massiv boshqarish elementlari yoki indikatorlardan tashkil topganmi yo'qmi shuni oldin aniqlash zarur. So'ngra massiv shablonini (array shell) sonli, mantiqiy, manzilli yoki satrli tipdagi ob'yekti bilan birlashtiriladi. Massiv shablони "Boshqarish elementlari" palitrasining Modern >> Array Matrix&Cluster ichki palitrasida joylashtirilgan(2.4.1-rasm).

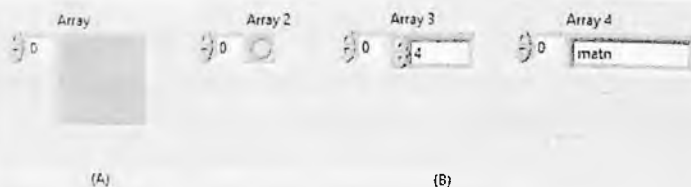


2.4.1-rasm.Massiv shablони

Massiv elementlarini yaratish va ask etish

Massivni yaratish uchun ma'lumotlar ob'yektini yuzadagi panelda joylashtirish zarur. Controls(boshqarish elementlari) panelidan sichqonchani o'ng tugmasi yordamida ushbu ma'lumotlar ob'yektiga ixtiyoriy elementni joylashtirishingiz mumkin. Elementni aks etish oynasi hajmini o'zgartiradi, bu tiplar aniqlanganligini bildiradi (2.4.2-rasm), lekin siz birorta ham element qo'shmasangiz u rangini to'q rangda saqlab turadi.

Shunga e'tibor qaratish zarurki, massivning barcha elementlari boshqarish elementi yoki indikator bo'lishi mumkin.



2.4.2-rasm. Yuzadagi panelda massiv shablони (A) va uchta aniqlangan (tipi) massiv (B) ko'rinishlari keltirilgan

Massiv shablони yuzadagi panelga joylashtirilganda uning terminali diagramma blokida qora rangda bo'lib, bu aniq bo'lmagan tip rangini anglatadi. Terminalda kvadrat qavslar (terminal kompakt ko'rinishda holatida) yoki massiv ikonkasi (terminal ikonka ko'rinishi holatida) ko'rinishlari LabVIEW da massiv tuzilmasini ifodalovchi usul hisoblanadi (2.4.3-rasm). Massivga ma'lumotlar tipi berilganda (massiv shablони ichiga boshqarish elementi yoki indikatorni joylashtirish), diagramma bloki terminali tarkibiga kiritilgan element rangi va yozuvini (qavslar saqlanadi) o'zlashtiradi (2.4.3-rasm). E'tibor qarating: diagramma blokida massiv o'tkazgichlari oddiy o'tkazgichlarga qaraganda qalinroq bo'ladi.

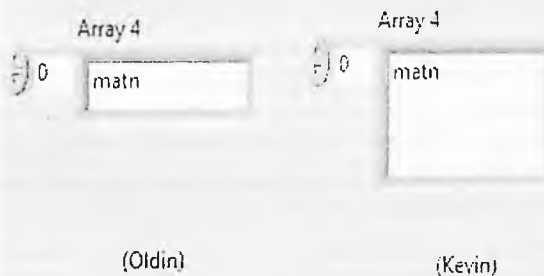
Massiv tipi berilgach siz massiv elementlari qiymatlarini kiritishingiz mumkin. Ma'lumotlarni kiritish uchun matnni kiritish yoki boshqarish vositalaridan foydalanishingiz mumkin. Sonli qiymatlar uchun element ifodalangan oynadagi yo'naltirgichko'rsatkichlari orqali qiymatni oshirish yoki kamaytirish orqali kiritish ham mumkin.

Massivdagi ob'yekt elementlari hajmini o'zgartirish uchun chegaralaridagi markerlardan foydalanish mumkin (har bir tomon o'rtasida va burchaklaridagi markerlar). Massiv chegarasida ham

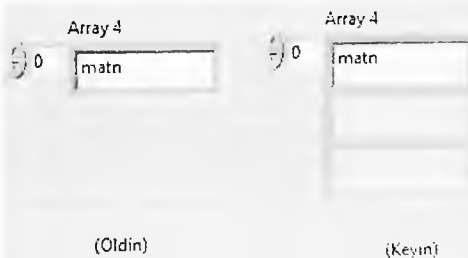
markerlar paydo bo'ladi, lekin ular elementni emas, balki massiv o'lchamlarini o'zgartiradi (2.4.5-rasm). Element markerlari kichikroq bo'lib, ular massiv qobig'ining ichida joylashgan bo'ladi (2.4.4-rasm).



2.4.3-rasm. Massivlarni diagramma blokida akslanishi.



2.4.4-rasm. Massiv elementi hajmini o'zgartirish



2.4.5-rasm. Massiv qobig'i hajmini o'zgartirish

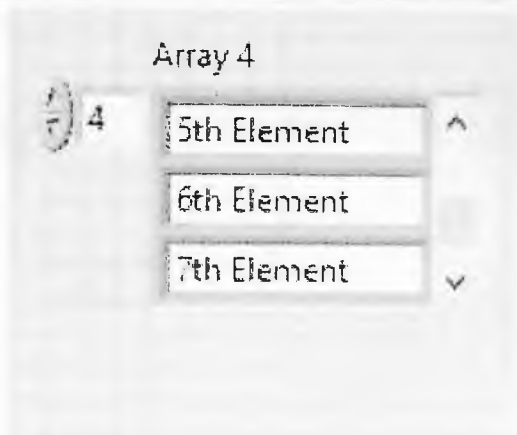
Massiv o'lchamlarini o'zgartirish degani massivdagi elementlar sonini aks etish o'zgartirishni bildiradi. Bir o'lchovli massivni vertikal yoki gorizontal yo'nalishda o'zgartirish mumkin. Bu o'zgartirishlar massiv ma'lumotlariga ta'sir etmaydi va faqat hajmiga ta'sir ko'rsatadi. Formal tarzda bir o'lchamli massiv *LabVIEW* da qanday ask etilishidan qat'iy nazar qator sifatida qaraladi. Massiv o'lchamlarini o'zgartirishda massivga yangi ma'lumotlar qo'shilmaydi.

Diagramma blokida konstantalardan iborat massivni xuddi sonli, mantiqiy va matnli konstantalarni yaratish singari yaratish mumkin. *Function* palitrasining *Massiv(Array)* bo'limining *Doimiy massiv(Array constant)* opsiyasini tanlash konstantalardan iborat massiv yaratish imkonini berib, keyinchalik siz unga konstantalarni joylashtirasiz.

Massivlarni boshqarish elementi (akslantirish, konstantalar) dagi elementlarni o'chirish(tozalash) uchun sichqonchani o'ng tugmasi massiv indikator ustida tanlab(massiv elementi ustida emas) *Data operations >> Empty array* opsiyasini tanlash zarur.

Massiv elementini o'chirish yoki yangi element qo'shish kerak bo'lsa, massiv ob'yektining yordamchi menyusidan *Data operations >> Insert Element Before* yoki *Data operations >> Delete Element* opsiyalarini tanlash zarur.

Massivning birinchi ko'rinayotgan elementining indeksini o'zgartirish massiv bo'ylab harakatlanish imkonini beradi, ammo o'tkazib ko'rish chizig'idan foydalanish maqsadga muvofiq bo'lar edi(2.4.6-rasm). O'tkazib ko'rish chizig'i massiv chegarasida ko'rinish uchun yordamchi menyudan: *Visible Items >> Vertical Scrollbar* yoki *Visible Items >> Horizontal Scrollbar* opsiyalarini tanlash mumkin.



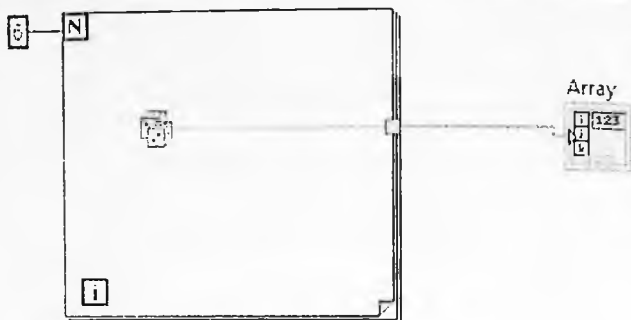
2.4.6-rasm. Massivning o'tkazib ko'rish chizig'i

Avtoindekslashni ishlatish

Iteratsiyalar soni aniq tsikllar va shartga ko'ra tsikllarga kiritilayotgan massiv elementlariga avtomatik ravishda ketma-ket murojaatni amalga oshirishi mumkin yoki bittadan element qo'shgan holda chiqarishda yangi massivni to'plash imkonini beradi. Bu –avtoindekslash deyiladi.

Shuni unutmash kerakki: standart holatga binoan iteratsiyalar soni aniq tsikllarda avtoindekslash yoqilgan holatda bo'lsa, shartga bo'yicha tsikllarda bu avtoindekslash o'chirilgan rejimda bo'ladi.

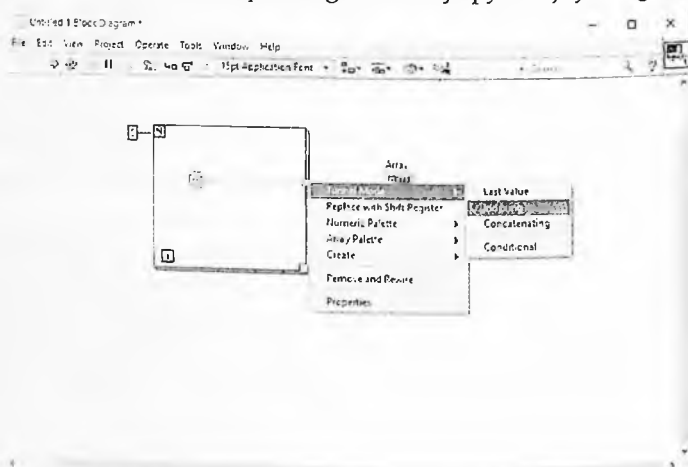
2.4.7-rasmda ko'rsatilganidek iteratsiyalar soni aniq tsikllarda massivni uning chegaralarida avtomatik ravishda yaratadi. Har bir iteratsiya massivning keyingi elementini yaratadi. Tsikl yakunida chiqarilayotgan massiv akslantirish elementida hosil bo'ladi. Tsikl yakunlanmaguncha, massiv elementlariga murojaat qilib bo'lmaydi.



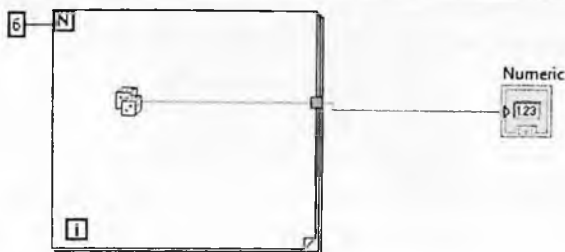
2.4.7-rasm. Iteratsiyalar soni aniq tsikl chegarasida avtoindekslash

Agar iteratsiyalar soni aniq tsikllarda massiv yaratish zaruriyati bo'lmasa va siz aniq sonlarni o'zingiz ko'rsatmoqchi bo'lsangiz, unda tsikl o'ng chegarasidagi [] qavslar yordamchi menyusidan avtoindekslash rejimini o'chirishingiz ham mumkin (2.4.8-rasm).

2.4.9-rasmda tsiklda avtoindekslash o'chirilgan shuning uchun tsikl chiqishida faqat oxirgi taxminiy qiymat joylashgan.

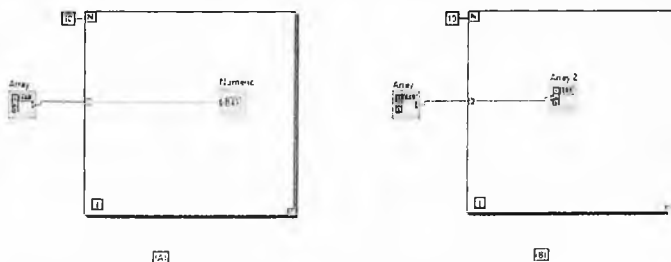


2.4.8-rasm. Avtoindekslash rejimini o'chirish.



2.4.9-rasm. Avtoindekslash rejimi o'chirilgan holatdagi tsiklning ishlashi

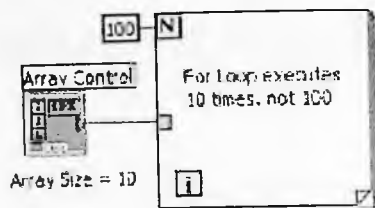
Avtoindekslash tsiklga massivni kiritish jarayonida ham ishlatiladi. Agar avtoindekslash yoqilgan bo'lsa (2.4.10-rasm [A holat]), tsikl har bir iteratsiyada massivning keyingi indeksiga o'tadi. Agar avtoindekslash o'chirilgan bo'lsa (2.4.10-rasm [B holat]), unda massiv to'liqligicha tsiklga kiritiladi.



2.4.10-rasm. Massivni tsiklda kiritishda avtoindekslash yoqilgan rejim(A) va avtoindekslash o'chirilgan rejim(B)

Agar siz iteratsiyalar soni aniq tsikllarda massivlarni kirish jarayonida avtoindekslash rejimini yoqqan bo'lsangiz, LabVIEW tsiklda takrorlashlar sonini massiv hajmiga teng deb oladi. Lekin agar siz o'zaro kesishuvchan qiymatlarni ko'rsatsangiz, masalan: massiv va iteratsiyalar sonini konstanta orqali ko'rsatish yoki ikkita turli hajmdagi massivni kiritish, unda LabVIEW iteratsiyalar soni eng kichik hoatini tanlaydi. Quyidagi 2.4.11-

rasmda iteratsiyalar sonini son emas balki massiv hajmi belgilanishi ko'rsatilgan, chunki massiv hajmi kichik.



2.4.11-rasm. Iteratsiyalar soni 10 ga teng

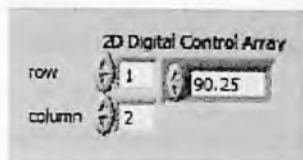
Ikki o'lchamli massivlar

Ikki o'lchovli (2D) massiv elementlari jadval ko'rinishida saqlaydi. Har bir elementning o'rnini 2 ta indeks bilan aniqlanadi: har biri 0 dan boshlanadigan vertikal va gorizontal joylashgan indekslar. 2.4.12-rasmda 4 ta qator va 6 ta ustundan iborat massiv tasvirlangan. Massivda jami 6×4 bo'lib, 24 ta element bor.

Siz "Add Dimension" opsiyasini massiv ob'yekti yordamchi menyusidan tanlash orqali massiv o'lchamlarini o'zgartirishingiz mumkin. 2.4.13-rasmda boshqarishning sonli elementlari ikki o'lchamli massivi ko'rsatilgan. Endi har bir elementni tasvirlash uchun 2 ta indeks mavjud. Siljitish uskunasi orqali ikki o'lchovli massiv elementlarini tasvirlovchi oyna hajmini kengaytirish mumkin. Bu bir vaqtda ko'p elementlarni ko'rish imkonini beradi.

	0	1	2	3	4	5
0						
1						
2						
3						

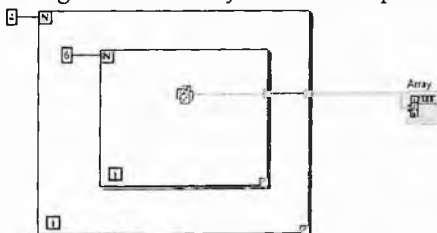
2.4.12-rasm. 6 ustun va 4 qatordan iborat ikki o'lchamli massiv jami 24 elementdan iborat.



2.4.13-rasm. Sonli boshqarish elementlaridan iborat ikki o'lchamli massiv

Ikki o'lchovli massivlarni yaratish

Yuzadagi panel orqali ma'lumotlarni kiritishni istamasangiz, ichma-ich joylashgan 2 ta iteratsiyalar soni aniq tsikllardan foydalanib ikki o'lchovli massivlarni yaratish mumkin. Bunda ichki tsikl ustunlarni hosil qilsa, tashqi tsikl satrlarni hosil qilish uchun bu qatorlarni qo'shib boradi. 2.4.14-rasmda avtoindekslashni qo'llab, tasodifiy sonlarning ikki o'lchovli massivini yaratadigan 2 ta iteratsiyalar soni aniq tsikl ko'rsatilgan.



2.4.14-rasm. Avtoindekslash orqali ikki o'lchovli massivni yaratish

Takrorlash va mustaqil ishlash uchun savollar:

1. Massivlar nima?
2. Yuzadagi panelda massivlarni yaratish jarayoni qanday amalga oshiriladi?
3. Massiv elementlari va ularning turlari haqida nimani bilasiz?
4. Massivlar ustida operatsiyalar qanday amalga oshiriladi?
5. Ikki o'lchamli massivlar qanday yaratiladi?

§ 2.5. Klasterlar. O'zaro almashinuvchan massivlar va klasterlar

Klasterlar

Massiv kabi *klaster (cluster)* lar ham elementlar to'plamidan iborat bo'ladi. Ammo massivlardan farqli ravishda, klasterlar har xil tipdagi elementlarni birlashtira oladi (*sonli, mantiqiy va hokazo*). Bu tushuncha C dasturlash tilidagi struct, C++ yoki Java tillaridagi class elementlari tushunchalariga mos keladi. Klasterlarni telefon kabelidagi simlar bog'lami kabi tasavvur qilish mumkin. Kabeldagi har bir sim klaster elementini ifodalaydi. Diagramma blokida klaster uchun bitta sim mavjud bo'lganligi (klasterda har xil tipdagi ma'lumotlar bo'lishiga qaramasdan) sababli, klasterlar ichki dasturlar uchun kerakli ulanadigan terminallar sonini kamaytiradi (2.5.1-rasm).



2.5.1-rasm. Ma'lumotlarning klasterga birlashtirilish sxemasi

Klaster elementlariga murojaat qilish uchun ularni to'liq bo'lish (unbundling) yoki elementlarning nomlari bo'yicha bo'lib chiqish orqali murojaat etish mumkin. Bo'lish usuli tanlangan funktsiyaga bog'liq hamda o'zining ishlatilish sohasiga ega. Klaster elementlarining bo'linishini telefon kabelidagi har xil rangli simlarning ochib tashlash sifatida tasavvur qilish mumkin. Dinamik ravishda o'lchamlarini o'zgartirishi mumkin bo'lgan massivlardan farqli ravishda klasterlarning o'lchamlari o'zgarmaydi (2.5.2-rasm).

Unbundling Data



2.5.2-rasm. Klasterni bo'lish sxemasi

Klaster terminallarini agar ularning tipi bir xil bo'lsagina birlashtirishingiz mumkin, ya'ni ikkala klasterlar ham bir xil sondagi elementlarga ega bo'lishi va mos elementlarning tipi o'zni bilan bir xil bo'lishi kerak. Berilganlar tiplari bir xil bo'lganda polimorfizm tamoyilini qo'llash mumkin.

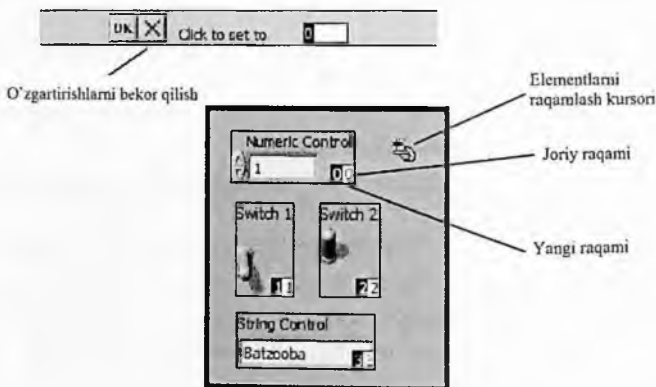
Klasterlar ko'pincha xatolarni qayta ishlash jarayonida uchraydi.

Klasterlar uchun boshqarish va akslantirish elementlarini yaratish Yuzadagi panelda "Massiv, matritsa va klaster" (*Array, Matrix and Cluster*) bo'limidan klaster(*cluster*) shablonini joylashtirish klaster yaratishingiz mumkin. Endi siz klaster ichiga yuzadagi panelning ixtiyoriy ob'yektini joylashtirishingiz mumkin. Massivlar holidagi kabi ob'yektlarni "Boshqarish elementlari(Controls)" palitrasidan olib yoki to'g'ridan-to'g'ri tortib olib kelib, klaster ichiga joylashtirish mumkin. Klaster tarkibi yoki faqat boshqarish elementlari yoki faqat indikatorlardan iborat bo'lishi mumkin. Boshqarish elementlari va indikatorlarni bir klasterga joylashtirish mumkin emas, chunki klasterning o'zi yoki boshqarish elementlaridan yoki indikatorlardan iborat bo'lishi mumkin. Klaster tarkibi boshqarish elementlari yoki indikatorlar bo'lishi birinchi joylashtirilgan elementning tipiga bog'liq.

Klaster elementlarini tartiblash

Shablon doirasida klaster elementlari joylashgan o'rnidan qat'iy nazar mantiqiy ketma-ketlikka ega. Klasterda joylashtirilgan birinchi ob'yekt 0-element hisoblanadi, ikkinchisi esa 1-element va hokazo. Bir element klaster ichidan o'chirilganda tartib avtomatik ravishda o'zgaradi. Agar birta klasterni boshqasi bilan birlashtirish kerak bo'lsa, klasterdagi elementlar tartibini aniq bilish kerak, chunki tartibi va tipi bir xil bo'lishi lozim.

Klaster ichida tartibni o'zgartirish uchun uning chegarasida yordamchi menyuni faollashtirib *"Reorder Controls in Cluster"* opsiyasi tanlanadi. Uskunalar panelida yangi tugmalar paydo bo'ladi hamda klasterning tashqi ko'rinishi o'zgaradi (2.5.3-rasm).



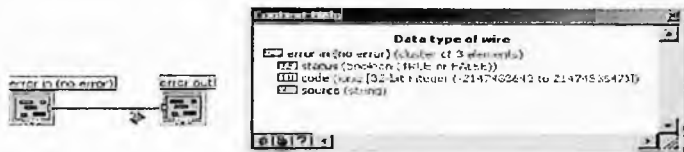
2.5.3-rasm. Klaster elementlarini tartiblash

Elementlarning oq oynalari ularning joriy tartib raqamini, qora oynalar esa yangisini ko'rsatadi. Element ustida sichqoncha ko'rsatkichini 1 marotaba tanlash bilan uning uskunalar panelida aks ettirilgan tartib raqami paydo bo'ladi.

 Revert tugmasi o'zgartirishlarni bekor qilib, oldingi holatga qaytaradi. Agar kerakli tartib o'rnatilgan bo'lsa, uni xotirada saqlab, OK tugmasi orqali yuzadagi panelga qaytish

mumkin. kasterdagi elementlarning tartibini ko'rish uchun, bog'lash uskunasi o'tkazgichga olib borilsa, oynada klasterdagi berilganlarning tiplari va elementlari yuqoridan-pastga qarab tartiblangan holda ko'rinadi (2.5.4-rasm). Help>>Show

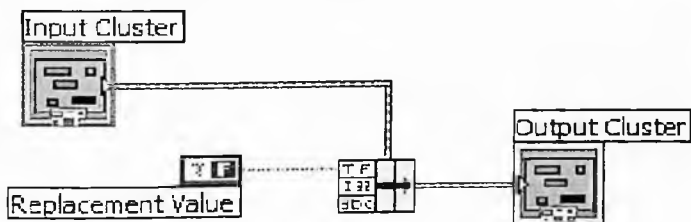
Context Help yoki <Ctrl+H>[Windows], <Meta+H> [Linux] tugmalari orqali yordam oynasi yuklanadi.



2.5.4-rasm. Yordam oynasida klaster elementlarining tartiblanishi

Klaster elementlarini almashtirish.

Klasterdagi biror elementni almashtirish uchun Birlashtirish funksiyasining yuqori qismidan o'rta terminali bilan klaster bog'lanadi, so'ngra funksiyaning o'lchamlari avtomatik tarzda o'zgaradi. Shundan so'ng, funksiyaning mos kirish nuqtalariga yangi qiymatlar beriladi. Klasterdagi elementni o'zgartiriladigan terminallarga ulanishi lozim (2.5.5-rasm).

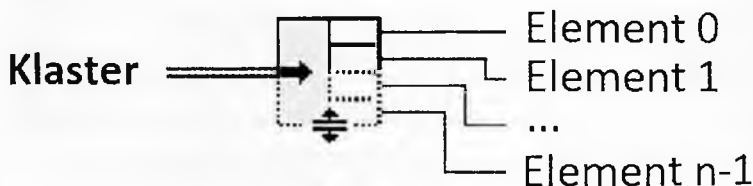


2.5.5-rasm. Birlashtirish bog'lami yordamida klaster elementlarini almashtirish

Klasterlarni bo'lib tashlash.

Programming >> Cluster, Class&Variant ichki palitrasidagi bo'lish (Unbundle) funksiyasi klasterni komponentlarga bo'lib

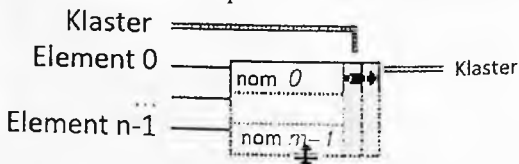
tashlaydi (2.5.6-rasm). Bo'linishdan keyin chiquvchi komponentlar yuqoridan pastga qarab klasterdagi ketma-ketlik bo'yicha joylashgan bo'ladi. Agar komponentlar bir xil tipda bo'lsa, ularni faqrlashning yagona yo'li bu ularning tartib raqamidir. Bu funktsiyani diagramma blokida joylashtirgandan so'ng, uning hajmini kattalashtirish kerakki, chiqish nuqtalari klasterning kirish nuqtalariga teng bo'lsin, aks holda bo'glashda xatolik yuz berishi mumkin.



2.5.6-rasm. Klasterni bo'lish funktsiyasi

Klasterlarni birlashtirish va nom bo'yicha bo'lib tashlash

Ko'pincha butun bir klasterni bo'lish talab etilmaydida lekin bir yoki ikki elementi zarur. Bu uchun nom bo'yicha bo'lish(unbundle by name) va nom bo'yicha birlashtirish (*Bundle by name*) funktsiyalaridan foydalanishimiz mumkin. Nomi bo'yicha birlashtirish funktsiyasi klasterlar bilan ishlash ichki palitrasida joylashgan. (2.5.7-rasm). Bu funktsiya elementlarga nom bo'yicha murojaat etadi tartib raqami bo'yicha emas. Siz bu funktsiya yordamida kerakli elementga nomi bo'yicha murojaat etishingiz mumkin. Bu funktsiya yangi klaster yaratmaydi balkim, klasterdagi bir elementni boshqasi bilan almashtiradi xolos.



2.5.7-rasm. Nom bo'yicha birlashtirish funktsiyasi

Nomi bo'yicha klasterni bo'lish funksiyasi xuddi shu ichki palitrada joylashgan bo'lib, siz ko'rsatgan nomli elementlarni sizga qaytaradi. Siz elementlarning tartib raqami yoki hajmini to'g'rilash haqida o'ylashingiz shart emas(2.5.8-rasm).

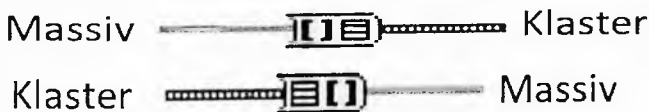
Siz nomi bo'yicha birlashtirish yoki nomi bo'yicha bo'lish funksiyalarini ishlatganda har bir element belgisiga ega bo'lishi kerak. Aks holda siz kerakli elementga murojaat etib bilmaysiz.



2.5.8-rasm. Nom bo'yicha klasterni bo'lish funksiyasi

Massiv va klasterlarni o'zaro almashtirish

Ba'zan klasterlarni massivga va aksincha almashtirish zaruriyati tug'iladi. Bu almashtirishlar bizga qachon kerak bo'ladi? LabVIEW da massivlar bilan ishlash funksiyalari klasterlarga qaraganda ko'p. Masalan, sizda tugmalar klasteri bo'lsa va siz ularni teskari tartibda joylashtirmoqchi bo'lsangiz Reverse 1D Array funksiyasidan foydalanishingiz mumkin. Siz Cluster to Array funksiyasi orqali klasterni massivga va aksincha Array to Cluster funksiyasi yordamida massivni klasterga almashtirishingiz mumkin. Bu funksiyalarni ishlatishda elementlar bir xil tipda bo'lishi muhim (2.5.9-rasm).

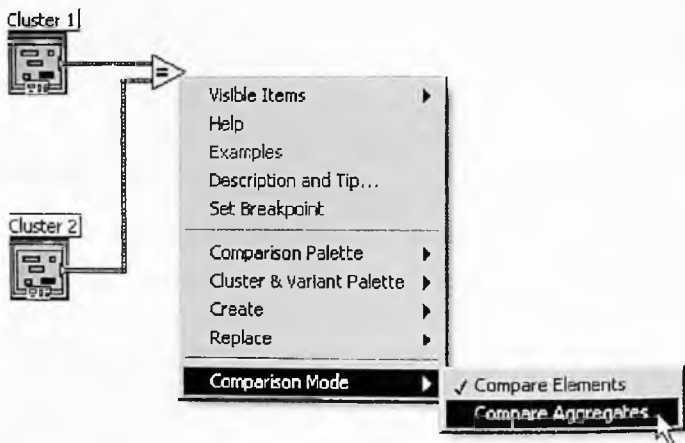


2.5.9-rasm. Massivni klasterga va klasterni massivga aylantiruvchi funksiyalar

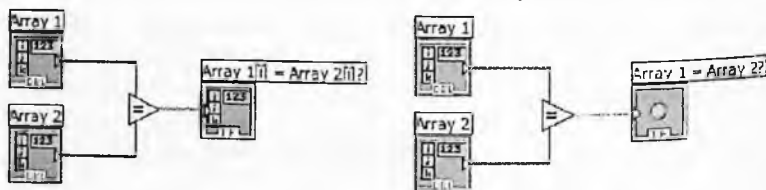
Massiv va klasterlarni taqqoslash

Ba'zi massivlar va klasterlar bilan ishlashdagi taqqoslash operatsiyalarida ikki rejim mavjud: Compare Aggregates(Ob'yektni taqqoslash) va Compare

Elements(Elementlar bo'yicha taqqoslash). Rejimni tanlash yordamchi menyudan amalga oshiriladi(2.5.10-rasm). Ob'yektlarni taqqoslash rejimida funktsiya yagona mantiqiy qiymat qaytarsa, elementli taqqoslashda esa mantiqiy qiymatlar klasteri yoki massivi hosil bo'ladi(2.5.11-rasm).



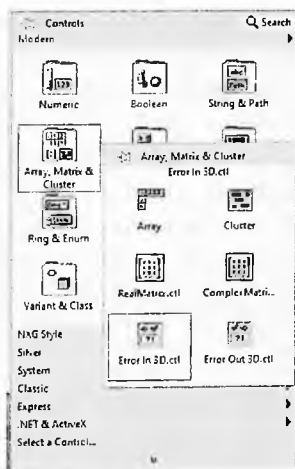
2.5.10-rasm. Tenglikni tekshirish amalida rejimlarni tanlash



2.5.11-rasm. Chap tomonda elementlarni taqqoslash natijasi o'ng tomonda esa ob'yektlarni taqqoslash natijasi

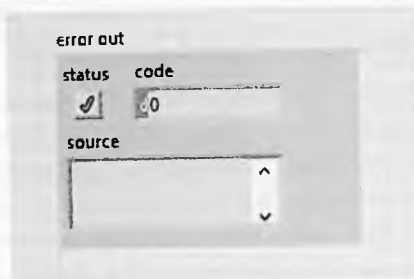
Klaster va xatolarni qayta ishlash.

Modern>>Array, Matrix&Cluster (boshqaruvchi element-Error In va indikator- Error Out) palitrasining maxsus boshqaruvchi elementlari tasvirlangan. Bu-ko'pchilik VP va funktsiyalarning standart kiritish-chiqarish qiymatlari. Xatolar klasteri-bu *LabVIEW* dagi qiymatlarning maxsus tipi (*status*) mantiqiy elementidan, uzun butun kod (I32 code) va manba qatoridan (source) iborat bo'lgan klaster). Undan *LabVIEW* da dasturlar bajarilish jarayonidagi xatolar to'g'risidagi axborotni uzatish uchun foydalaniladi. Xatolar va xatolarni qayta ishlash – bu tabiiy hol. Ularning paydo bo'lish sabablarini aniqlashimiz zarur.



2.5.12-rasm. Modern>>Array, Matrix&Cluster palitrasida boshqaruvchi element-Error In va indikator- Error Out.

Xato o'zi nima? Bu - resurslarning yo'qligi yoki kirish parametrlarining nomuvofiqligi sababli funktsiyaning zarur amalini bajara olmasligidir.



2.5.13-rasm. Xatolar klasteri Berilganlar tipi: xatolar klasteri

Xatolar klasteri quyidagi 3 ta elementdan iborat:

■ Status Boolean (status - mantiqiy elementi) xato mavjudligi to'g'risida signal beradi (ROST) yoki xato yo'qligi to'g'risida (YOLG'ON);

■ code 132 (uzun butun kod) –xatoning kodli nomeri:

■ Musbat kod-xato,

■ Manfiy kod-ogohlantirish,

■ 0-xato yo'q;

■ source string (manba - satrli o'zgaruvchisi) – xatolar tavsifi va ularning paydo bo'lish sabablari.

Xatolar klasterida xatoning tavsifi mavjud bo'ladi.

Xatolar va ularning *LabVIEW* da tahlili.

Standart qoidaga binoan xatolar bilan bog'liq VP ning quyidagi harakatlari

qabul qilingan:

■ Xatolar to'g'risida ma'lumot berildi (status=ROST), tozalashdan boshqa biror amal bajarilmaydi,

■ Fayllarga murojaatlar yopiladi,

■ Qurilma bilan ishlash to'xtatiladi, bog'lanishlar yopiladi,

■ Tizim xavfsiz rejimga o'tkaziladi,

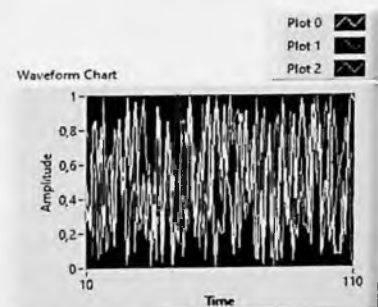
■ VP ning ichidagi xato. Xatolar to'g'risida ma'lumot kiruvchi termanalda qayd etilmagan bo'lsa, chiquvchi xatolar klasteriga yoziladi. Bu holda xato o'zgarishlarsiz uzatiladi.

Takrorlash va mustaqil ishlash uchun savollar:

- 1.Klaster nima?
- 2.Klasterni massivdan faqrqi nimada?
- 3.Klasterlar qanday yaratildi?
- 4.Klasterlar bilan ishlaydigan bog'lamlar qaysilarni bilasiz?
- 5.Klaster elementlarini tartiblash haqida nima bilasiz?
- 6.Massivdan klasterga va aksincha o'tkazish nima uchun kerak?

§ 2.6. LabVIEW vizual aks etish vositalari

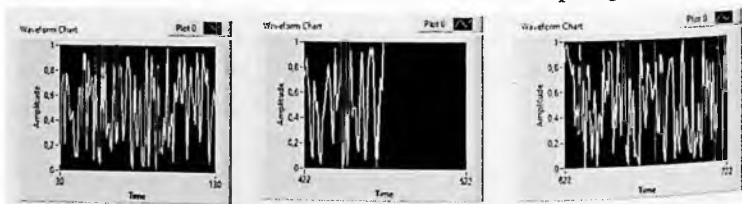
Grafikdagi egri chiziq Y ning X ga bog'liqligini grafik aks etilishi hisoblanadi. Ba'zan Y berilgan qiymatlarni ifodalasa, X vaqtni bildiradi. Waveform Chart grafiklarni aks etish elementi Controls panelining Graph ichki palitrasida joylashgan bo'lib, bir yoki bir necha egri chiziqlarni aks etish imkonini beradi. Bu element odatda tsikllarda keng qo'llaniladi. LabVIEW da otsillogrammada aks etish grafiki ob'yekti turi bir ob'yektdan iborat (*Waveform Chart*) bo'lishiga qaramay, unda 3 rejimdagi yangilash rejimi mavjud. Quyidagi 2.6.1-rasmda ko'p nurli grafiklarni otsillogrammada aks etish grafiki ko'rsatilgan.



2.6.1-rasm. Ko'p nurli grafikni otsillagrammada aks etilishi

Otsillogrammada nurlarni aks etish ob'yekti uch yangilash rejimiga ega: panoramali (stript chart mode), vaqtli (scope chart mode) va markerli vaqt (sweep chart mode), bu uchala rejim 2.6.2-rasmda keltirilgan. Yangilash rejimini o'zgartirish yoki tanlash uchun otsillogrammada sichqonchani o'ng tugmasi bilan murojaat etib, paydo bo'lgan yordamchi menyudan Advanced >> Update Mode opsiyasidan foydalanishingiz mumkin. Agar siz yangilash rejimini ishlash jarayonida o'zgartirmoqchi bo'lsangiz otsillogramma grafigida sichqoncha o'ng tugmasini tanlab, yordamchi menyudan yangilanish rejimini tanlashingiz mumkin.

Panoramali rejimda qog'ozli lenta diagrammasi singari ko'rinish paydo bo'ladi. Vaqtli va markerli vaqt rejimlarida esa, otsillografga o'xshagan teskari yo'nalishdagi ko'rinishga ega. Vaqtli rejimda egri chiziq otsillografning o'ng chegarasiga yetganda avtomatik grafik tozalanadi va chap chegaradan qayta chizish boshlanadi. Markerli vaqt bo'yicha rejimi ham xuddi vaqt bo'yicha rejimiga o'xshab ishlaydi, faqat egri chiziq o'ng chegaraga borganda o'chib ketmaydida: yangi ma'lumotlarni kelib tushishi vertikal chegara-marker bilan belgilanadi. Albatta ularni farqini yaxshiroq sezish uchun ularni ishlatib ko'rish maqsadga muvfiq.

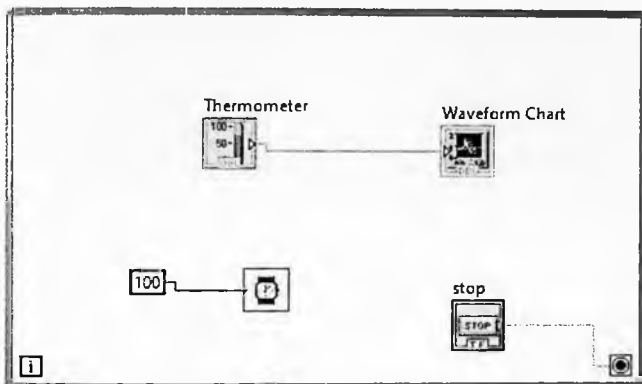


2.6.2-rasm. Otsillogramma grafini yangilash rejimlari

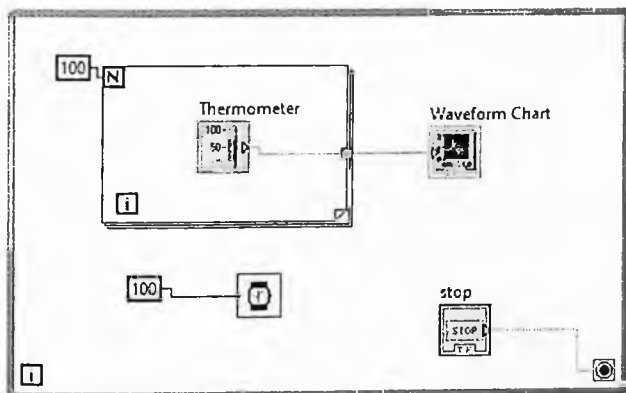
Otsillogrammani qurishda X o'qi nuqtalari doimiy qadam bilan teng taqsimlangan. LabVIEW da biz bunday grafiklarni aks etish uchun funktsiya qiymati va qadamni ko'rsatishimiz yetarli, har bir qadam uchun argument shart emas. Otsillogrammada grafikni aks etishning eng oddiy usuli unga skalyar kattalikni

ulash hisoblanadi (2.6.3-rasm). Bunda har bir qadamda yana bir nuqta chiziladi.

Bundan tashqari skalyar kattalik o'rniga ma'lumotlar massivini ham ko'rsatish mumkin (2.6.4-rasm).



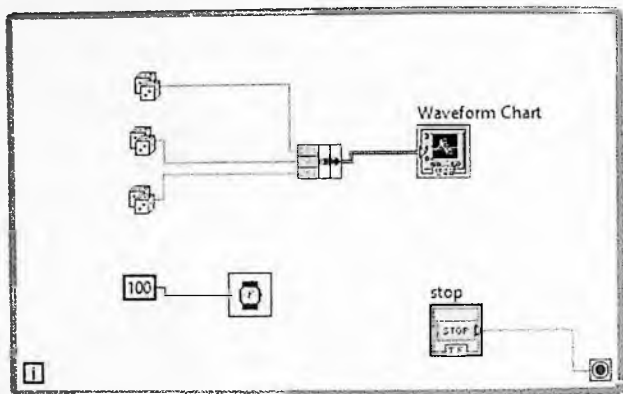
2.6.3-rasm. Har bir qadamda bir nuqta bilan yangilanadi



2.6.4-rasm. Har bir qadamda bir nechta nuqta bilan yangilanadi

Otsillogrammada grafiklarni aks etish ob'ekti bir nechta nurni aks etish imkoniyatiga ega. Ammo diagramma blokida bir

nechta manbaani grafik terminali bilan bog'lay olmasligizni inobatga olib ularni Programming >> Cluster&Variant ichki palitrasidan Birlashtirish (*Boundle*) funktsiyasidan foydalanib klasterga birlashtirib, so'ngra otsillogrammga aks etish grafik terminaliga bog'lash to'g'ri bo'lar edi (2.6.5-rasm).



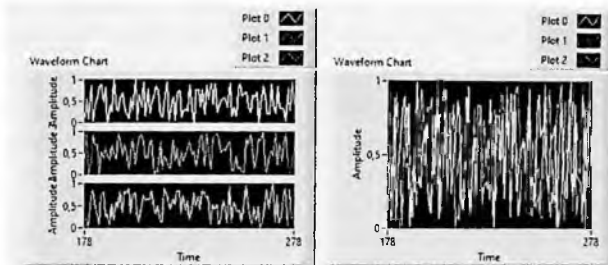
2.6.5-rasm. Otsillogrammada bir nechta nurni aks etish uchun ma'lumotlarni birlashtirish

Otsillogramma to'ri siljitish chizig'iga ega bo'lib, uni ko'rinadigan yoki ko'rinmadigan holatga keltirish mumkin. U eskirgan ma'lumotlarni ko'rish uchun ishlatiladi, ya'ni grafik indikator chegarasidan chiqib ketgan ma'lumotlarni ko'rish uchun ishlatiladi.

Ba'zan grafik indikatoridan oldingi grafik va ma'lumotlarni o'chirish hamda tozalash zarur bo'ladi. Bu uchun otsillogramma yordamchi menyusining Data Operations >> Clear chart opsiyasidan foydalaning.

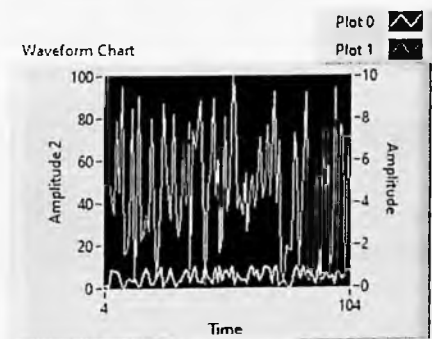
Agar sizning otsillogramma to'ringizda bir nechta nur mavjud bo'lsa, ularni nisbatan yagona Y o'qida aks etish mumkin yoki har bir grafik uchun alohida o'zining Y o'qini ishlatish mumkin. Aks etishning qaysi bir turini tanlash uchun

Stack Plots(aloxida ask etish) yoki Overlay Plots(birgalikda aks etish) opsiyalarini tanlash mumkin. Quyidagi rasmda ularning farqi keltirilgan (2.6.6-rasm).



2.6.6-rasm. Chap tomonda aloxida o'ng tomonda esa birlashtirilgan grafiklarni aks etish formasi

Agar siz bir grafik indikatorida bir nechta egri chiziqlarni birgalikda aks etish rejimida aks etmoqchi bo'lsangiz, lekin ularning Y o'qlari turli shkalalarda bo'lsa, unda nima qilish mumkin? Siz Y o'qi uchun aloxida shkala yaratishingiz mumkin, bu uchun mos o'qqa (masalan Y o'qi) sichqoncha o'ng tugmasini tanlab, Duplicate Scale opsiyasini tanlash mumkin(2.6.7-rasm).

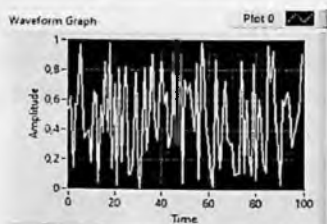


2.6.7-rasm. Ikkilangan shkalali otsillogramma grafigi

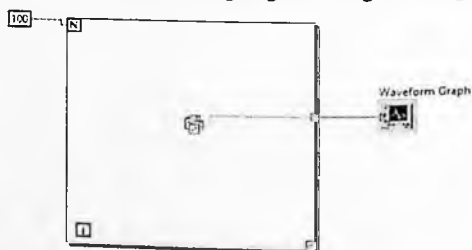
Otsillogramma grafiklari

Otsillogramma to'rlarida grafiklar interfaol ravishda chizilsa, otsillagramma grafiklari ulardan farqli ravishda darhol ma'lumotlarni aks etishmaydi. Ular berilgan qiymatlarga yangilarini qo'shish imkoniyatiga ega emas. LabVIEW grafiklar bilan ishlash uchun bir nechta grafik turlarini taqdim etadi: otsillogramma grafiklari, ikki kordinatali grafiklar, intensive grafiklar, 3 o'lchovli va hokazo.

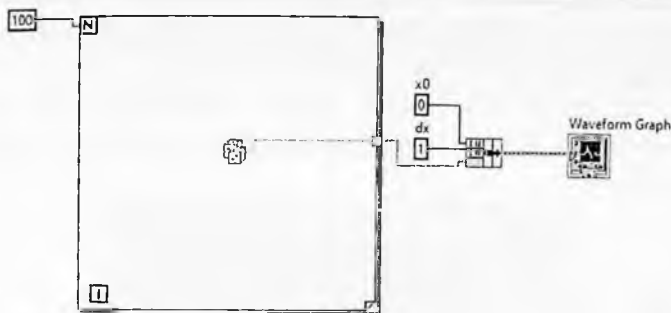
Quyidagi rasmda (2.6.8-rasm) turli uskunalardan iborat grafik aks etilgan bo'lib, otsillogramma grafigi faqat aniq grafiklarni chizib aks etish imkonini beradiki(X o'qiga mos nuqtada yagona Y qiymati), bunda X o'qi teng taqsimlangan hisoblanadi. Otsillogramma grafigi nuqtalar teng taqsimlangan massiv qiymatlarini aks etish uchun ideal uskuna hiboalanadi.



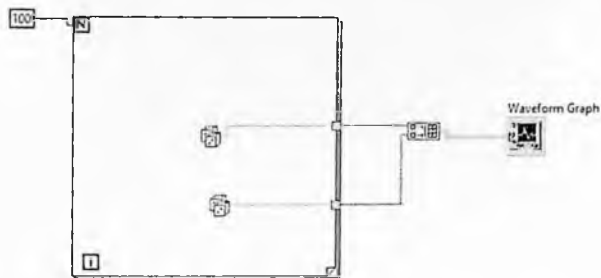
2.6.8-rasm. Turli uskunalarga ega otsillogramma grafigi



2.6.9-rasm. Bir o'lchamli massiv otsillogramma grafigida beriladi va nuqtalar to'plami akslanadi.



2.6.10-rasm. Qiymat parametrlarini ko'rsatishimiz uchun massivdan tashqari X0(boshlang'ich qiymat) va dX(qadamni) ko'rsatishimiz mumkin.



2.6.11-rasm. Otsillogramma grafigida bir nechta egri chiziqlarni aks etish uchun ikki o'lchamli massivni bog'lash zarur.

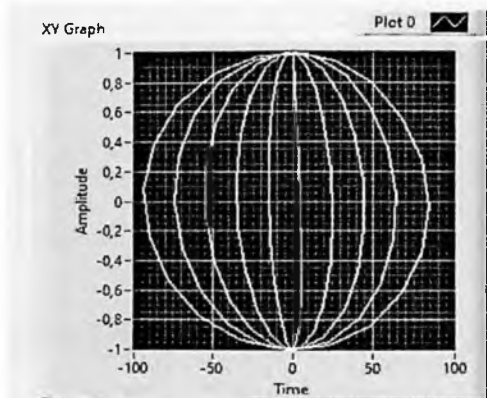
Ikki kordinatali grafiklar

Oldingi rejalarda ko'rib o'tilgan grafiklar otsillogrammalarni yaratish orqali yaratilar edi, ya'ni ularning nuqtalari aniq vaqt intervalidan keyin olinar edi. Ammo agar siz ma'lumotlarni olmasangiz yoki biror-bir matematik funktsiya grafigini chizmoqchi bo'lsangiz (bir X qiymati uchun bir nechta Y qiymati), unda siz (X,Y) kordinatada nuqtalarni aks etishga majbur bo'lasiz.

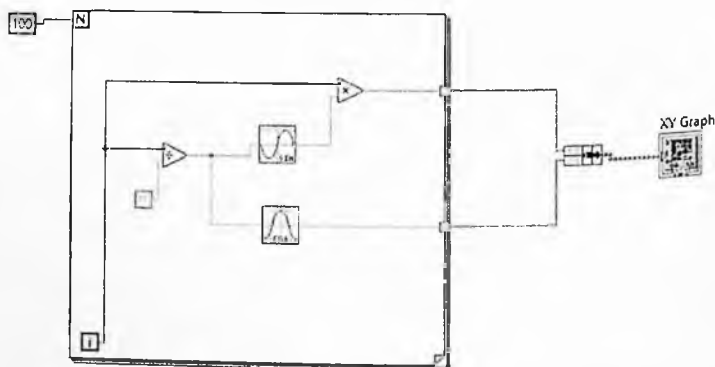
Ikki kordinatali(XYGraph) grafiklar xuddi shunday grafiklarni yaratish imkonini beradi. XYGraph ob'yekti kirish qismiga mos ma'lumotlar tipini bog'lash kerak. Bir nurli ikki

kordinatali otsillagramma(XY) va unga mos digramma blokidagi terminal 2.6.12-rasm va 2.6.13-rasmda ko'rsatilgan.

Ikki kordinatali grafik kirish qismiga bir klasterga birlashtirilgan X massiv(yuqori kirish) va Y massiv(quyi kirish) bog'lanadi.

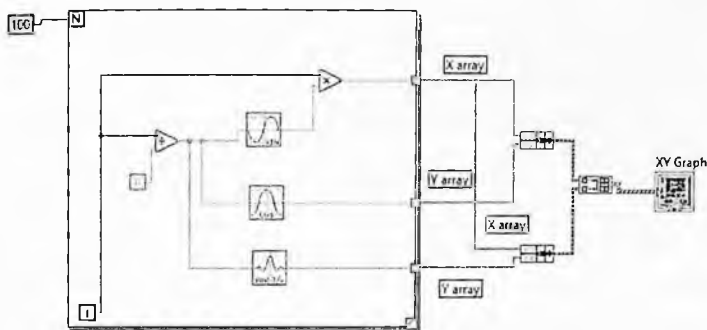


2.6.12-rasm. Ma'lumotlarga ega ikki kordinatali grafikning yuzadagi paneldagi ko'rinishi.



2.6.13-rasm. Ma'lumotlarni shakllantirish va grafikda aks etish digramma bloki.

Ikki kordinatorli grafiklarda bir nechta egri chiziqni akslantirish uchun X va Y qiymatli klasterlar massivini yaratish kerak(2.6.14-rasm).



2.6.14-rasm. Ko'p egri chiziqli ikki kordinatali grafiklarni shakllantirish diagramma bloki.

Grafiklar va otsilloграмма to'rlari komponentlari

Grafiklar va otsilloграмма to'rlari ajoyib xususiyatlarga ega bo'lib, ulardan foydalanib grafiklarni aks etishni moslashingiz mumkin. Ular quyidagilar:

1. Grafikni sozlash;
2. Kursor;
3. To'ring asosiy setkalari;
4. To'ring ikkinchi darajali setkalari;
5. Grafika palitrasi;
6. Kursorni boshqarish uskunasi;
7. Kursorni sozlash;
8. Masshtabni sozlash;
9. Argument shkalasi (X, abtsissa);
10. Qiymatlar shkalasi (Y, ordinata);
11. Belgi.

Takrorlash va mustaqil ishlash uchun savollar

1.Waveform Chart grafiklarni aks etish elementining abtsissa o'qida odatda qanday qiymat(parametr) joylashgan bo'ladi?

2.Waveform Chart grafiklarni aks etish elementining nechil xil rejimda aks etish rejimi mavjud?

3.Waveform Chart grafiklarni aks etishning birinchi rejimi qanday nomlanadi?

4.Waveform Chart grafiklarni aks etishning ikkinchi rejimi qanday nomlanadi?

5.Waveform Chart grafiklarni aks etishning uchinchi rejimi qanday nomlanadi?

§ 2.7. Qatorlarni o'rganish va kiritish/chiqarish ichki uskunalari

Qatorlarni aks etish turlari

Matnli qatorlarni kiritish va aks etish boshqaruv elementlari va indikatorlar bir necha foydali opsiyalarga ega. Masalan, ular aks etilmaydigan belgilarni qabul qilishi va aks etishi mumkin, xususan tabulyatsiya yoki qator boshiga qaytish belgilari. Agar siz yordamchi menyudan Normal Display o'rniga Codes Display opsiyasini tanlasangiz, ba'zi belgilar teskari slesh("\") va undan keyin ASCII-kod orqali ko'rinadi (2.7.1-jadval).

2.7.1-jadval.

LabVIEW slesh(boshqarish) kodlari

Kod	LabVIEW da bajarilishi
<u>\\00-\\FF</u>	8 bitli belgini 16-sanoq tizimidagi qiymati (harflar kata harflar bilan yozilishi kerka)
\\b	Bir belgi orqaga qaytish
\\n	Yangi qator
\\r	Qator boshiga keltirish
\\t	Tabulyatsiya

\s	Probel
\\	Teskari slesh

Matnli qatorlarni aks etish rejimlarini tanlash yoki o'zgartirish matn tarkibini o'zgartirmaydi, faqat ularni aks etishlihi o'zgartiradi. Kodlangan rejim dasturni otladkalash va qurilmalar, ketma-ket portlar va boshqa uskunalar bilan ishlash uchun boshqaruv belgilarini aniqlash uchun zarur.

Matnli qatorlarni bundan tashqari Password Display(yashirin qatorli) rejimi ham mavjud bo'lib, kiritilgan belgi o'rniga "*" ni aks etadi va ularni hech kim ko'rmaydi.

Yuzadagi panelda "*" lar ko'rsatilsa hamki, diagramma blokida real belgilar o'qiladi. Bunday aks etish siz VU ni himoyalashda zarur. Bundan tashqari matnli qatorlarni 16 lik sanoq tizimida aks etish rejimi ham mavjud. Rejimlar quyidagi rasmda keltirilgan(2.7.1-rasm).

2.7.1-rasm. Matnli qatorni aks etishning to'rt rejimi

Ro'yxatlar oynasi

LabVIEW da maxsus tuzilma bo'lib, qaysiki u ikki o'lchamli matli qatorlar massivni aks etadi. Siz matnli qatorlar jadvalini Controls elementlar palitrasining (List&Tables) ichki palitrasidan topishingiz mumkin (2.7.2-rasm). Jadvallarda qator va ustunlar sarlavhalari mavjud bo'lib, ularni ko'rindaigan yoki ko'rinmaydigan etish mumkin.

Normal String Display

LabVIEW dasturi

Backslash String Display

LabVIEW\dasturi

Password String Display

Hex String Display

4C61 6256 4945 5720
6461 7374 7572 69

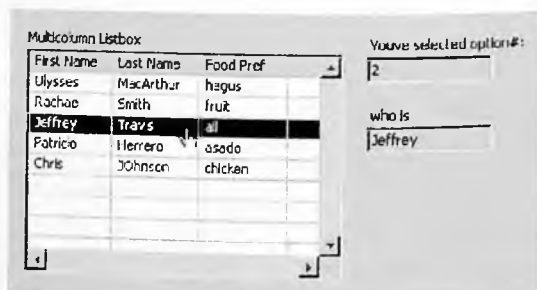
Index Display Table columns header

rows	0		x	x**2	sqrt(x)	
columns	0	0	0.0000	0.0000	0.0000	
		1	1.0000	1.0000	1.0000	
		2	2.0000	4.0000	1.4142	
		3	3.0000	9.0000	1.7321	
		4	4.0000	16.0000	2.0000	
		5	5.0000	25.0000	2.2361	
		6	6.0000	36.0000	2.4495	

rows header rows scrollbar columns scrollbar

2.7.2-rasm. Jadvalga misol

LabVIEW da ro'yxatlarni saqlovchi oynalarning ikki ko'rinishi mavjud: ro'yxatlar-oynasi(ListBox) va Ko'p ustunli ro'yxat oynasi(Multi-column Listbox). Ular ham List&Tables ichki palitrasida joylashgan. Ro'yxatlar oynasi xuddi jadvalga o'xshaydi, lekin ishlash paytida o'zini boshqacha tutadi. Tahrirlash rejimida ushbu ro'yxat oynaga ixtiyoriy matnni kiritish mumkin. VU ni ishlatish paytida to'plam ichidan menyudan tanlash singari harakatlanadi. Ro'yxat qiymati- bu tanlangan qator yoki qatorlar(agar yordamchi menyudan bir necha qatorni tanlash rejimi yoqilgan bo'lsa: Selection Mode >> 0(1) or More Items 0(1)) raqami (2.7.3- rasm).



2.7.3-rasm. Ro'yxat qiymatlarini tanlash.

Agar matn juda uzun bo'lsa va jadval yacheykasi yoki ro'yxat yacheykasiga sig'masa, VU ni ishga tushurganda bu elementga sichqonchani ko'rsatkichini etganizda matn to'liq shiphshitish oynasida paydo bo'ladi.

Bir ustunli ro'yxatda bir necha qatorli matnni kiritish uchun yordamchi menyudan Multi-line Input buyrug'ini tanlang.

Yordamchi menyudan Autosize Row Height opsiyasini tanlash shriftni o'zgartirganda yoki qo'shimcha qator qo'shilganda ro'yxat o'z hajmini avtomatik o'zgartiradi.

Matnli qatorlarni qayta ishlash funktsiyalarini ishlatish

Matnli qatorlar massivlar singari *LabVIEW* da tayyor funktsiyalarni ishlatish mumkin. Bu funktsiyalar Function palitrasining String ichki palitrasida joylashgan. Bu paragrafda bu funktsiyalarning ba'zilari bilan tanishamiz, qolganlari bilan esa keyinchalik mustaqil tanishingiz mumkin.

String Length(qator uzunligi) funktsiyasi matnli qatorda mavjud belgilar sonini qaytaradi(2.7.4-rasm).



Normal String Display
LabVIEW dasturi

Uzunligi
15

2.7.4-rasm. Matnli qator uzunligi funktsiyasini ishlatish

Concatenate Strings(qatorlarni birlashtirish) funktsiyasi barcha kiruvchi qatorlarni bir matnli qatorga birlashtiradi (2.7.5-rasm). Oddiy qatorlardan boshqa bir o'lchamli massvilarni ulab birlashtirish mumkin.



2.7.5-rasm. Matnli qatorlarni birlashtirish funksiyasini ishlatish

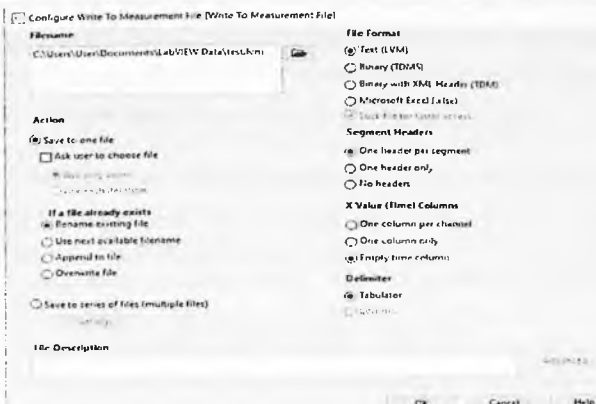
Fayllarga ma'lumotlarni kiritish/o'qish

Kiritish/chiqarish (I/O) operatsiyalari fayldan ma'lumotlarni o'qishga va ularga yozishga yordam beradi. Bu operatsiyalar Function palitrasining Programming >> File I/O ichki palitrasida joylashgan. Fayllar bilan ishlaydigan funksiyalar uchun faylni joylashgan manzilini ko'rsatish kerak. Agar siz fayl manzilini saqlovchi o'zgaruvchini ulamagan bo'lsangiz, unda bu funksiyalar dialog oynasini chaqirishida va bu oynada fayl tanlanishi yoki kiritilishi zarur.

Fayllar bilan ishlash funksiyalari uchun fayl tanlangandan keyin, ular faylni yaratishadi yoki ochishadi, fayldan ma'lumotlarni o'qiydi yoki yozadi, so'ngra faylni yopadi. Faylga ma'lumotlarni saqlashning yana bir keng tarqalgan usuli matnli fayllarni formatlash hisoblanadi. Bu juda qulay hisoblanadi, chunki keyinchalik bu fayllarni istalgan jadval protsessori orqali ochib qayta ishlashingiz mumkin.

Fayllarni o'qish va yozish ekspress bloki. LabVIEW da fayllardan ma'lumotlarni tezkor va interfaol o'qish va yozish uchun Write to Measurement File va Read From Measurement File ekspress bloklari mavjud.

Bu ekspress bloklar ixtiyoriy ma'lumotlar bilan ishlashi mumkin. Faylga yozish ekspress bloki dialog oynasi sozlanmasida ma'lumotlarni faylga yozish formatini tanlash imkonligi mumkin (2.7.6-rasm va 2.7.7-rasm).



2.7.6-rasm. Faylga yozish ekspres blokining sozlash dialog oynasi



2.7.7-rasm. Fayldan o'qish ekspres blokini sozlash dialog oynasi

Fayldan o'qish ekspres bloki sozlash dialog oynasida siz faylni formatini tanlashdan boshqa Read File Now tugmasini tanlash orqali fayldan bir qism ma'lumotlarni darhol o'qib aks etish imkonini beradi.

Agar siz fayllarga yozish va o'qish ekspres funksiyalari siz uchun yetarli emas deb o'ylasangiz, Write to Spreadsheet File (Jadvalli formatga yozish) va Read From Spreadsheet File (Jadvalli

fayllardan o'qish) funktsiyalarini ishlatish orqali jadval tipidagi fayllardan ma'lumotlarni o'qishingiz yoki yozishingiz mumkin.

Agar sizga faylga oddiy matnli ma'lumotlarni saqlash yoki o'qish zarur bo'lsa, unda siz LabVIEW ning Function palitrsi Programming>>File I/O ichki palitrasidagi Write to Text File yoki Read from Text File funktsiyalaridan foydalanishingiz mumkin.

Xuddi mantli fayllar singari siz binary(ikkilik) ma'lumotlarni ham faylda saqlashingiz yoki o'qishingiz mumkin. Bu katta sonlarni matnli sonlarga saqlashga qaraganda kichikroq hajmni egallaydi. Bu uchun Read from Binary File va Write to Binary File funktsiyalaridan foydalanishingiz mumkin.

Takrorlash va mustaqil ishlash uchun savollar:

- 1.Qatorlar nima uchun ishlatiladi?
- 2.Qatorlarni aks etish rejimlari va ular nima uchun kerak?
- 3.Jadvallar va ro'yxatlar nima uchun foydalaniladi?
- 4.Matnli qatorlarni qayta ishlash bog'lamlari haqida nimani bilasiz?
- 5.Fayllarga ma'lumotlarni yozish va o'qish bog'lamlari va ularning turlari haqida nimani bilasiz?

II-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI

AMALIY MASHG'ULOT №4

Avtomatik indekslash orqali massivlarni yaratish.

Ishdan maqsad: Avtomatik indekslash orqali massivlarni yaratish, LabVIEWdagi massivlarni boshqarish funksiyalari bilan tanishish.

Qisqa nazariy ma'lumotlar

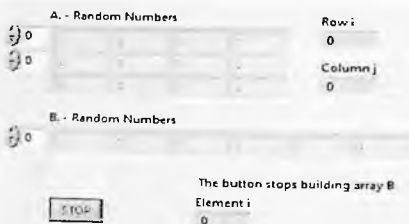
Amaliy mashg'ulotda chekli sondagi iteratsiyali tsikllarda va shartga ko'ra hosil qilinadigan tsikllarda avtoindekslashni qo'llab, massivlarni yaratish masalasi ko'riladi.

1. Amaliy mashg'ulot o'qituvchisidan *Building Arrays.vi* fayli misolini yuklang. Bu uskuna 2 ta massivni yaratadi va yuzadagi panelda akslantiradi: 2 o'lchovli massivni chekli sondagi iteratsiyadan foydalanib, bir o'lchovli massivni shartga ko'ra tsikl orqali hosil qiladi. Chekli sondagi iteratsiyali tsiklning takrorlanishlar soni aniq bo'ladi, shartga ko'ra hosil qilinadigan tsikllarni to'xtatish uchun *Stop* tugmasidan foydalaniladi (yoki tsikl 101 marta takrorlanishdan keyin o'zi to'xtaydi.)

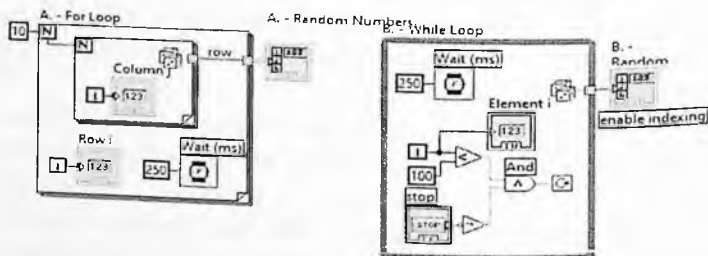
2. Yuzadagi panelga e'tibor bering (2.8.1-rasm.), so'ngra blok-diagrammaga o'ting (2.8.2-rasm.). Chekli sondagi iteratsiyali tsikllar avtoindekslashni qo'llab, 2 o'lchovli massivning satr va ustunlarini o'z chegaralarida qanday yaratishiga e'tibor bering. Tsikl chegarasidan tashqarida avtoindekslangan o'tkazgichning o'zgarishiga ham e'tibor bering.

3. Shartga ko'ra hosil qilinadigan tsikllardan ma'lumotlarni olishdan oldin, tasodifiy son mavjud bo'lgan kiritish/chiqarish nuqtasida o'ng tugmani tanlab, "Включить индексирование (*Enable Indexing*)" opsiyasini tanlang. Qanday ishlashini ko'rish uchun, kiritish/chiqarish nuqtasida o'ng tugmani

tanlab, *Отключить индексирование(Disable Indexing)* opsiyasini tanlang. Tsikldan chiqayotgan o'tkazgich ajratiladi. O'tkazgichni tiklash uchun kiritish/chiqarish nuqtasida o'ng tugmani tanlab, *Включить индексирование(Enable Indexing)* opsiyasini tanlang. Bu tsiklda foydalanuvchi Stop tugmasini bosmagan holda ham, ma'lum sondagi iteratsiya (101 marta takrorlanish) dan keyin albatta tsiklni to'xtatishini ta'minlovchi mantiqiy algoritm ishlatilgan. Agar foydalanuvchi Stop tugmasini bosmasa va tsikl 101 martadan kam takrorlangan bo'lsa, tsikl takrorlanishda davom etadi. Agar bu shartlardan birortasi buzilsa, tsikl to'xtatiladi. Nima uchun tsikl 101 marta takrorlanadi? Chunki tsikldan chiqish sharti har doim tsikl oxirida tekshiriladi. 100 marta takrorlanishdan keyin $i=99$, chunki 0 dan boshlanadi va tsikl takrorlanishda davom etadi. 101 marta takrorlanishdan keyin schetchikning qiymati 100 ga teng bo'ladi va tsikl to'xtatiladi.



2.8.1-rasm. VU ning yuzadagi paneli.

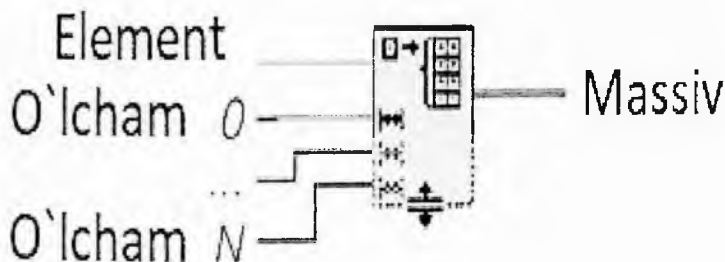


2.8.2-rasm. VU ning diagramma bloki.

4.VU ni ishga tushiring. Shartga ko'ra hosil qilinadigan tsiklni Stop tugmasi orqali to'xtatish lozim. Massiv to'liq yaratilmaguncha paneldagi akslantirish elementi yangilanmaydi.

Massivlar bilan ishlash uchun funktsiyalar.

LabVIEW da massivlarni boshqarish imkonini beruvchi funktsiyalar mavjud (Programming>> Array, Программируемые массивы bo'limi). *LabVIEW* da massivlar va barcha tuzilmalarda birinchi elementning indeksi-0, ikkinchisniki -1, va h.k. Massivlarning initsializatsiyasi funktsiyasi (Initialize Array)



2.8.3-rasm. Foydalanuvchi tanloviga ko'ra n-o'lchovli massiv elementlari yaratiladi va to'ldiriladi.

Размер массива (Array Size-massiv o'lchami) funktsiyasi kiritilayotgan massiv elementlari sonini aniqlaydi. Agar kiritilayotgan massiv n-o'lchovli bo'lsa, u holda funktsiya bir o'lchovli n ta elementdan iborat massivni qaytaradi.

Создать массив (Build Array-massiv yaratish) funktsiyasi konfiguratsiyasiga ko'ra ikkita massivni kombinatsiyalaydi yoki birlashtiradi yoki massivga qo'shimcha elementlarni qo'shadi. *Создать массив* funktsiyasida ikki turdagi kiritish usuli mavjud: massiv uchun va element uchun. Bu funktsiya orqali massivlarni bir vaqtning o'zida ham berilganlar massividan, ham skalyar kattaliklardan yaratish mumkin.

Подмножество массива (Array Subset-massiv osti) funksiyasi indeks(index) dan boshlab, soni uzunlik(length)ga teng elementlardan iborat massiv qismini qaytaradi.

Выборка из массива (Index Array –massivdan tanlash) funksiyasi massivning ixtiyoriy elementiga murojaatni ta'minlaydi.

Удалить из массива (Delete From Array –massivdan o'chirish) funksiyasi massivning berilgan indeks(index)dan boshlab, berilgan uzunlik(length)dagi elementlardan iborat massiv qismini o'chiradi. *Подмножество массива* (massiv osti) funksiyasi kabi massivning berilgan qismini qaytaradi va chiqishda boshlang'ich o'zgartirilgan massiv ham ko'rinadi.

Quyidagi misolda ikkita massivni birlashtiruvchi va yangi massiv hosil qilingan massiv o'rtasidagi elementni qirqib oluvchi VUni yaratish jarayoni ko'riladi.

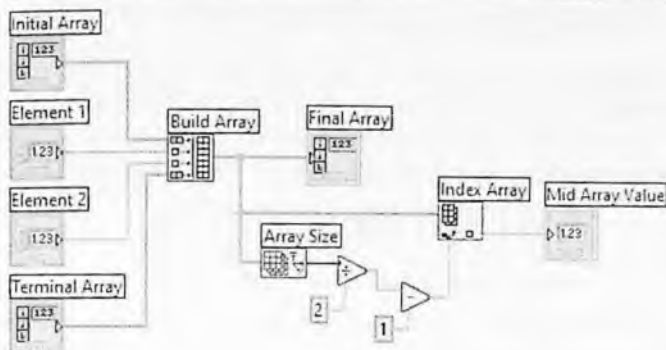
1. *Amaliy mashg'ulot o'qituvchingizdan* Array Exercise.vi VUni yuklang. Bu uskunaning yuza panelida ikkita kiruvchi massiv (har biri 3 tadan elementdan iborat), boshqarishning ikkita sonli elementi, hamda chiquvchi massiv joylashgan. Yangi massivni hosil qilish uchun VU ikkita massivni va boshqarish elementlari qiymatini quyidagi tartibda birlashtiradi:

-Boshlang'ich massiv + (1-element) + (2-element) + oxirgi massiv.

2.2.8.4-rasmdagi kabi diagramma blokini quring. Funksiyaning mos terminallarini topish uchun kontekst yordam oynasidan foydalaning.

Создать массив (Massiv yaratish) funksiyasi (*Programming>> Array(Massiv)* palitrasi) bu misolda boshlang'ich ma'lumotlarni yangi massivni hosil qilish uchun quyidagi tartibda birlashtiradi:

Boshlang'ich massiv + (1-element) + (2-element) + oxirgi massiv.



2.8.4-rasm. Diagramma blokidagi funksiya chap tomondagi ikonka sifatida tasvirlangan.

Число элементов массива (Massivdagi elementlar soni-Massiv palitrasi) funksiyasi birlashtirilgan massivdagi elementlar sonini qaytaradi.

Выборка массива (massivdan tanlash) funksiyasi massivning o'rtasida joylashgan elementni qaytaradi.

LabVIEW **Создать массив** (Massiv yaratish) funksiyasi orqali massiv yaratadi. So'ngra massivning o'rtasidagi elementi indeksining qiymati - massiv uzunligining yarmi minus bir kabi hisoblanadi.

3.Yuza panelga qaytib, VU ni ishga tushuring. Kiruvchi ma'lumotlarning har xil kombinatsiyasini hosil qiling.

4.VUni saqlab, yoping.

AMALIY MASHG'ULOTNI BAJARISH TARTIBI

1. Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajaring va natijalarni tahlil qiling.

2. Avtomatik indekslash orqali yaratilgan massivlarni ekranga chiqarib uni yaratish texnologiyasini tushuntiradi.

AMALIY MASHG'ULOT №5

Tartibni o'zgartirish. Ichki massivni aniqlash.

Ishdan maqsad: Massivdagi elementlar tartibini o'zgartirish, ichki massivni aniqlash va massiv elementlarini ko'paytirish amallarini o'zlashtirish.

Nazariy qism

Massiv va klasterlarning o'zaro almashinuvi

Ba'zi hollarda massiv va klasterlarni o'zaro almashtirish qulay hisoblanadi. Buning qulayligi shundaki, *LabVIEW* da klasterlardan ko'ra massivlar bilan ishlaydigan funktsiyalar ko'proq. Masalan, yuzadagi panelda tugmalar klasteri joylashgan bo'lsin. Tugmalarning joylashish tartibini teskarisiga o'zgartirish uchun "Кластер в массив" (*Cluster to Array*) funktsiyasi yordamida klasterni massivga o'zgartirib, so'ngra "Обращение индексов одномерного массива" (*Reverse 1D Array*) funktsiyasi yordamida tugmalarning joylashish tartibi teskarisiga o'zgartiriladi. Teskari amal - massivni qaytib klasterga o'zgartirish uchun "Массив в кластер" (*Array to Cluster*) funktsiyasidan foydalaniladi (2.9.1-rasm, 2.9.2-rasm).

Massiv  Klaster

2.9.1-rasm. "Array to cluster" funktsiyasi diagramma blokida.

Klaster  Massiv

2.9.2-rasm. "Cluster to array" funktsiyasi diagramma blokida.

Massiv va klasterlarni taqqoslash

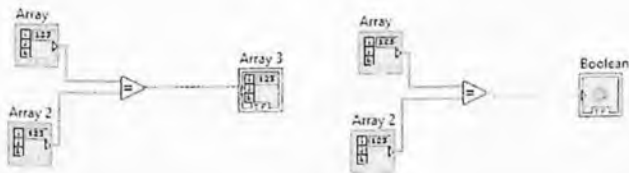
Massiv va klasterlar bilan ishlashda ba'zi taqqoslash funktsiyalarining 2 ta rejimi mavjud: *Compare Aggregates* (Сравнить объект) va *Compare Elements* (Сравнить по элементно). Rejimni tanlash 2.9.3-rasmda ko'rsatilganidek, funktsiyaning kontekst menyusi orqali amalga oshiriladi.



2.9.3-rasm. "Tenglik" polimorf funktsiyasini tanlash rejimi.

Ob'yektlarni taqqoslash rejimida, agar ob'yektning barcha elementlari uchun shart bajarilsa, funktsiya 1 ta ROST mantiqiy qiymatni qaytaradi.

Elementlar bo'yicha taqqoslash rejimida funktsiyaning chiqishida o'lchami kirishdagi kabi bo'lgan mantiqiy qiymatlarning massiv va klasteri, bunda har bir elementi – mos elementlarni taqqoslash natijasi hisoblanadi. 2.9.4-rasmda "Tenglik" funktsiyasi uchun 2 ta rejimda taqqoslash natijalari keltirilgan.



2.9.4-rasm. "Tenglik" polimorf funktsiyasi, ob'yektlar sifatida (o'ngda) va 2 ta massivni har bir elementi bo'yicha taqqoslaydi (chapda).

Tartibni o'zgartirish.

100 ta tasodifiy sondan iborat massivdagi tartibni o'zgartiruvchi VU ni yarating. Masalan, 0-element 99-element bo'lsin, 98-element 1-element bo'lsin va h.k. Yaratilgan VU ni Reverse Random Array.vi deb nomlang.

Ichki massivni aniqlash.

100 ta tasodifiy sondan iborat massivning bir qismini, masalan, 10-indeksdan 50-indeksgacha bo'lgan elementlarni ekranga chiqaruvchi VU ni yarating.

Yaratilgan VU ni Subset Random Array.vi deb nomlang.

Massiv elementlarini ko'paytirish.

Bir o'lchovli massivni kiritib, so'ngra har bir juft elementlarni ko'paytiradigan (0 va 1 dan boshlab) va ekranga chiqaruvchi VU ni yarating. Masalan, elementlari 1,23,10 va 5,7,11 sonlaridan iborat 2 ta massiv kiritilganda, natija 23, 50, 77 sonlaridan iborat massiv bo'lsin. Yaratilgan VU ni Array Pair Multiplier.vi deb nomlang.

AMALIY MASHG'ULOTNI BAJARISH TARTIBI

1.Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajaring va natijalarni tahlil qiling.

2.Elementlar tartibni o'zgartirish, massiv elementlarini ko'paytirish orqali yaratilgan massivlarni ekranga chiqarib, ularni yaratish texnologiyasini tushuntiring.

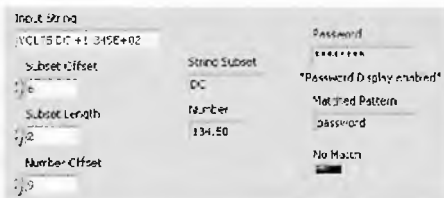
AMALIY MASHG'ULOT №6

Qatorlarni tahlil etish.

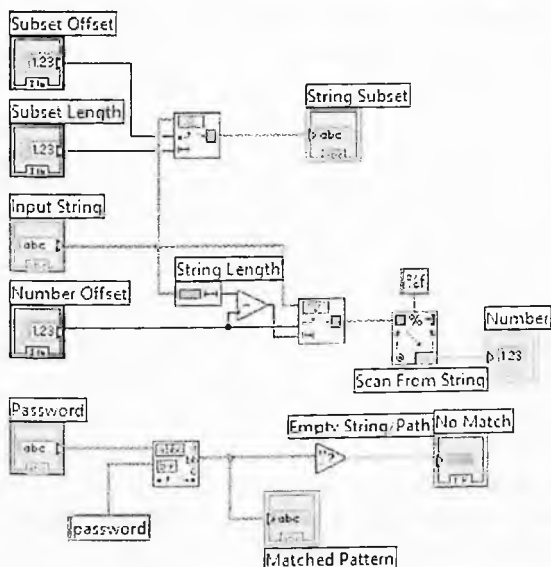
Ishdan maqsad: Boshqarish elementlari opsiyalari va qatorlarni aks etishni, qatorlar bilan ishlash uchun LabVIEW funksiyalarini ishlatish texnologiyalarini o'zlashtirish.

Qatorning bir qismini ajratish orqali uzun qatorli ma'lumotlarni tahlil qiladigan va qatorning shu qismidagi sonli belgilarni sonli qiymatlarga o'zgartiradigan VU yaratimg.

1.Konteks menyusidan *Скринное отображение* (Password Display - Yashirin ko'rinishlar) funksiyasini tanlash yo'li bilan qatorga Parol (Password) ni shunday sozlangki, u faqat yulduzchalarni ko'rsatsin.



2.10.1-rasm. Ushbu mashg'ulotning VU yuzadagi paneli ko'rinishi.



2.10.2-rasm. Ushbu mashg'ulotning VU diagramma bloki sxemasi.

 **Qator qismini ajratish** funksiyasi, ma'lum bir simvoldan boshlab kiritilayotgan qatordan berilgan uzunlikdagi qator qismini qaytaradi

 **Qatorni ko'rib chiqish** funksiyasi, haqiqiy sonli simvollardan tarkib topgan qatorni (0 dan 9 gacha, +, -, E va davr) songa o'zgartiradi.

 **Qator shabloni** funksiyasi, foydalanuvchi paroli qatorini berilgan parol qatori bilan taqqoslaydi. Agar moslik bo'lsa, qator ekranda ko'rinadi, aks holda, ekranda bo'sh qator ko'rinadi. **Bo'sh qator /yo'l?** (Empty String ? Path) funksiyasi

 **Taqqoslash** palitrasidan ROST mantiqiy qiymatni qaytaradi, agar

 **Qator shabloni** funksiyasidan chiqarish qator qismi shablonida bo'sh qatorni topsa.

Qator uzunligi (*String Length*) funksiyasi qatordagi belgilar sonini

2.VU ni ko'rsatilgan ma'lumotlar bilan ishga tushiring. Kirish qatoridan DC qator ostini ajratish amalga oshirilishiga e'tibor bering. Shuningdek, qatorning sonli qismi tahlil qilinib, songa o'zgartirilgan. Boshqarish elementlarining turli xil qiymatlarida bajarib ko'ring (qatorlar ham massivlar kabi 0 dan boshlab indekslanadi).

Parol qatori faqat "yulduzchalar"ni ko'rsatayotganligiga e'tibor bering. **Qator shabloni** funksiyasi kirish parolini qatordagi parol (hozirgi holatda belgilardan tarkib topgan <<password>>) taqqoslaydi va ularni qaytaradi, agar u topilsa. Agarda moslik bo'lmasa, funksiya bo'sh qatorni qaytaradi.

3.Fayl menyusidan Yopish funksiyasini tanlab VU ni yoping. VU ni "Parse String.vi" nomi bilan saqlang.

AMALIY MASHG'ULOTNI BAJARISH TARTIBI

1. Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajarib va natijalarni tahlil qiling.

2. *LabVIEW* da yaratilgan qatorlarni ekranga chiqaring va ular bilan ishlash texnologiyasini tushuntiring.

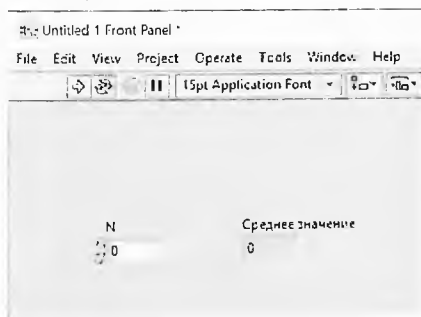
TAJIRIBA MASHG'ULOTI №3

Tsikl tuzilmalari va registrlardan foydalanib masalalar yechish.

Ishdan maqsad: Bir necha tasodifiy sonlarni o'rtacha qiymatini hisoblovchi virtual uskuna yaratish. Tsikllar va siljuvchan registrlarni qo'llagan holda dasturlarni yaratish.

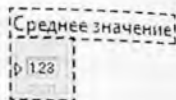
Tajribani bajarish ketma-ketligi:

1. Yangi bo'sh virtual uskuna yarating va uning yuzadagi panelida birta sonli boshqarish elementi hamda birta sonli indikator joylashtiring.



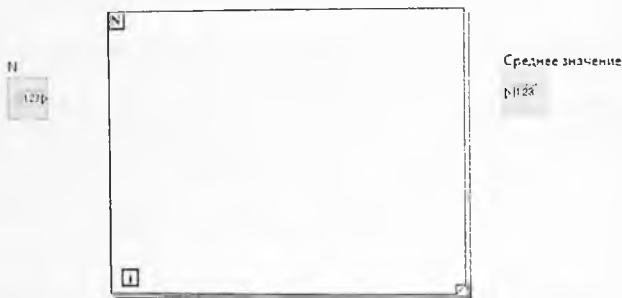
2.11.1-rasm. Mashg'ulotdagi VU ning yuzadagi paneli ko'rinishi.

2. Virtual uskunani diagramma blokiga o'ting va sonli terminallarni quyidagi rasmdagidek joylashtiring.



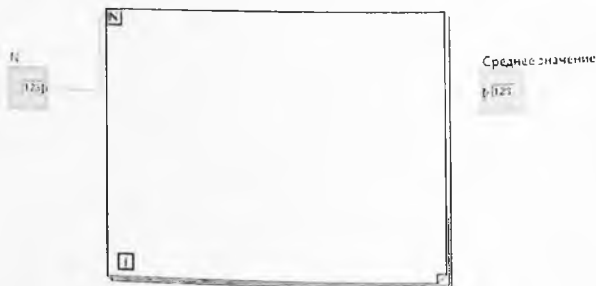
2.11.2-rasm. VU ning diagramma blokidagi terminallari.

3. Diagramma blokida sonli terminallar o'rtasida For Loop tuzilmasini quyidagi rasmdagidek joylashtiring.



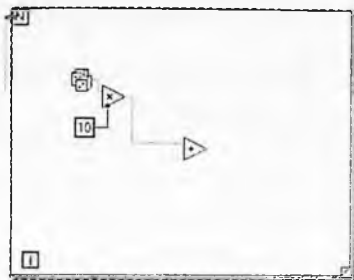
2.11.3-rasm. VU ning diagramma blokida For Loop tuzilmasini joylashtirish.

4. For Loop siklidagi N terminalini sonli boshqarish elementi terminali bilan bog'lang.



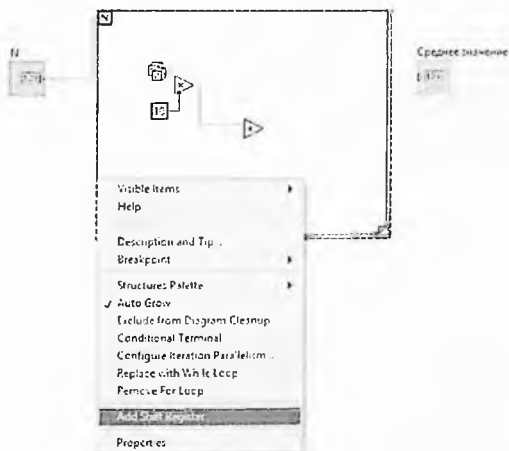
2.11.4-rasm. VU ning diagramma blokida For Loop tuzilmasiga takrorlanishlar sonini ko'rsatish.

5. For Loop tuzilmasi ichiga quyidagi sxemadagi singari sxemani tuzing:



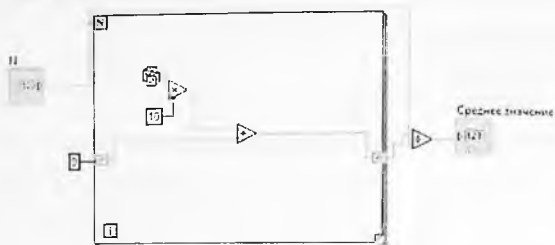
2.11.5-rasm. VU ning diagramma blokida For Loop tuzilmasi tarkibi.

6. For Loop siklini tanlang va uning chegarasiga sichqonchani o'ng tugmasini tanlang hamda paydo bo'lgan yordamchi menyudan «Add shift Registr» opsiyasini tanlang.



2.11.6-rasm. VU ning diagramma blokida For Loop tuzilmasida siljovchan registr larni joylashtirish.

7. Barcha terminallarni ulagandan so'ng, diagramma blokida sizga quyidagi rasmda k'orsatilgan k'orinishidagi sxema hosil bo'lishi kerak.



2.11.7-rasm. VU ning diagramma blokining yakuniy ko'rinishi.

8. Dastruni ishga tushurib tekshirib ko'ring.

Talabalar uchun mustaqil bajarish uchun variantlar:

1. Hisoblang $S = \sum_{i=1}^n \frac{0,4}{i(i+1) + \sqrt{i}}$

2. Hisoblang $S = \sum_{j=1}^n \frac{5 + j^2}{6j + 2,7}$

3. Hisoblang $S = \sum_{k=1}^n \frac{1}{k + 1,2}$

4. Hisoblang $S = \sum_{i=1}^n \frac{35}{\sqrt{i+1} + \sqrt{i}}$

5. Hisoblang $S = \sum_{i=1}^n \frac{1}{|\cos i + i|}$

6. Hisoblang $S = \sum_{i=5}^n \frac{1}{\cos i + \sin i}$

7. Hisoblang $S = \sum_{k=1}^n \frac{k + 0,4}{\sqrt{k} + \sqrt{k}}$

8. Hisoblang $S = \sum_{k=1}^n \frac{k+1}{k^2+2}$

TAJRIBA MASHG'ULOTI №4

Variant tuzilmasi va massivlardan foydalanib masalalar yechish.

Ishdan maqsad: Tsikllar tuzilmasidan foydalanib, tasodifiy qiymatlarni o'rtacha qiymatni aniqlash dasturini yaratish. Nolga bo'lish jarayoni tekshirish dasturini tayyorlash.

Vazifa:

1.O'rtacha qiymatni aniqlash dasturini yaratish. Bunda 0 dan 10 gacha bo'lgan oraliqda tasodifiy sonlar olinsin va ularni o'rtacha qiymati hisoblansin. Tasodifiy sonlarni tanlash jarayonida(tsiklda) tanlangan tasodifiy son qiymati sonli indikatora ko'rsatilsin va sonni ko'rib olish uchun 1 soniya kutish funksiyasi mavjud bo'lsin.

Tasodifiy qiymatlar soni

Tasodifiy son

O'rtacha qiymat

3

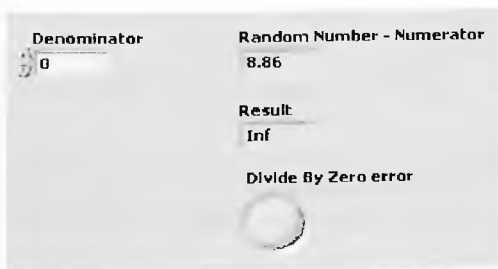
8,39204

7,91754

2.12.1-rasm. Mashg'ulotning VU si yuzadagi paneli ko'rinishi.

2.Nolga bo'lishni tekshiruvchi dastur yaratish.

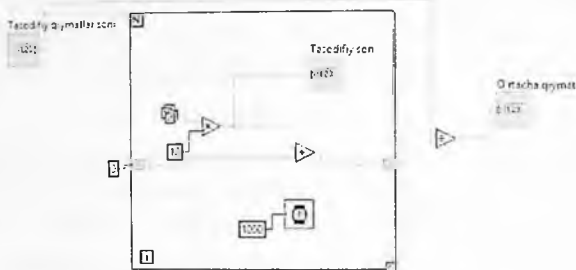
Variant tuzilmasidan foydalaning!



2.12.2-rasm. Nolga bo'lishni tekshiruvchi VU ning
yuzadagi paneli ko'rinishi

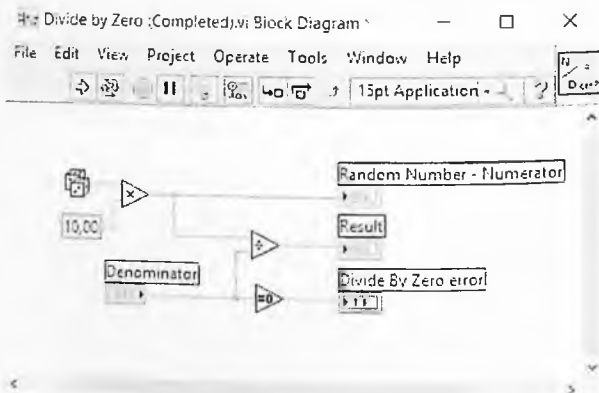
Ishni bajarish tartibi:

1.Yuzadagi panelni 1-topshiriqqa asosan tayyorlang va quyidagi diagramma bloki orqali o'rtacha qiymatni hisoblang. Vazifani ikkala tsikldan foydalanib bajarish ham mumkin.



2.12.3-rasm. Birinchi vazifa bo'yicha VU ning diagramma bloki sxemasi.

2.Nolga bo'lishni tekshirish dasturini yaratish uchun quyidagi diagramma blokidan foydalanish mumkin. Ushbu vazifani bajarishda variant tuzilmasidan foydalanish ham mumkin.



2.12.4-rasm. Nolga bo'lish vazifasi uchun diagramma bloki sxemasi

TAJRIBA MASHG'ULOTI №5

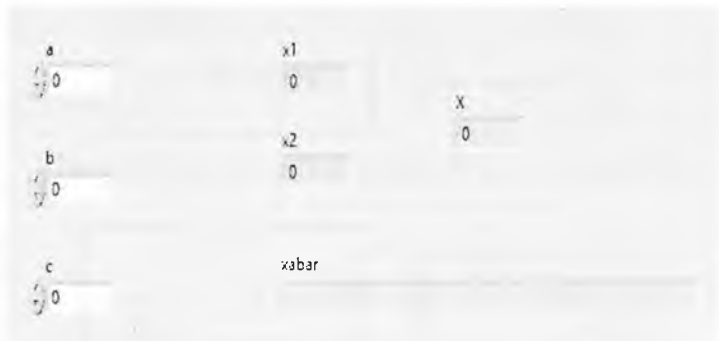
Qatorlarni qayta ishlashni o'rganish.

Ishdan maqsad: Variant tuzilmasi va formula tugunidan foydalangan holda virtual uskunalarni yaratish.

Tajribani bajarish ketma-ketligi:

1. Yangi virtual uskunani yarating.

2. Elementlarni yuzadagi panelga quyidagi rasmda ko'rsatilganidek joylashtiring, ya'ni bunda 3 ta sonli boshqarish elementi, 3 ta sonli indikator va bitta matnli indikator joylashtirasiz.



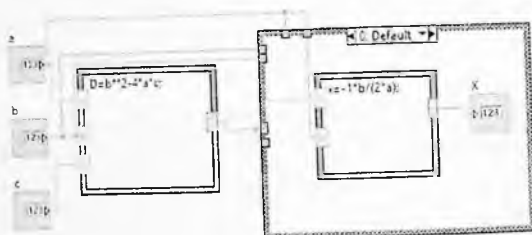
2.13.1-rasm. Mashg'ulotning VU si yuzadagi paneli ko'rinishi.

3. Diagramma blokiga o'ting va quyidagi rasmlarda ko'rsatilgan sxemani yarating. Ma'lumki, variant tuzilmasining variantlarida barcha variantlar ekranda bir vaqtning o'zida ko'rsatilmaydi, shuning uchun quyidagi uchta variant uchun $D > 0$ (diskriminant noldan katta), $D = 0$ (diskriminant nolga teng) va

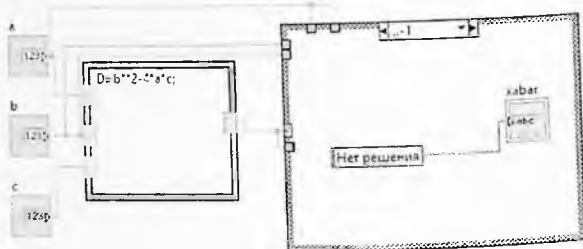
$D < 0$ (diskriminant noldan kichik) holatlar uchun sxemalar keltirilgan.



2.13.2-rasm. VU ning $D > 0$ holatidagi diagramma bloki sxemasi.



2.13.3-rasm. VU ning $D = 0$ holatidagi diagramma bloki sxemasi.



2.13.4-rasm. VU ning $D < 0$ holatidagi diagramma bloki sxemasi.

4. Yaratilgan virtual uskunani sinab ko'ring.

Talabalar uchun topshiriqlarning variantlari:

1. a, b va c uchta sondan eng kattasini toping.

2. X qiymatini hisobga olib, funktsiyaning qiymatini hisoblang:

$$y = \begin{cases} x^2 + 4x - 7, & x < 2, \\ \frac{1}{x^2 + 4x - 7}, & x \geq 2. \end{cases}$$

1.

$$f(x) = \begin{cases} 0, & x \leq 0, \\ x^2 - x, & 0 < x \leq 1, \\ x^2 - \sin \pi x^2, & x > 1. \end{cases}$$

4.

$$f(x) = \begin{cases} x^2 - 3x + 8, & x^2 - x \leq 1, \\ \frac{1}{x^2 - 3x + 8}, & x^2 - x > 1; \end{cases}$$

2.

$$f(x) = \begin{cases} -4, & x < 0, \\ x^2 + 3x + 4, & 0 \leq x < 1, \\ (x^2 + 3x + 4)^2 - 1, & x \geq 1; \end{cases}$$

3.

$$5. \quad Z = \frac{\min(x, y) + 0,5}{1 + \max^2(x, y)}.$$

$$6. \quad u = \frac{\min\left(\frac{x+y+z}{3}, xyz\right)}{1 + \min^2\left(\frac{x+y+z}{3}, xyz\right)}.$$

7. $x_1, x_2, x_3, y_1, y_2, y_3$ sonlari berilgan. Ushbu nuqtalardan uchburchakning mos $(x_1, y_1), (x_2, y_2)$ va (x_3, y_3) nuqtalari bo'lishi mumkinligini aniqlovchi dastur yarating.

8. Bikvadrat tenglama $ax^4 + bx^2 + c = 0$ ildizlarini topish dasturini tuzing.

II-MODUL BO'YICHA TEST SAVOLLARI

1.LabVIEW da asosiy necha xildagi tsikllar mavjud?

- A.2
- B.3
- C.1
- D.4

2.LabVIEW da tuzilishlar nima?

- A.Oddiy dasturlash tillari operatorlariga o'xshab virtual uskuna bajarilishini boshqaradi
- B.diagrammalar blokini izohlash uchun juda qulay vositadir
- C.matn hisoblanib, u boshqaruvchi element yoki indikatorni izohlaydi
- D.yuzadagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloke

3.LabVIEW da tsikllarning birinchi turi(iteratsiyalar sonini qayd etish) tuzilishi nomini to'g'ri belgilang?

- A.For Loop
- B.While Loop
- C.Case Structures
- D.Event Structures

4.LabVIEW da tsikllarning ikkinchi turi(shart bo'yicha tsikl) tuzilishi nomini to'g'ri belgilang?

- A.While Loop
- B.For Loop
- C.Case Structures
- D.Event Structures

5.LabVIEW da For Loop siklini takroran bajarilishi tsikl tarkibiga kirgan massivga bog'liqligi?

- A.Avtoindekslashga bog'liq
- B.For Loop sikliga bog'liq
- C.While Loop sikliga bog'liq
- D.Terminalga berilgan iteratsiyalar soni qiymatiga bog'liq

6.LabVIEW da For Loop siklini muddatdan oldin to'xtatish mumkin?

- A.Maxsus terminal yordamida
- B.While Loop siklida mavjud
- C.Terminalga berilgan iteratsiyalar soni qiymatiga bog'liq

D.Avtoindekslashga bog'liq

7.LabVIEW da While Loop sikli standart holatda qaysi mantiqiy qiymatda takrorlash to'xtatiladi?

- A.True
- B.False
- C.And
- D.Or

8.LabVIEW da While Loop siklini "yo'q" mantiqiy qiymatda takrorlash to'xtatilishini ta'minlash maqsadida yordamchi menyuning qaysi opsiyasidan foydalanish zarur?

- A.Continue if True
- B.Continue if False
- C.Stop if True
- D.Stop if False

9.LabVIEW da iteratsiyalar sonini qayd etish siklini muddatdan oldin shart bo'yicha to'xtatish imkoniyatini ta'minlash uchun yordamchi menyuning qaysi opsiyasidan foydalanish zarur?

- A.conditional terminal
- B.Continue if True
- C.Continue if False
- D.Stop if True

10."Shart bo'yicha tsikl doimo bir marotaba bo'lsa hamki bajariladi, iteratsiyalar sonini qayd etish sikli esa bir marotaba ham bajarilmasligi mumkin" degan fikr to'grimi?

- A.While Loop va For Loop siklida
- B.Faqat while loop siklida
- C.Faqat 1-qism to'g'ri
- D.Faqat 2-qism to'g'ri

11.LabVIEW da siljuvchan registrlarni tsikl tuzilishlarining yordamchi menyuning qaysi opsiyasi yordamida paydo qilish mumkin?

- A.Add shift register
- B.conditional terminal
- C.Continue if True
- D.Stop if True

12.Quyidagilardan qaysi biri LabVIEWda belgini yashirish (ko'rinmaydigan) qilish imkonini beradi?

- A. kontekst menyudan Visible items >> Caption bandini tanlash

- B.kontekst menyudan Create >> Caption bandini tanlash
- C.kontekst menyudan Create >> Label bandini tanlash
- D.kontekst menyudan Visible items >> Label bandini tanlash

13.Bir vaqtning o'zida yuzadagi panel va diagrammalar-blogi oynasini ko'rish uchun.....tugmalar tanlanadi:

- A.Ctrl+T
- B.Ctrl+N
- C.Ctrl+C
- D.Ctrl+A

14. Bir vaqtning o'zida yuzadagi panel va diagrammalar-blogi oynasini ko'rish uchun..... bandini tanlash kerak:

- A.Window menyusidan Title Left and Right bandi
- B.Tools menyusidan Title Left and Right bandi
- C.Window menyusidan menyusidan Show Block Diagram bandi
- D.Tools menyusidan Title Left and Right bandi

15.LabVIEW tuzilishlarda faqatgina o'tkazgich va tuzilish chegarasi kesishgan nuqtada yaratiladi. U nima?

- A."tunnel"
- B."bog'lam"
- C."hodisa"
- D."avtoindekslash"

16.Variant tuzilmasini ishlatganda standart qanday variantlar avtomatik mavjud bo'ladi?

- A.True va False
- B.Sonli
- C.0,1
- D.0 dan 9 gacha

17.Formula bog'lamiga qiymat kiritish uchun yordamchi menyuning qaysi opsiyasidan foydalanish kerak?

- A.Add Input
- B.Add Output
- C.Add Input Value
- D.Add Output Value

18.Formula bog'lamiga qiymat chiqarish uchun yordamchi menyuning qaysi opsiyasidan foydalanish kerak?

- A.Add Output
- B.Add Input

- C.Add Input Value
- D.Add Output Value

19.Formula bog'lamida har bir formula ifodasi oxirida qaysi belgi bilan yakunlanishi zarur?

- A. ;
- B. :
- C. ,
- D. .

20.Formula bog'lami tarkibida shart bo'yicha hisoblashlarni amalga oshirish imkoniyati mavjudmi?

- A.Maxsus ifoda yordamida
- B.MAxsus terminal yordamida
- C.Variant tuzilmasini ishaltish orqali
- D.Tsikllarni qo'llagan holatda

21.Variant tuzilmasini dan yuklash mumkin.

- A. Functions palitrasining Programming >> Structures ichki palitrasi;
- B.Controls palitrasining Modern >> Structures ichki palitrasi;
- C.Controls palitrasining Modern >> String ichki palitrasi;
- D.Functions palitrasining Programming >> String ichki palitrasi;

22. Agar Variant tuzilmasining selektorga sonli yoki qatorli ma'umotlar ulangan bo'lsa, unda tuzilmavariantlarga ega bo'ladi.

- A. cheksiz;
- B. 2 ta;
- C. 0 va 1;
- D. 3 ta;

23.Add Case After va Add Case Before opsiyalari orqali nima bajariladi?

- A.Variant qo'shiladi;
- B.Tunnel qo'shiladi;
- C.O'tkazgich qo'shiladi;
- D.Shart qo'shiladi;

24.Dialog oynalari ochiq bo'lganda LabVIEW ning boshqa oynasini faollashtirib bo'ladimi?

- A.Dialog oynasini yopish zarur;
- B.2 ta oynani;
- C.Vaqtincha;
- D.1 ta oynani.

25. Agar avtoindekslash o'chirilgan bo'lsa, unda.....?

- A. massiv to'liqligicha tsiklga kiritiladi;
- B. tsikl har bir iteratsiyada massivning keyingi indeksiga o'tadi;
- C. tsikl har bir iteratsiyada massivning 1- indeksiga o'tadi;
- D. massiv to'liqligicha tsiklga kiritilmaydi.

26. LabVIEW da bir o'lchamli massivda maksimal nechta element joylashgan bo'lishi mumkin?

A. 2^{31}

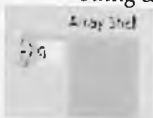
- B. 1 mln.
- C. 32 ming
- D. 100

27. LabVIEW da massiv elementlari tartibi doim nechadan boshlanadi?

- A. 0
- B. 1
- C. Siz ko'rsatgan qiymatdan
- D. LabVIEW tomonidan o'zi belgilanadi

28. LabVIEW da massivlarni yuzadagi panelda yaratish uchun qaysi paneldan foydalanish kerak?

- A. Modern >> Array Matrix & Cluster
- B. Modern >> Numeric
- C. Modern >> Boolean
- D. Modern >> String & Path



29. massiv tipini aniqlang?

- A. Aniqlanmagan
- B. Sonli
- C. Matnli
- D. Boolean

30. LabVIEW da massiv shablonini yuzadagi panelda joylashtirilganda va uning tipi aniq bo'lmaganda diagramma blokida uning terminali qaysi rangda bo'ladi?

- A. Qora
- B. Oq

C.Qizil

D.Ko'k

31.LabVIEW da massiv terminalini oddiy terminallardan nimasi bilan farqlash mumkin ?

A.Kvadrat qavslar mavjudligi

B.Qora rangda bo'lishi

C.O'tkazgich qalinligi

D.O'tkazgich mayinligi

32.LabVIEW da diagramma blokida massiv o'tkazgichlari oddiy o'tkazgichlaridan nimasi bilan farqlanadi?

A.Qalinligi

B.Mayinligi

C.Rangi

D.O'tkazgichdagi uchlarida tunelalrning mavjudligi

33.LabVIEW da avtoindekslash nima uchun zarur?

A.Massiv elementlariga tsikl ichida avtomatik tartib raqami bo'yicha murojaatlash uchun

B.Massiv elementlarini qayta raqamlash uchun

C.Massiv lementlarini o'chirish uchun

D.Massiv elementlarini birlashtirish uchun

34.LabVIEW da tsiklda 100 marotaba takrorlashlar soni ko'rsatilgan va unda 37 elementdan iborat bir o'lchamli massiv avtoindekslash yoqilgan holatda ulangan. Tsikl necha marotaba takrorlanadi?

A.37

B.100

C.0

D.137

35.LabVIEW da tsiklda 100 marotaba takrorlashlar soni ko'rsatilgan va unda 137 elementdan iborat bir o'lchamli massiv avtoindekslash yoqilgan holatda ulangan. Tsikl necha marotaba takrorlanadi?

A.100

B.37

C.0

D.137

36.LabVIEW da tsiklda 100 marotaba takrorlashlar soni ko'rsatilgan va unda 37 va 45 elementlardan iborat bir o'lchamli massivlar

avtoindekslash yoqilgan holatda ulangan. Tsikl necha marotaba takrorlanadi?

A.37

B.100

C.0

D.137

37.LabVIEW da klasterning massivdan farqi nimada?

A.Elementlar turining har xilligi

B.Elementlar sonining ko'pligi

C.Elementlar turining bir xilligi

D.Elementlarni tartiblash 1 dan boshlanadi

38.Noto'g'ri jumlani toping: Klasterlar terminallarini birlashtirish..... mumkin.

A.agar ularning tipi har xil bo'lsa;

B.agar ularning tipi bir xil bo'lsa;

C.ikkala klasterlar ham bir xil sondagi elementlarga ega bo'lsa;

D.mos elementlarning tipi o'rni bilan bir xil bo'lsa.

39.Klasterda joylashtirilgan birinchi ob'yekt hisoblanadi.

A.0-element;

B.1-element;

C.oxirgi element; D. n-1 element.

40.Kontekst menyudagi "Reorder Controls in Cluster" bandining vazifasi nima?

A.Klaster ichidagi elementlarni tartiblash;

B.Klaster ichidagi elementlarni o'sish tartibida joylashtirish;

C.Klaster ichidagi elementlarni kamayish tartibida joylashtirish;

D.Klaster ichidagi elementlar tartibni o'zgartirish;

41.Nima uchun massiv va klasterlarni o'zaro almashtirish kerak bo'ladi?

A.LabVIEW da klasterlardan ko'ra massivlar bilan ishlaydigan funksiyalar ko'proq;

B.Massivdagi elementlar tipi bir xil;

C.Klasterdagi elementlar tipi har xil;

D.LabVIEW da massivlardan ko'ra klasterlar bilan ishlaydigan funksiyalar ko'proq;

42.LabVIEW da matnli qatorlarni yuzadagi panelda aks etishning necha xil rejimi mavjud?

- A.4
- B.2
- C.3
- D.5

43.LabVIEW da matnli qatorlarni yuzadagi panelda aks etishning birinchi rejimi qasi rejim?

- A.Normal Display
- B.Codes Display
- C.Password Display
- D.HEX Display

44.LabVIEW da matnli qatorlarni yuzadagi panelda aks etishning ikkinchi rejimi qasi rejim?

- A.Codes Display
- B.Normal Display
- C.Password Display
- D.HEX Display

45.LabVIEW da matnli qatorlarni yuzadagi panelda aks etishning uchinchi rejimi qasi rejim?

- A.Password Display
- B.Normal Display
- C.Codes Display
- D.HEX Display

46.LabVIEW da matnli qatorlarni yuzadagi panelda aks etishning to'rtinchi rejimi qasi rejim?

- A.HEX Display
- B.Normal Display
- C.Codes Display
- D.Password Display

47. Yuzadagi panelda matnli boshqarish elementi joylashtirilgan va unda "Dasturlash tillari bazasida texnik ilovalar" yozuvi kiritilgan. Bundan tashqari yuzadagi panelda sonli indikator ham joylashtirilgan. Diagrammalar blokida terminallar orasida String Length funksiyasini qo'llash natijasida sonli indikatorlarda nima akslanadi?

- A.43
- B.Dasturlash tillari bazasida texnik ilovalar
- C.Dasturlash tillari
- D.100

48. LabVIEW fayllardan ma'lumotlarni tezkor o'qish yoki yozish uchun qaysi funktsiyalardan foydalanishni maslahat beradi?

- A. Write to Measurement File va Read From Measurement File
- B. Write to Spreadsheet File va Read From Spreadsheet File
- C. Write to Text File va Read from Text File
- D. Read from Binary File va Write to Binary File

49. LabVIEW fayllardan jadval ko'rinishidagi(formati) ma'lumotlarni o'qish yoki yozish uchun qaysi funktsiyalardan foydalanishni maslahat beradi?

- A. Write to Spreadsheet File va Read From Spreadsheet File
- B. Write to Measurement File va Read From Measurement File
- C. Write to Text File va Read from Text File
- D. Read from Binary File va Write to Binary File

50. LabVIEW fayllardan matnli ma'lumotlarni o'qish yoki yozish uchun qaysi funktsiyalardan foydalanishni maslahat beradi?

- A. Write to Text File va Read from Text File
- B. Write to Measurement File va Read From Measurement File
- C. Write to Spreadsheet File va Read From Spreadsheet File
- D. Read from Binary File va Write to Binary File

51. LabVIEW fayllardan ikkilik sanoq tizimidagi ma'lumotlarni o'qish yoki yozish uchun qaysi funktsiyalardan foydalanishni maslahat beradi?

- A. Read from Binary File va Write to Binary File
- B. Write to Measurement File va Read From Measurement File
- C. Write to Spreadsheet File va Read From Spreadsheet File
- D. Write to Text File va Read from Text File

III-MODUL. KOMPYUTERGA MA'LUMOTLARNI KIRITISH CHIQARISH. MA'LUMOTLARNI YIG'ISH VA LABVIEW USKUNALARINI BOSHQARISH.

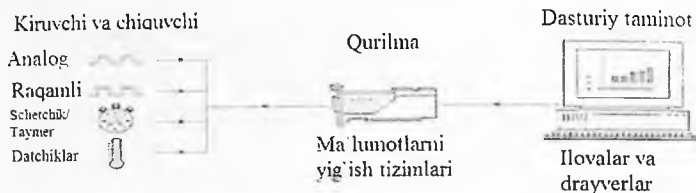
§ 3.1. KOMPYUTERGA MA'LUMOTLARNI KIRITISH/CHIQARISH

Qanday qilib kompyuter tashqi dunyo bilan ulanadi?

Sizda yaxshi personal kompyuter bor deb faraz qilaylik va siz *LabVIEW* ilovasi yordamida kompyuter chegarasidan tashqaruda biror ishni bajarish uchun ishlatmoqchisiz. Bunda kompyuterda nima ishni bajarishingizdan qat'iy nazar kompyuterga ma'lumotlarni kelib tushish yo'lini toppishiz kerak. Ko'pincha buning uchun bir necha imkoniyatlar bor, lekin eng yaxshi yechim sizni o'ziz qancha va qanday harakat qilishizga bog'liq bo'ladi. Apparat qismini sotib olishdan oldin, qanaqa siganllarni o'lchashni istashingizni tahlil qilib ko'ring. Ma'lumotlarni yig'ish tizimlarini yaratishda birinchi yechishingiz kerak bo'lgan narsa shuki - multimetr kabi an'anaviy tashqi qurilmalarni ishlatish kerakmi degan savolga javob berishingiz kerak. Siz fizik qurilmani ma'lumotlarni yig'ish va qayta ishlash uchun ishlatasizmi yoki *LabVIEW* da hamma vazifalarni o'zida jamlagan ma'lumotlarni yig'ish platalari bilan ishlash uchun virtual qurilma yaratishdan foydalanasizmi?

Narx, jadval bo'yicha ishlash va egiluvchanlik kabi tushunchalar ham qaror qabul qilishda muhim o'rin egallaydi. Masalan, past quvvatli elektr signal bilan bog'liq ba'zi bir ma'lumotlarni hisob-kitob qilish uchun siz oddiy kirish/chiqish platalaridan foydalanishingiz mumkin. Siz bu maqsadga to'g'ri keladigan virtual qurilma yaratadigan dastur yozib bilasiz. Umumiy holda olganda sizga o'ziga mujassam kirish/chiqish platasini olishingiz aloxida qurilma olishdan ko'ra ancha arzonroq

tushadi. Lekin sizda agar tayyor qurilma bo'lsa, albatta sizga bu qurilma bilan quvvatni o'lchab kompyuter portiga uzatish yaxshiroq yechim bo'ladi.



3.1.1-rasm. Ma'lumotlarni yig'ish qurilmasi yordamida kompyuterni real dunyo bilan ulash umumiy sxemasi.

Va nihoyat agar siz ma'lumotlarni kiritish/chiqarish platasini sotib olmoqchi bo'lsangiz yoki sotib olgan bo'lsangiz uni 100% imkoniyatidan ishlatishga harakat qiling. Ko'pchilik odamlar kiritish/chiqarish platali kompyuterlarining to'liq potentsialini hisobga olmaydilar. Faqatgina bitta plata bilan siz o'zizga kerak bo'lgan ko'plab virtual qurilmalar tuzishiz mumkin. *LabVIEW* virtual qurilmalari kompyuter ishlatadigan kartalarga bog'liq bo'lmagan holda ishlaydi.

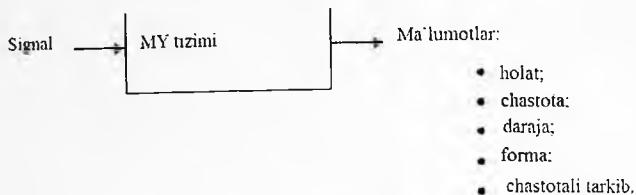
Signallar

Ma'lumotlarni yig'ishga to'la sho'ng'ishdan oldin, o'zi nimani biz qabul qilmoqchi bo'lganimiz haqida gapirishimiz kerak. Signal - bu oddiy fizik kattalik bo'lib, uning amplitudasi va vaqt ichida o'zgarishi ma'lumotni saqlaydi. *Vaqt parametrlari - eng asosiysi*. Vaqt - hamma o'lchovlarda eng asosiy kattalik bo'lib hisoblanadi. Istalgan jarayonda xox u vaqt ichida dvigatel temperaturasi o'zgarishi bo'lsin, xox filtrlangan audio signal qanday bo'lishini ko'rmoqchi bo'lsak yoki gaz o'zing optimal sig'imiga ega bo'lganda klapanlarni yopish bo'lsin, ma'lumotlarni yig'ish va boshqarish uchun vaqt eng asosiy kattalik bo'lib hisoblanadi. Doimiy amplitudali signallar ham bir xil bo'lib qolmaydi. Agar ular vaqt ichida o'zgarmasa, biz uning

kattaligini doim bilar edik. Unday bo'lsa uni o'lchash ham kerakmas bo'lar edi. Signalarni vaqt parametrlari bo'yicha yig'ish dasturlarni tuzishga ikkita muhim sabab mavjud. Birinchidan sizga kompyuter qanday vaqt oralig'ida o'lchashlarni amalga oshiradi. Ikkinchidan protsessor vaqti ketadigan boshqa amallar uchun vaqtni ajratsih kerak: masalan: ma'lumotlarni faylga yozish yoki ularni fayldan o'qish kabi amallarga ketadigan vaqtini ajratib bi'lishi kerak. Agar ma'lumotlarni sekundiga 1 yoki 2 marotaba yoki undan ham kam marotaba o'qish kerak bo'lsa va sizga yuqori aniqlik kerak bo'lmasa, siz LabVIEW ning tanlanishlar chastotasini boshqarish vazifasi bilan dasturni o'zidan sozlash imkonigingiz mavjud.

Faraz qilaylik sizga qandaydir o'lchashlarni olib borish kerak. Signalni apparat o'zgartgich bilan yoki kirish/chiqish platasi bilan o'zgartirish uchun uni birinchi elektr-quvvat yoki tokka o'zgartirishiz kerak. Bunday vazifani o'zgartgichlar bajaradi(datchiklar). Masalan siz temperaturani o'lchamoqchi bo'lsangiz, siz uni qandaydir yo'l bilan quvvat ko'rinishiga keltirib, uni kiritish/chiqarish platasi o'qib biladigan formatga aylantirishiz kerak. Termodinamikaning ba'zi tamoyillari va materialning fizik xususiyatini hisobga olib uni elektr signalga o'zgartiruvchi temperatura o'zgartgichlarning ko'p turlari mavjud. Fizik kattalik elektr signal ko'rinishida kelganidan so'ng, siz kerakli ma'lumotlarini olish uchun uni o'lchash imkoniyatiga ega bo'lasiz, qaysiki bir yoki undan ortiq parametrlar orqali uzatiladi, bularga: holat, kattalik, tezlik, forma va chastotaviy spektr kiradi.

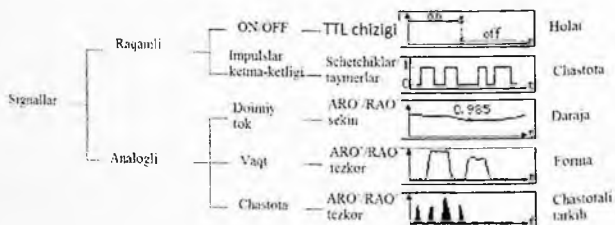
Umuman olganda, hamma signallar analogli va vaqt ichida o'zgaradi. Ammo o'lchash usulini muhokama qilish uchun siz berilgan signalni klassifikatsiyalashtirishingiz kerak.

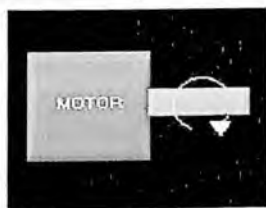
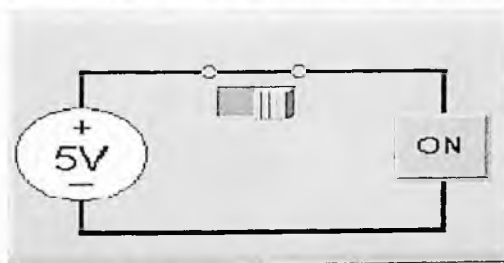
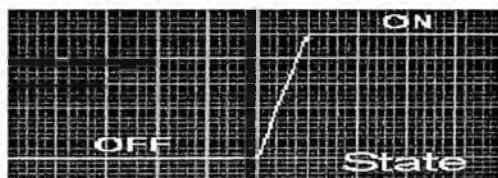
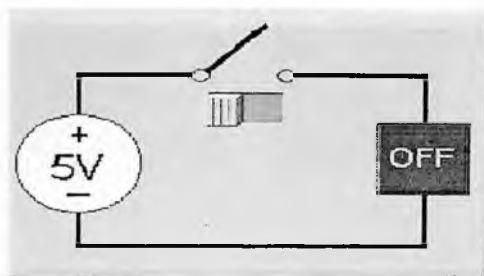


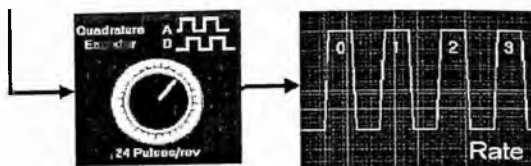
3.1.2-rasm. Ma'lumotlarni yig'ish tizimi signallarni ma'lumotga o'tkazadi, qaysiki ularni dasturiy ta'minotni ishlatish mumkin

Signallarni klassifikatsiyalashni ma'lumotlarni uzatishi bo'yicha amalga oshirish qabul qilingan. Signallarning besh turi mavjud: Birinchi navbatda signallar analog yoki raqamli bo'lishi mumkin. Raqamli(yoki ikkilik) signal faqat ikkita diskret holatga ega – yuqori va past. Analog signal esa aksincha vaqt amplitudasida uzluksiz o'zgarish ma'lumotlarini saqlaydi.

Mutaxassislar odatda raqamli signallarni ikki ko'rinishga, analog signallarni esa uchinchi ko'rinishga ajratishadi. Ikki raqamli signalni ko'rinishi – bu yuqori darajadan (on) past darajaga (off) o'tkazish va impulslar seriyasi ko'rinishidagi signallar. Uch analog signal ko'rinishi: doimiy signal sifatida ifodalanadi, vaqt hududida(time domain) o'zgaruvchan signal, chastota hududida(frequency domain) o'zgaruvchan signal(3.1.3-rasm). Signalning ushbu 5 ko'rinishi informatsiyani uzatish 5 ko'rinishiga mos keladi: holat, chastota, daraja, forma, chastotali spektr.







3.1.3-rasm. Signallarni ko'rinishi.

Signallar shakllantirish va o'tkazish

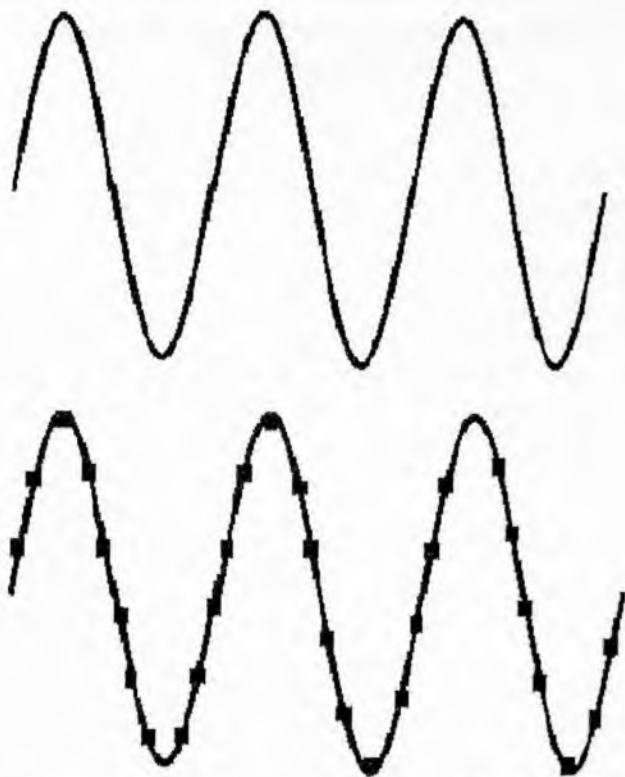
Qanday sizignalni olishni aniqlagandan so'ng, o'tkazuvchi (datchikning) chiqishini kiritish/chiqarish platasiga tog'ri ulash zarur. Signal ma'lumotlarni yig'ish platasiga yetib borguncha, unda shu qadar shovqin juda ko'p bo'ladiki, signal xatolik keltiradi. Signal plataga juda kichik shovqin bilan kuchlanish chegarasida(+5V dan to -5V gacha yoki 0-10 V oralig'ida) bo'lishi kerak va tokda(20mA atrofida). Maxsus qayta ishlash zarur bo'lgan signallar yoki juda ko'p signallarni generatsiyalaydigan o'rnatmalar uchun NI kompaniyasi SCXI (Signal Conditioning eXtensions for Instrumentation) tizimini ishlab chiqqan.



3.1.4-rasm. NI kompaniyasi SCXI (Signal Conditioning eXtensions for Instrumentation) tizimi uskunolari.

Signallarni qayta ishlashning turlarining eng keng tarqalganlari bular:

- kuchaytirish;
- o'tkazgichni boshqarish;
- linerazatsiyalash;
- elektr zanjirlarini izolyatsiyalash;
- filtrlash.



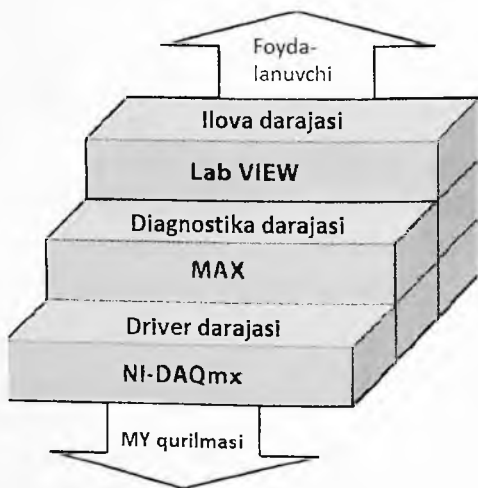
3.1.5-rasm. Uzlusiz signalni qayta ishlash ko'rinishi.

Ma'lumotlarni yig'ish tizmi o'lchash apparat qismini tanlash va konfiguratsiyalash

Agar siz qanday turdagi signallarni o'lchashni bilsangiz, kiritish/chiqarish platasini tanlash vaqti keldi (agar u sizning talablarizga javob bersa). Agar siz *abVIEW* ni ishlatsangiz, unda NI kompaniyasi yaratgan platalardan foydalanishni maslahat beramiz.

Barcha platalar o'z drayverlariga ega, ya'ni quyi darajadagi kod bo'lib, kompyuterni plata unda ulangan deb hisoblashga ishonitiradi. NI ning barcha platalari yagona paketda drayverlarni birlashtirgan. Bu maxsulot NI-DAQmx deyiladi. NI-DAQmx va *LabVIEW* o'rtasida bog'lovchi dastur funktsiyalanadi hamda u MAX (Measurement and Automation Explorer) deb nomlanadi.

MAX Windows operatsion tizimining dasturiy interfeysi hisoblanib, u barcha NI platalariga murojaat etish imkonin beradi. Bu dasturni siz *LabVIEW* da platani ishlatishdan oldin sozlashingiz zarur. MAX asosan apparat qismni konfiguratsiyalash va testlash uchun ishlatiladi.





3.1.6-rasm. MAX dasturi asosiy oynasi. NI-DAQmx

NI-DAQmx – NI kompaniyasining ma’limotlarni yig’ish uskunalari uchun drayverlar to’plami hisoblanib, quyidagi yutuqlarga ega:

- yaxshilangan holat modeli;
- ko’poqimli drayver;
- to’xtab qolishda yuqori barqarorlik;
- juda oddiy sozlash;
- LabVIEW da VIU kamligi;

MAX da uskunalarini sozlash haqida ma’lumot keltirish vaqti keldi. Agar sizda bunday uskuna bo’lmasa, NI-DAQmx sizga uskunani imitatsiyalash imkonini beradi. MAX ning My System >> Devices and Interfaces >> NI-DAQmx devices bo’limida barcha real va imitatsiyalanadigan uskunalar ro’yxati mavjud bo’ladi.

Takrorlash va mustaqil nazorat savollari:

- 1.Qanday qilib kompyuter tashqi dunyo bilan ulanadi?
- 2.Signallar nima va ularning turlari haqida nimani bilasiz?
- 3.Signallar qanday shakllantiriladi?
- 4.Ma’lumotlarni yig’ish tizmi o’lchash apparat qismini tanlash va konfiguratsiyalash nima uchun kerak?
- 5.NI-DAQmx nima?

§3.2. Ma'lumotlarni yig'ish va LabVIEW uskunalarini boshqarish.

Analog va raqamli ma'lumotlarni yig'ish

Measurement I/O >> DAQ mx – Data Acquisition(NI-DAQmx ma'lumotlarni yig'ish) ichki palitrasida ma'lumotlarni yig'ish VIU lari va boshqa uskunalari joylashgan bo'lib, ular LabVIEW ning asosiy yutuqlaridan biri hisoblanadi(3.2.1- rasm).



3.2.1-rasm. NI-DAQmx ma'lumotlarni yig'ish palitrasi.

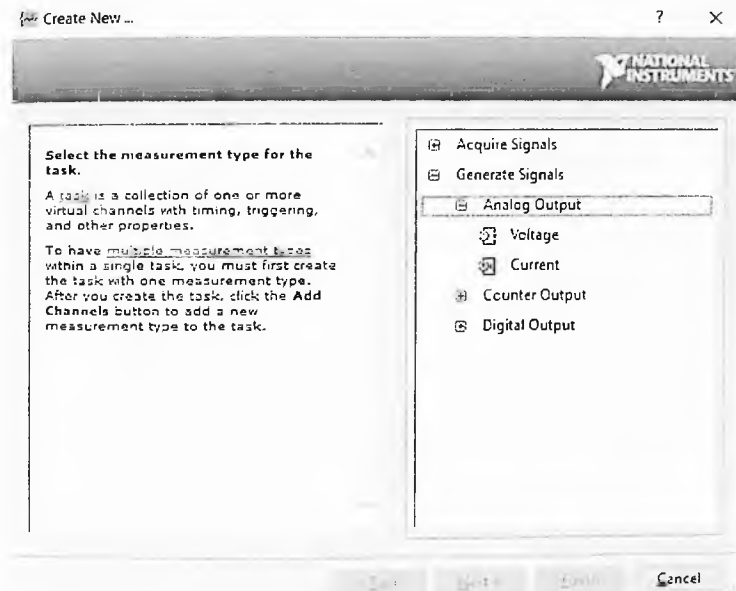
DAQ Assistantdan foydalanish

Analogli va raqamli qiymatlarni kiritish-chiqarish LabVIEW da DAQ Assistant tufayli juda oddiy bajariladi. DAQ Assistant - bu EVU bo'lib, NI-DAQmx masalalarini yaratadi, tahrirlaydi va ishga tushuradi (3.2.2-rasm).



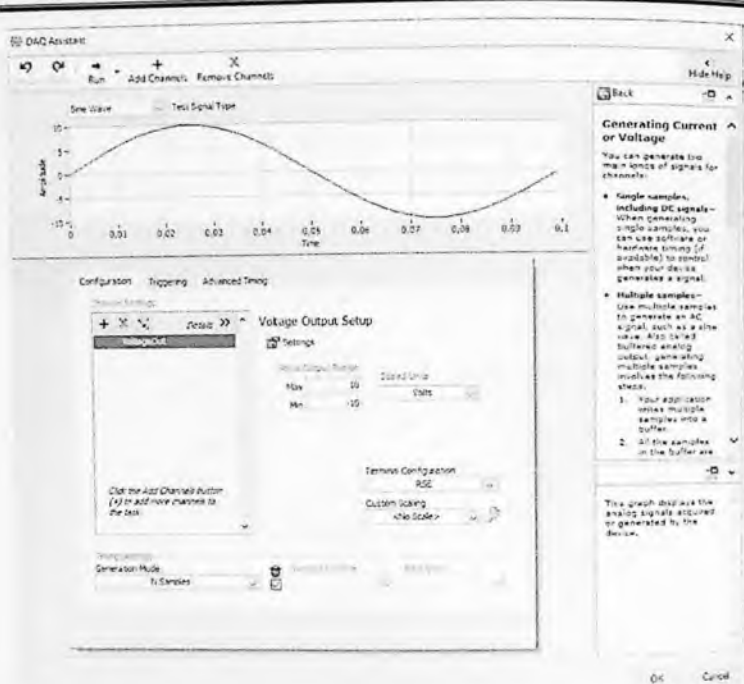
3.2.2-rasm. DAQ Assistant EVU

Bu EVU ning sozlash oynasi ochilganda (masalan, uni diagramma blokida joylashtirganda), 3.2.3-rasmda ko'rsatilgandek, signalni o'lchash (input) yoki generatsiya qilish (output) mumkin.



3.2.3-rasm. DAQ Assistant – sozlash oynasi

Bu oynada NI-DAQ mx qo'llab-quvvatlaydigan ma'lumotlarni yig'ish yoki generatsiyalashning ixtiyoriy masalasini tanlash mumkin. Masala tanlangach, konfiguratsiya uchun NI-DAQmx sozlash oynasi ochiladi(3.2.4-rasm).



3.2.4-rasm. DAQ Assistant masalani konfiguratsiyalash oynasi.



3.2.5-rasm. NI-DAQmx vazifalari va ishlash umumiy sxemasi.

Oldingi paragrafda biz virtual kanallarni MAX tizimida yaratishni aytib otgan edik. Bu virtual kanallar sizning

ma'lumotlarni yig'ish uskunangizning fizik kanallariga mos kelib, ulardan NI-DAQmx VIU yordamida ma'lumotlarni o'qish yoki yozish mumkin. Endi vazifalar haqida gapiraib o'tamiz – ular ma'lumotlarni yig'ish dasturlarini yaratish qulay vositalari hisoblanadi.

Vazifalar –NI-DAQmx harakatlarini bajarilish asosi hisoblanadi. Bu bir yoki bir necha virtual kanallar to'plami bo'lib, mos taktlash, sinxronlash va boshqa xususiyatlarga ega.

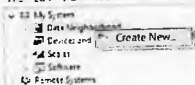
Real vazifa - bu sizning o'lchovingiz(signalni kiritish) yoki generatsiya(chiqarish). Vazifaning barcha kanallari bir tipda bo'lishi kerak, masalan: analogli kirish yoki taktli impulsli generatsiya. Ba'zi uskunalarda bir vazifaga turli uskunalarga tegishli fizik kanallarni joylashtirish mumkin. Bu vazifani amalga oshirish uchun quyidagilarni bajarish zarur:

- Vazifani yarating yoki yuklang;
- Kanallarni sozlang;
- Ishga tushurishga tayyorlash uchun aniq harakatlarni o'rnatish(agar zarur bo'lsa);
- Hisoblashlarni o'qing yoki yozing;
- Vazifani tozlang.

MAX da vazifani qo'shish uchun Data Neighborhood yordamchi menyusida "Create New .." punktini tanlang(3.2.6-rasm).

Paydo bo'lgan oynadan NI-DAQmx vazifasini(NI-DAQmx task) tanlang(3.2.7-rasm) va NEXT tugmasini tanlang.

O'lchash muhitini tanlang(3.2.8-rasm). Uskuna ro'yxatidan bir yoki bir necha fizik kanalani tanlang(3.2.9-rasm). Vazifaga nom bering(3.2.10-rasm) va FINISH tugmasini tanlang.



Data Neighborhood

Data Neighborhood provides access to descriptively named shortcuts to configured phy

What do you want to do?

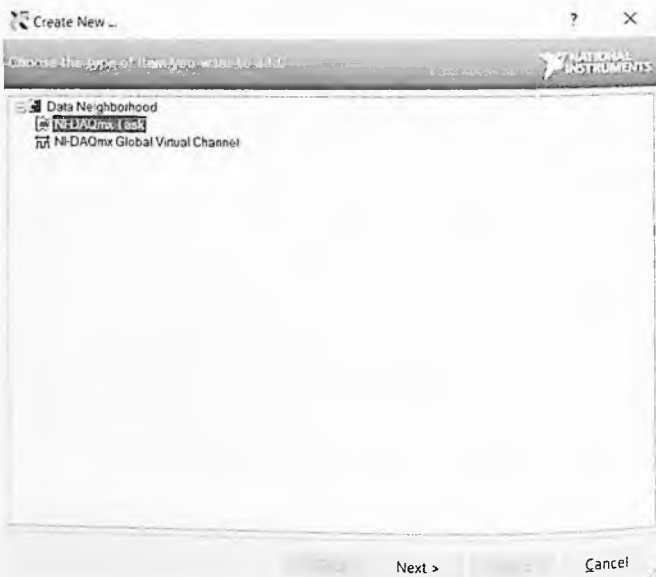
- ☐ Create a new shortcut
- ☐ Edit or remove a shortcut
- ☐ View or change properties

For more information about using Data Neighborhood with your NI product, refer to your launch from the Help menu or by pressing <F1>

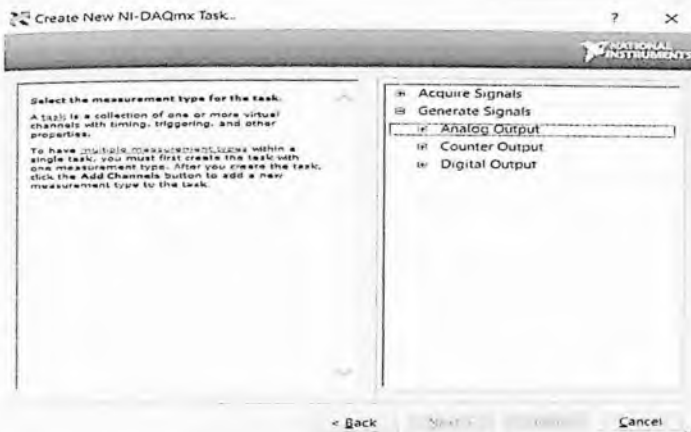
Submit feedback on this topic

Visit ni.com/support for technical support

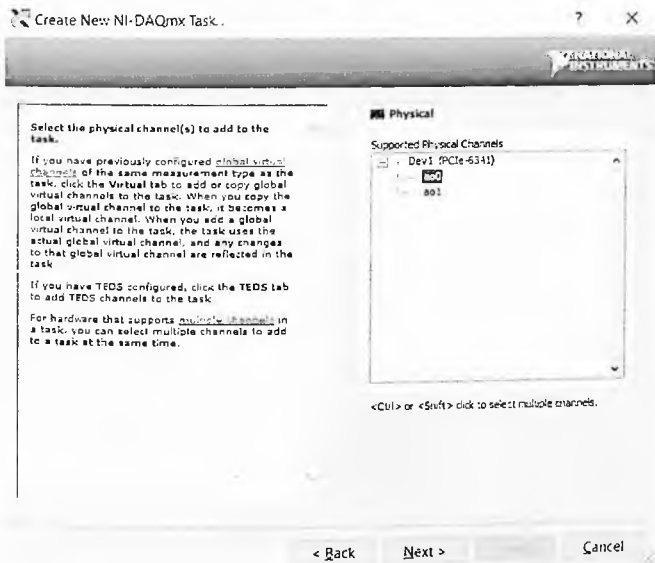
3.2.6-rasm. MAX da vazifani qo'shish uchun Data Neighborhood yordamchi menyusi



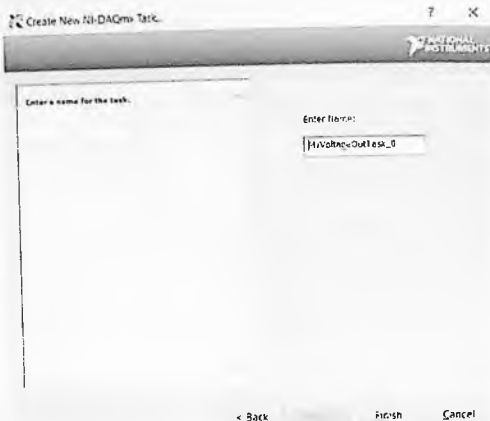
3.2.7-rasm. NI-DAQmx vazifasini yaratish



3.2.8-rasm. O'lchash muhitini tanlash.



3.2.9-rasm. Bir yoki bir necha fizik kanalani tanlash.



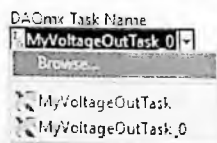
3.2.10-rasm. Vazifaga nom berish

Biz vazifa yaratishni ko'rib o'tgandan so'ng, undan *LabVIEW* da foydalanishni ko'rib o'tamiz. NI-DAQmx vazifa nomi konstantasini (*DAQmx Task Name constant*) *LabVIEW* da diagramma blokida joylashtiring. Bu element Measurement I/O >> DAQmx- Data Acquisition bo'limidan topishingiz mumkin (3.2.11-rasm).

Boshqarish elementi yordamida MAX da yaratilgan vazifani tanlash mumkin (3.2.12-rasm).

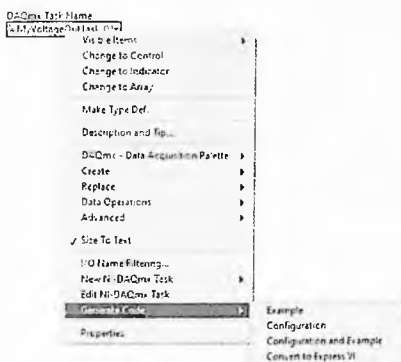


3.2.11-rasm. Diagramma blokida DAQmx vazifasini joylashtirish.

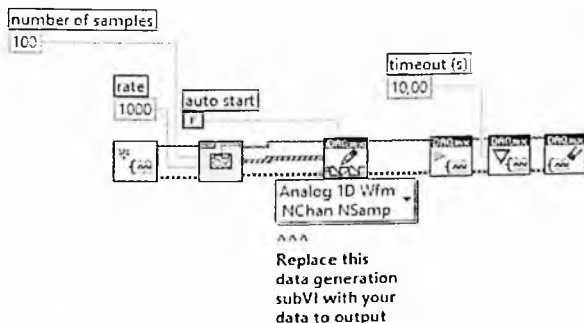


3.2.12-rasm. MAX dasturida yaratilgan vazifani diagramma blokida tanlash

Sizga NI-DAQmx vazifasi orqali ma'lumotlarni qabul qilish boshlashdan oldin qancha ish bajarish zarurligini bilasizmi? Bu ishni *LabVIEW* sizning o'rningizdan bajarishi mumkin. DAQmx vazifasi konstantasi yordamchi menyusidan Generate Code >> opsiyasi tarkibida kodni avtomatik generatsiyalash variantlari berilgan(3.2.13-rasm).



3.2.13-rasm. DAQmx vazifasi konstantasi yordamchi menyusidan Generate Code opsiyasi



3.2.14-rasm. DAQmx vazifasi konstantasi yordamchi menyusidan Generate Code opsiyasini ishlatgandan keying holat.

NI-DAQmx vazifasini LabVIEW da ishlatilishi

Har bir NI-DAQmx ning VU ishini tahlil qilishdan oldin NI-DAQmx VU sidan foydalanishni umumiy xaritasini keltirishga harakat qilamiz. Bu juda oson va bunda bir nechta qadamni eslab qolish zarur:

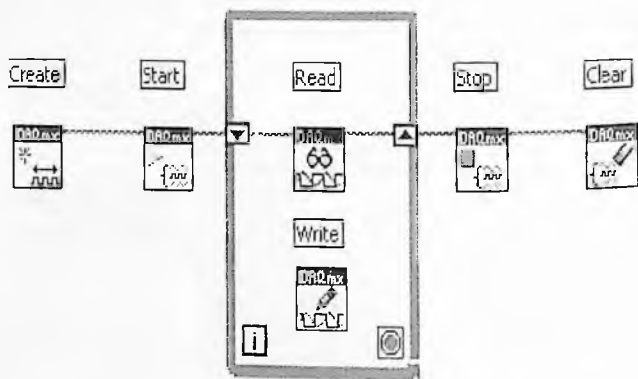
1. Vazifani yaratish (yoki NI-DAQmx vazifasiga murojaat);
2. Vazifani tayyorlash;
3. Kerakli miqdorda ma'lumotlarni o'qish yoki yozish;
4. Vazifani to'xtatish;
5. Vazifani tozalash.

Shunday qilib, deyarli barcha NI-DAQmx ilovalarining tuzilishi 3.2.15- rasmni eslatadi. Shuni esda saqlash kerak:

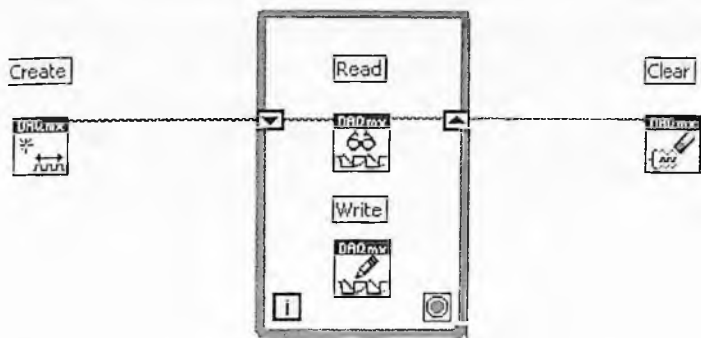
➤Tayyorlanishi (start) – majburiy bo'lmagan punkt, VU o'qish va yozish masalasini avtomatik tayyorlaydi;

➤To'xtatish (Stop) ham majburiy emas, chunki masalani tozalashdan oldin u avtomatik o'zi to'xtaydi.

Shuning uchun sizning NI-DAQmx ilovangiz 3.2.16-rasmdagidek ko'rinishda bo'lishi mumkin.



3.2.15-rasm. NI-DAQmx ilova tuzilishining diagramma bloki.

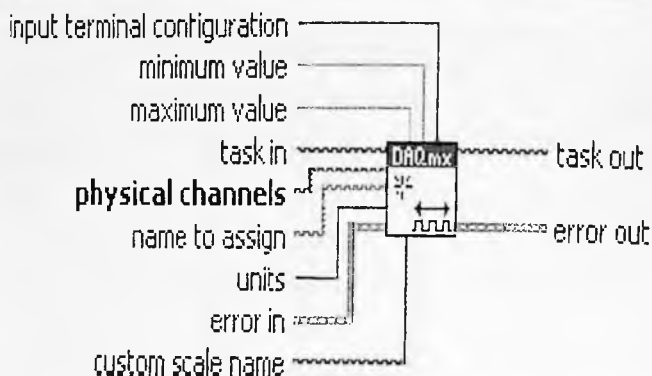


3.2.16-rasm. NI-DAQmx ilova tuzilishinign masalani tayyorlash va to'xtatish bosqichisiz diagramam bloki.

Virtual kanalni yaratish va NI-DAQmx masalasi

LabVIEW da NI-DAQmx bilan ishlashdagi birinchi qadam – bu vazifani yaratish. Qanday qilib MAX da vazifa yaratish va LabVIEW da undan NI-DAQmx masala nomi konstantasidan foydalanish haqida siz deyarli bilasiz. Masala yaratishning boshqa bir yo'li bu dasturiy usul – VU DAQmx Create Virtual Channel (Measurement I/O » DAQmx Data Acquisition ichki palitrasi).

Bu VU (3.2.17-rasm) virtual kanal yoki virtual kanallar to'plamini yaratadi va ularni vazifaga qo'shadi. Bu polimorfik VU ning variantlari masala turini tanlashga imkon beradi: analog yoki raqamli generatsiya yoki registratsiya, schetchiklar, masala turi: temperatuerani o'lchash, kuchlanishni generatsiyalash, hodisalarni hisobga olish, hattoki, foydalaniladigan datchik: temperaturani o'lchashda termopara yoki termoqarshilik.



3.2.17-rasm. VU DAQmx Create Virtual Channel elementi.

Takrorlash va mustaqil ishlash uchun savollar:

1. Analog va raqamli ma'lumotlarni farqi nimada?
2. DAQ Assistant nima?
3. DAQ Assistantdan nima maqsadda foydalaniladi?
4. NI-DAQmx vazifalari nima?
5. NI-DAQmx vazifasini LabVIEW da qanday maqsadda foydalaniladi?

§3.3. Mahalliy, global va tarmoq o'zgaruvchilari

Mahalliy o'zgaruvchilar

Mahalliy va global o'zgaruvchilar bular oddiy til bilan aytganda LabVIEW ning tuzilmalari hisoblanadi. Agar siz ilgari C yoki Pascal dasturlash tillariga o'xshagan dasturlash tillarida dastur tuzgan bo'lsangiz yoki o'rgangan bo'lsangiz, unda mahalliy va global o'zgaruvchilar g'oyasini darhol tushunasiz. Shu paytgacha biz LabVIEW da yuzadagi panel ob'yektidan ma'lumotlarni uning terminali yordamida diagramma blokida o'qib olishimiz(yoki yozish) mumkin edi. Ammo yuzadagi panel

elementlari diagramma blokida faqat bir terminalga ega bo'lishadi, lekin sizga bu yuzadagi panel elementi qiymatlarini diagramma blokining turli nuqtalaridan yoki boshqa VIU dan o'zgartirish yoki o'qib olish talab etilishi mumkin.

□ *Mahalliy o'zgaruvchilar* diagrammalar blokining turli nuqtalaridan o'tkazuvchi ma'lumotlarini ob'yekt terminaliga ulash imkoniyatingiz bo'lmagan holatda yuzadagi panel ob'yektiga murojaat etish imkonini beradi;

□ *Global o'zgaruvchilar* ixtiyoriy tipdagi qiymatlarga(yoki bir necha tipga bir vaqtda) bir necha VU orasida bir vaqtinich o'zida bir nechta VU baravar ishlaganda yoki VIU lar orasida signallarni uzatish imkoniyatingiz bo'lmaganda ularga murojaat etish imkonini beradi;

□ *Umumiy(tarmoq) o'zgaruvchilari* esa xuddi global o'zgaruvchilarga o'xshaydi, lekin ular turli mahalliy va tarmoq dasturlarida ishlaydi.



LabVIEW ning mahalliy o'zgaruvchilari Programming >> Structures ichki palitrasida joylashgan paneldan foydalanish mumkin(3.3.1-rasm). Siz birinchi marotaba mahalliy o'zgaruvchini tanlaganda "?" belgisi paydo bo'lib, bu o'zgaruvchi

aniqlanmaganini bildiradi. Siz mahalliy o'zgaruvchini ixtiyoriy terminal yoki yuzadagi panel ob'yektining yordamchi menyusidan Create >> Local Variable punktini tanlab hosil etishingiz ham mumkin.

3.1-rasm. Mahalliy o'zgaruvchilar palitrasi

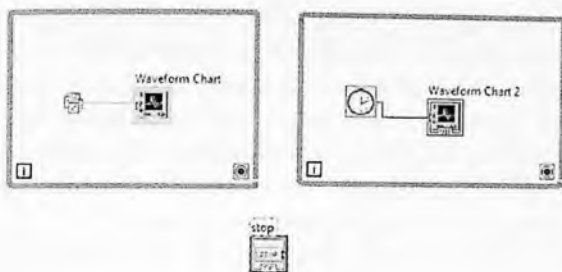


3.3.2-rasm. Yuzadagi panel ob'yektini diagramma blokida mahalliy o'zgaruvchi bilan bog'lanishi

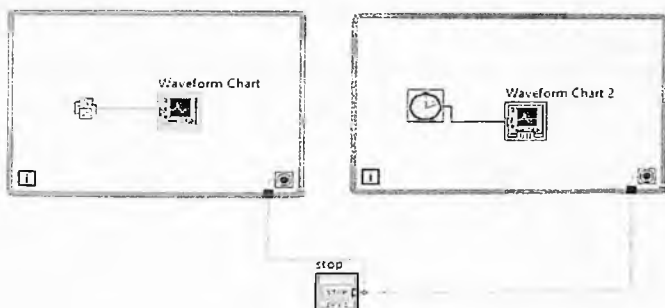
LabVIEW da mahalliy o'zgaruvchilarni ishlatishning bir nechta sabablari bo'lishi mumkin:

- ☐ Bir o'zgaruvchi yordamida parallel tsikllarni nazorat etish mumkin, buni boshqa usulda amalga oshirib bo'lmaydi;
- ☐ Ixtiyoriy boshqarish elementi indikator va aksincha boshqarish elementi sifatida ishlatilishi mumkin.

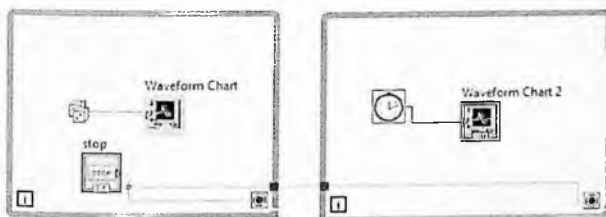
Ba'zan katta dasturlarda shunday holat yuzaga keladiki, sizga yuzadagi panel ob'ektlaridan ularning terminallariga to'g'ridan-to'g'ri ulanmasdan turib, ma'lumotlarni o'qish (yoki yozish) zaruriyati tug'iladi. Quyida klassik muammo keltirilgan: bizga ikki mustaqil While siklini bir mantiqiy element yordamida to'xtatish zarur bo'lsin(3.3.3-rasm).



3.3.3-rasm. Ikkita While sikli va bir mantiqiy "Stop" tugmasini ulash(muammo)



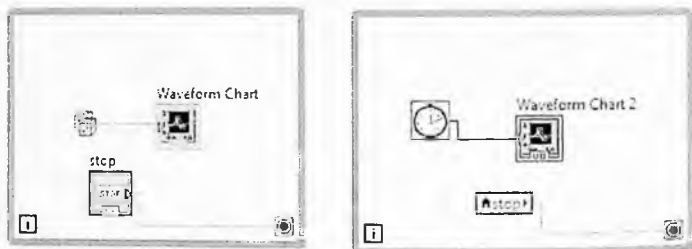
3.3.4-rasm. Tsikllardan tashqarida STOP tugmasini ikkita tsiklni to'xtatish uchun ulash



3.3.5-rasm. Bir tsikl ichidan STOP tugmasini ikkinchi tsiklni to'xtatish uchun ulash.

Bu muammoning yechimi albatta mahalliy o'zgaruvchilarni ishlatish hisoblanadi. Mahalliy o'zgaruvchilar ma'no jihatidan boshqa terminalning nusxasini diagramma blokida yaratishadi. Mahalliy o'zgaruvchilar yuzadagi panel ob'yektining eng yangi qiymatini o'zida saqlaydi. Shunday qilib, diagramma blokining turli bir nechta nuqtalarida bir boshqarish elementi yoki bir indikatorga murojaat etish imkoningiz tug'iladi.

Oldingi misolga qaytadigan bo'lsak, biz "Stop" tugmasini ikkala tsikl uchun ham ishlatishimiz mumkin, albatta birinchi tsiklda mantiqiy tugma terminalini tsiklni to'xtatish terminaligi ulab, unga bog'liq mahalliy o'zgaruvchini esa ikkinchi tsikl chiqish terminaliga bog'laymiz.



3.3.6-rasm. "Stop" tugmasi birinchi tsiklda va mahalliy o'zgaruvchi ikkinchi tsikl ichida ulash.

Mahalliy o'zgaruvchilarning yoqimli xususiyatlaridan biri – bu boshqarish elementlariga ma'lumotlarni yozishimiz mumkin va indikatorlardan ma'lumotlarni o'qib olishimiz mumkinligi. Mahalliy o'zgaruvchilar ikkita rejimda bo'lishadi: o'qish va yozish. Mahalliy o'zgaruvchi terminali bir vaqtning o'zida ikkala rejimdan birida bo'lishi mumkin, lekin siz bu o'zgaruvchining yana bir terminalini yaratish uni boshqa rejimga o'tkazish mumkin.

Avvalgi tushunchalar bilan chalkashmaslik uchun mahalliy o'zgaruvchilarni ishlatish paytida quyidagi formulani eslab qolish kifoya:

O'QISH rejimi = boshqarish elementi;

YOZISH rejimi = ask etish elementi;

Mahalliy o'zgaruvchi rejimini o'rnatish uchun uning terminali yordamchi menyuisida "Change to ..." punktidan foydalaning. Mahalliy o'zgaruvchi o'qish rejimida uning terminali qalin ramkaga ega, yozish rejimida esa mayin(3.3.7-rasm).

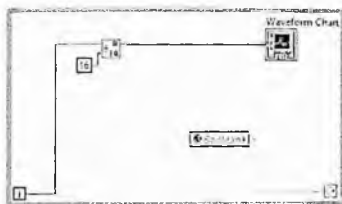
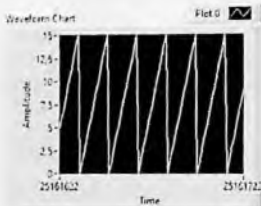
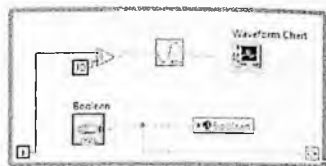
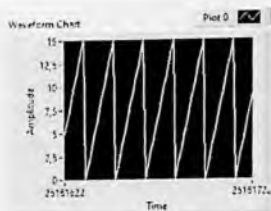


3.3.7-rasm. Mahalliy o'zgaruvchi o'qish va yozish rejimlarida.

Mahalliy o'zgaruvchilar juda yaxshi tuzilmalar bo'lishi bilan birga(bu haqiqatdan ham shunday), siz mahalliy o'zgaruvchilarni ishlatish jarayonida, ular bilan bog'liq: musobaqa holati bo'lishini oldini olishingiz zarur. Agar ikki va undan ko'p mahalliy o'zgaruvchilar yozish rejimida bir vaqtning o'zida ularda ma'lumot yozilsa, unda musobaqa holati yuzaga keladi.

Global o'zgaruvchilar

Eslasangiz agar *LabVIEW* da mahalliy o'zgaruvchilarni ishlatganda diagramma blokining turli nuqtasidan yuzadagi panel ob'ektlariga murojaat etish mumkin edi. Bu o'zgaruvchilar faqat bir VU ichida murojaatli. Tasavvur qiling, sizga bir VU dan boshqasiga ma'lumotlarni uzatish zarur bo'lsin, bunda siz VIU lardan foydalana olmaysiz. Masalan VU lar baravar ishlayotgandagi holat. Bu holatda siz global o'zgaruvchilarni ishlatishingiz mumkin. Aytib o'tish kerakki, global o'zgaruvchilar mahalliy o'zgaruvchilarga o'xshaydi, faqat ularning terminalida "dunyo" ikonkasi bor!



3.3.8-rasm. Global o'zgaruvchilarni ishlatish.

Mahalliy o'zgaruvchilar singari global o'zgaruvchilar tuzilmasini Programming >> Structures ichki palitrasidan toppish mumkin. Global o'zgaruvchi ham mahalliy o'zgaruvchi kabi ikki rejimda bo'ladi: o'qish va yozish. Faqat ulardan farqli ravishda bu o'zgaruvchilarga xoxlagan VU dan bemalol murojaat etish imkoniyati mavjud. Global o'zgaruvchilar bir nechta VU lar o'rtasida ma'lumotlarni almashinishning eng effektiv usullaridan biridir(albatta VIU ishlatib bo'lmaslik holatida).

Global o'zgaruvchi tuzilmasi diagramma blokida joylashtirilgandan so'ng, noaniq global o'zgaruvchi paydo bo'ladi. Siz ikonkaga sichqoncha chap tugmasi bilan ikki marotaba murojaat etganda, VU yuzadagi paneliga o'xshash panel paydo bo'ladi. Global o'zgaruvchini ixtiyoriy tipdagi va ixtiyoriy miqdordagi ma'lumotlarga ega yuzadagi panel deb tasavvur etish mumkin, bunda ularning diagramma bloki mavjud bo'lmaydi(3.3.9-rasm). Global o'zgaruvchi panelida ob'ektlarni

joylashtirganda har bir ob'yektga belgilar qo'yish esizdan chiqmasin, aks holda ularni ishlatib bilmaysiz!



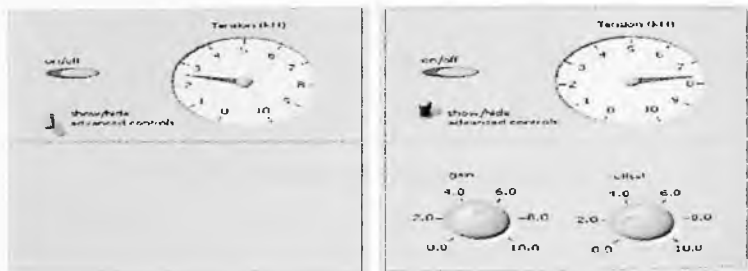
3.3.9-rasm. Global o'zgaruvchining yuzadagi paneli.

Global o'zgaruvchilarni xuddi VU kabi xotirada saqlash mumkin(ko'pgina professional dasturchilar global o'zgaruvchilar yuzadagi panelini nomlashda "_gb1" singari suffekslardan foydalanishadi). Oldin saqlangan global o'zgaruvchilardan foydalanish uchun Function palitrasidan Select VI opsiyasini ishlatishingiz mumkin. Global o'zgaruvchida qaysi komponentdan foydalanishni tanlash uchun uning yordamchi menyusidan foydalanish mumkin. Bunda bir vaqtning o'zida faqat bir ob'yektdan foydalanishingiz mumkin. Agar siz boshqa ob'yektga murojaat etmoqchi bo'lsangiz global o'zgaruvchidan nusxa olishingiz zarur.

Global o'zgaruvchilar ham oldin aytib o'tganimizdek ikki rejimda bo'lishi mumkin: o'qish va yozish. Siz rejimni almashtirishingiz uchun global o'zgaruvchi yordamchi menyusining Access Mode opsiyasidan foydalanishni maslahat beramiz.

Xususiyatlar bog'lami

Xususiyatlar bog'lami yordamida siz dasturingizni yanada kuchli va yoqimli etishga tayyorlashingiz mumkin. Xususiyatlar bog'lami sizga dasturiy jihatdan yuzadagi panel elementlari parametrlarini nazoratlash imkonini beradi, masalan rang, ko'rinishi, pozitsiyasi, sonli formati va hokazo. Kalitli so'z bu yerda *dasturiy jihatdan* – bu shuni bildiradiki, diagramma bloki algoritmiga asosan siz yuzadagi panel ob'yektlari xususiyatlarini o'zgartirishingiz mumkin.



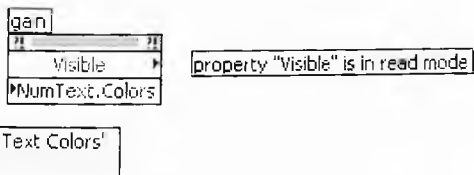
3.3.10-rasm. Yuzadagi panelda xususiyatlar panelidan foydalanish

Masalan siz tsifrblat rangini son qiymat o'sishi bilan ko'k rangdan yashil rangga o'zgartirish yoki foydalanuvchi qanday tugmalarni tanlaganiga qarab unga qaysi elementlarini ko'rsatishni va hattoki biror-bir fizik jarayonni ifodalab, ekranni animatsiyalashingiz mumkin. Xususiyatlar bog'lamini yaratish uchun yuzadagi panel ob'yekti yoki shu ob'yektning diagramma bloki terminali yordamchi menyusining Create >> Property Node ichki menyusidan foydalanishingiz mumkin(3.3.11-rasm). Bunda diagramma blokida shu ob'yekt nomi bilan mos terminal paydo bo'ladi.



3.3.11-rasm. Boshqarish elementi va indicator uchun murojaatli xususiyatlar

Mahalliy va global o'zgaruvchilar singari ob'jekt xususiyatlari o'qish yoki yozish rejimida bo'lishi mumkin (ba'zi xususiyatlar faqat o'qish rejimida bo'ladi). Xususiyatni rejimini o'zgartirish uchun uning yordamchi menyusidan Change to write/read opsiyasini tanlash kerak. Xususiyatlar bog'lamining alohida tomoni shundaki, siz diagramma blokida bir nechta xususiyatlari uchun bir terminalni ishlatishingiz mumkin (lekin bu terminal faqat bir yuzadagi panel elementiga tegishli bo'ladi). Bu terminalga yangi xususiyatlarni qo'shish uchun uning hajmini kattalashtiring(3.3.12-rasm).



3.3.12-rasm. Xususiyatlar bog'lami. Bunda Visible xususiyati o'qish rejimida bo'lsa, NumericText.Colors esa yozish rejimida.

Siz har bir xususiyatni alohida rejimini tanlashingiz mumkin. Bu uchun uning yordamchi menyusidan Change to Write yoki

Change to Read opsiyasini tanlashingiz mumkin. Siz bundan tashqari xususiyatlar bog'lamining barcha komponentlarini birdaniga o'zgartirishingiz ham mumkin (Change All to Write yoki Chage All to Read).

Ko'pincha sizga xususiyatlar bog'lamida bir nechta xususiyatni ishlatish zaruriyati tug'iladi. Har bir xususiyat uchun alohida xususiyatlar bog'lami yaratish o'rniga uning hajmini kattalashtirib foydalanish mumkin.

Metdolar bog'lami

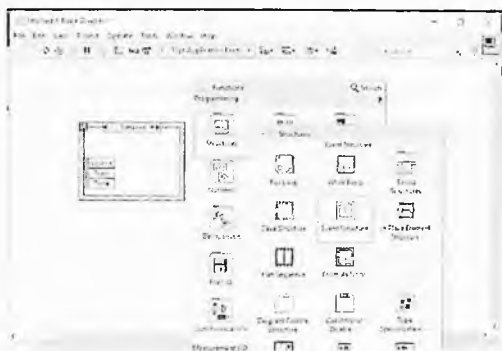
Metodlar bog'lami xususiyatlar bog'lamiga o'xshaydi. Metod bog'lamini yuzadagi panelda ishlatish, ma'no jihatidan VIU yoki biror-bir metod yoki funktsiyani ishlatishga o'xshaydi. Xususiyatlar bog'lami bilan metodlar bog'lami o'rtasidagi farq shundaki: metod bog'lami "biror-bir ish bajaradi", xususiyatlar bog'lami singari biror-bir o'zgaruvchi qiymatini almashtirmaydi. Metodlar bog'lamini yaratish uchun diagramma bloki terminali yoki yuzadagi panel ob'yekti yordamchi menyusidan Create >> Invoke node foydalaning!

Takrorlash va mustaqil ishlash uchun savollar

1. Mahalliy o'zgarivchilar nima?
2. Mahalliy o'zgarivchilardan qaysi maqsadda foydalanish mumkin?
3. Global o'zgaruvchilar nima?
4. Global o'zgaruvchilar mahalliy o'zgaruvchilardan nimasi bilan farqlanadi?
5. Xususiyatlar bog'lami nima?
6. Xususiyatlar bog'lamidan nima maqsadda foydalaniladi?

§3.4. Hodisaga asoslangan dasturlash.

Programming >> Structures ichki palitrasida joylashgan hodisa tuzilmasi juda ham kuchli instrumentlardan biri hisoblanadi. Bu tuzilma yuqori sifatli kodni yaratish imkonini berib, u biror-bir hodisa yuz berishini kutadi. Noeffektiv kod esa davriy ravishda hodisa yuz berishini tekshiradi.



3.4.1-rasm. Diagramma blokida hodisa tuzilmasini yaratish

Hodisa o'zi nima? LabVIEW da ixtiyoriy yuzaga keladigan vaziyatni hodisa deb atash mumkin. Masalan:

- *yuzadagi panelda biror-bir tugmani tanladiz;
- *sonli o'zgaruvchi qiymatini o'zgartirdingiz;
- *VU oynasida sichqoncha kursori joylashgan vaziyat;
- *kalvisha bosildi.

Hodisa tuzilmasisiz siz foydalanuvchi "Stop" tugmasini tanlaganini bilish uchun While sikli ichida doimiy uni so'rab o'tirishiz kerak(oldingi paragraflarda buni biz ko'rib o'tgan edik). Hodisa tuzilmasi yordamida esa siz doimiy so'rab o'tishini hojati yo'q, chunki VU hodisa yuz berganini biladi.

Hodisa tuzilmasi Variant tuzilmasiga juda o'xshab ketadi, chunki variant ramkalariga ega. Har bir hodisa varianti bir yok bir nechta hodisalarni qayta ishlashi uchun qayd etilishi mumkin.

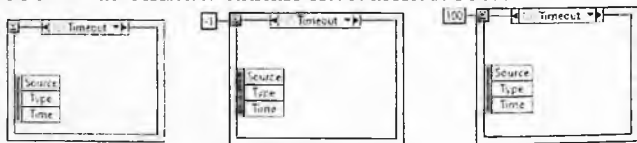
Hodisa tuzilmasi bajarilayotganda u ko'rsatilgan hodisa yuz berganini kutadi va shunga ko'ra bir variantni bajaradi.

Vaqt o'tishi bilan bog'liq hodisa

Diagramma blokida biz hodisa tuzilmasini yaratganda, u standart holat bo'yicha vaqt o'tishiga bog'liq holatda sozlangan bo'ladi.

Vaqt o'tishi bilan bog'liq hodisa – bu maxsus hodisa bo'lib, tuzilmada ko'rsatilgan boshqa hodisalar ko'rsatilgan vaqtda bajarilmagandan so'ng bajariladi.

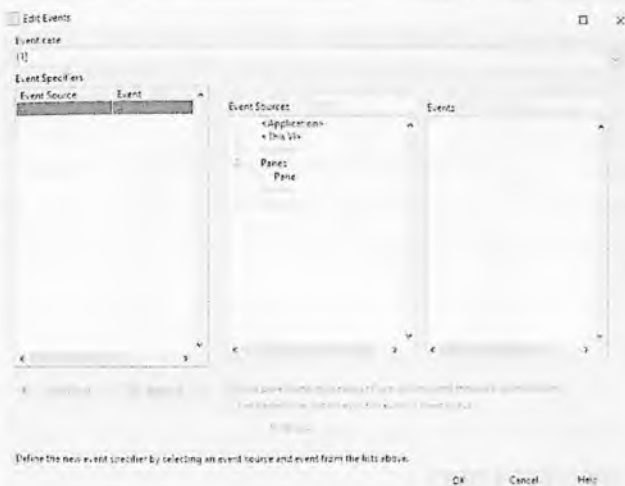
Vaqt kutishini siz tuzilma yuqori chap burchagida ko'rsatishingiz mumkin. Standart holat bo'yicha u "-1" ga teng, ya'ni "hech qachon" yoki "cheksiz kutish degani". 3.4.2-rasmda uch sozlanmada birinchi ikkitasi ekvivalent hisoblanadi.



3.4.2-rasmda 3 xildagi hodisalar konstruksiyasi keltirilgan. Bu rasmlardagi chap tomondan 1 va 2 – hodisalar ekvivalent hisoblanadi.

Hodisa tuzilmasini ishlatish

Yuzadagi panel boshqarish elementlari hodisalarini qayta ishlashni "Hodisa tuzilmasiga" qayd etish uchun hodisa tuzilmasi yordamchi menyusidan Add Event Case... opsiyasidan foydalaning. Sizning oldingizda hodisani sozlash dialog oynasi paydo bo'ladi.

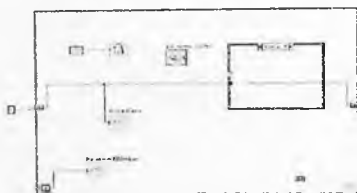


3.4.3-rasm. Hodisa tuzilmasining Add Event Case oynasi.

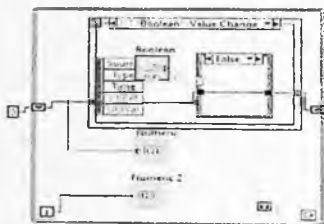
Bu oyna quyidagilardan iborat:

- ❑ Event Handler for Case – ushbu hodisa tuzilmasi barcha variantlari va raqamalarini saqlaydi. Siz ro'yxatdan ularni tanlashingiz mumkin;
- ❑ Event Specifiers – ushbu hodisa tuzilmasi uchun hodisalar to'liq ro'yxatini va ularning manbaalarini(dastur, VU, boshqarish elementi) ko'rsatadi;
- ❑ Add Event – ushbu hodisa tuzilmasi variant qo'llab-quvvatlovchi yangi hodisa yaratadi;
- ❑ Remove – ushbu hodisa tuzilmasi variant qo'llab-quvvatlovchi hodisani o'chirish imkonini beradi;
- ❑ Event Sources – yangi hodisalarni sozlash uchun sinflar bo'yicha saralangan hodisa manbaalarini ko'rsatadi;
- ❑ Events – Events Sources bo'limidan tanlangan manbaa uchun hodisalar ro'yxatini aks etadi;

□ Lock front panel until event case for this event completes—nomidan ma'lumki mos hodisa yuz berganda yuzadagi panelni bloklaydi. Hodisa tuzilmasi yordamchi menyusidan siz Edit Events Handled by this Case opsiyasini tanlab, hodisa variantini o'zgartirishingiz mumkin. Duplicate Event Case opsiyasidan hodisa variantini nusxalash mumkin yoki Delete Event Case opsiyasini tanlab hodisa variantini o'chirishingiz mumkin.



3.4.4-rasm. Hodisa tuzilmasidan foydalanilmagan holatda tugmani True holatga o'tkazandagi hisoblash sxemasi.

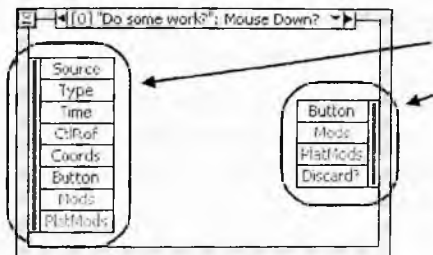


3.4.5-rasm. Hodisa tuzilmasidan foydalanilgan holatda tugmani True holatga o'tkazandagi hisoblash sxemasi.

3.4.5-rasmda hodisa tuzilmasi tugma qiymatini o'zgaryotganini ko'rsatyotgani keltirilgan.

Agar siz e'tibor qaratgan bo'lsangiz, hodisani qayta ishlash tuzilmasi ichida ba'zi terminallar mavjud. Bu terminallar chap va o'ng tomon qismlarida joylashgan. Bu terminallar hodisa ma'lumotlari bog'lami(Event Data Node) va hodisa filtrlari bog'lami(Event Filter Node) deb ataladi.

Bu bog'lamlar hodisalarga tegishli ma'lumotlarga murojaat etish va ularni o'zgartirish uchun ishlatiladi. Hodisa ma'lumotlari bog'lami sizga qo'shimcha ma'lumotlarni beradi: qachon hodisa yuz berdi, hodisa tipi, kursor kordinatalari va hokazo.



3.4.6-rasm. Hodisa ma'lumotlari va filtrlari mavjud hodisa tuzilmasi.

Hodisa filtrlari bog'lami qachon paydo bo'lishini tushunish uchun ikki tipdagi hodisalar mavjudligini bilish kerak: xabar-hodisalar va filtr-hodisalar. Faqat filtr hodisalar hodisa filtr bog'lamiga ega. Biz xabar-hodisani filtr-hodisadan faqlashni hodisalarni sozlash oynasida hodisadan oldin yo'naltirgich(strelka) rangidan farqlash mumkin.

Xabar-hodisalar oldida yashil rangdagi yo'naltirgich bo'lsa, filtr-hodisalar oldida qizil yo'naltirgichlar bo'ladi.

Takrorlash va mustaqil ishlash uchun savollar:

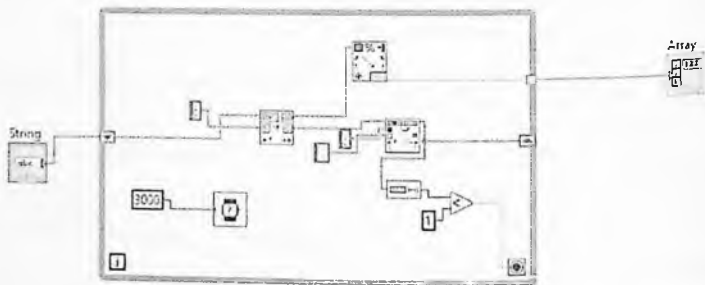
- 1.Hodisa tuzilmasi nima?
- 2.Hodisalar bilan bog'liq texnik ilovalar yaratish nima uchun kerak?
- 3.Vaqtga bog'liq hodisalar qanday amalaga oshiriladi?
- 4.Qanday hodisa holatlari mavjud?
- 5.Yangi hodisalarni qo'shish haqida nima bilasiz?
- 6.Hodisa ma'lumotlarini filtrlash nima uchun kerak?

§3.5. Diagramma-bloklari fragmentlaridan virtual ichki uskunalarni avtomatik yaratish. Kodlarni olib tashlash tuzilmalari.

Diagramma bloki fragmentidan virtual ichki uskunalarni avtomatik yaratish

Ilovalarni kam sonli xatoliklar bilan to'g'ri yozish uchun modullilik (modularity) tamoyildan foydalanish maqsadga muvofiq. *LabVIEW* da modullar sifatida virtual ichki uskunalar (VIU) qo'llaniladi.

Katta ilovalar yaratmoqchi bo'lsak, ularni kichik masalalarga bolib, alohida VIU - ga ajratish kerak. Edit-Create Sub VI menyu opsiyasi diagramma BLOKI fragmentini VIU ga oson o'zgartiradi. Faraz qilaylik, biz biror ilova bilan ishlayotgan bo'lsak va uning aniq biror qismini VIUga aylantirishiz kerak bo'lsin. Buning uchun VU ning diagramma bloki fragmentini ajratib Edit-Create Sub VI opsiyasini chaqirish kerak (3.5.1-rasm).



3.5.1-rasm. VIU ni yaratish uchun kod uchastkasini tanlash

LabVIEW ushbu fragmentni VIU ga o'zgartiradi va unga avtomatik tarzda boshqarish elementlarini va indikatorlarni yaratadi. Yangi VIU VU ning ajratilgan diagramma bloki qismini almashtiradi va *LabVIEW* avtomatik tarzda VIU ni mavjud o'tkazgichlarga ulaydi (3.5.2-rasm). Tavsiflangan opsiyadan

zaruriyat bo'lganda diagramma bloki fragmentlarini boshqa virtual uskunada takrorlash uchun ham foydalanish mumkin.



3.5.2-rasm. Edit >> Create SubVI opsiyasini tanlagandan so'ng, kod bloki o'rnida hosil bo'lgan VIU.

Create SubVI opsiyasini qo'llash qulay bo'lishi bilan birga quyidagi ayrim cheklanishlar ham mavjuddir:

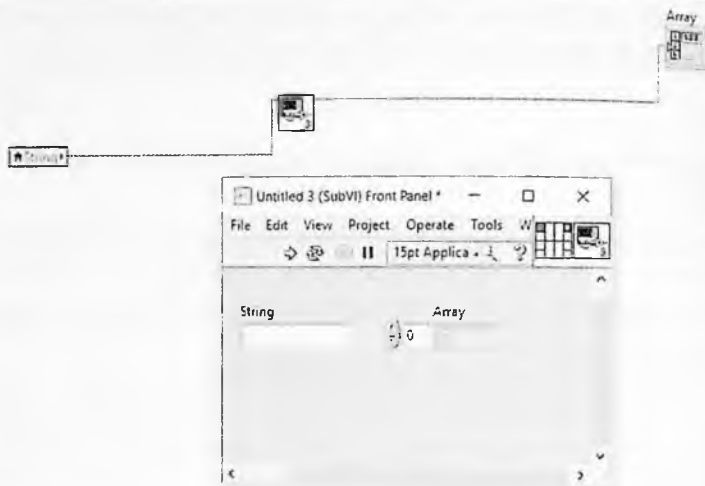
□ 28-tadan ko'p kirish/chiqishga ega bo'gan VIU yaratadigan diagramma bloki hududlarini o'zgartirish mumkin emas. Chunki, bu maksimal ruxsat etilgan ulanish panelidagi kirish va chiqishlar sonidir;

□ tuzilma ichida va uning tashqarisida tanlangan, lekin tuzilmaning o'zi tanlanmaganda diagramma bloki fragmentini o'zgartirish mumkin emas;

□ xususiyatlarning bog'langan bog'lamlari (panelning yuza tomonida boshqarish elementlari bilan bog'liq va murojaat uchun kiritish\chiqarishga ega bo'lmagan) bog'lanmaganlari bilan almashtiriladi. Element va uning tuguni orasidagi aloqalarni qo'llab-quvvatlash uchun ushbu elementga VU serveri murojaatnomasi yaratiladi va u yangi VIU ga uzatiladi. Ungacha va undan keyingi holat 3.5.3 va 3.5.4-rasmlarda keltirilgan.



3.5.3-rasm. Xususiyatlar bog'lamini VIU ga tanlash.



3.5.4-rasm. VIU dan keyingi holat.

Agar belgilangan fragment mahalliy o'zgaruvchilarga ega bo'lsa va tegishli boshqarish elementiga ega bo'lmasa, unda biror mahalliy o'zgaruvchi boshqarish elementiga chaqiriladigan VU ga qoladi hamda ma'lumotlarni VIU ga yoki VIU dan uzatadi.

VIU da diagramma bloki fragmentini o'zgartirishda u ilovani yaxlit o'zgartirishi bois, bunday hollarda qaror qabul qilish imkoniyati bo'lmaydi.

VIU ichida boshqarish elementi uchun birinchi mahalliy o'zgaruvchi o'qish va yozish instrumentiga aylanadi, keyingi mahalliy o'zgaruvchilar esa VIU dagi boshqarish elementiga murojaat qiladi. Yana bir bor ta'kidlab o'tish joizki, diagramma bloki elementidan VIU ni yaratishda agar u mahalliy o'zgaruvchiga ega bo'lsa, unda ehtiyot bo'lib ishlash talab etiladi.

Agar tanlangan diagramma bloki fragmenti mahalliy o'zgaruvchiga yoki tsikl tanasidagi yuza paneli terminallariga ega bo'lsa, unda ular tomonidan o'lchanadigan kattalik diagramma blokining boshqa joyida o'zgartirilishi mumkin. Shunday qilib,

VIU da tsikl o'zgartirilganda belgilangan fragmentning funktsionalligini o'zgartirish imkoniyatlari mavjud. Agar biz barcha mahalliy o'zgaruvchilarni tanlamasdan, balki ularning ayrimlarini yoki yuza paneli terminalini tanlasak, unda LabVIEW amalini bajarish yoki to'xtatish xususida ogohlantiruvchi xabar beradi.

Create SubVI opsiyasi kichik loyiha bilan ishlashda qulaydir. Lekin ish davomida loyiha murakkablashib boradi. Bunday holda diagramma bloklarning katta fragmentlarini VIU larga joylashtirishga harakat qilmaslik kerak. VIU ning bo'sh ishchi maydonini oshirish doim aniq belgilangan masalalarning maqsadlaridan kelib chiqqan holda bajariladi.

Kodni olib tashlash tuzilmasi nima?

Agar siz boshqa an'anaviy dasturlash tillarida dastur tuzgan bo'lsangiz, unda bir necha marotaba kodlariz qismini izohlashga duch kelgansiz. Masalan, C yoki Java dasturlash tillarida (*) orqali izoh qismini ochish va (/) orqali esa o'sha qismni yopishingiz mumkin. Bu qismlar dasturda bajarilmaydi. Yoki (//) belgisi yordamida bir qatorli izohlarni yaratishingiz mumkin.

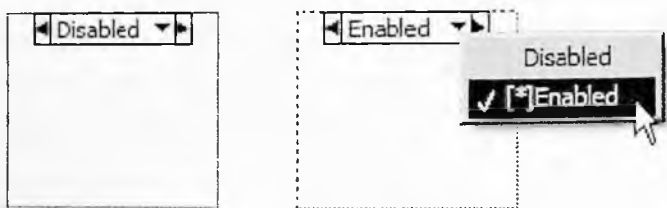
LabVIEW da grafik kodlar bo'lagini bajartirmoqchi bo'lmasangiz, unda Diagram Disable va Conditional Disable tuzilmalaridan foydalanishingiz mumkin. Bu tuzilmalarning ikkalasi ham Programming >> Structures ichki palitrasida joylashgan.

Variant tuzilmasi singari *Diagram Disable* va *Conditional Disable* tuzilmasi sizga bir necha diagramma bloki kodini yaratish imkonini beradi. Lekin siz e'tibor qaratishiz mumkinki, *Diagram Disable* va *Conditional Disable* tuzilmalari bajarish uchun ichki diagramma bloki kodini tanlash terminallariga ega emas. Chunki LabVIEW da tahrirlash jarayonida faqat bir freym ishlatish uchun aniqlanadi.

Diagram Disable tuzilmasi

Bu tuzilmani ishlatib, biz ko'pgina freymnlarni yaratishimiz mumkin va ularni istalgancha nomlashimiz mumkin. Biz birdan ko'p bo'lmagan freymni ishlatish uchun tanlashimiz mumkin, qolganlari esa izoh sifatida ishlatiladi. Diagram Disable tuzilmasiga sichqonchani o'ng tugmasini tanlab, Enable This Subdiagram yoki Disable This Subdiagram opsiyasini tanlab, u yoki bu freymni faollashtirish yoki izohlash mumkin.

Bir vaqtda faqat bir ichki diagramma faol bo'lishi mumkin, lekin albatta biror- bir freym faol bo'lishi shart degani emas. Siz freym faol yoki faol emasligini [*] belgisi orqali darhol bilib olishiz mumkin(3.5.5-rasm).

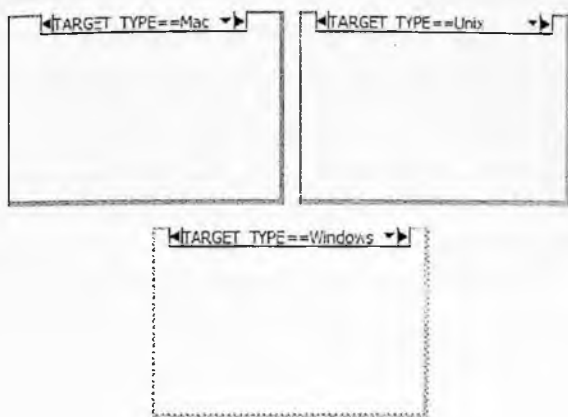


3.5.5-rasm. Diagram Disable tuzilmasi

Standart holat bo'yicha *LabVIEW* ichki diagrammalarni "Disabled" va "Enabled" deb nomlaydi. Sizda "Disabled" ismli bir necha ichki kod bo'lishi mumkin. Kodni o'chirib qo'yish nafaqat *LabVIEW* tomonidan uni bajarilmasligiga olib keladi, balki uni hatto kompilyatsiyadan ham o'tkazmaydi.

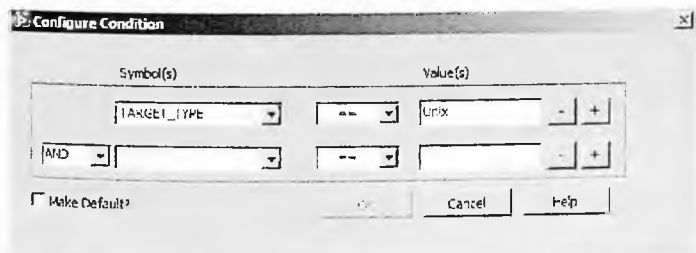
Conditional Disable tuzilmasi

Conditional Disable tuzilmasi oddiy *Diagramm Disable* tuzilmasidan farqladi. Birinchi farqi – *LabVIEW* bu tuzilmaning faqat bir freymni bajaradi. Ikkinchi farqi esa – *LabVIEW* qaysi freymni ishlatishni bilish uchun shartni tekshirib ko'radi. Freymni bajarish sharti uning nomida ko'rsatilgan(3.5.6-rasm).



3.5.6-rasm. Conditional Disable tuzilmasi.

Har bir freymning shartini freymga sichqonchani o'ng tugmasini tanlab, paydo bo'lgan menyudan "Edit Condition for This Subdiagram" optisyasini tanlash orqali tahrirlash mumkin. Bu sozlash dialog oynasini chaqiradi(3.5.7-rasm).



3.5.7-rasm. Conditional Disable tuzilmasi - shartni sozlash oynasi.

VU ni to'xtatish va dasturni bajartirish

LabVIEW bizga VU ni darhol to'xtatishning ikki funksiyasini taqdim etadi: Stop va LabVIEW dan chiqish (Quit LabVIEW). Bu funksiyalar Application Control ichki palitrasida joylashgan.

- *Stop* tugmasi mantiqiy kirish qiymatiga ega. True (ROST standart holat bo'yicha, agar kirish qismiga hech qanday ma'lumot ulanmagan bo'lsa) holatida VU va VIU lar barchasi to'xtaydi;

- *Quit LabVIEW* funktsiyasi nafaqat VU ishini to'xtatadi, balkim *LabVIEW* muhiti bilan ishni yakunlaydi, agarda uning kirish qismiga True(ROST) qiymati berilgan bo'lsa. Bu funktsiyani ishlatishga ehtiyot bo'ling!

Foydalanuvchi grafik interfeysi opsiyalari

LabVIEW da siz foydalanuvchilarning grafik interfeys komponentlari yordamida dasturingizga kuchli grafik imkoniyatlarni qo'shishingiz mumkin.



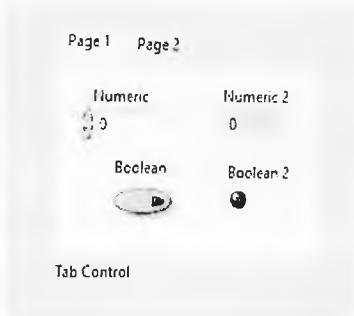
3.5.8-rasm. Yuzadagi panelda foydalanuvchi grafik interfeys elementlari paneli.



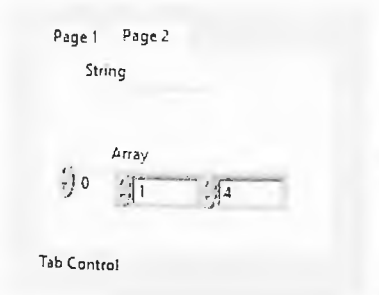
3.5.9-rasm. Yuzadagi panelda List&Table grafik interfeys elementlari paneli.



**3.5.9-rasm. Yuzadagi panelda
Container grafik
interfeys elementlari paneli**



**3.5.10-rasm. Yuzadagi panelda
Container grafik
interfeys elementlaridan
foydalanish**



**3.5.11-rasm. Yuzadagi panelda
Container grafik
interfeys elementlaridan
foydalanish.**



**3.5.12-rasm. Yuzadagi panelda
Container grafik interfeysi Splitter
elementidan foydalanish.**



3.5.13-rasm. Yuzadagi panelda grafik interfeysi Picture elementidan foydalanish



3.5.14-rasm. Yuzadagi panelda grafik interfeysi Graphics&Sound paneli

Takrorlash va mustaqil ishlash uchun savollar:

- 1.Virtual ichki uskuna nima?
- 2.Virtual ichki uskuna nima uchun kerak?
- 3.Diagramma bloki fragmentidan virtual ichki uskunalarni avtomatik yaratish qanday amalga oshiriladi?
- 4.Diagramma bloki fragmentidan virtual ichki uskunalarni avtomatik yaratishda qanday shartlar mavjud?
- 5.Kodni olib tashlash tuzilmasi nima?
- 6.Diagram Disable tuzilmasi nima maqsadda ishlatiladi?
- 7.Conditional Disable tuzilmasi nima maqsadda ishlatiladi?
- 8.VU ni to'xtatish va dasturni bajartirish nima maqsadda ishlatiladi?
- 9.Foydalanuvchi grafik interfeysi elementlari nima maqsadda ishlatiladi?

1. *Open/Create/Replace File (Programming>>File I/O)* faylni ochadi va unga *refnum* ni qaytaradi;

2. *Get File size (Programming>>File I/O>>Advanced File Functions)* fayl o'lchamini qaytaradi, bu o'lcham, sanalishi zarur bo'lgan, simvol (belgi)lar sonini aniqlashda foydalaniladi;

3. *Read from Text File (Programming>>File I/O)* – count kirishida ko'rsatilgan ma'lumotlarni fayldan, metkaning fayldagi joriy holatidan boshlab, son qilib qaytaradi (bizning holda 0). Bu funktsiya polimorf hisoblanadi. *Refnum* sifatida faylga berish va faylga bo'lgan yo'lni ko'rsatish mumkin. Agar yo'l ko'rsatilib, *refnum* chiqishda ulanmasa, unda faylga murojaat avtomatik tarzda bekiladi;

4. *Close File (Programming>>File I/O)* faylga *Open/Create/Replace File*

funktsiya bilan yaratilgan *refnum* ni bekitadi;

5. *Simple Error Handler.vi (Programming>>Dialog&User Interface)* – agar kiritish/chiqarish funktsiyalaridan birida xato bo'lsa, muloqot oynani ko'rsatadi;

6. Yuza panelga qayting va boshqarishning tegishli mos elementi orqali matnli faylga yo'lni ko'rsating. Katta (*LabVIEW* ga juda ko'p xotira sig'imi kerak bo'ladi) yoki binar (bu holda siz faqat tushunarsiz belgilarni ko'rasiz) faylni tanlamaganingizga ishonch hosil qiling;

7. VU ni ishga tushiring va avval ko'rsatilgan faylning tarkibini, hamda uning o'lchamini ko'ring;

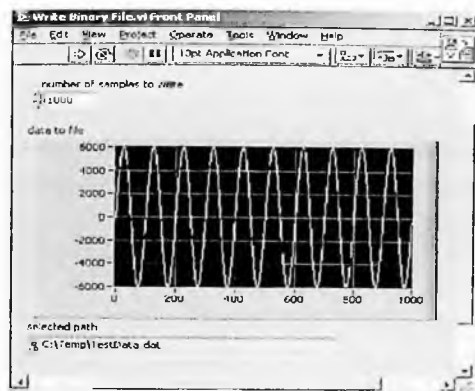
8. VU ni xotiraga yozing va yoping. Uni *Read Text File Advanced.vi* deb atang va MyWork papkada saqlang.

Siz matnli fayllarni ko'rib chiquvchini ilova yaratdingiz!

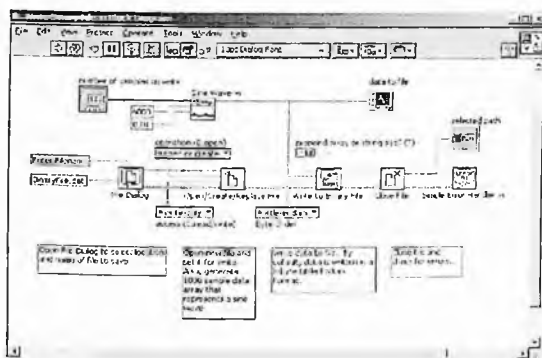
Binar fayllarni o'qish va yozish

Bu mashqda kirish/chiqish funktsiyalaridan foydalanib, binar fayl yoziladi va u tasodifiy nuqtalarda qaytadan o'qiladi.

1. Avval, 3.6.3-rasmda ko'rsatilganidek *Write Binary File Advanced.vi* nom bilan yuzadagi panelli VU yaratish:



3.6.3-rasm. Write Binary File Advanced.vi VU si yuzadagi paneli.



3.6.4.-rasm. Write Binary File Advanced.vi blok-diagramma

3.6.4.-rasmda ko'rsatilganidek blok-diagramma yaratish.

Sinusni ifodalovchi sonlar massivini yaratish uchun *Sine Wave.vi* (*Signal Processing*>>*Signal Generation*) funktsiyadan foydalaning. Faylni yaratish, ochish va yopish uchun *Advanced* palitradan fayllarni kiritish/chiqarish funktsiyalaridan

foydalaning. DBL massiv (ikkilamchi aniqlikdagi suzuvchi nuqtali son) binar faylga yozilayotganligiga e'tibor bering. Har bir DBL nuqta diskda 8 baytni egallaydi, bu axborot bizga faylni o'qish vaqtida kerak bo'ladi.

(Ma'lumotlarning har bir turi LabVIEW da qancha joy egallashini *Fundamentals>>HowLabVIEW Stores Data in Memory/* - ma'lumotlar olish seksiyasi orqali bilish mumkin).

Write Binary File Advanced.vi ni ishga tushiring va uning to'g'ri ishlayotganiga ishonch hosil qiling. Binar faylni oson topiladigan biror joyda saqlang. Agar Write to binary file funksiyasi fayllarni ochish va yopishni avtomatik tarzda bajarsa, unda faylni ochuvchi va yopuvchi alohida fayllar nima uchun kerak? Bunday yondoshuvdan quyidagi 2 holda foydalaniladi:

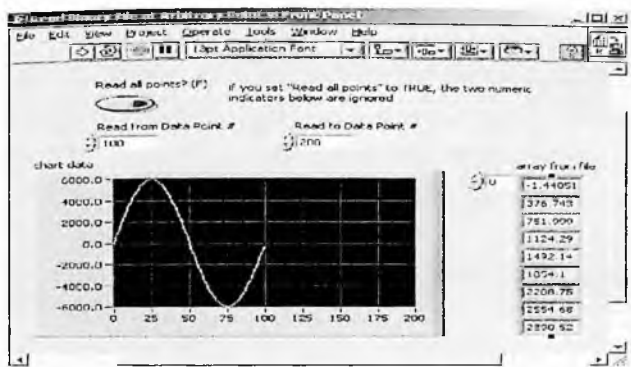
- Fayl oxiriga qo'shimcha biror ma'lumot yozish kerak; yozuvning alohida funksiyasi mavjud: fayl mazmunini almashtiradi;

- Yozuv markeri holatini boshqarish: ixtiyoriy joydan yozish;

- Tez oqimli yozish: fayl bir marta ochildi va ma'lumotlar, tezkor xotirada to'planmasdan, to'g'ridan-to'g'ri tez-tez diskka yoziladi. Keyin bir marta fayl yopildi.

Mustaqil topshiriq: *VU atrofida generatsiya va ma'lumotlarni yozish sharti bilan tsikl yarating. Fayl juda katta bo'lishi mumkin.*

3.Endi o'qish uchun VU yaratamiz. Faraz qilamiz, siz butun faylni emas, balki uning bir nuqtasidan boshqasigacha bo'lgan bir qismini o'quvchi VU yaratmoqchisiz. Masalan, sizda 1000 nuqtali (baytli emas) fayl bor va siz 100- 200- gacha bo'lgan nuqtalarni o'qimoqchisiz. 3.6.5.rasmda ko'rsatilganidek yuza panel' yarating va VU ni *Read Binary File Arbitrary Point.vi* deb saqlang.



3.6.5-rasm. Read Binary File Arbitrary Point.vi – yuza panel.

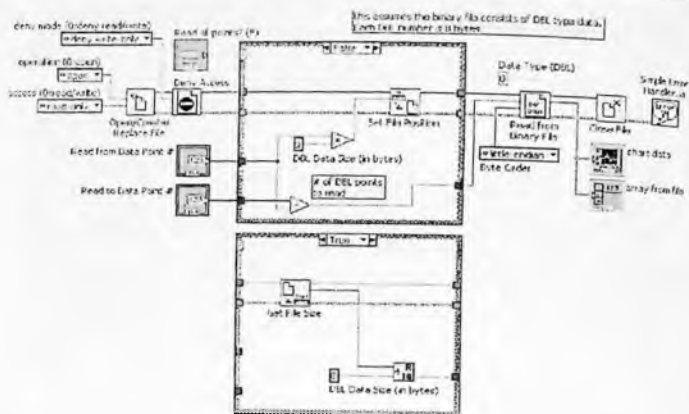
4. Binar faylning ixtiyoriy joyidan o'qish kaliti - *Set File Position* funktsiyasidir. Bu funktsiyadan metkani kerakli holatda o'rnatish uchun foydalanish mumkin. Metka holati nuqtalar bilan emas, baytlar bilan berilishini esda tuting. Bizning holda, binar faylda DBL-sonlar massivi bo'lgani uchun, har bir nuqtaga 8 bayt to'g'ri keladi. Agar metka 100-nuqtaga o'rnatilsa, uni 800-baytga o'rnatish kerak.

5. Qolgan nuqtalar sonini hisoblash uchun, *Read From File* funktsiyasida qancha nuqtani hisoblash kerakligini ko'rsatish kerak. Bu funktsiyada, o'zini har xil tutuvchi *count* kirish bor:

➤ Agar siz "*Data type*" kirishni ulasangiz (bu hol tavsiya etiladi), unda *count* kirishda hisoblashni baytlarda olib borish kerak, deb tushunadi. Bu holda avtomatik tarzda hisoblashlar baytlarda olib boriladi, chunki ma'lumotlar turi noma'lum;

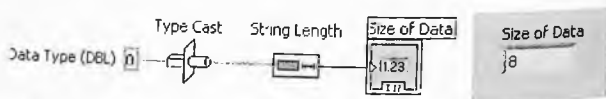
➤ Agar siz "*Data type*" kirishni ulamasangiz, unda *count* kirish siz unga baytni emas, nuqtalarni hisoblashni buyurgansiz, deb tushunadi, chunki unga ma'lumotlar turi baribir noma'lum.

6. Endi bu mashqni 3.6.6-rasmga qaramasdan bajarishga harakat qiling, e'tibor bering, Variantlar tasnifining TRUE varianti namunaviy ko'rgazma maqsadida keltirilgan.



3.6.6-rasm. Read Binary File Arbitrary Point.vi ning blok-diagrammasi

Keyinchalik *Tupe Cast* funksiyasi to'g'risida ma'lumot olganingizdan keyin, konstanta turini ko'p joyda ishlatish o'rniga ma'lumot elementlarining o'lchami (bizning holda DBL) dasturiy hisoblanishi uchun bu mashq blok-sxemasini o'zgartirishga to'g'ri keladi. Bu holda, baytlar sonini aniqlash uchun (*Tupe Cast* funksiyasining chiqishi) 3.6.7-rasmda ko'rsatilganidek, *String Length* funksiyasidan foydalanish mumkin:



3.6.7-rasm. *Tupe Cast* funksiyasidan ma'lumot elementlari o'lchamini tekshirish uchun foydalanish

AMALIY MASHG'ULOTNI BAJARISH TARTIBI

1. Nazariy ma'lumotlar bilan tanishib chiqing. Belgilangan vazifalarni ketma-ket bajaring va natijalarni tahlil qiladi.
2. Faylni yozish va uni tasodifiy nuqtalarda o'qish texnologiyasini tushuntiradi.

AMALIY MASHG'ULOT №8

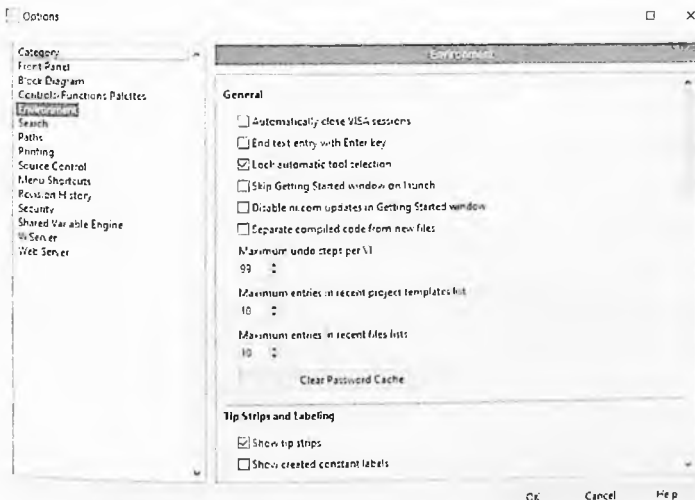
ICHKI VIRTUAL USKUNALARNI ISHLATISH

Ishdan maqsad: *LabVIEW* da virtual ichki uskunani o'rnatish va undan ilovalarda foydalanish texnologiyasini o'rganish.

Qisqa nazariy ma'lumotlar

Virtual ichki uskunalardan foydalanishdan oldin *LabVIEW*ni sozlash texnologiyasini tahlil etishimiz zarur. Sozlash deganda qidiruv marshrutini tanlash, ranglarni va shriftlarni o'rnatish, vaqt va sanani taqdim etish protseduralari tushuniladi.

3.7.1-rasmdan ko'rinib turibdiki sozlash kategoriyalarga bo'lingan.



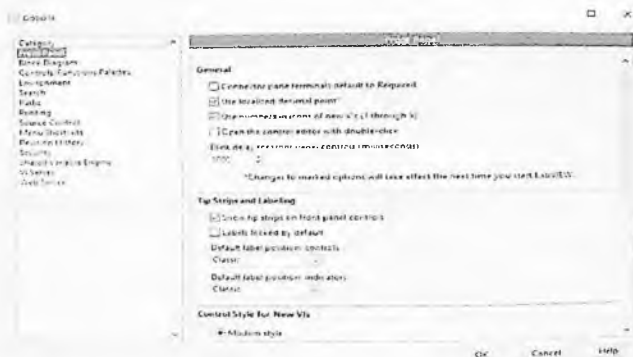
3.7.1-rasm. LABVIEW ni sozlash oynasi.

Sozlashning ko'p qo'llanadigan opsiyalariga quyidagilar kiradi:

-Yangi va o'zgartirilgan kategoriyalarda LabVIEWning oxirgi versiyalarida yangi yoki o'zgargan opsiyalari tarkibi keltirilgan.

-Yo'llar (Paths) opsiyalarida LabVIEW muhitini joylashtirishni so'ramasdan o'rnatish, kutubxonalarni qidirish yo'llari, foydalanuvchi VU lari va vaqtinchalik fayllarni saqlaydigan katalogni o'rnatish mumkin.

-Yuzadagi paneli kategoriyasi (Front Panel) bir qator muhim sozlashlarni o'z ichiga oladi (3.7.2-rasm).



3.7.2-rasm. Yuzadagi panelini sozlash bo'yicha LabVIEW opsiyalari.

-To'r bo'yicha tekislash (Alignment Grid) opsiyasida biz yuzadagi panelida va blok-diagrammada to'r chizig'ini sozlashimiz mumkin.

-Ranglar (Colors) opsiyasi yuzadagi paneli uchun foydalanuvchi aniq ko'rsatmagan taqdirda ranglarni o'rnatish, shuningdek diagramma bloklarini o'rnatish imkonini beradi. Agar LabVIEW yuzadagi panelini biror rangda bo'yamoqchi bo'lsangiz, aynan shu opsiya ishlatiladi.

-Bosmaga chiqarish (Printing) opsiyasi standart formatda bosmaga chiqarishni yoki PostScript formatda ma'lumotlarni chiqarish uchun qo'llaniladi.

-Окружение (Environment) opsiyasi muhit sozlash (just in time advice) imkonini beradi.

Oynaning tashqi ko'rinishini boshqarish va VU ni boshqarish ikkita bo'limdan amalga oshirilishi mumkin:

1)Virtual uskuna xususiyatlari (VI Properties) –panelning yuqori o'ng oynasidaga VU ning diagramma bloki kontekst menyusi ikonkasidan.

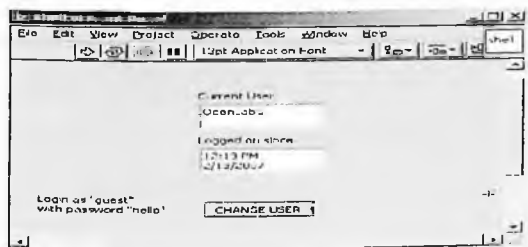
2)VIU ning tugunini o'rnatish (SubVI Node Setup) –VIU ning diagramma bloki kontekst menyusi ikonasidan.

VIU tugunini sozlash oynasi. VIU ning kontekst menyusidan sozlash opsiyasini tanlash asosida amalga oshiriladi.

AMALIY MASHG'ULOTINI BAJARISH TARTIBI

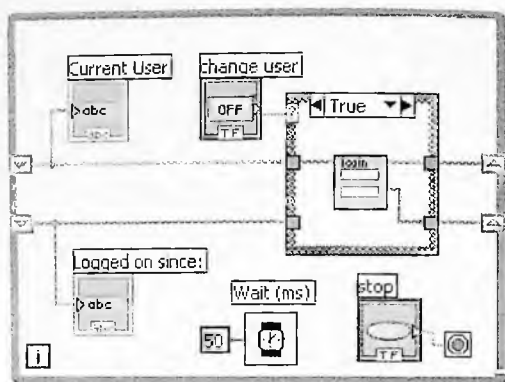
Amaliy mashg'ulotida algoritmlash protsedurasi uchun dastur qobig'ini yaratish bo'yicha VIU ning tugunini sozlash protsedurasidan foydalanish kerak. U keyingi boshqa protseduralarda ishlatilishi mumkin.

1.Ishni boshlashni 3.7.3-rasmda ko'rsatilgan yuzadagi panelni yaratishdan boshlang. U foydalanuvchini almashtirish tugmasini bosishdan keyin VIU ning oynasini chaqiradi. Ushbu yuqori darajadagi virtual uskunani Shell.vi deb nomlang.



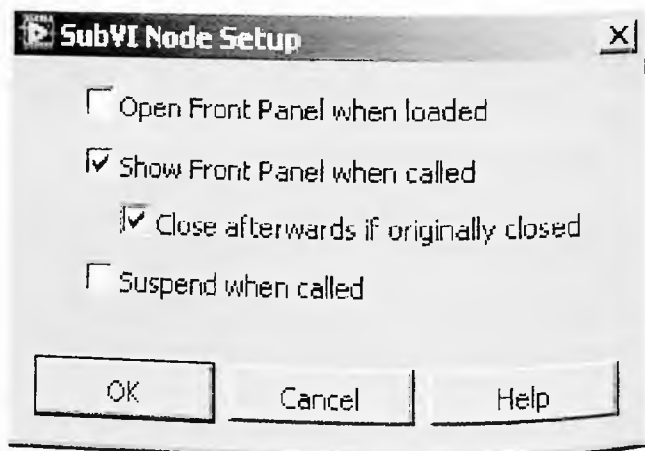
3.7.3-rasm. Mashg'ulotning VU yuzadagi paneli

2.3.7.4-rasmda keltirilgan diagramma blokini qurish uchun Login.vi ichki virtual uskunasiidan foydalaning (uni amaliyot o'qituvchisidan olasiz).



3.7.4-rasm. Mashg'ulotning VP diagramma bloki.

3.Login ichki virtual uskunaga sichqonning o'ng tugmasini bosing va 3.7.5- rasmda ko'rsatilganidek, ichki virtual uskunaning tugunini o'rnatng.



3.7.5-rasm. Login.VU uchun tugunni sozlash oynasi

AMALIY MASHG'ULOTINI TOPSHIRISH TARTIBI

Tjriba mashg'uloti №6

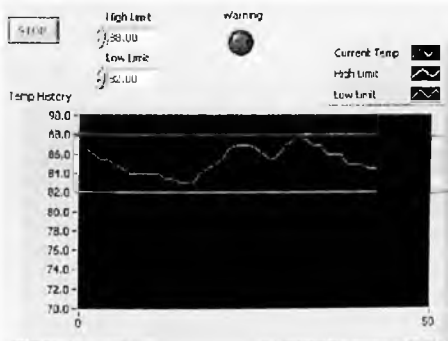
Yaratilgan virtual uskunalarining oyna ko'rinishini namoyish eting va uning imkoniyatlarini tushuntirib bering.

Datchiklardan ma'lumotlarni kompyuterga yig'ish va signallarni chiqarish

Maqsad: Haroratni o'lchaydigan va harorat minimal va maksimal qiymat chegarasidan chiqanda signal beruvchi virtual uskuna yarating.

Tajriba mashg'ulotini bajarish ketma-ketligi:

1. Yangi virtual uskuna yarating.
2. Yuzadagi panelda quyidagi rasmga o'xshash shaklda elementlarni joylashtiring.



3.8.1-rasm. Tajriba mashg'uloti virtual uskunaso yuzadagi paneli ko'rinishi.

3. Temperaturani sekundiga bir marta o'lchaydigan va sahna rejimida to'lqin shaklini tozalash natijalarini namoyish etadigan VU ni yarating. Agar Temperatura belgilangan chegaradan yuqori yoki pastroq bo'lsa, VU yuzadagi paneldagi LED ni yoqadi. Tozalash temperatura va uning yuqori va pastki chegaralarini

chizish kerak. Cheklovlar yuzadagi paneldan o'rnatilishi mumkinligiga ishonch hosil qiling.

4. Virtual qurilmaga Temperature Limit.vi. nom berib, kompyuter xotirasiga saqlang

5. Yaratilgan virtual uskunani sinab ko'ring.

Tjriba mashg'uloti №7

Masalalarni yechish uchun mahalliy o'zgaruvchilardan foydalanish

Ishdan maqsad: Bu darsda biz LabVIEW da masalalarni yechishda mahalliy o'zgaruvchilardan foydalanamiz.

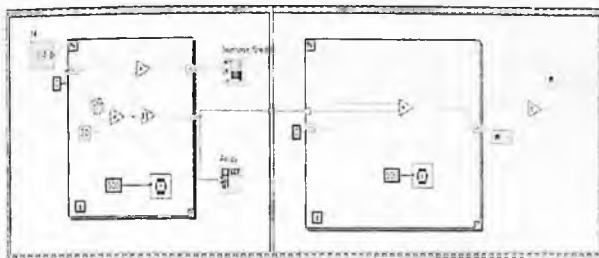
Vazifa: 0 dan 20 gacha bo'lgan oraliqda N (raqamli boshqaruv elementi) tasodifiy sonlarni yaratadigan va ularni massivga joylashtiradigan, shuningdek tasodifiy tanlangan sonlarni yig'ib, ularni ekranda Summa indikatorida ko'rsatadigan virtual uskuna yaratish. Xuddi shu sonlarning o'rtacha qiymatini topib, raqamli ko'rsatkichni ko'rsating.

Tajriba mashg'ulotini bajarish ketma-ketligi:

1. Yangi VU yarating.
2. Yuzadagi paneldagi elementlarni quyidagi rasmga o'xshash tarzda joylashtiring.
3. diagramma blokiga o'ting va quyidagi sxemani yarating



3.9.1-rasm. Tajriba mashg'uloti yuzadagi panel ko'rinishi.



3.9.2-rasm. Tajriba mashg'uloti diagramma bloki sxemasi.

Flat Sequence

N boshqarish elementining mahalliy o'zgaruvchisi

Summa/Sred indikatorining mahalliy o'zgaruvchisi

4. Ushbu diagramma blokida Flat Sequence tuzilishi qo'llaniladi, bu freymalar kabi tuzilmalar yoki operatsiyalarni o'z ichida ketma-ket bajarishga yordam beradigan tuzilma bo'lib, ya'ni birinchi navbatda birinchi kadrni, keyin ikkinchisi va hokazo bajaradi.

5. Dasturni ishga tushiring va VU ning ishlashini tekshiring.

Talabalar uchun topshiriqlar:

Vazifa: Virtual uskuna mavjud. Agar ushbu virtual uskunada siz "Voltage" sonli boshqarish elementi qiymatini 1 dan 30 ga o'zgartirsangiz, "Signal 1" indikator yonsin, agar siz "Voltage" sonli boshqarish elementi qiymatini 31 dan 60 ga o'zgartirsangiz, u holda "Signal 2" indikator yoqilsin va "Voltage" boshqarish sonli elementining qiymatini 61 dan 100 gacha o'zgartirganda "Signal 3" indikator yoqilsin. Ammo ushbu virtual uskunaning kamchiliklari shundaki, agar biz "Voltage" sonli boshqarish elementining qiymatini kamaytirsak, u holda yoqilgan indikator o'chmaydi. Sizning vazifangiz mahalliy o'zgaruvchilar yordamida ushbu kamchiliklarni tuzatishdir. Tayyor virtual uskunani fan bo'yicha telegram kanalidan yuklab oling yoki USB flesh-disk orqali fan o'qituvchidan olishingiz mumkin.

TAJRIBA MASHG'ULOTI №8

Hodisaga asoslangan virtual uskunalarni yaratish

Ishdan maqsad: VU ishini moslashuvchan boshqarish uchun hodisa tuzilmalaridan foydalanadigan ilovalarni yaratish tajribasiga ega bo'lish.

Nazariy ma'lumotlar

Hodisa o'zi nima? *LabVIEW* da ixtiyoriy yuzaga keladigan vaziyatni hodisa deb atash mumkin. Masalan:

- *yuzadagi panelda biror-bir tugmani tanladiz;
- *sonli o'zgaruvchi qiymatini o'zgartirdingiz;
- *VU oynasida sichqoncha kursori joylashgan vaziyat;
- *klavisha bosildi.

Hodisa tuzilmasisiz siz foydalanuvchi "Stop" tugmasini tanlaganini bilish uchun While sikli ichida doimiy uni so'rab o'tirishiz kerak(oldingi paragraflarda buni biz ko'rib o'tgan edik). Hodisa tuzilmasi yordamida esa siz doimiy so'rab o'tishini hojati yo'q, chunki VU hodisa yuz berganini biladi.

Hodisa tuzilmasi Variant tuzilmasiga juda o'xshab ketadi, chunki variant ramkalariga ega. Har bir hodisa varianti bir yok bir nechta hodisalarni qayta ishlashi uchun qayd etilishi mumkin. Hodisa tuzilmasi bajarilayotganda u ko'rsatilgan hodisa yuz berganini kutadi va shunga ko'ra bir variantni bajaradi.

Vaqt o'tishi bilan bog'liq hodisa

Diagramma blokida biz hodisa tuzilmasini yaratganda, u standart holat bo'yicha vaqt o'tishiga bog'liq holatda sozlangan bo'ladi.

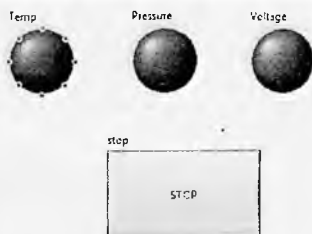
Vaqt o'tishi bilan bog'liq hodisa – bu maxsus hodisa bo'lib, tuzilmada ko'rsatilgan boshqa hodisalar ko'rsatilgan vaqtda bajarilmagandan so'ng bajariladi.

Vaqt kutishini siz tuzilma yuqori chap burchagida ko'rsatishingiz mumkin. Standart holat bo'yicha u "-1" ga teng, ya'ni "hech qachon" yoki "cheksiz kutish degani".

Vazifa: Dastur ilovasini har doim ishga tushirilganda 2 soniya kutadigan virtual uskuna yarating va keyin "Temp" indikatorini, keyin "Bosim" va keyin "Kuchlanish" ni yoqadigan va oxirida "STOP" tugmasini bosgandan so'ng, dastur barcha ko'rsatkichlarini yoqish virtual uskunasi yaratish.

Tajriba mashg'ulotini bajarish ketma-ketligi: Yuzadagi panel

1. Yangi virtual uskuna yarating.
2. Yuzadagi panel elementlarini yuzadagi panelingizda quyidagi rasmga o'xshash tarzda joylashtiring:



3.10.1-rasm. Tajriba mashg'uloti yuzadagi panel ko'rinishi. Diagramma bloki

3. Asosiy menyudan Window » Show Diagram ni ko'rsatish menyusini tanlash orqali diagramma blokiga o'ting va quyidagi sxemani yarating:



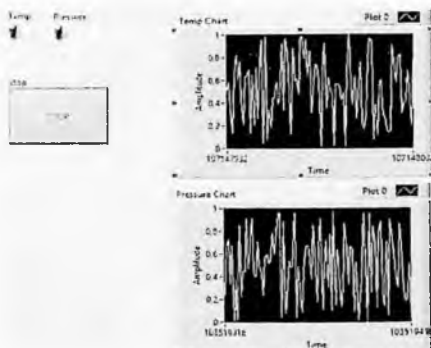
3.10.2-rasm. Tajriba mashg'uloti diagramma bloki sxemasi.

4. VU ni ishga tushiring va VU ni sinab ko'ring.

Talabalar uchun topshiriqlar:

Tasodifiy sonlar bilan "Temp Grafigi" va "Bosim Grafigi" aks etadigan VU yarating. "Temp" o'chirish/yoqish elementi(kaliti)ni tanlaganimizda, ya'ni y*oqib qo'ygan holatda "Temp Chart" grafigi ko'rinadigan "Temp" o'chirish/yoqish elementi(kaliti)ni o'chirib qo'yganimizda "Temp Chart" grafigi ko'rinadigan holatda bo'lishi kerak. Shuningdek, "Bosim" va "Bosim Grafigi" o'chirish/yoqish elementi(kaliti) uchun ham shunday va siz STOP tugmasini tanlaganimizda tasodifiy sonlardan iborat grafik shakllantirish to'xtasin.

VU ning yuzadagi paneli quyidagicha bo'ladi:



3.10.3-rasm. Tajriba mashg'uloti uchun vazafaning yuzadagi panel ko'rinishi.

TAJIRIBA MASHG'ULOTI №9

Virtual ichki uskunalarni yaratish.

Ishtdan maqsad: Ushbu mashqda biz virtual uskunani yaratamiz va ulardan dasturlarimiz va boshqa ilovalarimizda foydalanish tajribasiga ega bo'lish.

Vazifa: Butun sonlar massivini yaratish va ularning o'rtacha qiymatini, maksimal qiymatini va minimal qiymatini topish virtual

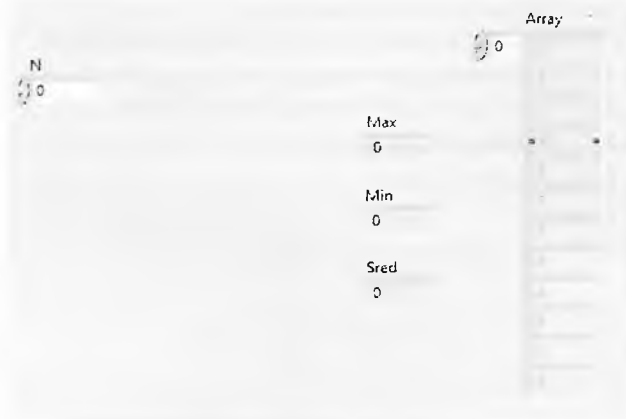
uskunasini yaratish. Keyin ushbu funktsiyalarning barchasini bajaradigan virtual ichki uskuna yaratish va ularni boshqa ilovalarda keng foydalanish uchun saqlang.

Tajriba mashg'ulotini bajarish ketma-ketligi:

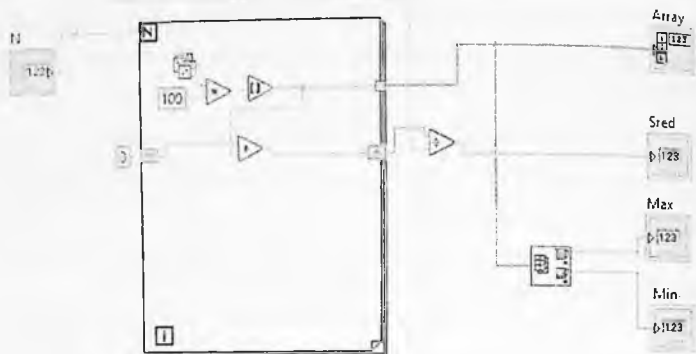
1. Yangi bo'sh VU yarating va darhol VU ni kompyuteringiz xotirasiga nom berib saqlang.

2. Elementlarni yuzadagi panelga joylashtiring, shunda yuzadagi panelingiz quyidagi rasmdagidek bo'lishi kerak.

3. Diagramma blokiga o'ting va quyidagi sxemani yarating.

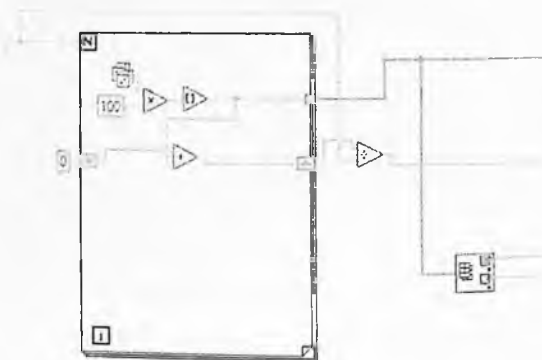


3.11.1-rasm. Tajriba mashg'uloti yuzadagi panel ko'rinishi.



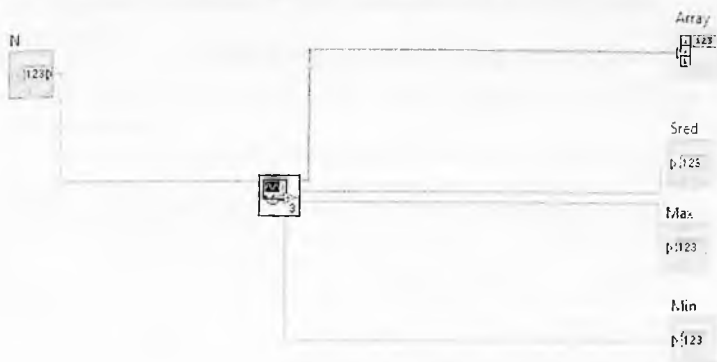
3.11.2-rasm. Tajriba mashg'uloti diagramma bloki sxemasi.

4. Diagramma blokidagi sxemaning boshqaruv elementlari va indikatorlaridan tashqari qolgan qismini belgilang. Masalan, quyidagi rasmda ko'satilganidek:



3.11.3-rasm. Tajriba mashg'uloti diagramma bloki sxemasidan bir qismini belgilash namunasi.

5. Diagramma blokidan bir qismi tanlangandan so'ng, Edit menyusidan Create SubVI ni tanlang. VU ni yaratgandan so'ng, diagramma blokingiz quyidagihca o'zgarishi kerak.



3.11.4-rasm. Tajriba mashg'uloti diagramma bloki sxemasidan bir qismini ichki virtual uskunani yaratishdan keying ko'rinishi.

6. Boshqalar foydalanuvchilar keyinchalik ushbu ichki virtual uskunadan foydalanishlari uchun yangi virtual ichki uskunani kompyuter xotirasiga saqlang.

7. Dasturni ishga tushiring va dasturning to'g'ri ishlashini tekshiring.

Talabalar uchun topshiriqlar:

Vazifa: Tenglamaning ildizlarini topadigan virtual ichki uskuna yarating ($ax^2+bx+c=0$).

III-MODUL BO'YICHA TEST SAVOLLARI

1. LabVIEW da signallarning nechta turi mavjud?

- A.3 ta;
- B.4 ta;
- C.5 ta;
- D.2 ta;

2.Hodisa tuzilmasini diagramma blokida qayerdan toppish mumkin?

- A.Programming >> Structures bo'limidan dan Event Structure
- B.Modern >> Structures bo'limidan Event Structure
- C.Programming bo'limidan Event Structure
- D.Modern bo'limidan Event Structure

3."Sonli o'zgaruvchi qiymatini o'zgartirdingiz" bu qanday hodisa bo'ladimi?

- A.Faol hodisa
- B.Passiv hodisa
- C.Yo'q bu harakat
- D.Yo'q bu vaziyat

4.Hodisa tuzilmasini diagramma blokida joylashtirganizda avtomatik qanday hodisaga moslangan bo'ladi?

- A.Vaqt o'tishi hodisasiga
- B.Tugma bosilishi hodisasiga
- C.Dasturni ishga tushurish hodisasiga
- D.Hech qanday hodisaga moslanmagan

5.Vaqt o'tishi hodisasiga moslangan hodisa tuzilmasida vaqtga qanday qiymat avtomatik berilgan bo'ladi?

- A.0
- B.1
- C.-1
- D.1000

6.Vaqt o'tishi hodisasiga moslangan hodisa tuzilmasida "-1" qiymati belgilangan bo'lsa hodisa tuzilmasi qancha vaqt kutadi yoki qachon bajariladi?

- A.Cheksiz kutadi
- B.0 millisekund kutadi
- C.Kutmaydi
- D.1000 millisekund kutadi

7.Hodisa tuzilmasiga yangi hodisani qayd etish uchun nima qilish kerak?

- A.Yordamchi menyudan Add Event Case... ni tanlash zarur
- B.Yordamchi menyudan Edit Events Handled by this Case ni tanlash zarur
- C.Yordamchi menyudan Duplicate Event Case ni tanlash zarur
- D.Yordamchi menyudan Delete Event Case ni tanlash zarur

8.Hodisa tuzilmasiga joriy hodisani o'zgartirish uchun nima qilish kerak?

A.Yordamchi menyudan Edit Events Handled by this Case ni tanlash zarur

- B.Yordamchi menyudan Add Event Case... ni tanlash zarur
- C.Yordamchi menyudan Duplicate Event Case ni tanlash zarur
- D.Yordamchi menyudan Delete Event Case ni tanlash zarur

9.Hodisa tuzilmasiga joriy hodisadan nusxa olish uchun nima qilish kerak?

A.Yordamchi menyudan Duplicate Event Case ni tanlash zarur

B.Yordamchi menyudan Add Event Case... ni tanlash zarur

C.Yordamchi menyudan Edit Events Handled by this Case ni tanlash

D.Yordamchi menyudan Delete Event Case ni tanlash zarur

10.Hodisa tuzilmasiga joriy hodisani o'chirish uchun nima qilish kerak?

A.Yordamchi menyudan Delete Event Case ni tanlash zarur

B.Yordamchi menyudan Add Event Case... ni tanlash zarur

C.Yordamchi menyudan Edit Events Handled by this Case ni tanlash zarur

D.Yordamchi menyudan Duplicate Event Case ni tanlash zarur

11.Hodisa tuzilmasida qanday qo'shimcha terminallar mavjud va qaysilar?

A.2 ta: ma'lumotlari bog'lami(Event Data Node) va hodisa filtrlari bog'lami(Event Filter Node)

B.2 ta: ma'lumotlari bog'lami(Event Data Node) va hodisa qidiruvi bog'lami(Event Find Node)

C.1 ta: ma'lumotlari bog'lami(Event Data Node)

D.1 ta: hodisa filtrlari bog'lami(Event Filter Node)

12. LabVIEW da necha turdagi hodisalar mavjud va qaysilar?

- A.2 turdagi: Xabar-hodisalar va filtr hodisalar
- B.2 turdagi: Xabar-hodisalar va qidiruv hodisalar
- C.1 turdagi: Xabar-hodisalar
- D.1 turdagi: Filtr hodisalar

13. Filtr hodisalarini xabar hodisalaridan qanday ajratib olish mumkin?

- A. Xabar-hodisalar oldida yashil rangdagi yo'naltirgich bo'lsa, filtr-hodisalar oldida qizil yo'naltirgichlar bo'ladi
- B. Xabar-hodisalar oldida qizil rangdagi yo'naltirgich bo'lsa, filtr-hodisalar oldida yashil yo'naltirgichlar bo'ladi
- C. Xabar-hodisalar oldida yashil rangdagi yo'naltirgich bo'lsa, filtr-hodisalar oldida ko'k yo'naltirgichlar bo'ladi
- D. Xabar-hodisalar oldida ko'k rangdagi yo'naltirgich bo'lsa, filtr-hodisalar oldida qizil yo'naltirgichlar bo'ladi

14. LabVIEW da necha xil o'zgaruvchilar va turlari mavjud?

- A.3 xil: mahalliy, global va tarmoqli
- B.2 xil: mahalliy va global
- C.3 xil: mahalliy, global va shartli
- D.1 xil: mahalliy

15. Mahalliy o'zgaruvchilariga to'g'ri ta'rif berilgan variantni tanlang.

A. diagrammalar blokining turli nuqtalaridan o'tkazuvchi ma'lumotlarini ob'yekt terminaliga ulash imkoniyatingiz bo'lmagan holatda yuzadagi panel ob'yektiga murojaat etish imkonini beradi;

B. ixtiyoriy tipdagi qiymatlarga bir necha VU orasida bir vaqtning o'zida bir nechta VU baravar ishlaganda yoki VIU lar orasida signallarni uzatish imkoniyatingiz bo'lmaganda ularga murojaat etish imkonini beradi;

C. xuddi global o'zgaruvchilarga o'xshaydi, lekin ular turli mahalliy va tarmoq dasturlarida ishlaydi;

D. Mahalliy tarmoqda ishlaydi

16. Global o'zgaruvchilariga to'g'ri ta'rif berilgan variantni tanlang

A. ixtiyoriy tipdagi qiymatlarga bir necha VU orasida bir vaqtning o'zida bir nechta VU baravar ishlaganda yoki VIU lar orasida signallarni uzatish imkoniyatingiz bo'lmaganda ularga murojaat etish imkonini beradi;

B.diagrammalar blokining turli nuqtalaridan o'tkazuvchi ma'lumotlarini ob'ekt terminaliga ulash imkoniyatingiz bo'lmagan holatda yuzadagi panel ob'ektiga murojaat etish imkonini beradi;

C.xuddi global o'zgaruvchilarga o'xshaydi, lekin ular turli mahalliy va tarmoq dasturlarida ishlaydi;

D.Internet tarmog'ida ishlaydi

17.Tarmoq o'zgaruvchilariga to'g'ri ta'rif berilgan variantni tanlang.

A.xuddi global o'zgaruvchilarga o'xshaydi, lekin ular turli mahalliy va tarmoq dasturlarida ishlaydi;

B.diagrammalar blokining turli nuqtalaridan o'tkazuvchi ma'lumotlarini ob'ekt terminaliga ulash imkoniyatingiz bo'lmagan holatda yuzadagi panel ob'ektiga murojaat etish imkonini beradi;

C.ixtiyoriy tipdagi qiymatlarga bir necha VU orasida bir vaqtinchi o'zida bir nechta VU baravar ishlaganda yoki VIU lar orasida signallarni uzatish imkoniyatingiz bo'lmaganda ularga murojaat etish imkonini beradi;

D.Korporativ tarmoqda ishlaydi

18.Bir o'zgaruvchi yordamida parallel tsikllarni nazorat etish uchun qanday o'zgaruvchilardan foydalanish mumkin?

A.Mahalliy

B.Global

C.Tarmoq(umumiy)

D.Tsiki o'zgaruvchisi

19.Ixtiyoriy boshqarish elementi indikator va aksincha boshqarish elementi sifatida ishlatilishi uchun qanday o'zgaruvchilardan foydalanish mumkin?

A.Mahalliy

B.Global

C.Tarmoq(umumiy)

D.Tsiki o'zgaruvchisi

20.O'QISH rejimidagi mahalliy o'zgaruvchi bu - ...

A.Boshqarish elementi

B.Indikator

C.Variant tuzilmasin

D.Tsiki

21.YOZISH rejimidagi mahalliy o'zgaruvchi bu - ...

A.Indikator

B.Boshqarish elementi

C.Variant tuzilmasin

D.Formul bog'lami

22.Xususiyatlar paneli nima uchun ishlatiladi?

A.Yuzadagi panel ob'yektlarini(boshqarish elementi va indikatorlar) xususiyatlari (masalan rangi, yozuvi, jolashuvi, ko'rinish/ko'rinmasligi va hokazo) dasturiy-dinamik(dastur ishlayotganda) o'zgartirish uchun;

B.Yuzadagi panel ob'yektlarini(boshqarish elementi va indikatorlar) xususiyatlari(masalan rangi, yozuvi, jolashuvi, ko'rinish/ko'rinmasligi va hokazo) o'zgartirish uchun;

C.Diagramma blokidagi terminallarni xususiyatlari(masalan rangi, yozuvi, jolashuvi, ko'rinish/ko'rinmasligi va hokazo) o'zgartirish uchun;

D.Diagramma blokidagi terminallarni xususiyatlari (masalan rangi, yozuvi, jolashuvi, ko'rinish/ko'rinmasligi va hokazo) dasturiy-dinamik(dastur ishlayotganda) o'zgartirish uchun;

23.Xususiyatlar bog'lamini qanday chiqarish kerak?

A.Yordamchi menyudan Create >> Property Node

B.Yordamchi menyudan Property Node

C.Yordamchi menyudan Invoke node

D.Yordamchi menyudan Create >> Invoke Node

24.LabVIEW da VIU(virtual ichki uskuna) nima?

A.bu VU(virtual uskuna) bo'lib, uni siz boshqa VU dan chaqirasiz

B.virtual uskunaning interfaol foydalanadigan interfeysi

C.LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi

D.foydalanuvchi kiritadigan ma'lumotlar

25.LabVIEW da tezkor VIU larni yaratish uchun qaysi buyruqdan foydalanish kerak?

A.Edit-Create Sub VI

B.File-Create Sub VI

C.Tools--Create Sub VI

D.VI-Create Sub VI

26.LabVIEW da tezkor VIU larni yaratish uchun birinchi navbatda qanday ishni bajarish zarur?

A.Diagramma blokidan kerakli sxema qismini tanlash kerak

B.Yuzadagi paneldan kerakli boshqarish elementlari va indikatorlarni tanlash kerak

C.Diagramma blokidan tsiklni tanlash kerak

D.Diagramma blokidan variant tuzilmasini tanlash kerak

27.LabVIEW da tezkor VIU larni yaratish uchun birinchi shart qanday?

A.28 ta dan kirish va chiqishga ega bo'lmashligi kerak

B.26 ta dan kirish va chiqishga ega bo'lmashligi kerak

C.24 ta dan kirish va chiqishga ega bo'lmashligi kerak

D.14 ta dan kirish va chiqishga ega bo'lmashligi kerak

28.LabVIEW da tezkor VIU larni yaratish uchun ikkinchi shart qanday?

A. Tuzilma ichida va tashqarisida elementlarni tanlamaslik kerak

B. For Loop sikli ichida va tashqarisida elementlarni tanlamaslik kerak

C. Variant tuzilmasi ichida va tashqarisida elementlarni tanlamaslik kerak

D. While Loop sikli ichida va tashqarisida elementlarni tanlamaslik kerak

29.LabVIEW da kodlarni olib olib tashlash tuzilmalari an'anaviy dasturashdagi qaysi konstruktsiya bilan mos keladi?

A.Izoh

B.Shart operatori

C.Tsikl operatori

D.O'cherish operatori

30.LabVIEW da kodlarni olib olib tashlash tuzilmalari turi nechta va qaysilar?

A.2 ta: Diagram Disable va Conditional Disable

B.2 ta: Diagram Disable va Case Disable

C.1 ta: Diagram Disable

D.1 ta: Conditional Disable

31.LabVIEW da kodlarni olib olib tashlash tuzilmalari qayerda joylashgan?

A.Programming >> Structures

B.Tools >> Modern

C.Programming

D.Modern

32.LabVIEW da kodlarni olib tashlash tuzilmalarida 2 ta variantni faollashtirish mumkinmi?

A.Faqat birta tuzilmani faollashtirish mumkin

B.Birorta variantni faollashtirmaslik mumkin

C.3 tani faollashtirish mumkin

D.2 tadan ko'p bo'lmaydi

33. LabVIEW da kodlarni olib tashlash tuzilmalarida faol variantni qanday bilish mumkin?

A. *belgisi mavjudligi

B. ? belgisi mavjudligi

C. + belgisi mavjudligi

D. -belgisi mavjudligi

34. LabVIEW da Diagram Disable va Conditional Disable tuzilmalaridan nima maqsadda foydalaniladi?

A. grafik kodlar bo'lagini bajartirmoqchi bo'lmasak;

B. mahalliy o'zgaruvchilardan foydalanish uchun;

C. global o'zgaruvchilardan foydalanish uchun;

D. grafik kodlar bo'lagini bajartirmoqchi bo'lsak;

IV-MODUL. DELPHI DASTURLASH TIZIMI BILAN TANISHISH.

§4.1. Delphi tizimi haqida ma'lumot. Delphi visual modeli

Delphi – dasturlarni ishlab chiqishning visual muhiti. Bu tizim dasturchidan dasturlashning barcha murakkabliklarini yashirib dastur yaratish jarayonini qiziqarli ishga aylantiradi. Oddiy dasturlarni, utilitalarni va ma'lumotlar bazasi bilan bog'liq dasturlarni yaratishda sizga registrlar, steklar va boshqa tushunchalar haqida o'ylashga hojat yo'q.

Delphi tizimi qobig'i

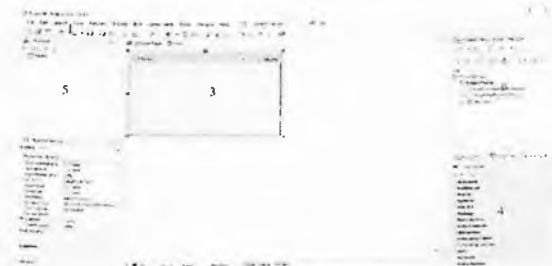
Delphi dasturlash tizimini ishga tushurish quyidagicha amalga oshiriladi:

Eski versiyalarida – Пуск | Все Программы | Borland Delphi Delphi

Yangi versiyalarida–Пуск |Все Программы (Приложения) Embarcadero Rad Studio XE3 | Delphi XE3

Delphi tizimi qobig'i bilan yaxshiroq tanishish uchun oddiy bo'sh loyiha yaratsak:

- File | New | VCL Forms Application.



4.1.1-rasm. Delphi tizimining asosiy oynasi.

Endi rasmda ko'rsatilganidek panellarni ko'rib o'tsak:

1.Dasturning bosh oynasi;

2.Ob'yektlar inspektori: **Properties**-xususiyatlar; **Events**-

hodisalar;

3.Forma;

4.Komponentlar palitrasi;

5.Komponentalar daraxti;

6.Loyihalar menedjeri.

Delphi visual modeli

Delphi tizimi – dasturlarni ishlab chiqishning visual muhiti hisoblanadi. Bu shuni bildiradiki, dasturning tashqi ko'rinishning ko'p qismini forma dizaynerida kerakli ob'yektlarni joylashtirish orqali sichqoncha yordamida amalga oshirib bo'ladi. Bunday harakatlar kubiklar yordamida uy qurishga o'xshab ketadi. Siz kelajakdagi dasturingiz tuzilishini qurasiz va bunda dasturning o'zida grafik interfeys qanday qurilishi haqida o'ylamaysiz ham.

Delphi dasturlarni ishlab chiqish muhiti dasturiy ilovalarni qurish ishini osonlashtiradi va dasturchilar hayotini osonlashtiradi. Bunda dasturchi ko'p diqqatini dasturning bajarilish mantig'iga va dasturning matematik qismini yaratishga bag'ishlashi mumkin.

Albatta amaliy jihatda komponentlarni ko'rib o'tish juda qiziq, lekin ob'yektga yo'naltirilgan dasturlash funktsiyalaridan to'la qonli foydalanish va uni ma'nosiga borish uchun ob'yektga yo'naltirilgan dasturlashning asosiy tushunchalarini bilib olsak maqsadga muvofiq bo'lar edi. Biz siz bilan ba'zi nazariy tushunchalar, ya'ni ob'yektlarga bog'liq, sinflarga bog'liq va Delphi komponentalar modeliga bog'liq nazariy tushunchalarni ko'rib o'tamiz. Bu asosiy fundament bo'lib, uni barcha dasturchilar tushunishlari zarur.

Protседurali dasturlash

Birinchi dasturlar

• *Listing 4.1. Chiziqli dastur matni*

• *Buyruq 1*

• *Buyruq 2*

• *Buyruq 3*

• *Buyruq N.*

Chiziqli dasturlar mantig'i

Listing 4.2. shartli o'tishga misol

Agar shart bajarilsa, unda 1 -buyruqqa o'tish, aks holda 3-buyruqqa o'tish. Buyruq 1

Buyruq 2

Buyruq 3

Buyruq N.

Bunday dasturlashda murakkab dasturlarni yaratish juda qiyin.

Keyingi bosqich protsedurali qarash. Bunday qarashda dasturning ma'lum bir kodlari alohida bloklarga (protседuralar) ajralishi mumkin edi. So'ngra bu blokni dasturning ixtiyoiy nuqtasidan chaqirish mumkin edi. Misol:

Listing 4.3. Protседurali dasturlash

1-Protседurani boshlanishi

Buyruq 1

Buyruq 2

1-Protседurani tugashi Dasturni boshlanishi Buyruq 1

Buyruq 2

Agar shart bajarilsa, 1-protседurani bajar. Buyruq 3

Dasturni tugashi.

Natijada bir va o'sha kodni dasturda bir necha marotaba ishlatish imkoniyati paydo bo'ldi. Dastur kodi oddiyroq va tuhunarliroq bo'ldi.

Listing 4.4. Parametrli protseduralarga misol

1-Protседura boshlanishi (1-o'zgaruvchi: qator) Buyruq 1

Buyruq 2

1-Protседurani tugashi Dasturni boshlanishi Buyruq 1

Buyruq 2

Agar shart bajarilsa, unda 1-Protседurani bajar. Buyruq 3 Dastur tugashi

Real misol:

Listing 4.5. Real misolga yaqin prosedurali dastur

1-Prosedura boshlanishi (1-o'zgaruvchi: butun son)

1-o'zgaruvchidagi sonning faktorialini hisoblash. Natijani ekranga chiqarish.

1-Prosedurani tugashi

Dasturni boshlanishi

1-Prosedura (10)

1-Prosedura (5)

1-Prosedura (8) Dasturni tugashi.

Lekin protsedura – bu hayot ikir-chikiri. Protседuradan ko'ra funktsiyani ishlatish ma'qulloq.

Funksiya – bu xuddi o'sha protsedura, faqat u qiymat qaytarishi mumkin, ya'ni o'zi bajargan ishni natijasi.

Listing 4.6. Funktsiyaga misol

1-Funktsiyani boshlanishi: Qaytariladigan qiymat tipi — butin son

Buyruq 1

Buyruq 2

1-Funktsiyani tugashi

Dasturni boshlanishi

Buyruq 1

Buyruq 2

Agar shart bajarilsa, unda 1-Funktsiyani bajar. Buyruq 3

Dasturni tugashi.

Obyektga yo'naltirilgan dasturlash

Dasturlash texnologiyasining keyingi bosqichi bu ob'yektga yo'naltirilgan dasturlash paydo bo'lishi bo'ldi. Bunda dasturchi nafaqat protsedura yoki funktsiyalardan foydalanadi, balkim butun sinflar bilan ishlaydi.

Sinf – xususiyat, metodlar va hodisalar to'plami. "To'plam" nima degani? Bu shuni bildiradiki sinf xususiyat, metodlar va hodisalardan iborat bo'lib, ular bu sinfni to'laqonli ishlashini ta'minlaydi. Keling tugmani tasavvur qilaylik:

- Xususiyatlar (rang, tugmadagi matn, matn shrifti va hokazo);

- Hodisalar (foydalanuvchi kiritishi hodisasi, masalan tugmaga bosish);

- Metodlar (tugmani ishlashini ta'minlovchi, masalan, matnni qayta chizish va hokazo).

Agar bularnig hammasini birlashtirsak, avtonom sinf paydo bo'ladi va u turli sharoitlarda ishlaydi. Bunda tugmani formaga joylashtirsak bo'ldi u ishlayveradi. Xususiyatar, hodisalar va metodlar qanday tashkil etiladi?

Xususiyat – bu o'zgaruvchilar bo'lib, sinfning holatiga ta'sir ko'rsatadi. Masalan, kengligi, balandligi.

Metodlar – bu o'sha protsedura va funktsiyalar bo'lib, sinf bajarishi mumkin bo'lgan ish (hisoblash). Masalan, protsedura biron-bir matnni ekranga chiqarishi mumkin. Bu protsedura o'sha metodi bo'lib, sinfga tegishli.

Hodisa – bu o'sha protsedura va funktsiya bo'lib, ma'lum bir hodisa yuzaga kelganda chaqiriladi, faqat bu protseduralar sinfga tegishli bo'lishi ham mumkin yoki undan tashqari bo'lishi ham mumkin. Masalan, ma'lum bir xususiyat o'zgarsa ma'lum hodisa yuz bersin va mos protsedura chaqiriladi.

Ob'yekt – bu sinf ekzemplari. Sinf yordamida ma'lum mazmunni belgilaysiz.

Endi xususiyat, metodlar va hodisalarni yagona butun sifatida ko'rib o'tsak. Yana misol sifatida tugmani olamiz. Bunday sinf quyidagi minimal to'plamdan iborat bo'lishi kerak:

Xususiyat:

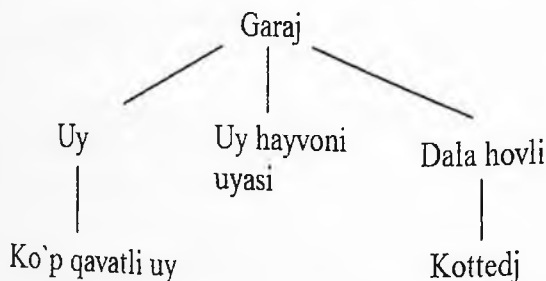
- Chap pozitsiya;
- Yuqori pozitsiya;
- Kengligi;
- Balandligi;
- Sarlavha; Metodlar:
- Tugmani yaratish;
- Tugmani o'chirish;
- Tugmani chizish; Hodisa:
- Tugma bosildi;
- Tugma sarlavhasi o'zgardi;

Komponentali model

Komponentalar – bu yanada mukammal ob'yektlar. Oddiy qilib aytganda komponentalar – bu ob'yektlar bo'lib, ular bilan vizual ishlash mumkin va bu uchun ularga kerakli xususiyat va metodlar mavjud.

Vorislik, Polimorfizm, Inkapsulatsiya

OYD ning eng buyuk yutuqlaridan biri bu – vorislik.



4.1.2-rasm. OYD vorislik misoli.

Polimorfizm

OYD yana bir imkoniyat bo'lib u juda qulaylik tug'diradi bu – Polimorfizm. Bu nima o'zi? Bunda turli ierarxiyadagi ob'yektlar bir xil hodisaga turlicha javob beradi. Bu hali hammasi emas. OYD lashda ajdodni metodini chaqirish qulay imkoniyati mavjud. Quyidagi misolni ko'rsak:

Listing 4.11. Ajdod metodini chaqirish

Garajni devorini yaratishga javob beruvchi protsedura.
Boshlanishi

Devorlarni yaratish

Tugash

Uyda devorlarni yaratishga javob beruvchi protsedura.

Boshlanishi

Ajdod ob'yekt metodini chaqirish. Devorga tasvirlarni osish.

Tugash

Inkapsulatsiya

Juda chiroyli va tushunarsiz so'z – Inkapsulatsiya.

Inkapsulatsiya – bu xususiyat bo'lib, u tufayli ma'lum bir qurish blokni ishlatuvchi ishlab chiquvchiga u qanday tuzilishi haqida bilishi shart emas.

Asosiy tayanch so'z va iboralar

Delphi, properties, events, component, loyiha, protsedura, funktsiya, ob'yektga yo'naltirilgan dasturlash, sinf, xususiyat, metod, vorsklik, polimorfizm, inkapsulyatsiya.

Takrorlash va mustaqil ishlash uchun savollar

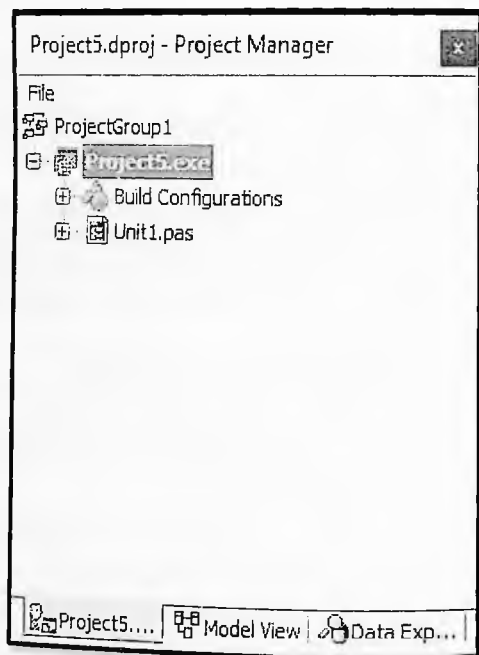
1. Delphi tizimi qobig'i haqida nimani bilasiz?
2. Delphi visual modeli haqida nimani bilasiz?
3. Protседurali dasturlash haqida yozing.
4. Ob'yektga yo'naltirilgan dasturlash nima?
5. Komponentali model nima?
6. Vorsklik, Polimorfizm, Inkapsulatsiya nima?

§4.2. Delphi dasturlash tizimi asoslari. Komponentlar bilan ishlash.

Delphida yaratilgan loyihaning nimadan iborat ekanligini tushunish uchun Delphi tizimida oddiy dasturni yaratib o'rganish maqsadga muvofiq.

Yaratiladigan yangi loyiha uning tarkibini o'rganish uchun eksperiment sifatida xizmat qiladi.

Bu uchun File | New | VCL Forms Application menyusidan foydalanamiz.



4.2.1-rasm. Delphi tizimida loyiha daraxti.

Loyihalar menedjerini ko'rib o'tsak, unda daraxt shoxalari paydo bo'ldi:

- Project Group X(Daraxt sarlavhasi)-loyihalar guruhi nomi;
 - Project(X).exe-loyiha nomi(ilo va);
 - Unit1.pas-modul. Loyiha modullardan tashkil topadi.
- Loyihaning har bir oynasi alohida modulda joylashgan bo'ldi;
- Unit1.dfm-vizual forma.

Loyihalar menedjeri bilan ishlaganda shuni inobatga olish kerakki, menedjerdan o'chirilgan fayllar real diskdan o'chirilmaydi. Ularni real diskdan o'chirmoqchi bo'lsangiz bu uchun fayl menedjerlaridan foydalanish kerak. Loyiha yaratilganda uni alohida papkaga saqlash maqsadga muvofiq va yangi yaratilgan loyihani saqlashda birinchi modulning ismlari so'rab saqlanadi keyin esa loyihaning ismi.

Loyihani saqlaganda Delphi tizimi loyihani *.dpr(yangi versiyalarida

*.dproject) kengaytmasida saqlaydi. Loyihani qaytib ochish zarur bo'lsa, siz shu

*.dpr kengaytmali faylni ochishingiz zarur. Ko'pgina yosh dasturchilar loyihani ochish uchun *.pas kengaytmali faylni ochishadi, bu esa xato. *.pas kengaytmali fayllar loyihaning faqat bir qismi xolos.

Agar siz bilan loyihaning saqlagan real diskdagi papkasini tahlil etsak unda bir qator fayllar paydo bo'lgan:

- *.cfg-loyihani sozlashlarini saqlovchi fayl;
- *.dof-loyihani opsiyalarini saqlovchi fayl;
- *.res-bunday kengaytmali fayllar loyiha resurslarini o'zida saqlaydi(maslan: ikonka, kursor va hokazo);
- *.dcu-loyiha modulining oraliq kompilatsiyalangan holati;
- *.ddp-loyiha modullarining diagrammalari.

Delphi tizimida ishlaganda dasturchilar odatda *.pas kengaytmali fayllarga o'zgartirish kiritib kod yozadilar.

*.dfm kengaytmali fayllar esa avtomatik generatsiyalanadi. Ba'zan ushbu fayllarga ham o'zgartirish kiritishga to'g'ri kelib qoladi.

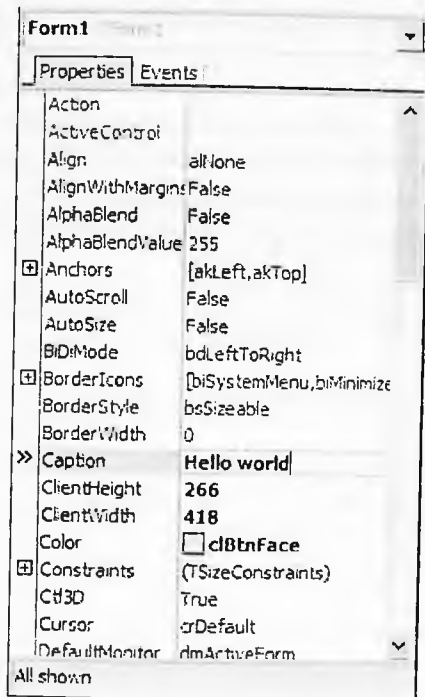
```

object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'Form1' ClientHeight = 201
  ClientWidth = 447
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET   Font.Color =
clWindowText Font.Height = -11
  Font.Name = 'Tahoma' Font.Style = [] OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
End

```

Loyihani kompilatsiyalash uchun Ctrl+F9 tugmasini yoki dastur yaratish uchun esa F9 tugmasini tanlash kifoya.

Loyiha kompilatsyalangandan so'ng, loyihani saqlagan papkada *.exe kengaytmali fayl hosil bo'ladi. Uni bimalol ishlatishimiz mumkin. Delphi tizimida hamma narsa ob'yekt sifatida qaralishini inobatga olsak, loyihamiz oynasi ham ob'yekt demak uni sarlavhasini bimalol "Ob'yektlar inspektori(Object Inspector)" ning xususiyatlar bo'limining "Caption"(Sarlavha)dan o'zgartirishimiz mumkin(Masalan: guruhingiz nomi).



4.2.2-rasm. Delphi tizimida Object Inspector oynasi.

Delphi dasturlash tili

Izohlar – dastur kodiga hech qanday ta'sir o'tkazmaydigan ixtiyoriy matn. U hech qachon kompilatsiya qilinmaydi va faqat kodni izohlash uchun ishlatiladi. Izohlar 2 usulda ishlatiladi:

- 2 sleshdan (//) keyin keladigan barcha matn izoh sifatida qaraladi(bunday usulda bir qatorlik izohlar ishlatiladi);

- Figurali qavslar { } ichidagi barcha matn izoh hisoblanadi (bunday usulda bir nechta qatordan iborat izohlar ishlatiladi).

// Bu izohmi?

Bu izoh yoki izoh emas?

Bu izohmi yoki izoh emas // Bu esa izoh?

{Bu esa yana

Izohmi?}

Loyihaning modulida kelajakdagi dastur kodi shabloni tayyor bo'ladi siz faqat uni to'ldirasiz. Agar siz dasturingizda 2 ta oyna bo'lishini xoxlasangiz yana bir forma qo'shasiz va bu oyna uchun yana bir modul paydo bo'ladi. Keling forma shablonini o'rganib chiqsak:

unit Unit1; //modul nomi

//interfeyslarni e'lon qilish qismi

Interface

uses //bu so'zdan keyin ulanadigan modullar nomi ketma-ket keladi

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs;

type //bu so'zdan keyin tiplar e'loni keladi

TForm1 = class (TForm) //TForm1 yangi ob'yektini e'lon qilish

//Bu qismda komponentlar va hodisalar qo'shiladi

private // bu so'zdan keyin ob'yektning yopiq ma'lumotlari e'lon qilinadi

{private declaration} //Delphi yordamchi so'zni generatsiyalaydi

{Bu yerda faqat TForm1 ob'yektiga murojaatli o'zgaruvchilar va metodlar e'lon qilinadi}

public //bu so'zdan keyin ob'yektning ochiq ma'lumotlari e'lon qilinadi

{public declaration} //Delphi yordamchi so'zni generatsiyalaydi

{Bu yerda faqat TForm1 ob'yektiga murojaatli o'zgaruvchilar va metodlar e'lon qilinadi} end;

var //Global o'zgaruvchilarni e'lon qilish

Form1:TForm1; //TForm1 ob'ekti tipiga mos Form1
o'zgaruvchisi e'loni

//amalga oshirinshni boshlash

Implementation

{ \$R *.dfm } //*.dfm faylini ulash end. //modulni tugashi

Delphida ma'lumotlar tipi

O'zgaruvchi-xotira hududi bo'lib, bunda ma'lum bir ma'lumot saqlanadi. Bu xotirada qanday turdagi malumot saqlanishini bilishimiz uchun biz kompilyatorga o'zgaruvchi tipini e'lon qilish jarayonida ko'rsatishimiz zarur.

4.2.1-jadval.

Delphi tizimida butun sonlar tiplari turlari ro'yxati.

Tip	Qymatlar diapazoni	Xotira hajmi	Izoh
Integer	-2147483648... +2147483647	4 bayt	Ishorali
Cardinal	0...4294967295	4 bayt	Ishorasiz
Shortint	-128...+127	1 bayt	Ishorali
Smallint	-32768...+32767	2 bayt	Ishorali
Longint	-2147483648... +2147483647	4 bayt	Ishorali
Int64	$-2^{63} \dots +2^{63}-1$	8 bayt	Ishorali
Byte	0...255	1 bayt	Ishorasiz
Word	0...65535	2 bayt	Ishorasiz
Longword	0...4294967295	4 bayt	Ishorasiz

Delphi tizimida butun bo'lmagan sonlar tiplari turlari ro'yxati.

Tip	Qymatlar diapazoni	Sondagi maksimal qaram	Xotira hajmi
Real48	$2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$	11-12	6 bayt
Real	$5,0 \cdot 10^{-}$	15-16	8 bayt
Single	$1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$	7-8	4 bayt
Double	$5,0 \cdot 10^{-}$	15-16	8 bayt
Extended	$3,6 \cdot 10^{-}$ 4951 ... $1,1 \cdot 10^{4932}$	19-20	10 bayt
Comp	$-2^{63} \dots 1,2^{63}$	19-20	8 bayt
Currency	-9223372203685477.5 808.... +9223372203685477.5 807	19-20	8 bayt

Ramzli ma'lumotlar tipi

1		31	=	61		91	y	121
2		32	>	62	\	92	z	122
3	!	33	?	63	}	93	{	123
4	-	34	@	64	^	94		124
5	#	35	A	65	_	95	}	125
6	\$	36	B	66	*	96	~	126
7	%	37	C	67	a	97	*	127
8	&	38	D	68	b	98	Ъ	128
9		39	E	69	c	99	Г	129
10	(	40	F	70	d	100	.	130
11	)	41	G	71	e	101	г	131
12	*	42	H	72	f	102	-	132
13	+	43	I	73	g	103		133

	14		44	J	74	h	104	†	134
	15		45	K	75	i	105	‡	135
	16		46	L	76	j	106	€	136
	17	/	47	M	77	K	107	‰	137
	18	0	48	N	78	L	108	Љ	138
	19	1	49	O	79	M	109	«	139
	20	2	50	P	80	N	110	Њ	140
	21	3	51	Q	81	o	111	Ќ	141
	22	4	52	R	82	p	112	Ў	142
	23	5	53	S	83	q	113	Ў	143
	24	6	54	T	84	r	114	Ў	144
	25	7	55	U	85	s	115	•	145
	26	8	56	V	86	t	116	•	146

	27	9	57	W	87	u	117	•	147
	28	:	58	X	88	v	118	•	148
	29	:	59	Y	89	w	119	•	149
	30	<	60	Z	90	x	120	—	150
—	151	•	172	Б	193	Ц	214	л	235
	152		173	В	194	Ч	215	м	236
™	153	©	174	Г	195	Ш	216	н	237
Љ	154	Ї	175	Д	196	Щ	217	о	238
Ј	155	°	176	Е	197	Ъ	218	п	239
Њ	156	±	177	Ж	198	Ы	219	р	240
Ќ	157	‡	178	З	199	Ь	220	с	241
h	158	і	179	И	200	Э	221	т	242
u	159	ѓ	180	Й	201	Ю	222	у	243

	160	μ	181	К	202	Я	223	Ф	244
Ў	161	¶	182	Л	203	а	224	х	245
Ў	162	•	183	М	204	б	225	ц	246
J	163	ё	184	Н	205	в	226	ч	247
□	164	No	185	О	206	г	227	ш	248
Г	165	с	186	П	207	д	228	щ	249
і	166	•	187	Р	208	е	229	ъ	250
§	167	j	188	С	209	ж	230	ы	251
E	168	S	189	Т	210	з	231	ь	252
©	169	s	190	У	211	и	232	э	253
E	170	i	191	Ф	212	й	233	ю	254
•	171	A	192	Х	213	к	234	я	255

4.2.3-rasm. Ramzli ma'lumotlar ro'yxati.

Qatorli(matn) ma'lumotlar tipi

4.2.3-jadval.

Delphi tizimida qatorli(matnli) tiplar turlari ro'yxati.

Tip	Maksimal qator uzunligi(ramzlarda)	Xotira hajmi	Izoh
ShortString	255	2 dan 256 baytgacha	
AnsiString	231	4 baytdan 2Giga baytgacha	8 bitli
WideString	230	4 baytdan 2Giga baytgacha	UNICODE

Delphida qatorlar '(birtalik shtrix) belgilari orasida ko'rsatiladi. Misol:

```
Var  
Str:String; Begin  
Str:='Hello World';  
End;
```

Qator bu - ramzlar massivi ekan demak qatorni alohida ramziga o'zgaruvchi nomidan keyin kvadrat qavslarda[] ramz tartib raqamini ko'rsatish orqali murojaat etish mumkin.

Mantiqiy tip

Ushbu tip o'zgaruvchilar 2 qiymatdan birini qabul qilishlari mumkin: TRUE(rost) yoki FALSE(yolg'on). Misol:

```
Var b:boolean; Begin  
b:=true;  
End;
```

Massivlar

Massivni e'lon qilish uchun quyidagi strukturadan foydalaning:
O'zgaruvchi nomi:array [qiymatlar diapazoni] of massiv
elementlari tipi Misol:

Var b:array[1..99] of integer; Begin

b[1]:=1;

b[2]:=2; End;

Pchartipi

Biz faqat Pchar tipini ko'rib o'tishimiz zarur. Bu o'zgaruvchi tipi WinAPI funktsiyalarida ko'p ishlatiladi.

Pchar o'zgaruvchi tipi – qatorni boshlanishiga ko'rsatkich hisoblanadi.

Konstantalar (o'zgarmaslar)

Konstantalar – o'zgaruvchilar qiymati bo'lib, ularni o'zgartirish mumkin emas. Misol: Const PI=3.14;

Kuchli Variant tipi

Dasturni tuzish jarayonida o'zgaruvchiga qanday tip bo'lishini oldindan bilmasangiz, unda Variant tipidan foydalanishingiz mumkin. Bunda siz bu o'zgaruvchiga ixtiyoriy ma'lumotni o'zlashtirishingiz mumkin, dastur esa o'zi uning tipini aniqlaydi. Misol:

Var V:Variant; Begin

V:=5;

V:='Bu matn'; V:=True; V:=3.14; End;

Protsedura va funktsiyalar

Protsedura va funktsiya – kod bloki bo'lib, alohida blok sifatida ajratilgan.

Procedure Exam1(); Vari:Integer; //Mahalliy o'zgaruvchini e'lon qilish Begin

i:=10; //Mahalliy o'zgaruvchiga qiymat o'zlashtirish End;

Function Exam1(index:Integer):Integer; Begin
Result:=index*2; {index berilgan qiymatni 2 ga ko'paytirib
qaytaramiz} End; Begin a:=Exam1(100); End; Ob'yekt metodlari

Metodlar – bu protsedura va funktsiyalar bo'lib, biron-bir ob'yektga tegishli bo'ladi.

Metodlar ob'yektlar tarkibida e'lon qilinadi va bir necha tipda bo'ladi:

□ Static – oddiy protsedura va funktsiyalar. Ob'yekt vorislarida uni o'zgartirib bo'lmaydi;

□ Virtual – bunday metodlar ob'yekt vorislarida qayta e'lon qilinishi mumkin;

□ Dynamic – bu xuddi *Virtual* singari lekin xotira tejiladi;

□ Abstract – bunday metod faqat ob'yektni e'lon qilishda ishlatilib, uning realizatsiyasi bo'lmaydi.

Asosiy tayanch so'z va iboralar

Loyiha menedjeri, ilova, modul, forma, ob'yekt inspektori, izohlar, butun tip, ramzli ma'lumotlar, matli tip, mantiqiy tip, massivlar, Pchar, konstanta, variant tipi.

Takrorlash va mustaqil ishlash uchun savollar

1. Delphi tizimida loyiha nimadan iborat?
2. Loyihani takomillashtirish, o'zgartirish va tahrirlash uchun nima uchun .dpr kengaytmali fayldan foydalanish zarur? Nega bunda .pas yoki dfm kengaytmali fayllardan foydalanib bo'lmaydi?
3. Loyihada yana qanday tipdagi fayllarni uchratish mumkin va ular nima maqsadda xizmat qiladi?
4. Object Inspector oynasining "Properties" bo'limi vazifalari haqida nimani bilasiz?
5. Object Inspector oynasining "Events" bo'limi vazifalari haqida nimani bilasiz?

§4.3. Komponentlar bilan ishlash. Standart komponentlar palitrasi.

Delphi tizimida sinflar juda yaxshi o'ylangan. Bu paragrafda biz asosiy forma va uning xususiyatlari bilan tanishib chiqamiz. Lekin undan oldin biz sinflar ierarxiyasini ko'rib o'tsak:

- ☐ System.Object – barcha ob'yektlar uchun asos;
- ☐ Classes.TPersistent – o'zini boshqa sinfga ifodalash uchun sinf;
- ☐ Classes.TComponent – novizual sinf uchun funktsionallik sinfi;
- ☐ Controls.TControl – ko'rinishga ega barcha vizual komponentlar sinfi;
- ☐ Controls.TWinControl – identifikatorga ega barcha Windows komponentlar shu sinfdan kelib chiqadi;
- ☐ Forms.TScrollingWinControl – siljitib ko'rish imkoniyatiga ega komponentalar uchun asos sinf;
- ☐ Forms.TCustomForm – forma va dialog oynalar uchun asos sinf.

Formaning asosiy xususiyatlari va hodisalari

- ☐ ActiveControl – jimglik qoidasi bo'yicha formadagi qaysi komponent avtomatik faol bo'lishi ko'rsatiladi;
- ☐ Align –komponentni tekshlashtirish;
 - > AlNone-tekshlamaslik;
 - > AlBottom-Quyida chegara bo'yicha tekshlash;
 - > AlLeft-Chap chegara bo'yicha tekshlash;
 - > AlRight-O'ng chegara bo'yicha tekshlash;
 - > AlTop-Yuqori chegara bo'yicha tekshlash;
- ☐ AlphaBend – forma shaffoflikka ega yoki ega emasligi;
- ☐ AlphaBendValue – shaffoflik darajasi(0-255);
- ☐ AutoScroll – forma avtomatik skrollingga egaligini ko'rsatadi;

☐AutoSize – agar bu xususiyat TRUE bo'lsa forma o'z hajmini uning o'zining tarkibiga qarab moslashtiradi;

☐BorderIcons – oynada qanday tugmalar bo'lishini ko'rsatadi;

➤biSystemmenu-menyuni ko'rsatish;

➤biMinimize-oynani minimallashtirish tugmasi;

➤biMaximize-oynani maksimal kattalashtirish tugmasi;

➤biHelp-yordam oynani

☐BorderStyle – Oyna atrofini tuzlashini tanlash;

➤bsSizeable-oz hajmini o'zgartirishi mumkin;

➤bsDialog-dialog oynasiga o'xshaydi;

➤bsNone-chegarasi mavjud emas;

➤bsSingle-oddiy oyna lekin uning hajmini o'zgartirib bo'lmaydi;

☐Caption – oyna sarlavhasi;

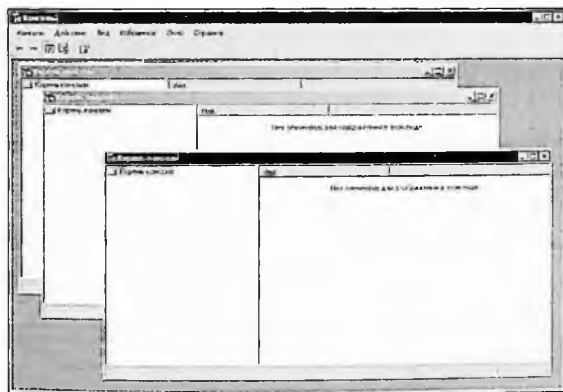
☐Color – forma rangi;

☐Cursor – formada kursorning ko'rinishi;

CrNone	Her	CrArrow	
CrCross		CrIBeam	
CrSizeNESW		CrSizeNS	
CrSizeNWSE		CrSizeWE	
CrUpArrow		CrHourGlass	
CrDrag		CrNoDrop	
CrHSplit		CrVSplit	
CrMultiDrag		CrSQLWait	
CrNo		CrAppStart	
CrHelp		CrHandPoint	
CrSize		CrSizeAll	

4.3.1-rasm. Formada sichqoncha ko'rsatkichlari ro'yxati.

- ☐ Enabled – forma murojaatligini ko'rsatadi;
- ☐ Font – oyna shrifti;
- ☐ FormStyle – forma stili;
- fsNormal-oddiy oyna;
- fsMDIForm-MDI oynalari uchun ota-ona formasi hisoblanadi;
- fsMDIChild-MDI bola formasi hisoblanadi;
- ☐ Height – balandligi;
- ☐ Hint – shipshitish yozuvi;



4.3.2-rasm. MDI forma ko'rinishi.

- ☐ Icon – forma ikonkasi;
- Load – yuklash;
- Save – saqlash;
- Copy – nusxalash;
- ...
- ☐ Menu – qaysi menyu oynaning asosiy menyusi ekanligini ko'rsatadi;
- ☐ Name – forma nomi(dastur kodida ishlatiladi);
- ☐ ShowHint – shipshitish yozuvlarini chiqarish yoki chiqarmaslikni bildiradi;

- ☐ Tag – yordamchi butun sonli xususiyat;
- ☐ Visible – forma ko'rinadigan yoki ko'rinmaydiganligini bildiradi;
- ☐ Position – dastur ishlaganda oynani paydo bo'lish pozitsiyasi;
 - > PoDefault-windows o'zi belgilaydi;
 - > PoDefaultPosOnly – Windows faqat oyna joylashuvini o'zi belgilaydi(hajmini emas);
 - > PoDefaultsizeOnly – Windows faqat oyna hajmini o'zi belgilaydi(joylashuvini emas);
 - > PoDesign – forma hajmi va joylashuvi dizayndagidek;
 - > PoDesktopcenter-ishchi stol o'rtasida;
 - > PoMainCenter-asosiy oyna o'rtasida;
 - > PoScreenCenter-monitor o'rtasida;
- ☐ Width – forma kengligi;
- ☐ WindowState – forma ishlagandagi holati(WsNormal, WsMaximized, WsMinimized).

Windows hodisali modeli

Windows OT ning barcha ishi hodisalarga asoslangan. Windows yadrosi ichida hodisalar navbatini yaratadi. Qachonki biror-bir dastur yoki qurilma o'zining holatini o'zgartirsa va bu haqida operatsion tizimga xabar bermoqchi bo'lsa OT bu hodisa xabarini navbatga joylashtiradi. So'ngra OT uni qayta ishlaydi.

4.3.1-jadval.

Windows hodisalarini qayd etish jadvali ko'rinishi.

Hodisa	Ilova identifikatori	Qo'shimcha
Klavisha bosildi	261	A
Ekranni qayta chizish	385	(12,46,336,267)
Sichqoncha siljidi	261	(365,451)

Delphi tizimida barcha komponentlar hodisalar orqali ishlaydi. Siz dasturchi sifatida har xil holatlar uchun doim hodisalar qayta ishlovchisini yaratasiz. Masalan, klaviaturadagi tugmani bosish hodisasi uchun qayta ishlovchini yaratishingiz mumkin va unda qanaqadir buyruqlarni bajartirishingiz mumkin. Masalan ma'lum bir klaviatura tugmasini bosganda biron-bir oynani chiqarish. Oddiy qilib aytganda hodisa qayta ishlovchisi – bu oddiy protsedura yoki funktsiya bo'lib, ma'lum bir hodisa yuz berganda bajariladi.

Asosiy forma hodisalari

Biz siz bilan formaning asosiy hodisalarini ko'rib o'tamiz.

4.3.2-jadval.

Formaning asosiy hodisalari ro'yxati.

Hodisa	Izoh
onActivate	Dastur faol bo'lgan hodisa
OnClick	Foydalanuvchi sichqoncha chap tugmasi bilan formaga tanlaganda
OnClose	Oyna yopilayotganda yuz beradi
OnCloseQuery	Oyna yopishidan oldin yuz beradi
OnCreate	Oyna yaratilayotganda yuz beradi
OnDbClick	Foydalanuvchi sichqoncha chap tugmasi bilan 2 marotaba chiqillatganda yuz beradi
OnDeactivate	Oyna faol bo'lmaganda yuzaga keladi
OnHide	Oyna no'malum bo'lganda yuzaga keladi
OnKeyDown	Klaviatura tugmasi bosilganda yuzaga keladi
OnKeyPress	Klaviatura tugmasi bosib so'ngra qo'yib yuborilganda yuzaga keladi
OnKeyUp	Klaviatura tugmasi qo'yib yuborilganda yuzaga keladi
OnMouseDown	Sichqoncha tugmasi bosilganda yuz beradi
onMouseMove	Sichqoncha ko'rsatkichi ob'yekt ustiga bo'lganda yuz beradi
onMouseUp	Sichqoncha tugmasi bosib so'ngra qo'yib yuborilganda yuz berdi

OnMouseWhell	Sichqoncha g'ildiragi harakatlanganda yuz beradi
OnMouseWhellDown	Sichqoncha g'ildiragi pastka harakatlanganda yuz beradi
OnMouseWhellUp	Sichqoncha g'ildiragi yuqoriga harakatlanganda yuz beradi
OnPaint	Oynani qayta chizish hodisasi
OnResize	Oyna hajmini o'zgarish hodisasi
OnShortCut	Issiq klavisha bosilganda yuz beradi
OnShow	Oyna ko'rsatilishi boshlanganda yuz beradi

Komponentlar palitasi

Delphi tizimida juda ko'p komponentlar mavjud, ularni joylashuvini eslab qolish uchun guruhlash maqsadga muvofiq bo'lgani sabab Delphi tizimida shunday qulaylik yo'lga qo'yilgan. Bu guruhlar quyidagilar:

- ☐ Standart-Windows komponentlariga o'xshaydi;
- ☐ Additional-qo'shimcha komponentlar;
- ☐ Win32-faqat Win32 OT mavjud komponentlar;
- ☐ System-tizimga murojaatni osonlashtiruvchi komponentlar guruhi;

- ☐ DataAccess-ma'lumotlar bazasiga murojaat komponentlar guruhi;

- ☐ DataControl-ma'lumotlar bazasi bilan ishlash komponentlar guruhi;

- ☐ dbExpress-BDE ma'lumotlar bazasiga murojaat komponentlari o'rniga kelgan komponentlar guruhi;

- ☐ BDE-ma'lumotlar bazasiga ulanishning eski komponentlar guruhi;

- ☐ ADO-Microsoftning ma'lumotlar bazasiga ulanish komponentlar guruhi;

- ☐ InterBase-InterBase ma'lumotlar bazasiga ulanish komponentlar guruhi;

☐ WebServices-Internet tarmog'iga ulanish komponentlar guruhi;

☐ InternetExpress-Internet tarmog'iga ulanish komponentlar guruhi;

☐ FastNet-tarmoq komponentlari;

☐ Dialogs-standart dialoglarga murojaat qiluvchi komponentlar guruhi;

☐ Win3.11-Win3.11 ga murojaat komponentlari guruhi;

☐ Samples-turli xil komponentlar guruhi.

Asosiy tayanch so'z va iboralar

Forma, sinflar ierarxiyasi, ob'yekt xususiyatlari, MDI oyna, Windows hodisali modeli, forma hodisalari, komponentlar guruhi.

Takrorlash va mustaqil ishlash uchun savollar

1. Ob'yektga yo'naltirilgan dasturlash tillarida sinflarning paydo bo'lishi va ularning ierarxiyasi haqida nimani bilasiz?

2. Formaning asosiy xususiyatlari haqida nima bilasiz?

3. Windows OT hodisali modeli qanday tuzilgan?

4. Formaning asosiy hodisalari haqida nima bilasiz?

5. Ob'yektga yo'naltirilgan dasturlash tillarida nega komponentlar guruhlariga bo'lingan?

6. Barcha ob'yektlar komponent sifatida xizmat qiladimi?

§4.4. Ishchi ilovalarni yaratish

Asosiy menyuni yaratish

Dasturda asosiy menyuni yaratish uchun TMainMenu komponentasidan foydalaniladi. Uning quyidagi xususiyatlari mavjud:

☐ AutoHotKeys-tezkor klavishlarni avtomatik yaratishni belgilaydi yoki avtomatik yaratmaslikni belgilaydi;

☐ Images-menyu punktlarida rasmlarni joylashtirish ro'yxati;

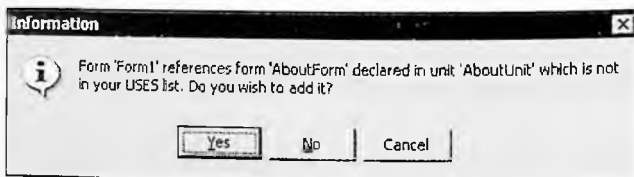
- ☐ Items-menyuni ichki punktlarini yaratish xususiyati;
- ☐ AutoMerge-ichki oynalar menyulari bilan avtomatik birlashtirish xususiyati.

Farzand oynalarni yaratish

Yangi formani yaratish uchun File menyusidan New bo'limi tarkibidan Form punktini tanlash zarur. Shunda Delphi bizga bo'sh forma yaratib beradi. Ikkinchi formaning moduli Unit2.pas bo'ladi, 3-formaning moduli esa Unit3.pas va hokazo. Boshqa oynaning chaqirish uchun formaning nomidan so'ng show yoki showmodal metodidan foydalaning.

Showmodal metodining show metodidan farqi shundaki, bunda oyna faol bo'ladi va unga boshqaruv beriladi. Bu oynani yopmaguncha siz boshqa oynaga o'tib bo'lmaydi.

Biroq siz dasturni ishlatmoqchi bo'lganizda Delphi tizimi xatolik chiqaradi va ochmoqchi bo'lgan formangiz topilmadi deb aytadi hamda unitni uses bo'limiga qo'shishni so'raydi. Siz rozilik bering!



4.4.1-rasm. Boshqa forma unitini chaqiruvchi forma uses bo'limiga qo'shishni so'rash dialog oynasi.

Delphi tizimi avtomatik implementation kalitli so'zidan keyingi qatorda uses kalitli so'zidan keyin sizning ochmoqchi bo'lgan formangiz modulini qo'shib qo'yadi.

Formalar o'rtasida ma'lumot almashinish

Bu formalar nimaga kerak agar ular o'rtasida ma'lumotlarni uzatib bo'lmasa? Biz biror-bir oynani chaqiramiz va foydalanuvchi unga ma'lumotlar kiritishini xoxlaymiz. Shundan so'ng bu kiritilgan ma'lumotlarni asosiy oynaga olishni xoxlaymiz.

Dialog oynasida foydalanuvchi qaysi tugmani tanlaganligini bilish uchun quyidagi **MessageDlg** funksiyasini ko'rib o'tamiz.

Bu metod 3 ta parametrga ega:

- ☐ Dialog oynasiga chiqadigan yozuv;
- ☐ Oyna turi:
 - ❖ **mtWarning**-ogohlantirish;
 - ❖ **mtError**-xatolik;
 - ❖ **mtInformation**-xabar berish;
 - ❖ **mtConfirmation**-tasdiq oynasi;
 - ❖ **mtCustom**-ixtiyoriy;
- ☐ Tugmalar; Tugmalar quyidagilar bo'lishi mumkin:
 - ❖ **MB_YesNo**-YES va NO tugmasi;
 - ❖ **MB_YesNoCancel**-YES, NO va CANCEL tugmasi;
 - ❖ **MB_YesAllNoAllCancel**-YES, YES ALL, NO va NOALL tugmalari;
 - ❖ **MB_OkCancel**-OK va CANCEL tugmasi;
 - ❖ **MB_AbortRetryIgnory**-Abort, Retry va Ignory tugmalari;
 - ❖ **MB_AbortIgnory**-Abort va Retry tugmalari.
- ☐ Javoblar: quyidagilar bo'lishi mumkin:
 - ❖ **mrNone**-tugma tanlanmagan(0);
 - ❖ **mrOK**- OK tugmasi tanlangan(1);
 - ❖ **mrCancel**-Cancel tugmasi tanlangan(2);
 - ❖ **mrAbort**-Abort tugmasi tanlangan(3);
 - ❖ **mrRetry**-Retry tugmasi tanlangan(4);
 - ❖ **mrIgnory**-Ignory tugmasi tanlangan(5);
 - ❖ **mrYes**-Yes tugmasi tanlangan(6);
 - ❖ **mrNo**-No tugmasi tanlangan(7);

- ❖ mrAll-All tugmasi tanlangan(8);
- ❖ mrNotoAll-No to All tugmasi tanlangan(9);
- ❖ mrYestoAll-Yes to All tugmasi tanlangan(10);
- ❖ mrClose-Close tugmasi tanlangan(11);

Ko'p oynali MDI-oynalar

Ko'p oynali MDI-oynalar nima? Bu asosiy oyna tarkibida bir nechta ichki oynalarni saqlaydi. Ichki oynalar nomodal oynalarga o'xshaydi va ular asosiy oynani bloklamaydi, faqat ularning ko'rinish soxasi asosiy oyna chegarasi bilan chegaralangan. Ular xuddi asosiy oyna ichidagidek ko'rinadi.

Ichki MDI-oynalar quyidagi metodlari mavjud:

- ☐ caNone-hech nima bo'lmaydi va oyna yopilmaydi;
- ☐ caHide-oyna yashiriladi, lekin oyna o'chmaydi va xotiradan tozalanmaydi;
- ☐ caMinimize-yopmaslik, oynani minimallashtiradi.

Oynani initsializatsiyalash

Oynalar qanday initsializatsiyalanishini bilish uchun Project menyusida ViewSource punktini tanlang. Bunda 3 ta asosiy qatorga e'tibor berish zarur:

- Application.Initialize-dasturni ishga tayyorash;
- Application.CreateForm(TForm1, Form1)- formalarni yaratish;
- ☐ Application.Run-Dasturni ishga tushurish.

```

program Project5;

uses
  Forms,
  Unit1 in 'Unit1.pas' (Form1),
  Unit2 in 'Unit2.pas' (Form2);

{$R *.res}

begin
  Application.Initialize;
  Application.MainFormOnTaskbar := True;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

```

4.4.2-rasm. Loyihaning asosiy kirish nuqtasi kodi.

```

program Project5;

uses
  Forms, Windows,
  Unit1 in 'Unit1.pas' (Form1),
  Unit2 in 'Unit2.pas' (Form2);

{$R *.res}

begin
  Form2 := TForm2.Create(nil);
  Form2.Show;
  Form2.Repaint;

  Application.Initialize;
  Application.MainFormOnTaskbar := True;
  Application.CreateForm(TForm1, Form1);

  Sleep(1000);
  Form2.Hide;
  Form2.Free;

  Application.Run;
end.

```

4.4.3-rasm. Loyihaning asosiy kirish nuqtasi Splash forma bilan o'zgartirilgan kodi.

Asosiy tayanch soʻz va iboralar

Asisiy menyu, Main Menu komponenti, farzand oyna, formalar oʻrtasida maʼlumot almashinish, messagedlg, MDI-oy-nalar, oynalarni initsializatsiyalash,

Takrorlash va mustaqil ishlash uchun savollar

- 1.Asosiy menyuni yaratish jarayoni haqida nima bilasiz?
- 2.Farzand oynalarni yaratish jarayoni haqida yozing.
- 3.Formalar oʻrtasida maʼlumot almashinish jarayoni nima uchun kerak?
- 4.Koʻp oynali MDI-oy-nalar nima?
- 5.Oynani initsializatsiyalash nima uchun kerak?

§4.5. Obyektga yoʻnaltirilgan dasturlash asoslari usuli

Massivlar bilan ishlash – dasturlashning eng muhim va eng kerakli qirralaridan biri. Biz dasturlarimizda massivlarni juda koʻp ishlatamiz.

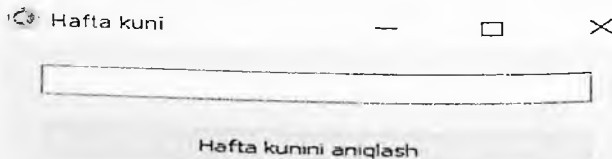
Masssiv nima oʻzi? Bu bir tipdagi maʼlumotlar toʻplami boʻlib, ular ketma- ket joylashgan. Delphida massiv array kalitli soʻzi bilan ifodalanadi.

Massivlarni eʼlon qilish: Var

Massiv_nomi : array [massiv uzunligi] of maʼlumotlar tipi

Misol: Var

R:array[0..11] of integer;



4.5.1-rasm. Massivlarni formada qoʻllash.

```

procedure TForm1.Button1Click(Sender: TObject);
var
  day: Integer;
  week: array[1..7] of String;
begin
  week[1] := 'Yakshanba';
  week[2] := 'Dushanba';
  week[3] := 'Seshanba';
  week[4] := 'Chorshanba';
  week[5] := 'Payshanba';
  week[6] := 'Juma';
  week[7] := 'Shanba';

  day := DayOfWeek(Date);
  Edit1.Text := week[day];
end;

```

4.5.2-rasm. Massivlarni dastur kodida qo'llash.

Ikki o'lchamli massivlarni e'lon qilish: Var

Massiv_nomi : array [massiv uzunligi, massiv uzunligi] of
ma'lumotlar tipi

Misol: Var t:array[0..3,0..4] of integer;

Fayllar bilan ishlash

Fayllar

Fayl – bu nomga ega disk hududi.

Fayllar

Matnli	Binar	Papkalar (kataloglar)
Faqat matn, boshqarish ramzlariga ega emas (32 kichik kodlar)	Ramzlar kodlari jadvalining ixtiyoriy ramzlarini saqlaydi	
ASCII (har bir ramzga 1 bayt)	*.doc, *.exe,	
UNICODE (har bir ramzga 2 bayt)	*.bmp, *.jpg,	
*.txt, *.log,	*.wav, *.mp3,	
*.htm, *.html	*.avi, *.mpg	

4.5.3-rasm. Dasturlashda oddiy fayllar turlari

«matnli fayl»
o'zgaruvchi tipi:
var f: text;

assign(f, 'qq.txt');

reset(f); {o'qish uchun}

rewrite(f); {yozish uchun}

read (f, n); {n qiymatini kiritish}

write (f, n); {n qiymatini yozish}

writeln (f, n); {yangi qatarga o'tish bilan yozish}

close(f);

4.5.4-rasm. Faylar bilan ishlash operatsiyalari ketma-ketligi.

Xususiyatlari:

- Fayl nomi faqat assign buyrug'ida keltiriladi, faylga murojaat fayl o'zgaruvchisi orqali amalga oshiriladi;
- O'qish uchun ochilayotgan fayl mavjud bo'lishi kerak;
- Yozish uchun ochilayotgan fayl xotirada mavjud bo'lsa uning eskisi o'chiriladi va ustidan yoziladi;
- Ma'lumotlar faylga matnli ko'rinishda yoziladi;
- Dastur ishi tugaganda barcha fayllar avtomatik yopiladi;
- Faylni yopgandan so'ng f o'zgaruvchini boshqa fayl bilan ishlash uchun yana qaytib ishlatish mumkin;

assign (f, 'qq.txt');
reset (f);

Fayl oxiri
(and of file, EOF)

12 5 45 67 56•

read (f, x);

12 5 45 67 56•

4.5.5-rasm. Fayllar bilan ishlaganda ma'lumotlarni o'qishning ko'rgazmali tamoyili.

Tiplarga keltirish

Ixtiyoriy dasturda bir tipdagi ma'lumotlarni boshqa tipga o'tkazish zaruriyati tug'iladi. Tiplarni o'tkazish 2 ko'rinishga ega:

- ☐ Mos bo'lmagan tiplarni o'tkazish;
- ☐ Mos tiplarni o'tkazish;

Mos bo'lmagan tiplarni o'tkazishga masalan matnni songa o'tkazishni

keltirish mumkin. Masalan: "12345" ni songa.

Bir tipdan boshqa tipga o'tkazish funksiyalari:

- ☐ StrtoInt(matn)-matndan butun songa;
- ☐ InttoStr(butun_son)-butun sondan matnga;
- ☐ StrtoDef(matn,xatolik yuz bergandagi qiymat)-matndan

songa aylantirish;

- ☐ DatetoStr(sana)-sanani matnga aylantiradi;
- ☐ StrtoDate(matn)-matnni sanaga aylantirish;
- ☐ TimetoStr(vaqt)-vaqtni matnga aylantirish;

Bir tipdan boshqa tipga o'tkazish funksiyalari:

- ☐ StrtoTime(matn)-matnni vaqtga aylantirish;
- ☐ round-kasr sonni butunga aylantirish;
- ☐ FloattoStr(kasr_son)-kasr sonni matnga aylantirish;
- ☐ StrtoFloat(matn)-matnni kasr songa aylantirish.

☐ FormatDateTime-sana va vaqtni ma'lum bir formatga keltirish. Bu funktsiyani to'liqroq ko'rsak. Funktsiyaning 2 ta parametri mavjud:

- ♦ Sana va vaqt keltirish kerak bo'lgan format matni;
- ♦ O'tkazilishi kerak bo'lgan TDateTime o'zgaruvchi.

FormatDateTime funktsiyasini birinchi parametri juda qiziq.

Quyida ularning formati- keltirilgan:

- ☐ d-sana kunini nolsiz ko'rsatish(1,2,..30,31);
- ☐ dd-sana kunini nol bilan ko'rsatish(01,02,..30,31);
- ☐ ddd-hafta kunini qisqa formada ko'rsatish(Пн,Вт,Ср,...);

☐ dddd-hafta kunini to'liq formada

ko'rsatish(Понедельник,Вторник,Среда,...);

- ☐ m-oyni nolsiz ko'rsatish(1,2,...11,12);
- ☐ mm-oyni nol bilan ko'rsatish(01,02,...11,12);
- ☐ mmm-oyni qisqa formada ko'rsatish(Ян,Фе,Ма,...);
- ☐ mmmm-oyni to'liq formada ko'rsatish (Январь, Февраль, Март,...);
- ☐ yy-yilni 2 xonali sonda ko'rsatish(98,99,00,01,...);
- ☐ yyyy-yilni to'liq formada ko'rsatish (2016);
- ☐ h-nol qo'shmasdan soatni ko'rstish;
- ☐ hh-soatni nol bilan ko'rsatish;
- ☐ n-daqiqani qisqa formada ko'rsatish;
- ☐ nn-daqiqani to'liq formada ko'rsatish;
- ☐ s-soniyani qisqa formada ko'rsatish;
- ☐ ss-soniyani to'liq formada ko'rsatish;
- ☐ z-millisekundlarni qisqa formada ko'rsatish;
- ☐ zz-millisekundlarni to'liq formada ko'rsatish;
- ☐ am/pm-12 soatlik vaqt ifodasi.

Tizim Reestri bilan ishlash

Ba'zan dasturlarimizda tizim Reestri bilan ishlashga majbur bo'lamiz. Bu uchun biz TRegIniFile ob'yektidan foydalanishimiz mumkin. Bu ob'yekt bilan ishlash uchun **uses** bo'limiga Registry modulini qo'shish kerak. TRegIniFile tipidagi o'zgaruvchi ishlatishimiz zarur.

Bu o'zgaruvchini initsializatsiyalash uchun TRegIniFile ob'yektining **Create** metodidan foydalanish mumkin. Masalan:

RegIni:=TRegIniFile.Create('Software');

Jimlik bo'yicha siz Reestrning **HKEY_CURRENT_USER** bo'limiga murojaat etasiz. **Create** metodining bir parametri mavjud qaysiki u **HKEY_CURRENT_USER** ichki bo'limini ochadi.



4.5.6-rasm. Windows tizim oynasi.

Reestrning boshqa ichki bo'limini ochish uchun OpenKey metodidan foydalanish mumkin. Masalan:

RegIni.OpenKey('Microsoft');

OpenKey metodining 2 ta parametri mavjud:

☐ Ochish kerak bo'lgan ichki bo'lim nomi;

☐ Agar bunday ichki bo'lim mavjud bo'lmasa uni yaratish kerakmi(true,false);

Endi Reestrda o'qish va yozish bilan shug'ullansak. O'qish uchun quyidagi metodlar mavjud:

☐ ReadBool-mantiqiy qiymatni o'qish;

☐ ReadInteger-butun son qiymatini o'qish;

☐ ReadString-matnni o'qish.

Bu metodlar bir-biriga o'xshash va faqat 3-parametrda farq bor. ReadString metodini ko'rib o'tsak hammasi tushunarli bo'ladi.

ReadString metodining 3 ta parametri mavjud:

☐ Ichki bo'lim nomi (masalan siz HKEY_CURRENT_USER\Software\Microsoft bo'limida turibsiz va uni ichidan ichki MySoftware bo'limidan ma'lumot o'qimoqchi bo'lsangiz parametr sifatida MySoftware ni ko'rsatishiz zarur);

- ☐ Parametr nomi;
- ☐ Agar shunday parametr bo'lmasa jimlik bo'yicha qiymat.

Masalan:

```
Str:=RegIni.ReadString('MySoftware','Path','c:\');
```

Xuddi shunday yozish amalga oshiriladi. Yozish uchun quyidagi metodlar ishlatiladi:

- ☐ WriteBool-mantiqiy qiymatni yozish;
- ☐ WriteInteger-butun qiymatni yozish;
- ☐ WriteString-matn qiymatni yozish;

Masalan:

```
RegIni.WriteString('MySoftware','Path','c:\Windows');
```

Reestr bilan ish tugagandan so'ng Free buyrug'i yordamida uni yopish mumkin. Masalan:

```
RegIni.Free;
```

Reestr bilan ishlaganda yana quyidagi metodlar yordamga kelishi mumkin:

- ☐ DeleteValue-reestrda qiymatni o'chirish;
- ☐ RootKey-asosiy bo'limni ko'rsatadigan xususiyat.

DeleteValue metodiga misol: `RegIni.DeleteValue('Width');`

```
var
```

```
RegIni:TRegIniFile;
```

```
begin
```

```
RegIni:=TRegIniFile.Create('Software');
```

```
RegIni.RootKey:=HKEY_LOCAL_MACHINE;
```

```
RegIni.OpenKey('VR-online', true);
```

```
RegIni.WriteString('Razd', 'Param', Edit1.Text);
```

```
RegIni.Free;
```

```
end;
```

4.5.7-rasm. Windows tizim bilan ishlash kodiga misol.

Asosiy tayanch so'z va iboralar

Massivlar, fayllar bilan ishlash, assign, tiplarga keltirish, tizim reestri, registry moduli, TregIni tipi.

Takrorlash va mustaqil ishlash uchun savollar

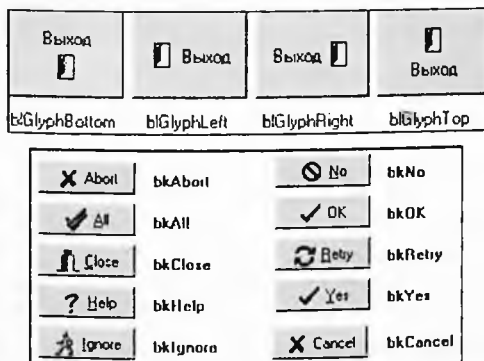
1. Massivlar bilan ishlash haqida nima bilasiz?
2. Ko'p o'lchamli massivlar qaysi holatlarda ishlatiladi?
3. Fayllar bilan ishlash haqida nima bilasiz?
4. Tiplarga keltirish nima uchun kerak?
5. Tizim Reestri bilan ishlash nima uchun kerak?

§4.6. Qo'shimcha komponentlar tahlili

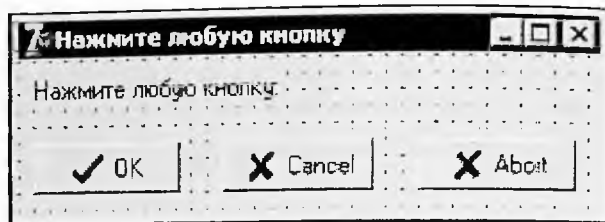
Nostandart dasturlarni yaratish jarayonida ba'zan nostandart ob'ektlar zarur bo'ladi. Bunda bizga qo'shimcha komponentlar to'plamidan foydalanishimiz mumkin. Ushbu komponentlar Delphi tizimining Additional yoki Win32 komponentlar bo'limida joylashgan. Ularni ko'rib o'tsak:

□ Qo'shimcha tugmalar (TSpeedButton va TBitBtn)-matndan boshqa

tasvirni ham oladi, bir-biridan farqi TBitBtn klaviaturdan fokusni olib biladi TSpeedButton esa yo'q;



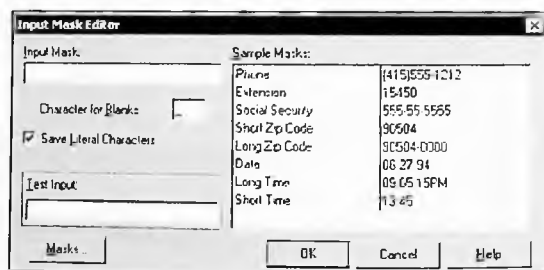
4.6.1-rasm. TBitBtn tugmasidan foydalanish misoli.



4.6.2-rasm. TBitBtn tugmasidan foydalanish misoli.

□ Kiritishni niqoblash (TMaskEdit) - foydalanuvchi ma'lum bir formatdagi ma'lumotlarni kiritishi uchun ishlatiladi. Uning asosiy xususiyatlari:

❖ EditMask-niqoblash shabloni(namunasi);



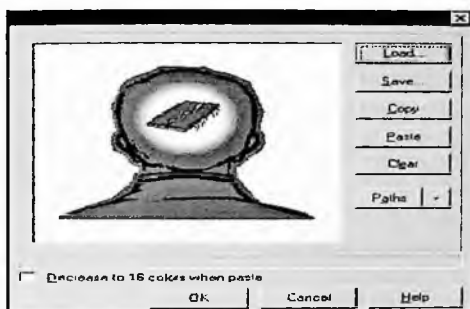
4.6.3-rasm. TMaskEdit komponentining EditMask xususiyati oynasi.

Jadvallar (TStringGrid, TDrawGrid)-ba'zan dasturlarda jadval ko'rinishida ma'lumotlarni kiritish yoki aks etish zarur, masalan Excelga o'xshab. Uning asosiy xususiyatlari:

- ❖ ColumnCount-ustunlar soni;
- ❖ FixedCols-biriktirilgan ustunlar soni;
- ❖ FixedRows-biriktirilgan qatorlar soni;
- ❖ RowCount-qatorlar soni.

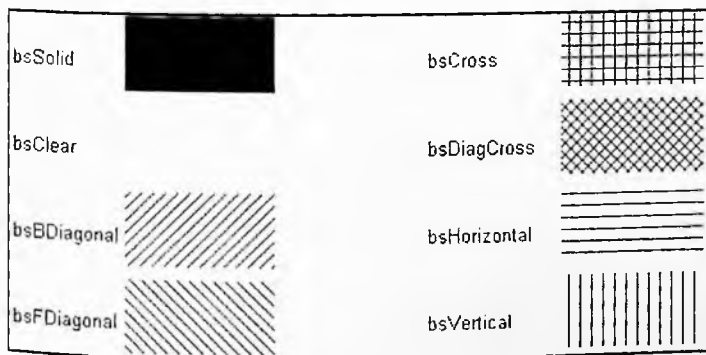
□ TImage, TShape, TBevel komponentlari-dasturda grafik imkoniyatlardan foydalanmoqchi bo'lsangiz ushbu komponentlardan foydalanishingiz mumkin.

- ❖ Timage.Picture-tasvirni yuklash xususiyati;
- ❖ Timage.AutoSize-tasvir hajmini oladi;
- ❖ Timage.Stretch-komponent hajmi bo'yicha kengayadi;

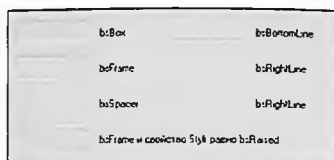
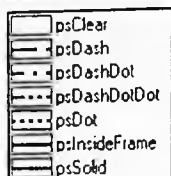


4.6.4-rasm. TImage komponentining Picture xususiyati oynasi.

- ❖ TShape.Shape-figura turini belgilaydi;
- ❖ TShape.Brush-bo'yash stili;
- ❖ TShape.Pen-qalam stili;
- ❖ TBevel.Shape-figura turi;
- ❖ TBevel.Style-aks etish stili;



4.6.5-rasm. TShape komponenti bo'yash stili ro'yxati.



4.6.6-rasm. TBevel komponenti xususiyatlari qiymatlari ro'yxati.

□ TCheckBox-ListBox komponenti singari faqat har bir qatorda belgi(razochka) qo'yish yoki olib tashlash mumkin. Uning xususiyatlari:

- ❖ Items-ro'yxatni shakllantirish;
- ❖ Columns-ustunlar soni;



4.6.7-rasm. TCheckBox komponenti xususiyatlari qiymatlari ro'yxati.

□ Ajratish chizig'i (TSplitter) – bir oynani qismlarga ajratish, bunda oyna qismlarini siljitish orqali hajmini o'zgartirish mumkin.



4.6.8-rasm. TSplitter komponenti ishlatish misoli.

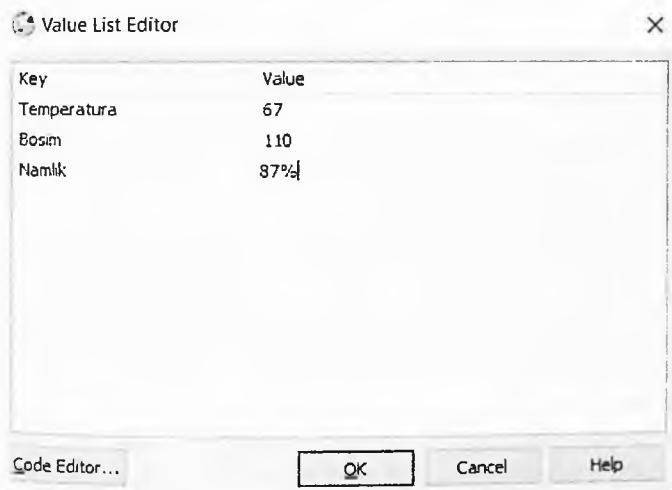
☐ **Ko'p qatorli yozuv (TStaticText)** – Label komponentiday faqat bir nechta qatorda yozuvni aks etish mumkin. Uning xususiyatlari:

- ❖ **AutoSize**-avtomatik hajmini belgilash;

☐ **Parametrlar muharriri (TValueListEditor)** – Parametrlar qiymatlarini kiritish komponenti. Uning xususiyatlari:

- ❖ **FixedCols**-biriktirilgan ustun indeksi;

- ❖ **Strings**-xususiyatlar nomi(ro'yxati);



4.6.9-rasm. TValueEditor komponenti qiymatlarni sozlash oynasi.

☐ **Bo'limlar to'plami (TTabControl)** – ba`zan formada bir nechta bo'limni ko'rsatishga to'g'ri kelib qoladi. Uning xususiyatlari:

- MultiLine-bir nechta qatorli bo'limlar to'plami;

- HotTrack-sichqoncha ko'rsatkichi bo'lim ustiga etilganda ajralib ko'rsatiladi;

- Style-bo'limlarni aks etish stili (ts Tabs, ts Buttons, ts FlatButton);

4.6.10-rasm. TTabControl komponentini ishlatish misoli.

- ❖ TabHeight-bo'limlar joylashuvi balandligi;
- ❖ TabIndex-joriy bo'lim raqami;
- ❖ TabPosition-bo'limlar nomlanishining joylashuvi (tpBottom, tpLeft, tpRight, tpTop);
- ❖ Tabs-bo'limlar ro'yxati.

□ Saxifalar to'plami (TPageControl) – baʼzan formada bir nechta saxifani ko'rsatishga to'g'ri kelib qoladi. Unga sichqonchani o'ng tugmasini tanlab quyidagi operatsiyalardan foydalanish mumkin:

- ❖ NewPage-yangi saxifa yaratish;
- ❖ NextPage-keyingi saxifaga o'tish;
- ❖ PreviousPage-oldingi saxifaga o'tish;
- ❖ DeletePage-joriy saxifani o'chirish;

Uning xususiyatlari:

- ❖ ActivePage-faol saxifa;
- ❖ ActivePageIndex-faol saxifa raqami;
- ❖ PageCount-saxifalar soni;
- ❖ MultiSelect-bir nechta saxifani tanlash mumkin;

4.6.11-rasm. TPageControl komponentini ishlatish misoli.

□ Tasvirlar to'plami (TImageList) – juda ko'p tasvirlar yig'ish komponenti.

□ Qiymat siljitgich (TTrackBar) – qiymat siljitgichlar foydalanuvchilar ma'lum qiymatlar diapazonidan tanlash kerak bo'lganda ishlatiladi. Uning xususiyatlari:

- ❖ Max-maksimal qiymati;
- ❖ Min-minimal qiymat;
- ❖ Orientation-siljitgich yo'nalishi;
- ❖ Position-pozitsiyasi;
- ❖ SelStart-tanlashni boshlash qiymati;
- ❖ SelEnd-tanlashni tugash qiymati;
- ❖ SliderVisible-siljitgich ko'rinishi yoki ko'rinmasligi;
- ❖ TrickMarks-ko'rsatkich joylashuvi.

Gorizonta! TrackBar



Vertikal TrackBar



Gorizonta! TrackBar



4.6.12-rasm. TTrackBar komponentini ishlatish misoli.

□ Jarayon indeksatsiyasi (TProgressBar) – jarayonni borishini aks etish komponenti(maslan fayllarni nusxalash). Uning xususiyatlari:

- ❖ Max-maksimal qiymati;
- ❖ Min-minimal qiymat;
- ❖ Position-pozitsiyasi;



4.6.13-rasm. TProgressBar komponentini ishlatish misoli.

□ Oddiy animatsiya (TAnimate) – ba`zan dasturlarda fydalanuvchi zerikib qolmasligi uchun kichik animatsiyalarni aks etish kerak (maslan fayllarni nusxalash). Uning xususiyatlari:

- ❖ FileName-AVI faylni tanlash;
- ❖ CommonAVI-odatiy animatsiyalar ro'yxati (aviCopyFile, aviCopyFiles, aviDeleteFile,...);
- ❖ Active-ijro etish.



4.6.14-rasm. TAnimate komponentini ishlatish misoli.

❑ **Sanani tanlash (TDateTimePicker)** – baʼzan dasturlarda foydalanuvchi tomonida aniq sanani tanlash kerak boʻladi (maslan tugʻilgan sana). Uning xususiyatlari:

- ❖ Date-tanlangan sana;
- ❖ DateFormat-sana formati(dfShort,dfLong);
- ❖ MaxDate-maksimal sana;
- ❖ MinDate-minimal sana.



4.6.15-rasm. TDateTimePicker komponentini ishlatish misoli.

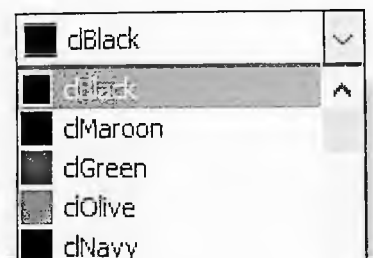
❑ **Taqvim (TMonthCalendar)** – baʼzan dasturlarda taqvimni koʻrsatishga toʻgʻri keladi. Uning xususiyatlari:

- ❖ FirstDayOfWeek-birinchi kun sifatida hafta kuni;
- ❖ MultiSelect-sanalar diapazonini tanlash mumkin;
- ❖ ShowToday-joriy sanani koʻrsatadi;
- ❖ ShowTodayCircle-aylana tugmachaga tanlanganda joriy sanaga oʻtadi;
- ❖ WeekNumber-hafta raqamini koʻrsatadi.

❑ **Elementlar daraxti (TTreeView)** – baʼzan dasturlarda maʼlumotlarni daraxt koʻrinishida ifodalash zarur boʻlganda ishlatiladi;

- ❑ **Yozuvli kiritish maydoni (TLabelEdit)** – bu komponent o'zida ham Label ham Edit komponentlarini birlashtirgan;
- ❑ **Ranglar palitrasi (TColorBox)** – baʼzan dasturlarda ranglarni tanlash va tanlangan rangga ob'ektni bo'yash zarur. Uning xususiyatlari:

- ❖ DefaultColorColor-standart tanlangan rang;
- ❖ Selected-tanlangan rang;
- ❖ ColorNames-ro'yxatdagi ranglar nomi;
- ❖ Colors-ranglar ro'yxati.



4.6.16-rasm. TColorBox komponentini ishlatish misoli.

- ❑ **Tizim burchagi ikonkasi (TSystemTray)** – baʼzan dasturimizni tizim burchagiga joylashtirishni xoxlaymiz(maslan soat oldiga). Uning xususiyatlari:

- ❖ Animate-tizim burchagida animatsiya hosil bo'lishi yoki yo'q, bu animatsiya Icons xususiyatidagi tasvirlarni namoyish etadi;

- ❖ AnimateInterval-ikonkalar ko'rsatilishi orasidagi vaqt masofasi;

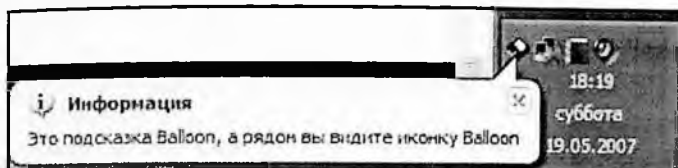
- ❖ BallonFlags-ikonkaning shipshitish stili;

- ❖ BallonHint-ikonka shipshitish yozuvi;

- ❖ BallonTimeOut-shipshitish matnini ko'rsatish vaqt intervali;

- ❖ BallonTitle-shipshitish sarlavhasi;

- ❖ Visible-ikonkani avtomatik tizim burchagiga chiqarish.



4.6.17-rasm. TSystemTray komponentini ishlatish misoli.

Asosiy tayanch so'z va iboralar

Additional, Win32, TSeepdButton, TBitBtn, TMaskEdit, TStringGrid, TImage, TCheckListBox, TSplitter, TPageControl, TTrackBar, TProgressBar, TDateTimePicker.

Takrorlash va mustaqil ishlash uchun savollar

1. Qo'shimcha komponentlar ro'yxati va vazifalari haqida yozing.
2. Qo'shimcha komponentlar xususiyatlari haqida yozing.
3. Eng ko'p foydalaniladigan qo'shimcha imkoniyatli komponentlar ro'yxatini keltirsangiz.
4. Qo'shimcha komponentalar asosan qanday holatlarda ishlatiladi?
5. Loyiha interfeysida nostandart elementlarni qo'llashda nimadan foydalaniladi?

IV-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI

AMALIY MASHG'ULOT №9

Delphi tizimi haqida ma'lumot. Delphi visual modeli.

Ishdan maqsad: Delphi dasturlash tizimi asoslari bilan tanishish. Loyiha yaratish, saqlash, ochish va o'zgartirish malaka va ko'nikmalariga ega bo'lish.

Nazariy ma'lumotlar

Delphi dasturi- dasturlash tillari orasida visual ko'rinishga ega bo'lgan dasturlash tilidir. Bu esa dasturlash jarayonini yengillashtirish uchun xizmat qiladi. Delphi dasturi oynasi quyidagilardan tashkil topgan:



4.7.1-rasm. Delphi dasturlash tizimi bosh oynasi.

1. *Dasturning asosiy oynasi;*
2. *Ob'ektlar inspektori:* Komponent bilan ishlash va boshqarish uchun 2 ta bo'limdan tashkil topgan:

Properties — Xususiyatlar. Tanlangan komponentlarga xos xususiyatlar mavjud.

Events —Hodisalar. Tanlangan komponentning keltirilgan holatlari uchun dastur kodini yozish mumkin.

3.*Forma;*

4.*Komponentlar politrasi;*

5.*Komponentlar daraxti;*

6. *Loyiha menejeri.*

Delphi muhiti bosh oynasining sarlavha satri standart ko'rinishiga ega bo'lib, unda ilovaning nomi, unda yaratilayotgan loyiha nomi va oynani yashirish, o'lchovini o'zgartirish va uni yopish uchun xizmat qiladigan uchta tugma joylashgan.

2. Delphining asosiy menyusi quyidagi bo'limdan iborat bo'lib, ular quyidagilardir.

 Delphi 10.2

File Edit Search View Refactor Project Run Component Tools Window Help

4.7.2-rasm. Delphi tizimining asosiy menyusi.

File menyusi buyruqlari:

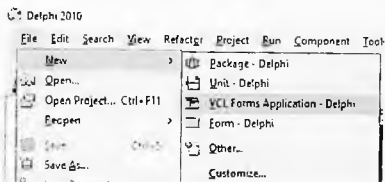
	New-yangi loyiha, forma va Unit yaratish; Open/Open Project- Yaratilgan loyiha faylini ochish; Save, Save as, Save all- Yaratilgan loyihani xotirada saqlash; Close/Close all- loyiha formasini yopish/loyihaning barcha fayllarini yopish; Print- chop etish; Exit-dasturdan chiqish.
	<i>Edit menyusi buyruqlari:</i> Undo- bajarilgan amalni bekor qilib, bir qadam orqaga qaytish; Redo- bajarilgan amalni bekor qilish holatini bekor qilish; Cut/Copy/Paste/Delete- Belgilangan qismni qirqib olish/Nusxalash/Joylashtrish/O'chirish; Select all- barcha komponentlarni va berilganlarni belgilash.
	<i>Run menyusi buyruqlari:</i>

Button1. Caption := TimeToStr(Time); end;

Sichqoncha bilan button tugmasi bosilganda uning sarlavhasida joriy vaqt akslanadi.

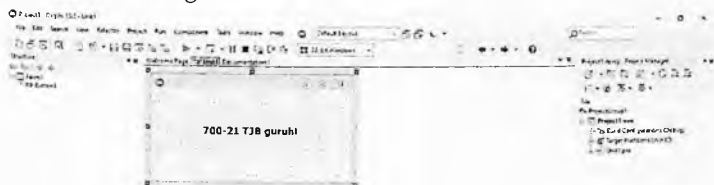
Ishni bajarish tartibi:

1. Delphi tizimini ishga tushiring va yangi loyiha yarating.



4.7.3-rasm. Delphi tizimida yangi loyiha yaratish.

2. Dasturning oynasidan Tool Palette -> Standart -> Button komponentini joylashtiring va uning sarlavhasini guruhning nomi bilan almashtiring.



4.7.4-rasm. Loyiha formasida komponentlarni joylashtirish.

3. Uni kompyuter xotirasida saqlang. *File menyusi -> Save all.*

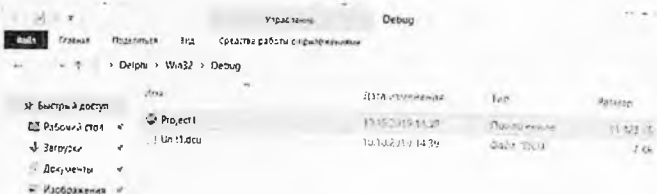
4. Dasturni ishga tushuring yoki kompilatsiya qiling. *Run menyusi -> Run without debugging* buyrug'i.

5. Loyihani yoping yoki Delphi dasturlash tizimini yoping. *File menyusi -> Exit* buyrug'i.

6. Loyihani o'zgartirish uchun oching va Button komponentasi sarlavhasini do'stingiz o'qiydigan boshqa guruh nomiga almashtiring. *File menyusi -> Open Project* buyrug'i, *Button komponentini tanlab -> Object Inspectors -> Caption* bo'limlari.

7.Dasturni ishga tushuring va kompilatsiya qiling. *Run menyusi-> Run without debugging buyrug'i.*

8.Hosil bo'lgan .exe kengaytmali faylni o'qituvchiga ko'rsating.



4.7.5-rasm. Loyiha natijasi sifatida exe faylini hosil bo'lishi.

Bajarish uchun topshiriqlar:

1. Formaga standart komponentlar palitrasidan 2 ta Button komponentini yuklang. Komponentlar sarlavhasini fakultet va yo'nalish nomi qisqartmasiga o'zgartiring. 1-tugma bosilganda uning sarlavhasi 2-tugma sarlavhasiga o'zgartirilsin.
- 2.Formaga standart komponentlar palitrasidan 1 ta Button komponentini yuklang. Komponentlar sarlavhasini familiyangiz o'zgartiring. Tugma bosilganda uning sarlavhasi guruh nomiga o'zgartirilsin.
- 3.Formaga standart komponentlar palitrasidan 1 ta Button komponentini yuklang. Komponentlar sarlavhasini sharifingizga o'zgartiring. Tugma bosilganda uning sarlavhasi ismingizga nomiga o'zgartirilsin.
- 4.Formaga standart komponentlar palitrasidan 2 ta Button komponentini yuklang. Komponentlar sarlavhasini OK va Cancel ga o'zgartiring. Button 1 ning onclick hodisasi uchun dastur yozing. Bunda 2-tugma bosilganda uning sarlavhasi 1-tugma sarlavhasiga o'zgarsin.
- 5.Formaga standart komponentlar palitrasidan 2 ta Button komponentini yuklang. Komponentlar sarlavhasini OK va No ga o'zgartiring. Button 1 ning onclick hodisasi uchun dastur yozing. Bunda 1-tugma bosilganda uning sarlavhasi Yes ga o'zgarsin.

AMALIY MASHG'ULOT №10

Standart komponentlardan foydalanish.

Ishdan maqsad: Delphi muhitida Standart komponentlar palitrasining komponentlari yordamida dasturlar yaratish.

Nazariy ma'lumotlar

Standart komponentlar palitrasida quyidagi komponentlar mavjud:

- 1) TLabel – yozuv; komponent xususiyatlari quyidagilar:
 - Color-komponent rangini aniqlaydi;
 - Fon- komponent matni xususiyatlarini aniqlaydi;
 - Visible-komponentning ko'rinish qiymatini aniqlaydi;
 - Caption-komponent sarlavhasini aniqlaydi;
 - Alignment-komponentning joylashuvini (tekislanishi) aniqlaydi;
- 2) TEdit – qator (matn) kiritish maydoni;
 - Name- komponentning nomini aniqlaydi;
 - Alignment-komponentning joylashuvini (tekislanishi) aniqlaydi;
- 3) TText – qatorli matn kiritish maydoni;
 - Text-komponentda kiritilgan matnni aniqlaydi;
 - Color-komponent rangini aniqlaydi;
 - Fon- komponent matni xususiyatlarini aniqlaydi.
- 4) TMemo – ko'p qatorli matn kiritish maydoni;
 - Lines-ko'p satrli matn kiritish bo'limi;
 - Name- komponentning nomini aniqlaydi.
- 5) TCheckBox – belgilash elementi;
 - Caption-belgilash elementi qiymati sarlavhasini aniqlaydi;
 - Checked-komponent tanlanganining mantiqiy qiymatini aniqlaydi;
- 6) TRadioButton – tanlash elementi;

- Caption-belgilash elementi qiymati sarlavhasini aniqlaydi;
- Checked-komponent tanlanganining mantiqiy qiymatini aniqlaydi;

- Name- komponentning nomini aniqlaydi.

7)TListBox – ro'yxat;

- Name- komponentning nomini aniqlaydi;

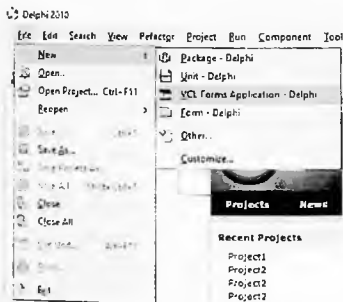
- Items- ro'yxat elementlarini kiritish maydoni.

Amaliy mashg'ulot bo'yicha topshiriq:

Edit komponentiga kiritilgan sonlar ustida 4 ta arifmetik amal sarlavhasiga mos tugmalarda bajarilishi dasturi tuzilsin.

Ishni bajarish tartibi:

1. Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating.



4.8.1-rasm. Delphi tizimida yangi loyiha yaratish.

2.Quyidagi rasmda ko'rsatilganidek forma interfeysini tayyorlang(4.8.2- rasm):

Buning uchun formaga 4 ta Label komponentini yuklang,
 Label1->Caption->1-son=, Label2->Caption->2-son=, Label3->Caption->Natija, Label4->Caption-> bo'sh qoldiring.

3.2 ta Edit komponentini yuklang, komponentning Text xususiyatini tozalab, bo'sh qoldiring.

4. 4ta Button komponentlarini yuklang. Komponentlarning Caption xususiyatini 4.8.2-rasmda ko'rsatilgan tartibda o'zgartiring.

Form1

1-son=

2-son=

Qo'shish Ayirish

Ko'paytirish Bo'lish

Natija

4.8.2-rasm. Amaliy mashg'ulot ilovasi interfeysi.

Dasturda "Qo'shish", "Ayirish", "Ko'paytirish" va "Bo'lish" tugmalariga sichqoncha bilan murojaat etganda Natija sarlavhali Label komponentiga kiritilgan sonlarni amallari javobi chiqsin!

Buning uchun : **Qo'shish** sarlavhali Button komponentiga quyidagi kodni kiriting:

Label4.caption:=Inttostr(strtoint(Edit1.text)+ strtoint(Edit2.text));

Ayirish sarlavhali Button komponentiga quyidagi kodni kiriting:

Label4.caption:=Inttostr(strtoint(Edit1.text)- strtoint(Edit2.text));

Ko'paytirish sarlavhali Button komponentiga quyidagi kodni kiriting:

Label4.caption:=Inttostr(strtoint(Edit1.text) strtoint(Edit2.text));*

Bo'lish sarlavhali Button komponentiga quyidagi kodni kiriting:

Label4.caption:=Inttostr(strtoint(Edit1.text)/strtoint(Edit2.text));

Bajarish uchun topshiriqlar

1. Delphi dasturida if shart operatoridan foydalanib, Bo'lish sarlavhali tugma uchun bo'luvchi 0 ga teng bo'lganda Natija bo'limiga "Sonni 0 ga bo'lish mumkin emas" matnli xabar chiqarish dasturini tuzing.

2. Delphi dasturida For operatoridan foydalanib Edit1 ga kiritilgan songacha bo'lgan sonlar yig'indisini hisoblab natijani chiqarish dasturini tuzing.

3. Delphi dasturida For operatoridan foydalanib Edit2 ga kiritilgan sonning faktorialini hisoblab natijani chiqarish dasturini tuzing.

4. Delphi dasturida if shart operatoridan foydalanib Ayirish sarlavhali tugma uchun agar Edit1 dagi son Edit2 dagi sondan kata bo'lsa Edit1-Edit2 amalini, aks holda Edit2-Edit1 ni hisoblab natijani chiqarish dasturini tuzing.

5. Delphi dasturida Edit1 ga kiritilgan sonning kvadratini Edit2 ga kiritilgan sonning kubini hisoblab natijalar yig'indisini Label4 ga chiqarish dasturini tuzing.

AMALIY MASHG'ULOT №11

Ob'yektga yo'naltirilgan dasturlash asoslari usullarini qo'llash.

Ishdan maqsad: Delphi muhitida standart komponentlar palitrasi va ob'yektga yo'naltirilgan dasturlashning asosiy usullarini qo'llagan holda dastur yaratish.

Nazariy ma'lumotlar

Delphida massivlar bilan ishlash

Massivlar bilan ishlash – dasturlashning eng muhim va eng kerakli qirralaridan biri.

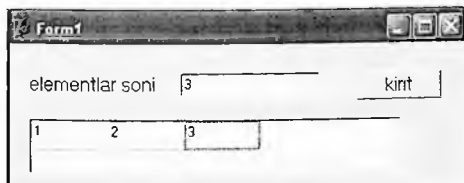
Delphida massiv array kalitli so'zi bilan ifodalanadi. Massivlarni e'lon qilish: Var

Massiv_nomi : array [massiv uzunligi] of ma'lumotlar tipi

Misol: Var

R:array[0..11] of integer;

Delphida massivlar bilan ishlash uchun qo'llaniladigan komponent bu StringGrid komponentidir. Bu component Additional sahifasida joylashgan bo'lib, uning qator va ustunlarini Rowcount va Colcount xususiyatlaridan kiritish mumkin.



4.9.1-rasm. StringGrid komponentini formada ishlatish.

```
Procedure TForm1.Button1Click(Sender: TObject);  
var      N:integer;      begin      N:=StrToInt(Edit1.Text);  
StringGrid1.ColCount:=N; end;
```

Delphida fayllar bilan ishlash

Fayllar

Fayl – bu nomga ega disk hududi.

Fayllar

Matnli	Binar	Papkalar (kataloglar)
Faqat matn, boshqarish ramzlariga ega emas (32 kichik kodlar)	Ramzlar kodlari jadvalining ixtiyoriy ramzlarini saqlaydi	
ASCII (har bir ramzga 1 bayt)	*.doc, *.exe,	
UNICODE (har bir ramzga 2 bayt)	*.bmp, *.jpg,	
*.txt, *.log,	*.wav, *.mp3,	
*.htm, *.html	*.avi, *.mpg	

4.9.2-rasm. Dasturlashda oddiy fayllar turlari.

«matnli fayl»
o'zgaruvchi tipi:
var f: text;

```
assign(f, 'qq.txt');
```

```
reset(f); {o'qish uchun}
```

```
rewrite(f); {yozish uchun}
```

```
read ( f, n ); {n qiymatini kiritish}
```

```
write ( f, n ); {n qiymatini yozish}
```

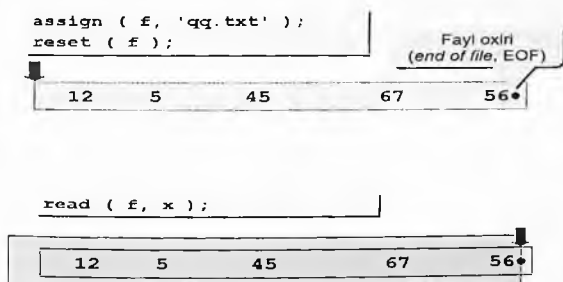
```
writeln ( f, n ); {yangi qatarga o'tish bilan yozish}
```

```
close(f);
```

4.9.3-rasm. Faylar bilan ishlash operatsiyalari ketma-ketligi.

Xususiyatlari:

- Fayl nomi faqat **assign** buyrug'ida keltiriladi, faylga murojaat fayl o'zgaruvchisi orqali amalga oshiriladi;
- O'qish uchun ochilayotgan fayl mavjud bo'lishi kerak;
- Yozish uchun ochilayotgan fayl xotirada mavjud bo'lsa uning eskisi o'chiriladi va ustidan yoziladi;
- Ma'lumotlar faylga matnli ko'rinishda yoziladi;
- Dastur ishi tugaganda barcha fayllar avtomatik yopiladi;
- Faylni yopgandan so'ng f o'zgaruvchini boshqa fayl bilan ishlash uchun yana qaytib ishlatish mumkin;



4.9.4-rasm. Fayllar bilan ishlaganda ma'lumotlarni o'qishning ko'rgazmali tamoyili.

Amaliy mashg'ulot bo'yicha topshiriq:

Delphida 10 ta tasodifiy sonni generatsiyalab, uni faylga yozish dasturini tuzing.

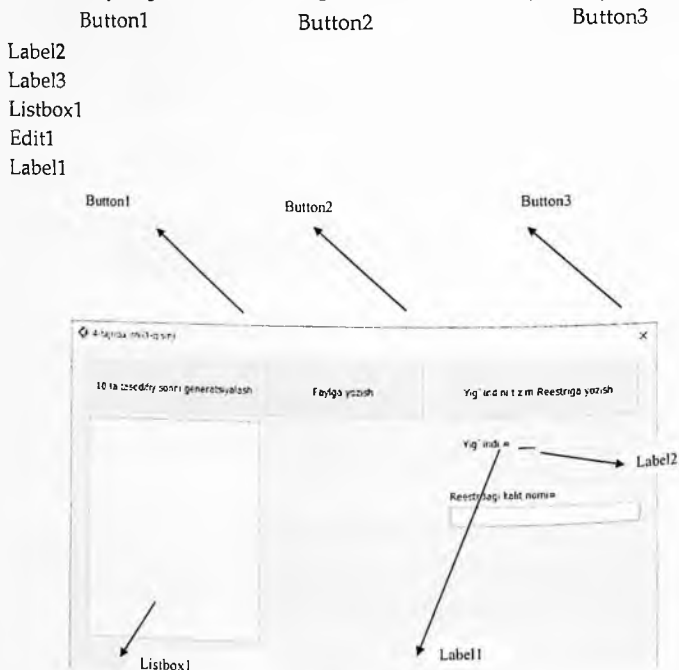
Ishni bajarish tartibi:

1. Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating va darhol alohida papkaga loyihani saqlang!



4.9.5-rasm. Delphi tizimida yangi loyiha yaratish.

2.Quyidagi rasmda ko'rsatilganidek forma interfeysini tayyorlang:



4.9.6-rasm. Delphi tizimida amaliy mashg'ulot oynasi.

3. Dasturda "Button1" komponenti "Click" hodisasiga quyidagi kodni kiriting, ya'ni Button1 hodisasi quyidagi ko'rinishda bo'lsin: procedure TForm1.Button1Click(Sender: TObject); var i:integer; begin

ListBox1.Clear; for i:=1 to 10 do
ListBox1.Items.Add(IntToStr(Random(100))); end;

4. Dasturni ishlatib ko'ring va "Listbox1" komponentida (0-100) qiymatlari oralig'ida 10 ta tasodifiy qiymat olinishiga amin bo'ling!

5. Loyihaga qayting. Dasturda "Button2" komponenti "Click(Sichqoncha bilan murojaat)" hodisasiga quyidagi kodni kiriting, ya'ni Button2 hodisasi quyidagi ko'rinishda bo'lsin:

procedure TForm1.Button2Click(Sender: TObject);

var F:TextFile; i:integer; begin

AssignFile(F,'tjbakt.txt'); Rewrite(F);

for i:=0 to 9 do begin Write(F,StrToInt(ListBox1.Items[i]));

Write(F,'); end; CloseFile(F); end;

6. Dasturni ishlatib ko'ring va loyiha papkangizdagi "exe" fayl joylashgan papkaga "tjbakt.txt" fayli hosil bo'lgan hamda uning ichiga 10 ta tasodifiy sonlar yozilganligiga amin bo'ling!

7. Loyihaga qayting. F12 tugmasi orqali modulga(dastur kodi yoziladigan oynaga) o'ting. Modulni eng boshiga o'tib, uses bo'limiga o'ting va "Winapi.Windows, Winapi.Messages, System.SysUtils" so'zlaridan keyin oxirida "," verguldan keyin "Registry" so'zini qo'shing.


```

Unit Unit1;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Registry;

type
  TForm1 = class(TForm)
    ListBox1: TListBox;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

4.9.7-rasm. Delphi tizimida amaliy mashg'ulot dastur kodi.

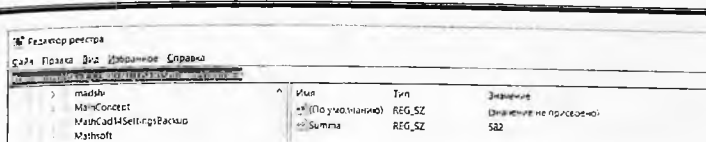
8. Dasturda "Button3" komponenti "Click(Sichqoncha bilan murojaat)" hodisasiga quyidagi kodni kiriting, ya'ni Button3 hodisasi quyidagi ko'rinishda bo'lsin:

```

procedure TForm1.Button3Click(Sender: TObject); var
i,sum:integer; RegIni:TRegIniFile; RegKey:DWORD; begin sum:=0;
for i:=0 to 9 do sum:=sum+StrToInt(ListBox1.Items[i]);
Label2.Caption:=IntToStr(sum);
RegIni:=TRegIniFile.Create('Software');
RegIni.WriteInteger('Tajriba-4', Edit1.Text,sum); RegIni.Free; end;

```

9. Dasturni ishlatib ko'ring va tizim reestrda "Компьютер\HKEY_CURRENT_USER\Software\Tajriba-4" manzilida "Edit1" komponentida kiritilgan matnli nomli parametr hamda uning qiymati yig'indi qiymatiga teng bo'lganligiga amin bo'ling! Tizim reestrini ishga tushurish uchun "Pusk" menyusiga (Win+R issiq klavishi) "regedit" deb yozing va "OK" tugmasini tanlang!



4.9.8-rasm. Windows OT da tizim reestri oynasi.

Bajarish uchun topshiriqlar

1.Dastur ishga tushganda va ushbu oynada "Button1" tugmasini tanlaganda loyiha papkasidagi "exe" fayl joylashgan papkadagi "tjbakt.txt" fayli ichida saqlangan sonlar "Listbox1" komponentiga chiqarilsin!

2.Dastur ishga tushganda va ushbu oynada "Button2" tugmasini tanlaganda "Listbox1" komponentiga chiqarilsin sonlar yig'indisi "Label2" komponentiga chiqarilsin!

3.Dastur ishga tushganda va ushbu oynada "Button3" tugmasini tanlaganda Tizim reestrda yizilgan qiymat va "Label2" komponentiga qiymat bir xil qiymat bo'lsa, "Checkbox1" komponenti belgilangan ☒ holatda bo'lsin, aks holda belgilanmagan ☐ holatda bo'lsin!

TAJRIBA MASHG'ULOTI №9

Komponentlar bilan ishlash

Ishdan maqsad: Delphi muhitida Standart komponentlar palitrasida mavjud bo'lgan komponentlar bilan tanishish, ularning xususiyatlaridan foydalanib Interfeys yaratish. Standart komponentlardan foydalanib dastur tuzish.

Nazariy ma'lumotlar

Delphi tizimida yaratilgan loyihaning nimadan iborat ekanligini tushunish uchun Delphi tizimida oddiy dasturni yaratib o'rganish maqsadga muvofiq. Loyihani kompilatsiyalash uchun Ctrl+F9 tugmasini yoki dastur yaratish uchun esa F9 tugmasini tanlash kifoya.

Loyiha kompilatsiyalangandan so'ng, loyihani saqlagan papkada *.exe kengaytmali fayl hosil bo'ladi. Uni bimalol ishlatishimiz mumkin. Delphi tizimida hamma narsa ob'yekt sifatida qaralishini inobatga olsak, loyihamiz oynasi ham ob'yekt demak uni sarlavhasini bimalol "Ob'yektlar inspketori(Object Inspector)" ning xususiyatlar bo'limining "Caption"(Sarlavha) o'zgartirishimiz mumkin(Masalan: guruhingiz nomi).

Izohlar – dastur kodiga hech qanday ta'sir o'tkazmaydigan ixtiyoriy matn. U hech qachon kompilatsiya qilinmaydi va faqat kodni izohlash uchun ishlatiladi. Izohlar 2 usulda ishlatiladi:

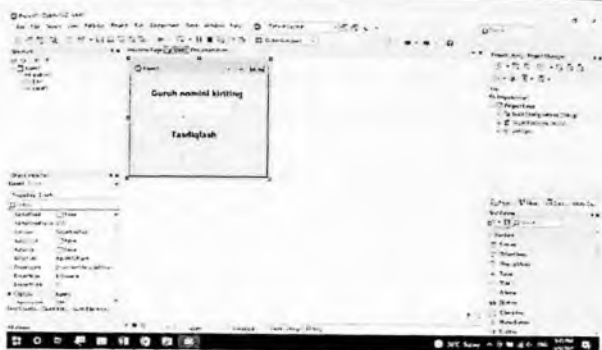
□2 sleshdan (//) keyin keladigan barcha matn izoh sifatida qaraladi(bunday usulda bir qatorlik izohlar ishlatiladi);

□Figurali qavslar { } ichidagi barcha matn izoh hisoblanadi (bunday usulda bir nechta qatordan iborat izohlar ishlatiladi).

Protsedura va funksiya – kod bloki bo'lib, alohida blok sifatida ajratilgan. *Procedure Exam1(); Var i:Integer; //Mahalliy o'zgaruvchini e'lon qilish Begin i:=10; //Mahalliy o'zgaruvchiga qiymat o'zlashtirish End;*

Ishni bajarish tartibi:

- 1.Delphi tizimini ishga tushiring.
- 2.Yangi loyiha yarating.
- 3.Dasturning oynasida Button, Edit, Label komponentlarini joylashtiring va uning sarlavhasini quyidagicha o'zgartiring.



4.10.1-rasm. Tajriba mashg'ulot ilovasi interfeysi

4.Loyihani kompyuter xotirasida saqlang.

5.Button komponentiga sichqonchani chap tugmasini ikki marotaba tanlang va hosil bo'lgan click hodisasi protsedurasida begin kalitli so'zadan oldin var kalitli so'zini kiritib, quydagi koddagidek o'zgaruvchilarni e'lon qiling:

i:integer; r:real; ch:char; s:string; v:variant;

6.Protseduraning begin va end kalitli so'zlari orasida e'lon qilingan o'zgaruvchilarga qiymatlarni quyidagi koddagidek beramiz: i:=10; //bu o'zgaruvchi faqat butun sonlarni qabul qiladi r:=3.14; //bu o'zgaruvchi butun va kasrli sonlarni qabul qiladi ch:='Z'; //bu o'zgaruvchi faqat 1 ta belgini qabul qiladi s:='Salom talaba'; //bu o'zgaruvchi qatorli matnni qabul qiladi v:=4; //bu o'zgaruvchi turli tiplarni qabul qiladi.

7. Berilayotgan qiymatni formada Edit komponentiga akslantirish uchun quyidagi dastur kodini kiriting: Edit1.text:= i:=10, r:=3.14, ch:=Z, s:=Salom talaba, v:=4';

8.Dasturni ishga tushuring yoki kompilatsiya qiling.

Bajarish uchun topshiriqlar:

1.Delphi tizimida loyiha yarating va 1 ta 0...4294967295 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta

2,9*10-39...1,7*1038 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta belgili, 1 ta matnli va 1 ta turli tipdagi(bu o'zgaruvchiga butun son o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

2.Delphi tizimida loyiha yarating va 1 ta -128...+127 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta 1,5*10-45...3,4*1038 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta belgili, 1 ta matnli va 1 ta turli tipdagi(bu o'zgaruvchiga butun bo'lmagan son o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

3.Delphi tizimida loyiha yarating va 1 ta -32768...+32767 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta 5*10-324...1,7*10308 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta belgili, 1 ta matnli va 1 ta turli tipdagi(bu o'zgaruvchiga simvol o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

4.Delphi tizimida loyiha yarating va 1 ta -2^63...+2^63-1 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta 3.6*10-4951...1,1*104932 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta belgili, 1 ta matnli va 1 ta turli tipdagi(bu o'zgaruvchiga matn o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

5.Delphi tizimida loyiha yarating va 1 ta 0...+255 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta 9223372203685477.5808 922337203685477.5807 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 1 ta belgili, 1 ta matnli va 1 ta turli tipdagi(bu o'zgaruvchiga butun son o'zlashtiring) o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

6. Delphi tizimida loyiha yarating va 2 ta 0...4294967295 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta 2,9*10-39...1,7*1038 oraliqdagi qiymatlarni qabul qiluvchi butun

o'zgaruvchi, 2 ta belgili, 2 ta matnli va 2 ta turli tipdagi(bu o'zgaruvchilarga biriga butun son ikkinchisiga kasrli son o'zlashtiring) o'zgaruvchilarni e'lon qiling va ularga qiymatlarni o'zlashtiring.

7.Delphi tizimida loyiha yarating va 2 ta -128...+127 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta 1,5*10-45...3,4*1038 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta belgili, 2 ta matnli va 2 ta turli tipdagi(bu o'zgaruvchilarga biriga kasrli son ikkinchisiga belgi o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

8.Delphi tizimida loyiha yarating va 2 ta -32768...+32767 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta 5*10-324...1,7*10308 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta belgili, 2 ta matnli va 2 ta turli tipdagi(bu o'zgaruvchilarga biriga butun son ikkinchisiga belgi son o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

9.Delphi tizimida loyiha yarating va 2 ta -2⁶³...+2⁶³-1 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta 3.6*10-4951...1,1*104932 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta belgili, 2 ta matnli va 2 ta turli tipdagi(bu o'zgaruvchilarga biriga butun son ikkinchisiga matn o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

10.Delphi tizimida loyiha yarating va 2 ta 0...+255 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta -9223372203685477.5808 922337203685477.5807 oraliqdagi qiymatlarni qabul qiluvchi butun o'zgaruvchi, 2 ta belgili, 2 ta matnli va 2 ta turli tipdagi(bu o'zgaruvchilarga biriga kasr son ikkinchisiga matn o'zlashtiring) tiplarni qabul qiluvchi o'zgaruvchi e'lon qiling va ularga qiymatlarni o'zlashtiring.

TAJRIBA MASHG'ULOTI №10

Ishchi ilovalarni yaratish

Ishdan maqsad: Delphi muhitida Standart komponentlar palitrasidagi komponentlardan foydalanib ishchi ilovalar yaratish, komponent xususiyatlarini o'zgartirishni o'rganish, komponentning holatlari uchun dastur yozishni o'rganish.

Nazariy ma'lumotlar

Dasturda asosiy menyuni yaratish uchun TMainMenu komponentasidan foydalaniladi. Uning quyidagi xususiyatlari mavjud:

☐ AutoHotKeys-tezkor klavishlarni avtomatik yaratishni belgilaydi yoki avtomatik yaratmaslikni belgilaydi;

☐ Images-menyu punktlarida rasmlarni joylashtirish ro'yxati;

☐ Items-menyuni ichki punktlarini yaratish xususiyati;

☐ AutoMerge-ichki oynalar menyulari bilan avtomatik birlashtirish xususiyati;

Yangi formani yaratish uchun **File** menyusidan **New** bo'limi tarkibidan **Form** punktini tanlash zarur. Shunda Delphi bizga bo'sh forma yaratib beradi. Ikkinchi formaning moduli Unit2.pas bo'ladi, 3-formaning moduli esa Unit3.pas va hokazo. Boshqa oynaning chaqirish uchun formaning nomidan so'ng **show** yoki **showmodal** metodidan foydalaniladi.

Ishni bajarish tartibi:

1.Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating.

2.Hosil bo'lgan oynaga quyidagi parametrlarni o'rnatish:

a.Oyna yuklanganda avtomatik kata hajmda bo'lsin (Object Inspector > Properties > WindowState > wsMaximized)

b.Shaffoflik darajasi 180 qiymatida bo'lsin (Object Inspector > Properties > AlphaBlend:True, AlphaBlendValue:180) c.Oynada

faqatgina yopish tugmasi bo'lsin (Object Inspector > Properties > BorderIcons > biMinimize:False, biMaximize:False)

d.Oynada 4.11.1-rasmda ko'rsatilgandek menyu hosil qiling (Tool Palette > Standart > TMainMenu)



4.11.1- rasm. Tajriba mashg'uloti ilovasi birinchi oyna interfeysi

Izoh: Loyihada boshqa oynalarni(formalarni) faollashtirish va dastur yuklanganda ularni avtomatik birinchi paydo bo'lishi uchun quyidagi menyudan foydalaning:Project >> Options ... >> Forms >> Main Form.



4.11.2-rasm. Loyihada asosiy formani belgilash oynasi

4.11.3,4.11.4,4.11.5,4.11.6-rasmlarda ko'rsatilgan menyularning har birida ichki menyular hosil qiling.



4.11.5-rasm. Tajriba mashg'uloti ilovasi 4- menyusi bo'limi



4.11.6-rasm. Tajriba mashg'uloti ilovasi 5- menyusi bo'limi

Ilova 2-oynasida Edit, PopupMenu, Label, CheckBox va RadioButton komponentlarini yuklang va berilgan oyna ko'rinishini hosil qiling.

Oynaga talablar:

1)Oyna yuklanganda avtomatik ekranni o'rtasida paydo bo'lsin. -FISH kiritish komponenti:

1)Maksimal 90 ta belgini kiritish imkoniyati bo'lsin.

2) Barcha harflarni katta harflar bilan yozish imkoniyati bo'lsin. -O'zim haqimda komponenti:

1) Yordamchi menyu bo'lsin.



4.11.7-rasm. Tajriba mashg'uloti ilovasi 2- oynasi.

2) Yordamchi menyu buyruqlari bajarilsin.

Bajarish uchun topshiriqlar:

1. MainMenu komponentidan foydalanib loyiha uchun asosiy menyu yarating va menyuning yordamchi buyruqlarini ham kiriting.

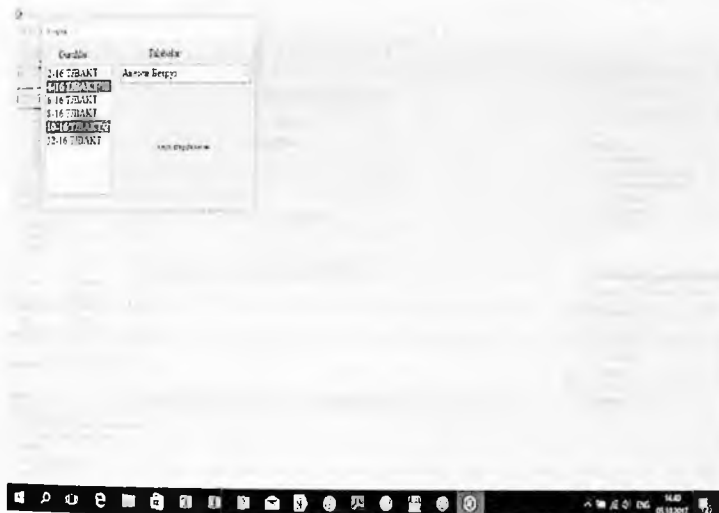
2. Loyiha uchun quyidagi oyna ko'rinishlarini hosil qiling. Oynaga talablar:

1) Oyna yuklanganda avtomatik ekranni o'rtasida paydo bo'lsin.

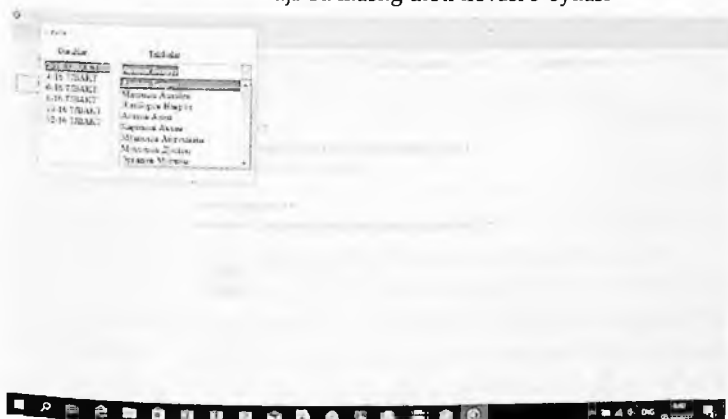
2) Oynada tizim tugmalari bo'lmasin.

-Guruhlar bo'limida:

Guruhlar ro'yxatidan bir nechta guruhni tanlash imkoniyati bo'lsin.



4.11.8-rasm. Tajriba mashg'uloti ilovasi 3-oynasi



4.11.9-rasm. Tajriba mashg'uloti ilovasi 4-oynasi

TAJIRIBA MASHG'ULOTI №11

Qo'shimcha komponentlar tahlilini olib borish

Ishdan maqsad: Delphi tizimida mavjud qo'shimcha komponentlardan foydalanilgan holda, texnik ilovalarni yaratishda foydalanuvchiga qulay dasturiy ta'minot interfeysini yaratishni o'rganish.

Nazariy ma'lumotlar

Qo'shimcha komponentlar to'plamidan foydalanish uchun Delphi tizimining Additional yoki Win32 komponentlar bo'limida joylashgan.

Qo'shimcha tugmalar(TSpeedButton va TBitBtn)-matndan boshqa tasvirni ham oladi, bir-biridan farqi TBitBtn klaviaturdan fokusni olib biladi TSpeedButton esa yo'q.

Kiritishni niqoblash (TMaskEdit)-foydalanuvchi ma'lum bir formatdagi ma'lumotlarni kiritishi uchun ishlatiladi. Uning asosiy xususiyatlari:

EditMask-niqoblash shabloni(namunasi);

Jadvallar (TStringGrid, TDrawGrid)-ba'zan dasturlarda jadval ko'rinishi ma'lumotlarni kiritish yoki aks etish zarur, masalan Excelga o'xshab. Uning asosiy xususiyatlari:

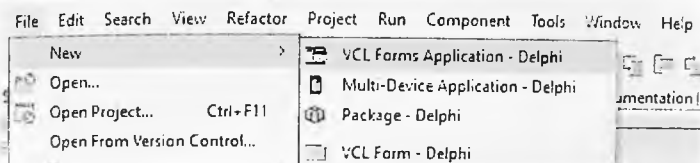
- ❖ ColumntCount-ustunlar soni;
- ❖ FixedCols-biriktirilgan ustunlar soni;
- ❖ FixedRows-biriktirilgan qatorlar soni;
- ❖ RowCount-qatorlar soni.

Ajratish chizig'i (TSplitter) – bir oynani qismlarga ajratish, bunda oyna qismlarini siljitish orqali hajmini o'zgartirish mumkin.

Ishni bajarish tartibi

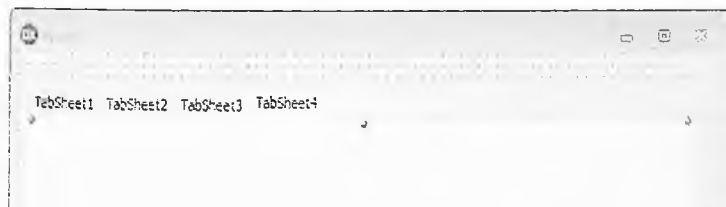
Qo'shimcha komponentlar yordamida quyidagi bo'limlardan iborat interfeysni yarating:

1. Delphi dasturini ishga tushiring.



4.12.1-rasm. Delphi tizimida yangi loyiha yaratish

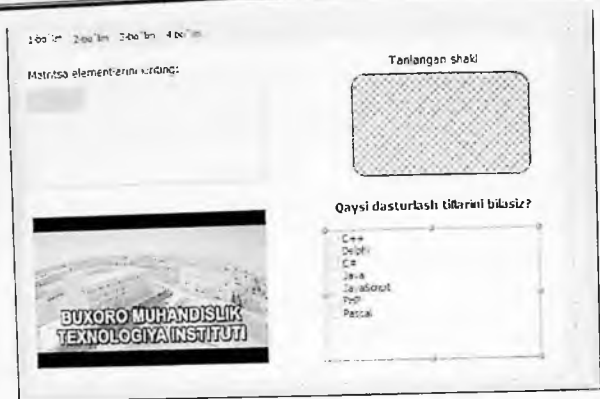
2. Win32 komponentlar palitrasidan PageControl komponentini formaga joylashtiring va butun ekran bo'yicha yoying. PageControl komponentini tanlang, o'ng tugmani bosing va yordamchi menyudan New Page bo'limini tanlang. Xuddi shunday amallarni bajarib 4 ta sahifa yarating.



4.12.2-rasm. PageControl komponentasidan foydalanib bir oyna tarkibida bir nechta saxifalar yaratish

3. Komponentni tanlab xususiyatlar oynasi, caption xususiyatidan TabSheet1 yozuvini 1-bo'lim deb o'zgartiring. Qolgan bo'limlar ham xuddi shunday o'zgartiriladi.

4. 1-bo'lim oynasiga StringGrid komponentasini yuklang (4 ta qator va 4 ta ustun). Shape komponentini yuklang. Komponentning shape xususiyatidan ixtiyoriy shaklni tanlang (stRoundRect), Brush xususiyati - Style bo'limidan shaklni bo'yash usulini tanlang, Color bo'limidan rangni tanlang.

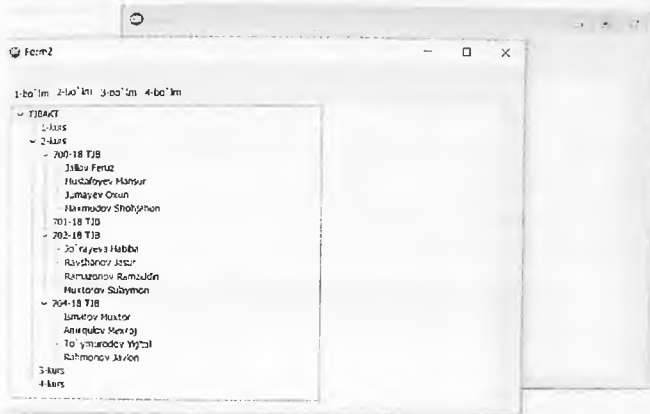


4.12.3-rasm. PageControl komponentasi 1-bo'limi ko'rinishi

5. TImage komponentini yuklang, rasm uchun kerakli maydon bo'yicha komponentni yoying. Stretch xususiyatini True qiymatiga o'zgartiring. Picture xususiyatidan rasmni manzilini ko'rsating va rasmni yuklang.

6. TCheckBox komponentini yuklang. Items xususiyatiga tanlanadigan ro'yxat elementlarini kiriting.

7. 2-bo'limni oching va TreeView komponentini yuklang. Items xususiyatini tanlang va hosil bo'lgan oynadan New Item tugmasini tanlang va Text bo'limiga TJBAKT yozuvini kiriting. TJBAKT uchun bo'limlarni kiritmoqchi bo'lsak New SubItem bo'limini tanlaymiz va Text bo'limiga 1-kurs yozuvini kiritamiz. Ro'yxatni davom ettirish uchun New Item bo'limini tanlaymiz va nomini yozamiz. 2-kurs uchun ichki ro'yxat chiqarmoqchi bo'lsak agar 2-kursni tanlaymiz va New SubItem bo'limi tanlab nomini kiritamiz, ushbu ro'yxatni davom ettirish uchun esa New Item bo'limini tanlaymiz. Agar biror bo'limda xatolik bo'lgan bo'lsa, Delete tugmasi orqali o'chirish mumkin. Quyidagi dastur ko'rinishini hosil qiling.



4.12.4-rasm. PageControl komponentasi 2- bo'limi ko'rinishi

8.3-bo'limni oching va Label, CheckListBox, Memo va Button komponentlarini yuklang.

1-bo'lim 2-bo'lim 3-bo'lim 4-bo'lim

Qaysi dasturlash tillarini o'rganishni xoxlaysiz?



4.12.5-rasm. PageControl komponentasi 3- bo'limi ko'rinishi

Quyidagi dastur ko'rinishini hosil qiling. Button tugmasi click hodisasi uchun quyidagi dastur matnini yozing. `var I : integer; begin Memo1.Clear; for I := 0 to CheckListBox1.Items.Count -1 do begin if CheckListBox1.Checked[i] then`

`Memo1.Lines.Add(CheckListBox1.Items[i]); end;`

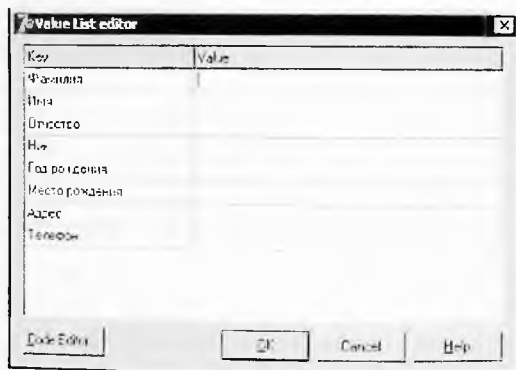
`Memo1.Lines.Add('Dasturlash tillari bo'yicha bilimingizni oshirishga harakat qilamiz'); end;`

Bajarish uchun topshiriqlar:

1. Delphi dasturlash tizimida yuqorida berilgan oynalarni hosil qiling.

2. 4-bo'lim uchun TValueListEditor (Parametrlar qiymatlarini kiritish komponenti) komponentidan foydalanib quyidagi oyna ko'rinishini hosil qiling.

3. Dasturni ishga tushiring va bo'limlarga qiymatlarni kiritishni amalga oshiring.



4.12.6-rasm. TValueListEditor komponentasi ko'rinishi.

IV-MODUL BO'YICHA TEST SAVOLLARI

1. Ob'yektga yo'naltirilgan dasturlashda sinf tushunchasi?

- A. Xususiyatlar, metodlar va hodisalardan iborat
- B. Xususiyatlar va metodlardan iborat
- C. Xususiyatlar va hodisalardan iborat
- D. Metodlar va hodisalardan iborat

2. Yozuv ob'yektining(TLabel) ekranga chiqariladigan matn rangi ob'yektga yo'naltirilgan dasturlash tilidagi ob'yektning qanday tushunchasiga mos keladi?

- A. Xususiyat
- B. Hodisa
- C. Metod
- D. Initsializatsiya

3. Matn kiritish ob'yektida(TEdit) matnni o'zgarishi ob'yektga yo'naltirilgan dasturlash tilidagi ob'yektning qanday tushunchasiga mos keladi?

- A. Hodisa
- B. Xususiyat
- C. Metod
- D. Initsializatsiya

4. BitBtn tugma ob'yekti Button tugma ob'yektidan barcha imkoniyatlarini o'zida mujassamlashtirgan! Ushbu fakt ob'yektga yo'naltirilgan dasturlash tilidagi ob'yektning qanday tushunchasiga mos keladi?

- A. Vorislik
- B. Polimorfizm.
- C. Inkapsulatsiya
- D. Initsializatsiya

5. BitBtn tugma ob'yekti Button tugma ob'yektidan tugma ustiga yozuvni chiqarishdan boshqa rasmni ham chiqaradi! Ushbu fakt ob'yektga yo'naltirilgan dasturlash tilidagi ob'yektning qanday tushunchasiga mos keladi?

- A. Polimorfizm
- B. Vorislik
- C. Inkapsulatsiya
- D. Initsializatsiya

6. object Form3: TForm3 Left = 293 Top = 55 BorderStyle = bsDialog Caption = '#1050#1080#1088#1080#1090#1080#1096'...' ClientHeight = 551 ClientWidth = 797 Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'Tahoma' Font.Style = [] OldCreateOrder = False Position = poDesigned PixelsPerInch = 96 TextHeight = 13 Ushbu kod qismi qaysi kengatmali fayldan olingan?

- A. .dfm
- B. .exe
- C. .pas
- D. .res

7. Form1.close; //Ushbu kod qismi qaysi kengatmali fayldan olingan?

- A. .pas
- B. .exe
- C. .dfm
- D. .res

8. Delphi tizimida Forma ob'yektining rangini o'zgartirmoqchi bo'lsangiz Delphi tizimining qanday oynasidan foydalaniladi?

- A. Object Inspector
- B. Project manager
- C. Main menu
- D. Tool palette

9. Delphi tizimida Forma ob'yektiga sichqoncha ko'rsatkichini etganda dastur kodini yozmoqchi bo'lsangiz Delphi tizimining qanday oynasidan foydalaniladi?

- A. Object Inspector
- B. Project manager
- C. Main menu
- D. Tool palette

10. Qaysi qatorda Delphi dasturlash tilida to'g'ri izoh yozilgan?

- A. {close}
- B. /close
- C. [close
- D. close]

11. $a := 4 * 3 + 5$; "a" o'zgaruvchi tipini to'g'ri tanlang?

- A. Integer

- B. Char
- C. String
- D. Boolean

12. $a = 4 \frac{3}{5}$; "a" o'zgaruvchi tipini to'g'ri tanlang?

- A. Real
- B. Char
- C. String
- D. Boolean

13. $a = 'Z'$; "a" o'zgaruvchi tipini to'g'ri tanlang?

- A. Char
- B. Real
- C. String
- D. Boolean

14. $a = 'Zayniddin'$; "a" o'zgaruvchi tipini to'g'ri tanlang?

- A. String
- B. Real
- C. Char
- D. Boolean

15. Agar o'zgaruvchi oldindan qanday tipli ma'lumotlarni qabul qilishinni bilmasangiz qanday tipni tanlagan bo'lardiz?

- A. Variant
- B. Real
- C. String
- D. Boolean

16. Tsikl operatori dasturlashda keng qo'llaniladi! Agar tsikllar soni aniq bo'lsa, (masalsan 10 marta qaytib bajarish) unda qaysi tsikl operatoridan foydalaniladi?

- A. For
- B. While
- C. Repeat
- D. IF

17. Tsikl operatori dasturlashda keng qo'llaniladi! Agar tsikllar soni aniq bo'lmasa va shart birinchi tekshirilib buyruqlar keyin bajarilsa unda qaysi tsikl operatoridan foydalaniladi?

- A. While
- B. For
- C. Repeat

D. IF

18. Tsikl operatori dasturlashda keng qo'llaniladi! Agar tsikllar soni aniq bo'lmasa va buyruqlar avval bajarilib va shart keyin tekshirilsa unda qaysi tsikl operatoridan foydalaniladi?

A. Repeat

B. For

C. While

D. IF

19. Shart operatori dasturlashda keng qo'llaniladi! Agar bir son ikkinchi sondan kattaligini tekshirish uchun (maslan $a > b$) qanday operatoridan foydalaniladi?

A. IF

B. For

C. While

D. Repeat

20. $a:=3$; Case a of 0:Form1.Tag:=1; 2:Form1.Tag:=2; 3:Form1.Tag:=3;

4:Form1.Tag:=4; End; Ushbu operatoridan keyin Form1 ob'yekting Tag xususiyati nechiga teng bo'ladi?

A. 3

B. 1

C. 2

D. 4

21. $A:=\text{Length}(\text{'Ob'yektga yo'naltirilgan dasturlash'})$; A o'zgaruvchi qiymatini toping?

A. 35

B. 33

C. 34

D. 36

22. $A:=\text{copy}(\text{'Ob'yektga yo'naltirilgan dasturlash'},26,6)$; A o'zgaruvchi qiymatini toping?

A. Dastur

B. Ob'ekt

C. Yo'naltir

D. Dasturlash

23. $s:='Ob'yektga yo'naltirilgan dasturlash'; Delete(s,32,4)$; "S" o'zgaruvchi qiymatini toping?

- A. Ob`yektga yo`naltirilgan dastur
- B. Yo`naltirilgan dastur
- C. Dastur
- D. Dasturlash

24. `a:=pos('dastur','Ob`yektga yo`naltirilgan dasturlash',);` "a" o'zgaruvchi qiymatini toping?

- A. 26
- B. 24
- C. 25
- D. 27

25. `A:array[4..17] of integer;` Bu nechi o'lchamli massiv?

- A. 1
- B. 2
- C. 3
- D. 4

26. `A:array[4..17,1..8] of integer;` Bu nechi o'lchamli massiv?

- A. 2
- B. 1
- C. 3
- D. 4

27. "26.12.2016,Понедельник" deb chiqarish uchun `FormatDateTime` funksiyasiga qanday formatni ko'rsatish zarur?

- A. 'dd.mm.yyyy,dddd'
- B. 'dd.mm.yyyy,dddddd'
- C. 'dd.mmm.yyyy,dddd'
- D. 'dd.mm.yy,dddd'

28. "4:34:10 pm" deb chiqarish uchun `FormatDateTime` funksiyasiga qanday formatni ko'rsatish zarur?

- A. 'h:nn:ss am/pm'
- B. 'hh:nn:ss'
- C. 'hh:nn:ss am'
- D. 'h:nn:ss pm'

29. "434" matnni butun songa aylantirmoqchiman!

Qanday funksiyadan foydalanish kerak?

- A. `StrToInt`
- B. `IntToStr`
- C. `StrToDate`

- D. DatetoStr
30. 434 butun sonini mantnga aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. InttoStr
B. StrtoInt
C. StrtoDate
D. DatetoStr
31. '26.12.2016' sanani matnga aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. DatetoStr
B. StrtoInt
C. InttoStr
D. StrtoDate
32. '26.12.2016' matnni sanaga aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. StrtoDate
B. StrtoInt
C. InttoStr
D. DatetoStr
33. '16:46' vaqtni matnga aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. TimetoStr
B. StrtoTime
C. StrtoDate
D. DatetoStr
34. '16:46' matnni vaqtga aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. StrtoTime
B. TimetoStr
C. StrtoDate
D. DatetoStr
35. '3,14' matnni songa aylantirmoqchiman! Qanday funktsiyadan foydalanish kerak?
A. FloattoStr
B. InttoStr
C. StrtoInt
D. StrtoFloat

36. $a = \text{round}(16/5)$; "a" o'zgaruvchi nechiga teng bo'ladi?

- A. 3
- B. 4
- C. 5
- D. 6

37. $a = \text{round}(3/4)$; "a" o'zgaruvchi nechaga teng bo'ladi?

- A. 1
- B. 2
- C. 3
- D. 4

38. Tizim Reestri bilan ishlash uchun qanday ob'ektdan foydalanish mumkin?

- A. TRegIniFile
- B. TButton
- C. TFileStream
- D. TRegistry

39. Tizim Reestri bilan ishlash uchun qanday muduldan foydalanish mumkin?

- A. Registry
- B. WinAPI
- C. System
- D. Windows

40. Nostandart komponentlar asosan qanday komponentlar guruhiga kiritilgan?

- A. Addinional va Win32
- B. Standart va Win32
- C. Addinional va Standart
- D. Addinional va Win33

41. Telefon raqamni kiritish maydonini tashkil etish uchun quyidagi qanday komponentdan foydalanish mumkin?

- A. TMaskEdit
- B. TBirbtn
- C. TStringGrid
- D. TDrawGrid

42. Taqvimdagi sanani tanlashni osonlashtirish uchun qanday komponentdan foydalanish mumkin?

- A. TDateTimePicker

B. TCheckBox

C. TMaskEdit

D. TStringGrid

43. Ranglarni tanlashni tashkil etish uchun qanday komponentdan foydalanish mumkin?

A. TColorBox

B. TCheckBox

C. TMaskEdit

D. TDateTimePicker

44. Daraxt ko'rinishida ma'lumotlarni shakllantirish uchun qanday komponentdan foydalanish mumkin?

A. TTreeView

B. TCheckBox

C. TColorBox

D. TDateTimePicker

45. Dasturimizni tizim burchagiga paydo qilishimiz uchun qanday komponentdan foydalanish mumkin?

A. TSystemTray

B. TTreeView

C. TColorBox

D. TDateTimePicker

V-MODUL. OBEKTGA YO'NALTIRILGAN DASTURLASHDA KENGAYTIRILGAN TEXNOLOGIYALARIDAN FOYDALANISH.

§ 5.1. Obyektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan ishlash

Ma'lumotlar bazasi bilan ishlash Delphi tizimining asosiy yutuqlaridan biri hisoblanadi. Bu tizim ma'lumotlar bazasi bilan ishlash uchun ishlab chiqilmagan bo'lsa hamki, lekin ma'lumotlar bazasi bilan ishlashni osonlashtirishi odamni hayratga soladi. Delphi ma'lumotlar bazasi bilan ishlashdagi barcha qiyinchiliklarni yopish bilan birga, ma'lumotlar bazasi bilan ishlash uchun keng imkoniyatlarni yaratadi. Bu mavzuda biz Delphi tizimida ma'lumotlar bazasi bilan ishlashni ko'rib o'tamiz.

Misollar ko'rish uchun Access ma'lumotlar bazasini boshqarish tizimini va zamonaviy ma'lumotlar bazasi faylli formatini ishlatamiz. Ikkinchisi albatta ma'lumotlar bazasi emas lekin bugungi kunda ma'lumotlar almashinishi uchun zamonaviy standartga aylangan. Mavzuda bundan tashqari DBF va Paradox formatidagi keng tarqalgan jadvallar bilan ishlashni ko'rib o'tamiz. Ular keng tarqalganligi sababli ular bilan ishlashni ko'rib o'tish zarar keltirmaydi. Hattoki eng mashhur "1C" tizimi ham DBF tipidagi formatni ishlatadi. Shunday qilib, agar siz boshqa dastur ma'lumotlar bazasi bilan ishlaydigan dastur yaratmoqchi bo'lsangiz, unda majbur ushbu formatlarni bilishingiz zarur.

Relatsion ma'lumotlar bazasi nazariyasi

Ma'lumotlar bazasi (MB) mahalliy (mijozning kompyuterida o'rnatilgan, dastur ham shu kompyuterda ishlaydi) va uzoqlashgan (serverda o'rnatilgan - uzoqlashgan kompyuterda) turlariga bo'linadi. Ikkinchi tipdagi MB da mijoz-server yoki tarmoqli MB nomlanishlar ishlatiladi, chunki ma'lumotlarni qayta

ishlash mijoz kompyuterida joylashgan bo'lsa, ma'lumotlar esa serverda joylashgan bo'ladi.

Mijoz-server MB da ma'lumotlar server kompyuterda saqlanadi. Mijoz dasturlar ma'lumotlarga maxsus so'rov tili orqali murojaat etishadi(SQL- yoki uning kengaytmasi – Transact-SQL yoki PL/SQL). Bunda shuni inobatga olish kerakki, dastur faqat kerakli ma'lumotlarga ega bo'ladi va serverga faqat o'zgartirilgan ma'lumotlarni uzatadi.

Tarmoqli MB da ma'lumotlar serverda joylashgan bo'ladi, lekin har bir mijoz ma'lumotlarning nusxasini oladi. Bu holatda 2 ta kamchilik mavjud:

- Tarmoqqa katta yuklanish;

- Butunlikni ta'minlashning murakkab jarayoni.

MB bilan ishlashni tushunishimiz uhun ba'zi tushunchalarni aniqlashtirishimiz kerak. Ma'lumotlar bazasi jadvali – ikki o'lchamli massiv bo'lib, ustunlar va qatorlardan iborat. Ma'lumotlar bazasi – bu oddiy fayl bo'lib, unda bir yoki ko'p jadvallarni saqlash mumkin. Ko'pgina ma'lumotlar bazasi faqat bir jadvalni saqlashi mumkin(Paradox, dBase, XML), lekin bir faylda bir nechta jadvalni saqlash mumkin bo'lgan ma'lumotlar bazasi ham mavjud(masalan MS Access).

Bir necha xil ma'lumotlar bazasi mavjud bo'lib, lekin biz sizlar bilan relatsion ma'lumotlar bazasini ko'rib o'tamiz. Chunki bu tipdagi ma'lumotlar bazasi keng tarqalgan. Relatsion ma'lumotlar bazasi nima? Ma'lumotlar bazasi jadvallardan tashkil topgan va ular o'zaro bog'langan bo'ladi. Jadvallarda ustunlar ma'lumotlar saqlanishi kerak bo'lgan ma'lumotlar tiplari va nomlari sifatida xizmat qilsa, qatorlar ushbu ma'lumotlarni o'zini bildiradi.

Har bir jadvalda takrorlanmas ustun bo'lishi kerak, chunku u qatorni identifikatsiyalashi kerak. Bu ustun kalitli ustun deyiladi. Bu kalitsiz jadvalni o'zgartirish ko'p qiyinchiliklar va xatolarga olib keladi. Kalitli ustunlar bundan tashqari boshqa jadvallar bilan bog'lanishi uchun ham ishlatiladi. Agarda hattoki sizda

bog'langan jadvallar bo'lmasa hamki, ushbu kalitli ustun sizga kerak bo'ladi. Kalitli ustun(maydon) sifatida sonli maydonni ishlatish maslahat beriladi, imkoni bo'lsa (**autoincrement**) tipidagi ustunni.

Jadvaldagi ustunlar nomi takrorlanmas bo'lishi kerak, lekin sonli bo'lishi shart emas. Ularni istalgancha nomlashingiz mumkin, faqat ismlar takrorlanmasligi kerak(shu yerda kelajakda ma'lumotlar bazasi bilan dastur tuzishimizni inobatgaolib jadval ustunlari nomini o'zgaruvchilar nomlari qoidalariga moslab qo'yishni maslahat beramiz, masalan nomlanishi son bilan boshlanmasin yoki kirill alifbosi harflari mavjud bo'lmasin yoki maxsus belgilar ishlatilmasin ["_" belgisi istisno]). Har bir ustun (maydon) o'zning tipiga ega bo'lishi kerak.

Indekslar nima? Odatda ko'pincha jadvaldagi ma'lumotlarni o'zgartirish zarur, shuning uchun ularni birinchi navbatda tezkor qidirib topish kerak. Qidirish – juda qiyin operatsiya hisoblanadi, ayniqsa jadvalda juda ko'p ma'lumot bo'lsa. Indekslar bu ishlarni tezlashtirish imkonini beradi va saralash uchun keng ishlatiladi. Indekslar orqali jadvaldagi ma'lumotlar saralanadi. Indekslar ham jadvallarning bir turi. Ko'pincha oddiy asosiy ma'lumotlar bazasida saralanmagan ma'lumotlar joylashsa, indekslarda ular saralanadi.

Delphi tizimida ma'lumotlar bazasi bilan ishlash uchun bir nechta komponentlar to'plami mavjud. Ularning har biri ma'lum bir vazifalarni yechish uchun qulay. Nega bunchalik ko'p komponentlar to'plami? Ularning barchasi ma'lumotlarga murojaat etishning turli texnologiyalarini qo'llashadi va imkoniyatlari bilan farqlanadi. Bundan tashqari shunday komponentlar mavjudki, ularni doim ishlatamiz. Ushbu komponentlar guruhini qisqa ko'rib o'tsak.

□ **DataAccess** – bu komponentalar barchasi uchun umumiy va boshqa komponentlar bilan hamkorlikda ishlatib bo'ladi;

□ DataControls – jadvallardagi ma'lumotlarni aks etish va tahrirlash uchun komponentlar. Bu komponentlar ham qanday ma'lumotlarga murojaat qilish texnologiyasi qo'llanilgandan qat'iy nazar qo'llanilishi mumkin;

□ BDE – Borland kompaniyasining Borland Database Engine texnologiyasi asosida ma'lumotlarga murojaat qilish texnologiyasi ishlab chiqilgan. Bu texnologiya eski MB murojaat etish uchun qulay.

□ DBExpress – Borland kompaniyasining ma'lumotlarga murojaat etishning yangi texnologiyasi. Bu komponentalarni mijoz-server texnologiyasidagi dasturlarda ishlatish qulay;

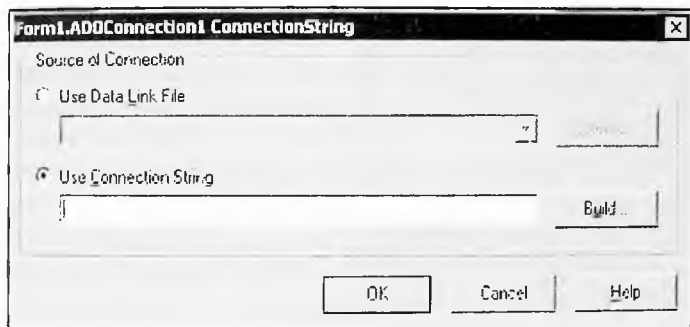
□ ADO(Active Data Object) – Microsoft kompaniyasi tomonidan ishlab chiqilgan ma'lumotlar bazasiga murojaatning yangi texnologiyasi. Juda yaxshi texnologiya, lekin uni odatda Microsoft MB bilan ishlatishni maslahat beriladi. Masalan MS Access yoki SQL Server bilan ishlashda yoki uni agar sizda noan'anaviy ma'lumotlar bazasi serveri bo'lsa unga bog'lanish uchun ODBC orqali amalga oshirish mumkin.

Ma'lumotlar bazasiga ulanish

MB ga ulanishni yaxshiroq tushunish uchun misol ko'rib o'tsak va bunda ADO MB ga murojaat etish texnologiyasidan foydalansak. Biz misolimizda MS Access MB bilan ishlaymiz va misol tariqasida MS Access tuzilgan tayyor MB ga ulanishni ko'rib o'tamiz.

Bu uchun birinchi navbatda formamizga ADOConnection komponentini joylashiriramiz va uning ConnectionString xususiyatidan foydalanamiz yoki komponentning ustiga sichqoncha bilan 2 marotaba murojaat etish mumkin. Ma'lumotlar bazasiga ulanishni ko'rishimiz uchun oddiy MS Access MB ga ulanish misolida ko'rib o'tsak. Oldin aytganimizdek bu uchun ADO MB murojaat texnologiyasi komponentalaridan

foydalanamiz. Bu uchun ADO komponentlar top'lamidan **ADOConnection** komponentidan foydalanamiz. MB sizga ulanish uchun ushbu komponentning **ConnectionString** xususiyatiga sichqoncha bilan 2 marotaba tezkor murojaat eting. Shunda quyidagi MB ga ulanish oynasi chiqadi.

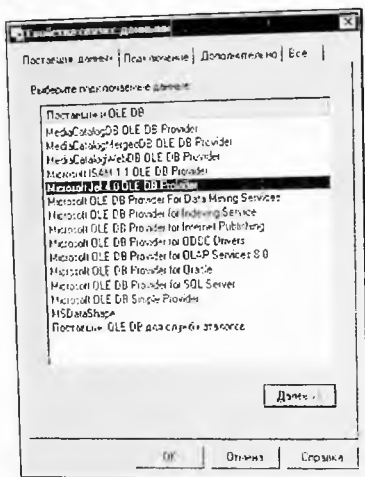


5.1.1-rasm. ADOConnection komponentasining ConnectionString xususiyatidan foydalanib MB ga ulanish turi oynasi.

Bu yerda bizda 2 ta tanlov bor:

- ☐ Maxsus fayl ishlatish;
- ☐ Ulanish qatorini ishlatish.

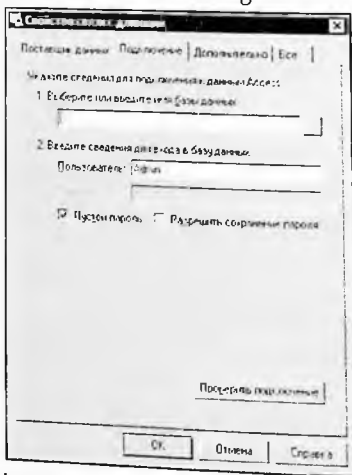
ConnectionString xususiyati oyna chiqaradi. Bunda 2-variantni tanlash maqsadga muvofiq. Bunda ulanish parametrlarini qo'lda kiritish mumkin. Lekin osonroq bo'lishi uchun ustadan foydalanish maqsadga muvofiq. Bu uchun Build tugmasini tanlaymiz. Bunda quyidagi oyna paydo bo'ladi. Bu oynada ma'lumotlar bazasiga murojaat qilish uchun ADO ning drayverlari ro'yxati keltirilgan. Agar biror bir drayver bo'lmasa – **Microsoft OLE DB Provider for ODBC Drivers** ni tanlashiz mumkin.



5.1.2-rasm. ADOConnection komponentasining ConnectionString xususiyatidan MB ga ulanish uchun mos drayverni tanlash oynasi

Bizning misolimizda biz MS Access MB ulanishimiz kerak shu uchun Microsoft Jet OLE DB Provider ni tanlaymiz. Bunday drayver albatta Office bilan birga o'rnatiladi odatda. Bazi holatlarda uning 2 ta versiyasi o'rnatiladi, shu uchun

yangisini tanlash maqsadga muvofiq (Microsoft Jet 4.0 OLE DB Provider). Shundan so'ng **Далее** (Next) tugmasi tanlanadi va Подключения bo'limiga o'tiladi.



5.1.3-rasm. ADOConnection komponentasining ConnectionString xususiyatidan MB manzili va boshqa parametrlarini sozlash oynasi

Подключение bo'limi ko'rinishi tanlangan drayverga bog'liq. Bizning misolimizda u yuqoridagi rasmdagidek ko'rinishda bo'ladi. Bu oynaning **Выберите или введите имя базы данных** (Select or enter a database name) maydonini to'ldirish kerak. Bu uchun ... tugmasidan foydalanib, sichqoncha bilan

tanlash orqali to'ldirish mumkin. Agar baza dasturning joriy papakasida joylashgan bo'lsa, unda to'liq manzil o'rniga MB faylini nomini keltirish kifoya.

Bundan tashqari quyidagi ustunlarni to'ldirish talab etislihi mumkin:

- ☐ Пользователь (User name);
- ☐ Пароль (Password);
- ☐ Пустой пароль (Blank password);
- ☐ Разрешить сохранение пароля (Allow saving password).

Ma'lumotlar to'ldirilib **Проверьте подключение (Test Connection)** tugmasini tanlab, tog'ri ulanish amalga oshirilganini tekshirish mumkin.

MB ga har bir murojaat paytida sizdan login va parol so'ramasligi uchun **ADOConnection** komponentining **LoginPrompt** xususiyatini **false** qiymatiga o'zgartiring. So'ngra **Connection** xususiyatini **true** qiymatiga almashtigach ulanish yuz beradi va Delphi tizimidan MB ga ulanish yuz beradi.

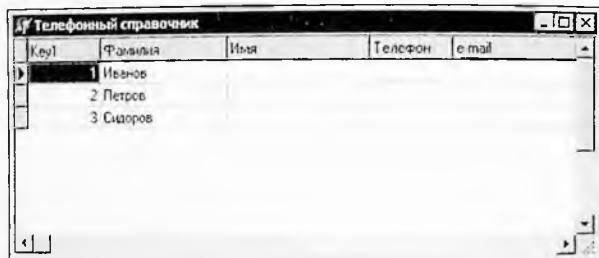
Ulanish amalga oshirilgach endi MB dagi har bir jadvalga ulanish mumkin. Bu uchun **ADO** palitrasidagi **ADOTable** komponentidan foydalanish mumkin.

ADOTable komponenti qaysi MBdagi jadvallarga ulanishini bilishi uchun uning **Connection** xususiyatiga **ADOConnection** komponentini ko'rsatish kerak. So'ngra **ADOTable** komponentining **TableName** xususiyatiga kerakli jadvalni ro'yxatdan tanlash kerak. Oxirgi qadam **ADOTable** komponentini **Active** xususiyatini **true** qiymatiga o'zgartirib jadvalni ochish mumkin.

Ochilgan jadval ma'lumotlarini Delphi formasida aks etish uchun **DataAccess** palitrasidan **DataSource** komponentidan foydalaning.

Ushbu komponentga qaysi jadvalni ask etish kerakligini ko'rsatish zarur. Bu uchun ushbu komponentning **DataSet** xususiyatiga ro'yxatdan **ADOTable** komponentini tanlash zarur.

Endi tayyor ma'lumotlarni turli ko'rinishda aks etishimiz mumkin. Jadval ma'lumotlarini ask etishning eng oddiy yo'li **DBGrid** komponenti orqali ularni Delphi formasida jadval ko'rinishida aks etish. Bu komponent ma'lumotlarni aks etish uchun uning **DataSource** xususiyatiga yuqorida keltirilgan **DataSource** komponenti nomini ro'yxatdan tanlash yetarli.



5.1.4-rasm. DBGrid komponentasining formada ko'rinishi.

Keling endi MB ga ulanishni xulosalaydigan bo'lsak, bunda qanday komponentlardan foydalandik va ular qanday rol o'ynaydi:

☐ **TADOConnection** – ma'lumotlar bazasi bilan ulanish. Bu ulanishni keyinchalik **ADOTable** singari komponentlar ishlatishi mumkin;

☐ **TADOTable** – MB jadvallarini ochish uchun komponent;

☐ **TDataSource** – Bu komponent **TADOTable** bilan ulanib, ma'lumotlarni oddiy komponentlarga ask etish uchun xizmat qiladi (Misol: **DBGrid** komponentida aks etish).

TADOTable komponentini juda ko'p kerakli xususiyatlari mavjud. Ularni qisqacha ko'rib o'tsak:

☐ **ReadOnly** – agar bu xususiyat **true** qiymatiga ega bo'lsa, unda jadval ma'lumotlarini o'zgartirib bo'lmaydi;

☐ **Filter** – filtrlash qatori;

☐ **Filtered** – jadval filtrlanganligini aniqlash;

☐ **IndexFieldNames** – indekslanadigan ustunlar nomi;

☐ RecNo-joriy tanlangan qator nomeri;
☐ RecordCount-jadvaldagi qatorlar soni;
☐ Sort-saralash qatori;
☐ FiledCount-ustunlar soni;
☐ Fields-bu xususiyat orqali jadvalning ixtiyoriy maydoniga murojaat etish mumkin;

☐ FieldValues-bu xususiyat orqali jadvalning ixtiyoriy qiymatiga ustun nomi orqali murojaat etish mumkin;

☐ FilterOption-filtrni sozlash. Bunda quyidagilardan birini tanlash mumkin:

❖ FoCaseInsensitive-filtrga harflar katta kichikligi ta'sir etmaydi;

❖ foNoPartialCompare-agar ushbu parametr tanlangan bo'lsa, taqqoslash filtriga ko'rsatilgan aniq qiymat bo'yicha olib boriladi;

☐ Modified-jadvaldagi ma'lumotlar o'zgartirilganligini aniqlash.

TADOTable komponentining juda ko'p kerakli xususiyatlari mavjudligini ko'rib o'tdik. Endi siz bilan ushbu komponentning metodlari bilan tanishsak:

☐ CancelUpdate-kesh xotirada saqlangan o'zgartirishlarni bekor qilish;

☐ DeleteRecords-jadvaldagi qatorlarni o'chirish. Metodda bir parametr ishlatiladi va u quyidagilardan birini tanlash imkonini beradi:

-arCurrent-faqat joriy qatorni o'chirish;

-arFiltered-faqat filterdan o'tgan qatorlarni o'chirish;

-arAll-barha qatorni o'chirish;

☐ Append-jadval oxirida yangi qator qo'shish;

☐ Cancel-joriy qatorni o'zgartirishni bekor qilish;

☐ Close-jadvalni yopish;

☐ Delete-joriy qatorni o'chirish;

☐ Edit-tahrirlash rejimiga o'tish;

- ☐ FieldByName-ustunni nomi bo'yicha topish;
- ☐ First-jadvalning 1-qatoriga o'tish;
- ☐ Insert-jadvalga yangi qator qo'shish;
- ☐ isEmpty-agar jadval bo'sh bo'lsa metod true qiymatini qaytaradi;

- ☐ Last-jadvalning oxirgi qatoriga o'tish;
- ☐ Next-jadvalning keyingi qatoriga o'tish;
- ☐ Post-barcha o'zgarishlarni saqlash;
- ☐ Prior-jadvalda joriy qatordan birta oldingi qatorga o'tish;
- ☐ Refresh-ma'lumotlar haqida malumotni yangilash;
- ☐ UpdateRecord-joriy qatorni yangilash.

Asosiy tayanch so'z va iboralar

Ma'lumotlar bazasi, relatsion MB, mijoz-server, TADOConnection, ConnectionString, draver, ODBC, TADOTable, indeks, TDataSource, TDBGrid.

Takrorlash va mustaqil ishlash uchun savollar

1. Relatsion ma'lumotlar bazasi nazariyasi haqida yozing.
2. Ma'lumotlar bazasiga ulanish jarayoni haqida yozing.
3. Ma'lumotlarni saralash jarayoni haqida yozing.
4. Ma'lumotlarni filtrlash jarayoni haqida yozing.
5. SQL so'rovlar tili nima?

§5.2. Ob'yektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan ishlash

Ba'zan ma'lumotlar bazasi (MB) bilan bog'liq dasturlarimizda ma'lumotlarni saralashga to'g'ri kelib qoladi. Bu uchun indeksli maydonlar mavjud va ular ishini osonlashtiradi. Indeksli ma'lumotlarni qidirish va saralashda juda qulay hisoblanadi, lekin ma'lumotlarni qo'shish va tahrirlashda tezlikka ta'sir etadi, chunki dastur qayta saralashga majbur bo'ladi. Shuning uchun indeksli zarur holatda ishlatish kerak. Qanday ustunlarni

indekslash kerak? Bu uchun qaysi ustun bo'yicha ko'p qidirish mavjud bo'lgan ustunlarini indekslash kerak.

MS Access ma'lumotlar bazasida ustunni indekslash uchun jadvalni konstruktor rejimiga ochib, mos ustunni tanlab oynaning quyi qismida quyidagi 3 variantdan birini tanlash kerak:

☐ Нет-ustun indekslashmagan;

☐ Да(допускается совпадение)-ustun indekslashgan va ikki (va undan ko'p) qiymatlar bir xil qiymatda bo'lishi mumkin;

☐ Да(совпадение не допускается)-agar bu variantni tanlasangiz, bu indekslangan ustunda ikki va undan ko'p bir xil qiymatni saqlab bilmaysiz.



5.2.1-rasm. MS Access MBBT da ustunlarni indekslash oynasi

Delphi tizimida ustun bo'yicha ma'lumotlarni saralab ko'rsatish uchun quyidagi misoldagidek koddan foydalanish mumkin:

ADOTable1.IndexFieldNames:= 'Фамилия'; yoki

ADOTable1.IndexFieldNames:= 'Телефон';

Ma'lumotlarni

filtrlash Filter xususiyatiga shartni kiritish tuzilishi: Ustun [taqqoslash operatori] 'qiymat'

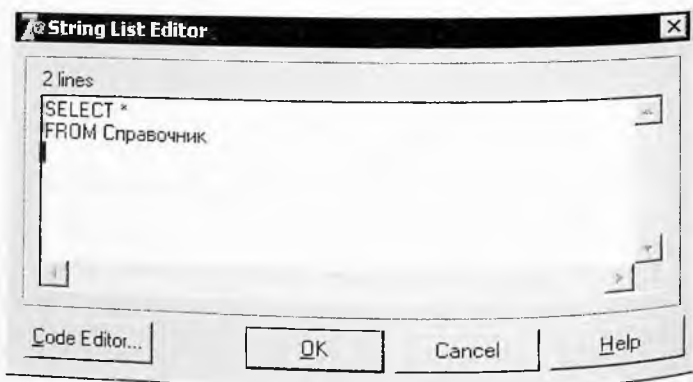
Misol: ADOTable1.Filter:= 'Familiya="Saidov"';

Yoki ADOTable1.Filter:= 'Familiya='+QuotedStr(Saidov);

Yoki ADOTable1.Filter:= 'Familiya='+QuotedStr(Saidov) or 'Telefon='+ QuotedStr(+998901234567);

SQL so'rovlar tili

Ma'lumotlar bazasiga Delphi tizmidan so'rovlarni amalga oshirish uchun TADOQuery komponenti ishlatiladi. Bu komponent ham ADO(yangi versiyasida dbGO) komponentlar bo'limida joylashgan. Bu komponentning ishi TADOTable komponentiga o'xshaydi lekin faqrqli tomoni mavjud albatta. Bu komponentning SQL xususiyatiga so'rovlarni kiritib bajartirish mumkin. So'rovlar ushbu xususiyatga kiritilgandan so'ng, uning Active xususiyatini true qiymatiga o'zgartirish orqali bajartirish mumkin.



5.2.2-rasm. TADOQuery komponentasi SQL so'rovlarni kiritish oynasi

```

procedure TMainForm.FindButtonClick(Sender: TObject);
begin
    DataModule1.FindQuery.Active:=false;
    DataModule1.FindQuery.SQL.Clear;
    DataModule1.FindQuery.SQL.Add('SELECT *');
    DataModule1.FindQuery.SQL.Add('FROM Справочник');
    DataModule1.FindQuery.SQL.Add('WHERE Телефон LIKE
        '''+FindTelephoneEdit.Text+''');
    DataModule1.FindQuery.Active:=true;

    FindResultForm.ShowModal;
end;

```

5.2.3-rasm. SQL so'rovlarni dastur kodida kiritish misoli.

So'rovlarni amalga oshirish uchun **ADOQuery** komponentini faollashtirish yetarli (**Active** xususiyati qiymatini **true** qiymatiga o'zgartirish). Bu juda yaxshi ishlaydi agar siz ma'lumotlar bazasi jadvallaridan ma'lumotlarni olmoqchi bo'lsangiz. Lekin agar jadvallarda ma'lumotlar o'zgartirilsa, o'chirilsa yoki jadvalning tuzilishi o'zgartirilsa (insert, update, delete va hokazo) unda **ADOQuery** komponentini **ExecSQL** metodini chaqirish kerak. Qachon faollashtirish kerak va qachon **ExecSQL** metodini chaqirish kerak? Javob juda oddiy: agar so'rov ma'lumotlarni qaytarsa faollashtirish kerak, agar so'rov o'zgartirishni amalga oshirsa **ExecSQL** operatoridan foydalaning.

Ma'lumotlar bazasiga dinamik ulanish

Tayyor ma'lumotlar bazasini ishlatish juda qulay, lekin foydalanuvchi boshqa bazaga ulanmoqchi bo'lsa nima yuz beradi? Bu nima uchun kerak degan savol tug'ilishi tabiiy. Masalan: Arxivdan qidirish kerak bo'lsa yoki ma'lumotlar bazasi manzili o'zgarsa va hokazo.

Ma'lumotlar bazasiga dinamik ulanish uchun **MS Access** **MB** ga ulanish qatorini tahlil etsak. Bunda minimal qator quyidagicha bo'ladi:

Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=baza.mdb; Persist Security Info=false

Bu qator 3 ta qismdan iborat:

- ☐ *Provider=Microsoft.Jet.OLEDB.4.0* – ta'minotchi nomi bo'lib, u orqali MB ga ulanish yuz beradi;
- ☐ *Data Source=baza.mdb* – MB ga yo'l va MB si fayli nomi;
- ☐ *Persist Security Info = false* – xavfsizlik bo'yicha ma'lumotni saqlash.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    if OpenFileDialog1.Execute then
        begin
            DataModule1.ADOConnection1.Close;
            DataModule1.ADOConnection1.ConnectionString:=
                'Provider=Microsoft.Jet.OLEDB.4.0;'+
                'Data Source='+OpenFileDialog1.FileName+
                ';Persist Security Info=false ';
            DataModule1.ADOConnection1.Open;
        end;
end;
```

5.2.4-rasm. Ma'lumotlarga bazasiga dinamik ulanish kodi misoli.

Ma'lumotlar to'plami va DataSource komponenti hodisalari

ADOTable komponentining juda ko'p hodisalari mavjud:

- ☐ AfterCancel-hodisa o'zgartirish bekor bo'lgandan keyin yuz beradi;
- ☐ AfterClose-hodisa jadval yopilgandan keyin yuz beradi;
- ☐ AfterDelete-hodisa qator o'chirilgandan keyin yuz beradi;
- ☐ AfterEdit-hodisa jadval o'zgartirish rejimiga o'tgandan keyin yuz beradi;
- ☐ AfterInsert-hodisa qator qo'shilgandan keyin yuz beradi;
- ☐ AfterOpen-hodisa jadval ochilgandan keyin yuz beradi;
- ☐ AfterPost-hodisa qator o'zgartirilishi saqlangandan keyin yuz beradi;

☐ AfterRefresh-hodisa jadval yangilangandan keyin yuz beradi;

☐ AfterScroll-hodisa jadvalda boshqa qatorga o'zgandan keyin yuz beradi.

☐ AfterOpen-hodisa jadval ochilgandan keyin yuz beradi;

☐ AfterPost-hodisa qator o'zgartirilishi saqlangandan keyin yuz beradi;

☐ AfterRefresh-hodisa jadval yangilangandan keyin yuz beradi;

☐ AfterScroll-hodisa jadvalda boshqa qatorga o'zgandan keyin yuz beradi.

Bundan tashqari Before so'zi bilan boshlanuvchi hodisalar mavjud bo'lib, ular oldindan yuzaga keladi.

DataSource komponenti ma'lumotlar to'plami (ADO Table, ADOQ uery va hokazo) va ma'lumotlar bilan ishlash komponentlari (TDBGrid, TDBEdit va hokazo) o'rtasida interfeys hisoblanadi. Bu komponentning hodisalari juda ko'p emas lekin juda kerakli:

☐ On DataChange-hodisa ma'lumotlar to'plamida o'zgartilgan yoki kursor boshqa qatorga o'tganda yuz beradi;

☐ On StateChange-hodisa ma'lumotlar to'plami holati o'zgarganda yuz beradi;

☐ On Update Data-hodisa joriy qator ma'lumotlari yangilanishi kerak bo'lganda yuz beradi.

Ma'lumotlar to'plamida pozitsiyalanish

Ko'pincha joriy ma'lumotlar to'plami ichidan ma'lum bir qatorni topish va unga o'tish zaruriyati tug'iladi. Buni tsikl operatorida Next metodidan foydalanib amalga oshirib bo'ladi lekin osonroq va tezroq yo'li mavjud. Bu uchun Locate operatoridan foydalanish mumkin.

Uning 3 ta parametri mavjud:

☐ Qidirilish amaga oshiriladigan ustun(ustunlar);

☐ Qidiriladigan qiymat(qiymatlar);

☐ Qidirish parametrlari.

Parametr sifatini 2 ta qiymatdan birini ishlatish mumkin:

`loPartialKey`-qidiriladigan qiymat to'liq mos kelmasligi mumkin, ya'ni bir qismi to'g'ri kelishi mumkin lekin boshidan boshlab;

`loCaseInsensitive`-registr (harflarning katta yoki kichikligi) inobatga olinmaydi.

Agar qiymat topilsa natija `true` qiymatini qaytaradi aks holda esa `false` qiymatini qaytaradi.

Misol: `If not AdoTable1.Locate('Familiya', Edit1.Text, []) then ShowMessage ('Qator topilmadi');` Yoki `Ado Table 1.Locate ('Familiya',Edit1. Text, [loPartialKey]);`

Yoki `Ado Table 1.Locate ('Familiya; Telefon', VarArray Of ([('Saidov','+998901234567')), []];`

Asosiy tayanch so'z va iboralar

Indeks, saralash, TADOQuery, so'rov, SQL tili, dinamik ulanish, TDataSource, ExecSQL, Locate, loPartialKey, loCaseInsensitive.

Takrorlash va mustaqil ishlash uchun savollar

- 1.Ma'lumotlarni filtrlash jarayoni haqida yozing.
- 2.SQL so'rovlar tili nima?
- 3.Ma'lumotlar bazasiga dinamik ulanish jarayoni haqida yozing.
- 4.Ma'lumotlar to'plami va `DataSource` komponenti hodisalari haqida yozing.
- 5.Ma'lumotlar to'plamida pozitsiyalanish jarayoni haqida yozing.

§5.3. Hisobotlarni tayyorlash

Ma'lumotlar bazasi bilan ishlashni o'rgangandan so'ng albatta hisobotlarni tayyorlash muhim hisoblanadi. Dasturning asosiy funktsiyalaridan biri bu hisobot tayyorlashidir. Ba'zi dastur hisobotsiz bu hech qanday qiymatsiz dastur hisoblanadi. Ushbu paragrafda hisobotlarni tayyorlashni ko'rib o'tamiz. Ko'pgina hisobotlarni tayyorlash ma'lumotlar bazasi bilan bog'liq bo'lib,

ko'pincha ushbu dasturlarda hisobotlarni tayyorlash juda kerak bo'ladi.

Excelda hisobotlar yaratish

Birinci navbatda biz Excel dasturiga hisobotlarni tayyorlashni ko'rib o'tamiz. Ma'lumotlar bazasidan Excel ga ma'lumotlarni eksportlash har bir dasturchiga ko'pincha kerak bo'ladi, chunki bilamizki ishchi kompyuterlarning ko'pchiligida Office paketi o'rnatilgan bo'ladi.

Office paketi dasturlari bilan ishlash uchun birinci navbatda Delphi modulining uses bo'limiga ComObj kutubxonasini qo'shish kerak. Shundan so'ng Office paketi dasturlari bilan ishlashimiz mumkin.

Excel bilan ishlash uchun dastur kodiga misol:

```
Var  
  XLApp, Sheet, ColumnVariant;  
  index, i: Integer;  
begin  
  XLApp:= CreateOleObject('Excel.Application');  
  XLApp.Visible:=true;  
  XLApp.Workbooks.Add(-4167);  
  XLApp.Workbooks[1].Worksheets[1].Name:='Order';  
  Column:=XLApp.Workbooks[1].Worksheets['Order'].Columns;  
  Column.Columns[1].ColumnWidth:=20;  
  Column.Columns[2].ColumnWidth:=20;  
  Column.Columns[3].ColumnWidth:=20;  
  Column.Columns[4].ColumnWidth:=20;  
  Column.Columns[5].ColumnWidth:=20;  
  Column:=XLApp.Workbooks[1].Worksheets['Order'].Rows;  
  Column.Rows[2].Font.Bold:= true;  
  Column.Rows[1].Font.Bold:=true;  
  Column.Rows[1].Font.Color:=clBlack;  
  Column.Rows[1].Font.Size:=14;  
  
  Sheet:=XLApp.Workbooks[1].Worksheets['Order'];  
  Sheet.Cells[1,2]:='Telefonni qayd qilish';  
  Sheet.Cells[2,1]:='Customer';  
  Sheet.Cells[2,2]:='Name';  
  Sheet.Cells[2,3]:='e-mail';  
  Sheet.Cells[2,4]:='Phone';  
  Sheet.Cells[2,5]:='Data paketi';
```

5.3.1-rasm. Delphi dasturlash tilida Excel dasturi bilan muloqot kodi misoli.

Endi bu kodlarni qisqa izohini berib o'tsak. Birinchi qator Excel ob'yektini yaratadi va uni XLApp o'zgaruvchisiga o'zlashtiradi. Bu o'zgaruvchining tipi Variant tipi bo'lib, u istalgan qiymatni olishi mumkin: qator, son, ko'rsatkichlar.

Dasturning ikkinchi qatori Excel dasturini ekranga ko'rsatadi. Dasturning 3-qatorida joriy Excel kitobiga yangi list qo'shadi. Dasturning 4-qatorida joriy kitobning 1-listini nomini o'zgartiradi.

Dasturning 5-qatorida Excel kitobining ustunlarini boshqarish uchun uning ustunlariga ko'rsatkichni olamiz.

Dasturning 6-qatoridan to 10-qatorigacha joriy Excel kitobiga joriy listning 1-ustunidan 5-ustunigacha bo'lganlarni kengligini 20 ga o'zgartiramiz.

Dasturning 11-qatorida Excel kitobining qatorlarini boshqarish uchun uning qatorlariga ko'rsatkichni olamiz.

Dasturning 12-qatoridan to 15-qatorigacha joriy Excel kitobiga joriy listning 1-qator va 2-qatorlarini parametrlarini o'zgartiramiz.

Dasturning 14-qatoridan to 20-qatorigacha joriy Excel kitobiga joriy listning mos yacheykalarini o'zgartirish kodi keltirilgan.

Agar siz Excel kitobining alohida yacheykalari bilan ishlamoqchi bo'lsangiz ularning manzilini ko'rsatishingiz zarur. Masalan:

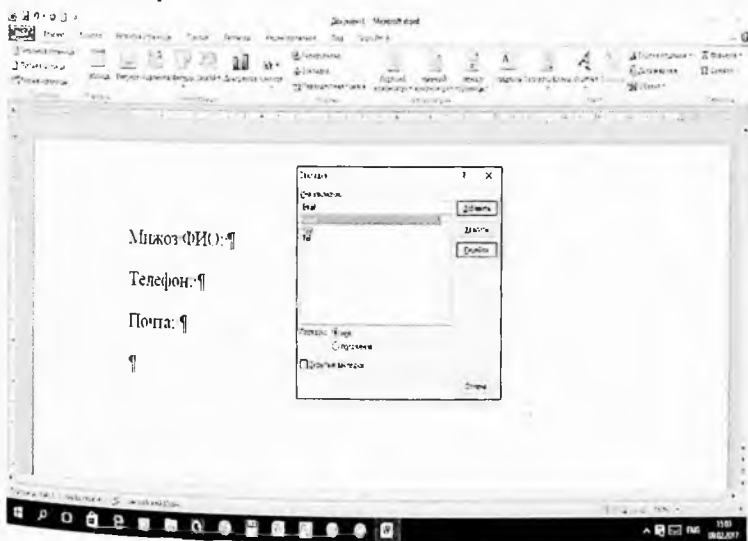
Sheet.Cells[qator, ustun].Font.Italic-kursiv shrift; Sheet.Cells [qator, ustun]. Font.Bold-yo'gon shrift; Sheet.Cells [qator,ustun]. Font.UnderLine-tagchiziq shrift; Sheet.Cells[qator, ustun]. Font.Size-shrift hajmi.

Word da hisobotlar yaratish

Office paketining Word dasturida hisobotlarni yaratishning tezkor va qulay usuli bo'lib, bu tayyor shablonlarni mos o'rinlarini matnlar va boshqa ma'lumotlar bilan to'ldirishdan iborat.

Faqat Word hujjatlarida biz dastur kodimizda kursorimiz pozitsiasini aniq ko'rsatishimiz zarur. Excel dasturi bilan ishlaganda yacheykalar aniq manzilga ega bo'lardi, lekin Word hujjatlarida esa bu unday emas (jadval tuzilishi istisno). Birta probelga adashish ham qo'pol xatoga olib keladi. Dasturiy jihatdan kursorni biz xoxlagan pozitsiyaga joylashtirmoqchi bo'lsak, "Закладка" lardan foydalanish mumkin.

Bu uchun masalan "FIO" matnidan keyin xodimni yoki mijozni **FAMILIYA, ISMI va SHARIFI** bilan to'ldirmoqchi bo'lsangiz, ushbu so'zdan keyin kursorni keltirib "Вставка" menyusining "Закладки" bo'limidan foydalanish mumkin. Albatta bu ishlar Word matn muharririda bajariladi. Zakladkalarni joyashtirib bo'lgach, Delphi dasturlash tili yordamida Word hujjatini boshqarishimiz mumkin.



5.3.2-rasm. Word matn muharririda Закладка lardan foydalanish oynasi
Word bilan ishlash uchun dastur kodiga misol:

```

var
  WordApp: TDocVariant;
begin
  try
    WordApp := CreateOleObject('Word.Application');
    WordApp.Documents.Open(
      ExtractFilePath(Application.ExeName) + 'DocTemplate.doc');
    WordApp.ActiveDocument.SaveAs(
      ExtractFilePath(Application.ExeName) + 'DocTemplate1.doc');
    WordApp.Visible := True;
    WordApp.Selection.GoTo(-1, wdAssigned, wdAssigned, 'F10');
    WordApp.Selection.TypeText('Shablon: Miskat');

    WordApp.Selection.GoTo(-1, wdAssigned, wdAssigned, 'Date');
    WordApp.Selection.TypeText(FormatDateTime('ddmmdd', Date));
  except
  end;
end;

```

5.3.3-rasm. Delphi dasturlash tilida Word dasturi bilan muloqot kodi misoli.

Word bilan ishlash kodi misolini tahlil etadigan bo'lsak, boshlang'ich kodlar xuddi oldingi Excel bilan bog'liq misoldagidek, yangi OLE ob'yekt yaratish bilan boshlanadi. Faqat bu Word dasturi uchun bo'ladi. Shundan so'ng **shablon** hujjatni yuklab oladi.

Endi biz **shablon** hujjatini o'zgartirishni boshlaymiz, lekin foydalanuvchi noxosdan uni saqlamasligi uchun quyidagi ikki yechimni qo'llashimiz mumkin:

Идентификационный номер документа	[Подготовил]
Дата документа: 8/10/2009	
Тема:	
СТУЖЕБНАЯ ЗАПИСКА	
Текст служебной	
Полнота выполнения	Фамилия И.О. (состав)

5.3.4-rasm. Word dasturida shablon shafi misoli.

☐ Faylga faqat o'qish uchun flagni o'rnatish va Word unda shablonni saqlashga yo'l qo'ymaydi hamda yangi nom qo'yishni so'raydi;

☐ Darhol hujjatni yangi nom bilan saqlash mumkin, bunda o'zgartirishlar yangi hujjatda amalga oshirilib shablon hujjat o'zgarmaydi.

Bizning misolimizda biz ikkinchi usulni ishlatamiz. Bunda *ActiveDocument.SaveAs* buyrug'idan foydalanamiz.

Shablonni o'zgartirmaslik yechimi topilgandan keyin esa **WordApp.Visible:=true** buyrug'i yordamida Word hujjatni ko'rsatish mumkin. Endi biz shablonni to'ldirishga tayyormiz. Bu uchun matnni kiritish o'rniga o'tishimiz zarur. Bunda oldin yaratgan "Заказка" lardan foydalanamiz va ularga o'tamiz. Bunda **Selection.GoTo** metodidan foydalanamiz. Matnni kiritish uchun esa **Selection.TypeText** metodidan foydalanamiz.

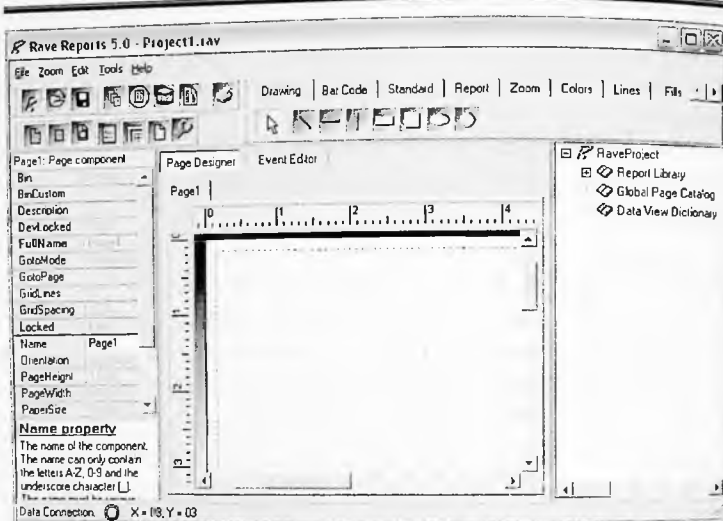
RAVE komponentlaridan foydalanib hisobotlar tayyorlash

RaveReport hisobotlar generatori **Nervona** firmasi tomonidan ishlab chiqilgan bo'lib, ushbu hisobotlarni yaratish texnologiyasi **Delphi7** va undan keyingi versiyalarda qo'llanilib kelinmoqda. **RaveReport** 3 qismdan iborat:

☐ Hisobotlarni generatsiyalash yadrosi hisobotni boshqarish, xomaki ko'rinishi va chop etishni ta'minlaydi;

☐ **RaveReport** ning hisobotlarni ishlab chiqish vizual muhiti hisobotlarni o'zini tayyorlab qo'yish uchun mo'ljallangan;

☐ **Delphi** tarkibidagi **Rave** komponentlari, ular hisobotlarni dasturdan boshqarishni amalga oshirishadi.



3.3.5-rasm. RaveReport dasturi bosh oynasi.

RaveReport hisobotlarini Delphi tizimi orqali boshqarishni ta'minlash uchun Rave bo'limida maxsus komponentlar mavjud. Rave ning barcha komponentlarini 4 guruhga ajratish mumkin:

- ☐ Hisobot loyihasini boshqarish;
- ☐ Ma'lumotlar bilan bog'lovchilar;
- ☐ Ekran yoki printeriga chiqarish;
- ☐ Biror-bir formatga eksportlash. Rave Report komponentlar

ro'yxati:

☐ RvProject-Delphi va Rave hisobotni o'zaro boshqarish uchun ishlatiladi;

☐ RvSystem-hisobotlar o'rtasida o'zaro harakat va chop etish hamda ekranga chiqarish bo'yicha boshqaruvni amalga oshiradi;

☐ RvCustomConnection-matnli fayl, Excel va boshqa tipdagi manbaalar bilan ulanish uchun kerak;

☐ RvDataSetConnection-MBBT ma'lumotlar to'plami bilan ulanish uchun kerak; kerak;

□RvTableConnection-Table tipidagi komponent bilan ulanish uchun kerak;

□RvQueryConnection-Query tipidagi komponent bilan ulanish uchun

□RvNDRWriter-keyinchalik xomaki ko'rsatish va printerga chiqarish uchun hisobotlarni tayyorlash uchun komponent;

□RvRenderPreview-ekranga hisobotni chiqarish uchun kerak;

□RvRenderPrinter-hisobotni printerga chiqarish uchun kerak;

□Rv Render PDF, RvRender HTML, Rv Render RTF, Rv Render Text- komponentlar guruhi bo'lib, shu .formatlarga eksportlaydi.

Asosiy tayanch so'z va iboralar

Hisobot, Office, ComObj, uses, Excel, Word, Variant, Закладка, shablon, OLE object, Rave Report.

Takrorlash va mustaqil ishlash uchun savollar

- 1.Excel da hisobotlar yaratish jarayoni haqida yozing.
- 2.Excel da dasturiy pozitsiyalanish qanday amalga oshirladi?
- 3.Word da hisobotlar yaratish jarayoni haqida yozing.
- 4.Word matn muharririda namuna(shablon) fayllarni o'zgarishini oldini olish uchun qanday usullar mavjud?
- 5.RAVE komponentlaridan foydalanib hisobotlar tayyorlash jarayoni haqida yozing.

§5.4. Oqimlar bilan ishlash.

Zamonaviy operatsion tizimlar ko'p vazifali hisoblanadi. Bu ular bir vaqtning o'zida bir nechta vazifani bir vaqtda bajarish mumkinligini bildiradi. Siz dasturni ishga tushurganda u uchun aloxida oqim ochiladi va bu oqimda u bajariladi. Lekin siz faqat shu oqim bilan chegaralangan degani emas. Ixtiyoriy vaqtda siz

qo'shimcha oqimlar yaratishingiz mumkinki, ular aloxida bajariladi.

Siz yangi dasturni yaratish jarayonida aloxida funktsiyalarning barchasini aloxida oqimga joylashtirish shart emas. Har bir yangi oqim dasturga murakkablik va barqarorsizlik keltiradi hamda bunday dasturlarni otladkalash juda qiyin bo'ladi. Bu yerda "yetarlicha kerakli" tamoyiliga amal qilish kerak, ya'ni ular zarur paytdagini ishlatish maqul. Qanday kodni aloxida oqimga joylashtirish kerak? Quyida biz bir nechta vaziyatlarni ko'rib o'tamizki, bu holatlarda aloxida oqimlarni ishlatish maqsadga muvofiq.

□ Biror-bir funktsiya asosiy jarayonga paralell ravisda bajarilishi kerak bo'lsa, aloxida oqim qo'llaniladi;

□ Biror-bir hisoblash ishlari juda uzoq davom etsa (bekor qilish imkonini ta'minlash uchun hisoblash kodi tarkibiga Application.ProcessMessages chaqiruvini joylashtirish mumkin);

□ Bajarilish vaqtida kod kritik bo'lganda.

Oddiy oqim yaratish

Yangi oqimarni yaratish uchun **File->New->Other** bo'limidan **Thread Object** punktini tanlash kerak.

Delphi tizimi bizga yangi oqim yaratib beradi va yangi oqimda bajariladigan yoki hisoblanadigan vazifalar uchun **Execute** protsedurasi shablonini tayyorlab beradi.

```

unit MyThread;

interface

uses Classes;

type
  TCountObj = class(TThread)
  private
    { Private Declarations }
  protected
    procedure Execute; override;
  end;

implementation

{ TCountObj }

procedure TCountObj.Execute;
begin
    { Place thread code here }

end;

end.

```

5.4.1-rasm. Delphi tizimida oqimlarni yaratish namuna kodi.

Oqimni ishga tushurish uchun asosiy formada quyidagi koddan foydalaning:

```

Procedure TForm1.ButtonClick(Sender:TObject);
begin
    co:=TCountObj.Create(true);          co.Resume;
co.Priority:=tpLower; end; Oqimni ishini to'xtatish uchun esa
asosiy formada quyidagi koddan foydalaning:

```

```

Procedure TForm1.ButtonClick(Sender:TObject); begin
co.Terminate; end.

```

Asosiy tayanch so'z va iboralar

Oqimlar, parallellik, Application.ProcessMessages, kritik bo'lim, thread, excute, Resume, tpLower, terminate.

Takrorlash va mustaqil ishlash uchun savollar

- 1.Oqimlar nima?
- 2.Parallel tizimlarda oqimlarning roli.
- 3.Oqimlar nazariyasi haqida nimani bilasiz?
- 4.Oddiy oqim yaratish jarayoni haqida yozing.
- 5.Oqimlarni ishga tushurish va to'xtatish

§5.5. Dinamik ulanadigan kutubxonalar.

DLL nima?

Dunyoning ko'p dasturchilari yozilgan kodni ko'p marotaba ishlatish yechimi bilan shug'ullanardi. Eng birinchi yechimlardan biri modulli dasturlashni ishlatish bo'ldi. Siz dasturning aloxida qismini dasturda tuzasiz, uni aloxida modul sifatida rasmiylashtirasiz, keyinchalik uni o'z dasturlarizda ishlatishingiz mumkin. Bu juda qulay va oddiy edi. Bu juda yaxshi qo'llanilardi toki ko'pvazifalik paydo bo'lmaguncha. Shundan so'ng bir necha muammolar paydo bo'ldiki, bunda yangi yechim talab etildi.

Muammo1: Agar modul 1MB hajmga ega bo'lsa va bu modulni bir necha dastur ishlatasa, unda tashqi xotirada shuncha marotaba ortiqcha modul hajmi mavjud bo'ladi!

Muammo2: Agar modul 1MB hajmga ega bo'lsa va bu modulni bir necha dastur ishlatasa, unda tezkor xotirada shuncha marotaba ortiqcha modul hajmi mavjud bo'ladi!

Yechim: Bu muammolarni yechish uchun puxta yechim o'ylab topildi. Bun yechimda modullar asosiy dasturga qo'shilmaydi, aloxida faylga saqlanadi va ixtiyoriy dastur kerakli paytda uni yuklab olishi va ishlatishi mumkin. Shunday qilib, DLL fayllar paydo bo'ldi. Bu DLL fayllarda protsedura va funktsiyalar, resurslar, grafiklar va boshqa ma'lumotlar saqlanadi.

DLL fayllarni yuklash jarayonida kamchiliklardan biri – bu ularni yuklash jarayonida ortiqcha vaqt sarflanadi.

DLL fayllarning yana bir yutug'i esa – dastur kodi bir necha fayllarga bo'linadi. Kelingki ushbu kodlardan birida xatolik yuz berdi. Bu holatda dasturni to'liq yangilash kerak emas, balki faqat bir DLL faylni yangilab uni foydalanuvchilarga tarqatish kifoya.

Dinamik ulanadigan kutubxonalarda 3 kamchilik mavjud:

□DLL-fayldagi kod asosiy dastur xotirasida bajariladi (DLL kod global o'zgaruvchilarni saqlab bilmasligi mumkin);

❑ Azaldan “dinamik ulanadigan kutubxonalar” protsedurali bo’lgan;

❑ Versiyani nazoratlash imkoniyatining mavjud emasligi.

Oddiy DLL yaratish

DLL kutubxona bu aloxida fayl ekan demak, Delphi tizimida DLL fayllar aloxida loyiha sifatida yaratilishi kerak. Yangi dinamik kutubxona yaratish uchun **FILE->New->Other** menyusidan foydalanish mumkin. Bu bo’limdan DLL Wizard punktini tanlash kerak. Modul quyidagi kod namunasini saqlaydi:

```
Library Project1; Uses SysUtils,Classes; {$R *.res} Begin End.
```

Misol tariqasida ikkita sonni qo’shish funksiyasini DLL faylga kiritsak: `Library Project1; uses SysUtils,Classes; Fucntion Summ(X,Y:integer):integer; StdCall; Begin`

```
Result:=X+Y; End;
```

`Exports Sum; {$R *.res} Begin End. function Sum; stdcall; external 'Project6.dll' name 'Sum';` Kodli qatorda DLL fayldagi Sum funksiyaga nom bo’yicha qidirish amalga oshirilmoqda. DLL fayl tarkibidagi protsedura yoki funksiyaga indeks orqali ham qidirishni amalga oshirsa bo’ladi. Bu uchun kodni quyidagicha almashtirish mumkin:

DLL loyihasida export kalitli so’zi va Sum funksiyasidan keyin index va raqam qo’yiladi:

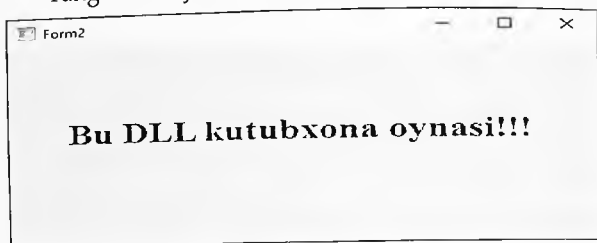
```
Exports Sum index 10; Asosiy modulda esa: function Sum; stdcall; external 'Project6.dll' index 10;
```

Formalarni DLL fayllarda saqlash

Endi biz DLL fayllarda formalarni saqlashni va ishlatishni ko’rib o’tamiz. Bu juda qulay hisoblanadi qachonki dasturning juda kam ishlatiladigan oynalarini DLL faylga joylashtirsak va kerak bo’lgandagina chaqirib olsak maqsadga muvofiq bo’ladi. Bunda asosiy dastur ortiqcha koddan xolis bo’ladi.

DLL fayl: `Library Project1; Uses SysUtils,Classes;`

{ $\$R$ *.res} Exports ShowAbout name 'ShowAbout'; Begin End.
Yangi forma yaratamiz: File->New Form.



5.5.1-rasm. DLL faylga joylashtirish uchun forma misoli.

DLL kod: var Form1: TForm1; Kodidan keyin quyidagi kodni qo'shamiz: procedure ShowAbout(Handle:THandle); export; stdcall; DLL kod: implementation { $\$R$ *.dfm} Kodidan keyin quyidagi kodni qo'shamiz: procedure ShowAbout (Handle:THandle); begin Application. Handle:= Handle; Form1:= TForm1. Create (Application); Form1 .Show Modal;

Form1.Free; end; Asosiy modulda: Uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

Kodidan keyin quyidagi kodni qo'shamiz: procedure ShowAbout(Handle:THandle) stdcall; var

Form2: TForm2; Kodidan keyin quyidagi kodni qo'shamiz: procedure ShowAbout; external 'About.dll' name 'ShowAbout';

Asosiy modulda: procedure TForm2.Button2Click(Sender: TObject); begin ShowAbout(Handle); end;

Kutubxonalarni dinamik yuklash

Oldingi misollar yaxshi va oson, lekin ularning bir xususiyati borki – **dinamik** kutubxonalar dastur ishga tushganda avtomatik yuklanadi. Bunda ikki **kamchilikni** kuzatish mumkin:

□ Dastur yuklanishi bir oz sekinlashadi, kutubxonadagi funksiyalar esa dastur ishlashi jarayonida umuman kerak bo'lmasligi mumkin;

❑ Dasturni DLL fayllarsiz taqdim etish mumkin, lekin bu ham yordam bermaydi, chunki dastur yuklanish paytida kutubxona yo'qligini aytib xatolik beradi.

Bu kamchiliklarni bartaraf etish uchun kutubxonalarni dinamik (zarur paytdagina) yuklashni amalga oshirib yechish mumkin. Uses Windows, Messages, Sys Utils, Variants, Classes, Graphics, Controls, Forms, Dialogs, Std Ctrls; //procedure ShowAbout(Handle:THandle) stdcall;

Type Kodidan keyin quyidagi kodni kiritish zarur: TShowA = procedure (Handle:THandle); stdcall; var Hnd DLLHandle: THandle; showabout: TShowA; begin try // DLL faylni dinamik yuklashhndDLLHandle := loadLibrary ('About.dll'); if hndDLLHandle < 0 then begin // qismdastur manzilini olish

@showabout := get ProcAddress (hnd DLLHandle, 'ShowAbout'); // agar qismdastur manzili topilsa if addr (showabout) < nil then begin // qism dasturni ishlatish ShowAbout (Application.Handle); end else // DLL faylda qismdastur topilmasashowMessage ('Qsim dastur topilmadi...');

end else // DLL fayl topilmasa show Message ('DLL fayl topilmadi...'); finally // DLL fayl xotiradan chiqarildi freeLibrary (hndDLLHandle); end; end;

Asosiy tayanch so'z va iboralar

DLL, xotira, modul, kutubxona, verisya, DLL Wizard, export, indes, forma, handle.

Takrorlash va mustaqil ishlash uchun savollar

- 1.DLL nima?
- 2.Oddiy DLL yaratish jarayoni haqida yozing.
- 3.Formalarni DLL fayllarda saqlash jarayoni haqida yozing.
- 4.Kutubxonalarni dinamik yuklash jarayoni haqida yozing.
- 5.Chetki DLL lardan foydalanish

§5.6. Ob'yektga yo'naltirilgan dasturlashda buffer xotiradan foydalanish

Dasturlarni testlash

Delphi tizimining ko'pgina komponentlari buffer xotira bilan ishlash uchun tayyor holatda bo'ladi. Bu asosan shunday komponentlarga ta'alluqligi, ularda ma'lum bir ma'lumotlar saqlanadi va foydalanuvchi ularni buffer xotiraga joylashtirib biladi.

Agar komponentda kiritilgan ma'lumotni buffer xotirada olish imkoniyati bo'lsa demak, unda quyidagi metodlar mavjud:

- ☐ CutToClipboard-buffer xotiraga qirqib olish;
- ☐ CopyToClipboard-buffer xotiraga nusxalash;
- ☐ PasteFromClipboard-buffer xotiradan joylashtirish.

Clipboard ob'yekti

Delphi tizimida buffer xotira bilan ishlash uchun **Clipboard** ob'yekti mavjud. Bu ob'yekt bo'lishiga qaramay, uni initsializatsiyalash shart emas (TApplication yoki TPrinter singari), balkim, uni hech qanday tayyorgarliksiz ishlatish mumkin. Bu uchun **uses** bo'limiga **clipbrd** so'zini yozish kifoya.

Bu ob'yektning xususiyatlari va metodlari juda ko'p emas.

SetTextBuf metodi matnini buffer xotiraga nusxalaydi. Parametr sifatida odatda komponent xususiyati ko'rsatiladi. Misol:

```
Clipboard.SetTextBuf(Pchar(Memo1.SelText));
```

Bu yerda matnni **Pchar** tipiga keltirish zarur chunki, bu metod Windows API funktsiyasini chaqirmoqda.

AsText metodi matn ko'rinishida bufderdagi matnni ko'rsatadi. Misol: `Memo1.SelText:=Clipboard.AsText;`

Clipboard ob'yektining boshqa metodlari bilan tanishsak:

- ☐ Assign-TPersistent tipiga mos ob'yektni buffer xotiraga belgilash;
- ☐ Clear-buffer xotira tarkibini tozalash;

□ HasFormat-buffer xotirada qanday tipdagi ma'lumotlar saqlanganligini tekshiradi:

- > CF_TEXT-matn ko'rinishida;
- > CF_BITMAP-tasvir ko'rinishida;
- > CF_METAFILEPICT-vektorli grafika;
- > CF_PICTURE-TPicture tipdagi ob'yekt;
- > CF_COMPONENT-bufferda komponent.

- SetComponent-buffer xotiraga komponentni joylashtirish;
- SetTextBuf-buffer xotiraga matnli ma'lumotni joylashtirish.

Tasvir va buffer xotira

Keling endi buffer xotirada tasvirlar bilan ishlashni ko'rib o'tsak. Biz bilamizki Delphi tizimida tasvirlar bilan ishlash uchun asosiy komponent TImage komponenti xizmat qiladi. TImage komponentidagi tasvirni buffer xotiraga joylashtirishimiz uchun quyidagi koddan foydalanishimiz mumkin:

```
Clipboard.Assign(Image1.Picture);
```

Buffer xotiradagi tasvirni TImage komponentasiga yuklash uchun quyidagi koddan foydalanish mumkin:

```
Image1.Picture.Assign(Clipboard); Otladkalash
```

Otladka – bu dasturni qadamli bajaritirish. Bu rejimda kodning har bir qatori bajariladi va darhol boshqaruv Delphi tizimiga qaytib o'tkaziladi. Delphi tizimi keyingi buyruqni bajarilishini kutib o'tiradi. Dastur ma'lum bir qadamda to'xtatilganda, siz o'zgaruvchilarni qiymatlarini ko'rishingiz mumkin va hattoki ularni qiymatini o'zgartirishingiz ham mumkin.

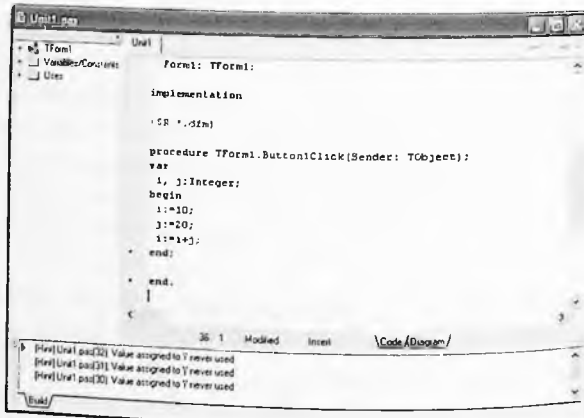
To'xtatish nuqtasi – dastur kodi qatori bo'lib, dastur shu qatorda bajarilishini to'xtatadi va boshqaruv Delphi tizimiga davom ettirish uchun beriladi.

To'xtash nuqtasini joylashtirish uchun kodning kerakli qatoriga kelib, <F5> tugmasini tanlash zarur va bu qator qizil rangga bo'yaladi.



5.6.1-rasm. Delphi tizimida to'xtatish nuqtasidan foydalanish oynasi.

Delphi tizimida kodning chap tomonida ko'k nuqta tursa, demak siz bu qatorni to'xtash nuqtasi sifatida ishlatishingiz mumkin. Agar bunday nuqta bo'lmasa, unda to'xtash yuz bermaydi.



5.6.2-rasm. Delphi tizimida to'xtatish nuqtasidan foydalanish mumkin bo'lgan kod qatorlari misoli.

Dasturni ishini keyingi qatorgacha davom etish uchun <F7> yoki <F8> tugmalarini tanlash mumkin. Agar tanlangan qator kodi sizning funktsiya yoki protsedurangiz bo'lsa, unda <F7> tugmasini tanlab, ushbu funktsiya yoki protsedura ichiga kirib, uni qatorma-qator bajartirish mumkin. <F8> tugmasi esa ichki dastur tarkibiga kirmasdan davom etadi.

Ba'zan xatoliklar ma'lum bir shartlar yuz berganda yuzaga kelishi mumkin. Masalan, tsikllarning qaysidir qadamida bo'lishi mumkin. Unda nima qilish kerak?

1000 marta takroriy qadamdan iborat tsikl bo'lsa, 1000 marotaba <F7> yoki <F8> tugmalariga murojaat etish juda noqulay.

Bu muammoni hal etish uchun eski dasturchilarni usulini qo'llash mumkin, ya'ni tsikl oxirida shart joylashtirish mumkin va shu shartga to'xtash nuqtasi joylashtiramiz. Masalan:

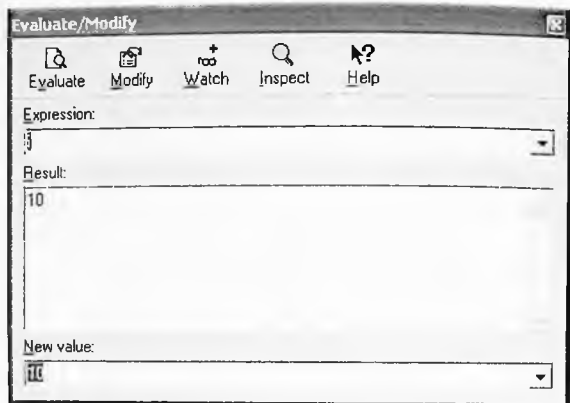
for i:=1 to 1000 do begin if i=999 then Bu qatorda to'xtash nuqtasi; operatorlar; end; Delphi tizimida bunday holatlarni inobatga olib, shart bo'yicha to'xtash nuqtalarini joylashtirish mumkin. Bu uchun "To'xtash nuqtasi" belgisi ustiga sichqoncha o'ng tugmasini tanlash zarur va paydo bo'lgan menyudan "Breakpoint properties" punktini tanlash kerak. Bizda oyna paydo bo'ladi.

Bu oynada 2 parametr bizga qiziqish uyg'otadi:

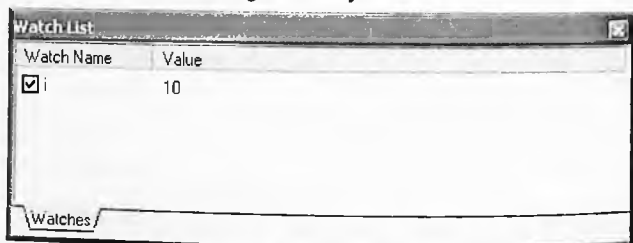
□ Condition – bu yerda shartni kiritish mumkinki, bu shartda to'xtash nuqtasi faollashadi. Bu shart Delphi kodiga o'xshaydi, masalan: i=999;

□ PassCount – o'tishlar sonidan keyin to'xtash zarurligi. Agar 1000 kiritsak, 999 marotaba to'xtamasdan o'tadi va 1000 qadamda to'xtash nuqtasi ishlaydi.

To'xtash nuqatsi ishlaganda biror-bir o'zgaruvchi qiymatlarini ko'rmoqchi bo'lsangiz, uni belgilab <Ctrl+F7> tugmasini tanlab ko'rish mumkin. Ushbu oynada New Value maydonida o'zgaruvchiga boshqa qiymat kiritish mumkin. O'zgaruvchi qiymatini ko'rishni yana bir usuli <Ctrl+F5> tugmasini tanlash.



5.6.3-rasm. Otladkalash jarayonida o'zgaruvchilar qiymatini o'zgartirish oynasi.



5.6.4-rasm. Otladkalash jarayonida o'zgaruvchilar qiymatini ko'rish oynasi.

Asosiy tayanch so'z va iboralar

Clipboard, buffer xotira, nusxalash, qirqib olish, joylashtirish, Pchar, SetTextBuf, AsText, WinAPI, Assing, TImage, otladka, to'xtatish nuqtasi, qadamba- qadam bajartirish, Condition, PassCount.

Takrorlash va mustaqil ishlash uchun savollar

1. Buffer almashinuv jarayoni haqida nimani bilasiz?
2. Clipboard ob'yekti nima?
3. Tasvir va buffer xotira haqida yozing.
4. Otladkalash jarayoni haqida yozing.
5. Dasturiy maxsulotni testlash jarayoni haqida yozing.

V-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI

AMALIY MASHG'ULOT №12

**Ob'yektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan
ishlashni o'rganish**

Ishdan maqsad: Delphi tizimida mavjud ma'lumotlar bazasiga ulanish ko'nikmasini hosil qilish va ma'lumotlar bazasidagi ma'lumotlar bilan ishlashni o'rganish. Ma'lumotlar bazasini to'ldirishda yaratilgan formadan foydalanishni tashkil qilish.

Nazariy ma'lumotlar

Tayyor ma'lumotlar bazasini ishlatish juda qulay, lekin foydalanuvchi boshqa bazaga ulanmoqchi bo'lsa nima yuz beradi? Bu nima uchun kerak degan savol tug'ilishi tabiiy. Masalan: Arxivdan qidirish kerak bo'lsa yoki ma'lumotlar bazasi manzili o'zgarsa va hokazo.

Ma'lumotlar bazasiga dinamik ulanish uchun MS Access MB ga ulanish qatorini tahlil etsak. Bunda minimal qator quyidagicha bo'ladi:

*Provider=Microsoft.Jet.OLEDB.4.0;Data Source=baza.mdb; Persist
Security Info=false*

Bu qator 3 ta qismdan iborat:

□ *Provider=Microsoft.Jet.OLEDB.4.0* – ta'minotchi nomi bo'lib, u orqali MB ga ulanish yuz beradi;

□ *Data Source=baza.mdb* – MB ga yo'l va MB si fayli nomi;

□ *Persist Security Info = false* – xavfsizlik bo'yicha ma'lumotni saqlash.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  if OpenDialog1.Execute then
  begin
    DataModule1.ADOConnection1.Close;
    DataModule1.ADOConnection1.ConnectionString:=
      'Provider=Microsoft.Jet.OLEDB.4.0;'+
      'Data Source='+OpenDialog1.FileName+
      ';Persist Security Info=false ';
    DataModule1.ADOConnection1.Open;
  end;
end;

```

5.7.1-rasm. Ma'lumotlarga bazasiga dinamik ulanish kodi misoli.

Amaliy mashg'ulot bo'yicha topshiriq:

Microsoft Accessda yaratilgan ma'lumotlar bazasini Delphi dasturida yaratilgan formadan to'ldirish.

Ishni bajarish tartibi:

1. Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating va darhol aloxida papkaga loyihani saqlang!



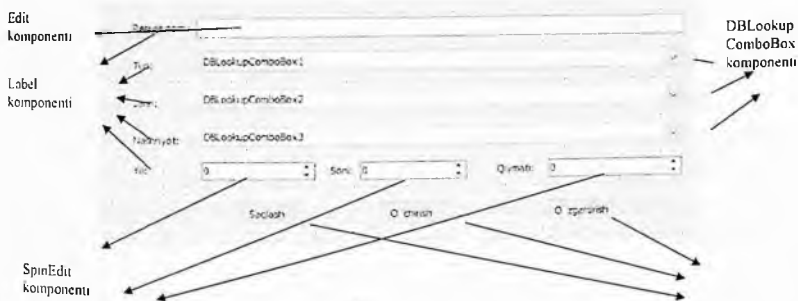
5.7.2-rasm. Delphi tizimida yangi loyiha yaratish.

2. Delphi tizimida yaratilgan ma'lumotlar bazasiga ulaning.
3. TADOTable, TDataSource va TDBGrid komponentalari yordamida

“Resurslar” jadvali formada aks eting.

Xod	Resurs_nomi	Tur	Jami	Tashkilot	Yil	Son	Qiymat
7	Elektronik 1000 usul	1	1	2	2005	5000	7355
8	Dasturlash	1	2	1	2013	4	45648

5.7.3-rasm. Ma'lumotlar bazasidagi Resurslar jadvali ma'lumotlarini aks etish formasi.



5.7.4-rasm. Amaliy mashg'ulotda foydalaniladigan forma va komponentlar.

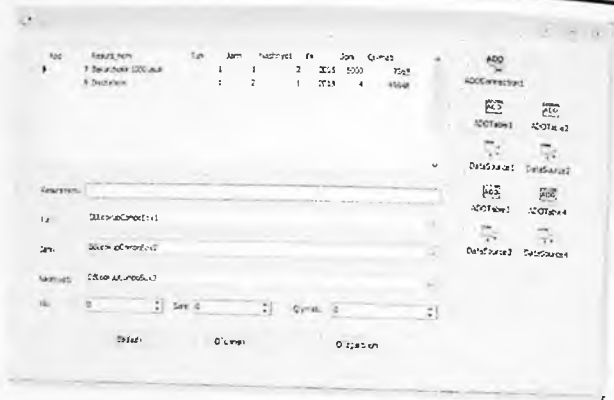
4. Formada TADOTable va TDataSource komponentlaridan foydalanib ma'lumotlar bazasidagi qolgan jadvallariga ham ulaning.

5. Formaning pastki qismida quyidagi ko'rinishni tayyorlang.

Edit komponenti DBLookup Combo Box komponenti

Label Komponenti SpinEdit komponenti Button komponenti

Umumiy ko'rinishda forma quyidagicha bo'lishi kerak:



5.7.5-rasm. Amaliy mashg'ulotdagi ilova yakuniy forma ko'rinishi.

5.DBLookup ComboBox1 komponentining quyidagi xususiyatlarini o'zgartiring:

- ListSource = DataSource2;
- ListField = TurNomi (yoki bazadagi Turi jadvalining 2-ustuni);

•KeyField = Kod (yoki bazadagi Turi jadvalining 1-ustuni);

6.DBLookupComboBox2 komponentining quyidagi xususiyatlarini o'zgartiring:

- ListSource = DataSource3;
- ListField = JanrNomi (yoki bazadagi Janrlar jadvalining 2-ustuni);

•KeyField = Kod (yoki bazadagi Janrlar jadvalining 1-ustuni);

7.DBLookupComboBox3 komponentining quyidagi xususiyatlarini o'zgartiring:

- ListSource = DataSource4;
- ListField = NashrNomi (yoki bazadagi Nashrlar jadvalining 2-ustuni);

•KeyField = Kod (yoki bazadagi Nashrlar jadvalining 1-ustuni);

8. Button1 ("Saqlash" sarlavhali) tugmaga Click hodisasiga ma'lumotlar bazasi "Resurslar" jadvaliga yangi ma'lumot kiritish uchun quyidagi kodini kiriting:

```
ADO Table1. Append; ADO Table1. Fields [1]. As String:=Edit1.  
Text; ADO Table1. Fields[2].AsInteger:=DBLookup ComboBox1.  
Key Value; ADO Table1. Fields[3]. AsInteger:=DBLookup  
ComboBox2. Key Value; ADO Table1.
```

```
Fields[4]. AsInteger:=DBLookup Combo Box3. Key Value;
```

```
ADOTable1.Fields[5].AsInteger:=SpinEdit1.Value;
```

```
ADOTable1.Fields[6].AsInteger:=SpinEdit2.Value;
```

```
ADOTable1.Fields[7].AsInteger:=SpinEdit3.Value;
```

```
ADOTable1.Post; ADOTable1. Close; ADOTable1.Open;
```

9. Button2 ("O'chirish" sarlavhali) tugmaga Click hodisasiga ma'lumotlar bazasi "Resurslar" jadvalidan belgilangan ma'lumot qatorini o'chirish uchun quyidagi kodi kiriting:

ADOTable1.Delete; 10. Dasturni ishga tushurib, ma'lumotlar bazasidagi "Resurslar" jadvaliga yangi ma'lumot qatorini qo'shish va o'chirishni tekshirib ko'ring. Bunda komponentlarda ma'lumotlar kiritilgandan keyin yoki DBGrid jadvalidan kerakli qator belgilangandan keyin tugmalar tanlash zarurligini unutmang!!!

Bajarish uchun topshiriq:

Microsoft Accessda yaratilgan ma'lumotlar bazasini Delphi dasturida yaratilgan formadan to'ldirish uchun dastur yozing.

1. Quyidagi ko'rinishdagi formani hosil qiling.

2. Talaba.mdb ma'lumotlar bazasiga dinamik ulanishni tashkil qiling.

3. Saqlash tugmasi bosilganda talaba haqidagi ma'lumotni bazaga kiritish uchun dastur kodini yozing.

4. O'chirish tugmasi bosilganda talaba haqidagi ma'lumotni bazadan o'chirilishi uchun dastur kodini yozing.

5.7.6-rasm. Amaliy mashg'ulot bo'yicha topshiriq formasi.

AMALIY MASHG'ULOT №13

Hisobotlar tayyorlash

Ishdan maqsad: Delphi tizimida mavjud ma'lumotlar bazasiga ulanish ko'nikmasini hosil qilish va ma'lumotlar bazasidagi ma'lumotlar bilan ishlashni o'rganish. Ma'lumotlar bazasi ma'lumotlaridan hisobotlar yaratishni o'rganish.

Nazariy ma'lumotlar

Ma'lumotlar bazasi bilan ishlashni o'rgangandan so'ng albatta **hisobotlarni** tayyorlash muhim hisoblanadi. Dasturning asosiy funksiyalaridan biri bu hisobot tayyorlashdir. Ko'pgina hisobotlarni tayyorlash ma'lumotlar bazasi bilan bog'liq bo'lib, ko'pincha ushbu dasturlarda hisobotlarni tayyorlash juda kerak bo'ladi.

Birinchi navbatda biz Excel dasturiga hisobotlarni tayyorlashni ko'rib o'tamiz. Ma'lumotlar bazasidan Excelga ma'lumotlarni eksportlash har bir dasturchiga ko'pincha kerak bo'ladi, chunki **bilamizki** ishchi kompyuterlarning ko'pchiligida Office paketi o'rnatilgan.

Office paketi dasturlari bilan ishlash uchun birinchi navbatda Delphi modulining **uses** bo'limiga **ComObj** kutubxonasini

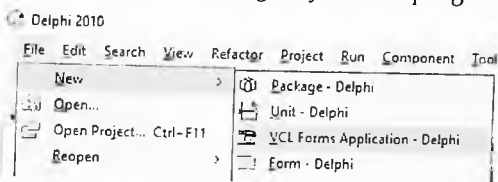
qo'shish kerak. Shundan so'ng Office paketi dasturlari bilan ishlashimiz mumkin.

Amaliy mashg'ulot bo'yicha topshiriq:

Ma'lumotlar bazasidagi ma'lumotlarni Word va Excel dasturlariga exportlash orqali hisobotlar yaratish.

Ishni bajarish tartibi:

1. Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating va darhol aloxida papkaga loyihani saqlang!



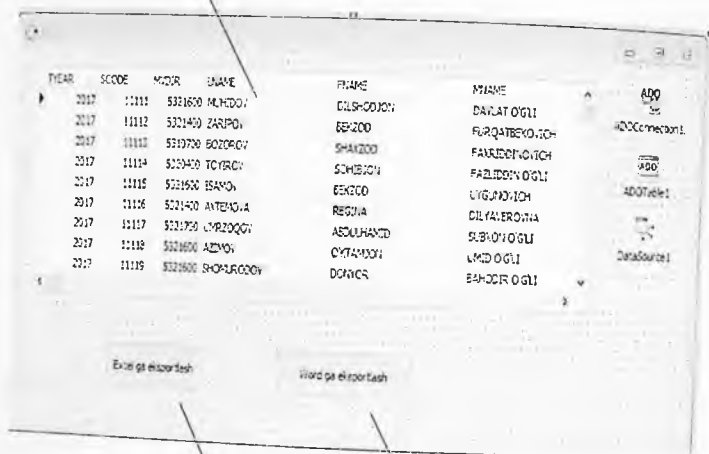
5.8.1-rasm. Delphi tizimida yangi loyiha yaratish.

2. Loyihada papkada joylashgan Baza.mdb ma'lumotlar bazasi fayliga ulanishni tashkil eting.
3. Formada TADOTable, TDataSource va TDBGrid komponentlarni joylashtiring hamda bazadagi "Talabalar" jadvali bilan ulanib, DBGrid komponentiga ma'lumotlarni ko'rsating.
4. Formani quyidagi interfeysini tayyorlang:
DBGrid Button Button
5. Formaga F12 tugmasini tanglang va ushbu forma moduliga o'ting.
6. uses kalitli so'zidan keyin kelgan modular eng oxirida ComObj so'zini kiriting.
7. Klaviaturadan F12 tugmasini tanlang va qaytib formaga o'ting.
8. "Excel ga eksportlash" sarlavhali tugmaning "Click" hodisasiga "TYEAR", "MVDIR", "LNAME", "FNAME", "MNAME", "PSSER", "PSNUM", "BDATE", "SAYEAR" va "Tel" ustunlari bo'yicha ma'lumotlarni Excel dasturiga chiqarish dastur

```

kodini kiriting: procedure TForm1.Button1Click(Sender: TObject);
var XlApp:Variant;      i:Integer; Begin  ADOTable1.First;
XlApp:=CreateOleObject('Excel.Application');
XlApp.WorkBooks.Add(-4167);

```



5.8.2-rasm. Amaliy mashg'ulotda foydalaniladigan forma va komponentlar.

```

XlApp.WorkBooks[1].Worksheets[1].Name:='Talabalar';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,1]:='TYEAR';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,2]:='MVDIR';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,3]:='LNAME';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,4]:='FNAME';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,5]:='MNAME';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,6]:='PSSER';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,7]:='PSNUM';
XlApp.WorkBooks[1].Worksheets[1].Cells[1,8]:='BDATE';

```

```

XLApp.WorkBooks[1].WorkSheets[1].Cells[1,9]:='SAYEAR';
XLApp.WorkBooks[1].WorkSheets[1].Cells[1,10]:='Telefon';
for i:=1 to ADOTable1.RecordCount do begin XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,1]:=ADOTable1.Fields
[0].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,2]:=ADOTable1.Fields
[2].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,3]:=ADOTable1.Fields
[3].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,4]:=ADOTable1.Fields
[4].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,5]:=ADOTable1.Fields
[5].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,6]:=ADOTable1.Fields
[6].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,7]:=ADOTable1.Fields
[7].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,8]:=ADOTable1.Fields
[8].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,9]:=ADOTable1.Fields
[9].AsString; XLApp.
  WorkBooks[1].WorkSheets[1].Cells[i+1,10]:=ADOTable1.Field
s[15].AsString; ADOTable1.Next; end; XLApp.visible:=true; end;
9. Excelga eksportlash dasturini ishga tushurib tekshirib
ko'ring. (Agar Excelga eksportlash biroz vaqt olsa kuting!) 10.
Word matn muharririni oching va quyidagi rasmda keltirilgan
shablonni tayyorlang hamda loyiha EXE kengaytmali fayl
pakasiga SPRAVKA.docx nomi bilan saqlang:

```



5.8.3-rasm. Amaliy mashg'ulotda Word matn muharririda hisobot tayyorlash shabloni.

11. "Word ga eksportlash" sarlavhali tugmaning "Click" hodisasiga "TYEAR", "MVDIR", "LNAME", "FNAME", "MNAME", "PSSER", "PSNUM", "BDATE", "SAYEAR" va "Tel" ustunlari bo'yicha ma'lumotlarni Word dasturiga chiqarish dastur kodini kiriting: procedure TForm1.Button2Click(Sender: TObject); var WordApp,doc:Variant; begin try WordApp:= CreateOle Object ('Word.Application'); doc:= WordApp. documents.Open (Extract File Path (Application.ExeName) +'Spravka.docx'); Word App. Active Document. SaveAs(ExtractFilePath(Application.ExeName)+'Spravka1.docx'); WordApp.visible:=true; WordApp. Selection. GoTo (-1,unAssigned,unAssigned,'TYEAR'); WordApp.Selection.TypeText(ADOTable1.Fields[0].AsString); WordApp.Selection.GoTo (-1,unAssigned, unAssigned,'MVDIR'); WordApp.Selection.TypeText(ADOTable1.Fields[2].AsString); WordApp.Selection.GoTo(-1,unAssigned,unAssigned,'LNAME'); WordApp.Selection.TypeText(ADOTable1.Fields[3].AsString); WordApp.Selection.GoTo(-1,unAssigned,unAssigned,'FNAME'); WordApp.Selection.TypeText(ADOTable1.Fields[4].AsString); WordApp.Selection.GoTo

```

(-1,unAssigned,unAssigned,'MNAME');
WordApp.Selection.TypeText(ADOTable1.Fields[5].AsString);
WordApp.Selection.GoTo(-1,unAssigned,unAssigned,'PSSER');
WordApp.Selection.TypeText(ADOTable1.Fields[6].AsString);
WordApp.Selection.GoTo(-
1,unAssigned,unAssigned,'PSNUM');WordApp.Selection.TypeText(ADOTable1.Fields[7].AsString);
WordApp.Selection.GoTo
(-1,unAssigned,unAssigned,'BDATE');
WordApp.Selection.TypeText(ADOTable1.Fields[8].AsString);
WordApp.Selection.GoTo(-1,unAssigned,unAssigned,'SAYEAR');
WordApp.Selection.TypeText(ADOTable1.Fields[9].AsString);
WordApp.Selection.GoTo(-1,unAssigned,unAssigned,'Tel');
WordApp.Selection.TypeText(ADOTable1.Fields[15].AsString);
Except ShowMessage ('Xatolik yuz berdi!'); end;
WordApp:=unassigned; end;

```

Bajarish uchun topshiriqlar

1. "Word ga eksportlash" sarlavhali tugmaning "Click" hodisasiga Ismi, Familiyasi, Sharifi, Tug'ilgan yili ustunlari bo'yicha ma'lumotlarni Word dasturiga chiqarish dastur kodini yozing;
2. "Excelga eksportlash" sarlavhali tugmaning "Click" hodisasiga Ismi, Familiyasi, Sharifi, Tug'ilgan yili, Telefon raqami ustunlari bo'yicha ma'lumotlarni Excel dasturiga eksportlash dastur kodini yozing.

AMALIY MASHG'ULOT №14

**Ilovalarni yuklashni va bajarilishini tezlashtirish uchun
dinamik ulanadigan kutubxonalarni qo'llash**

Ishdan maqsad: Delphi tizimidan foydalanib DLL fayllarini yaratish va ulardan foydalanish ko'nikmalariga ega bo'lish.

Nazariy ma'lumotlar

DLL fayllarda protsedura va funktsiyalar, resurslar, grafiklar va boshqa ma'lumotlar saqlanadi. DLL fayllarning yana bir yutug'i esa – dastur kodi bir necha fayllarga bo'linadi. Kelingki ushbu kodlardan birida xatolik yuz berdi. Bu holatda dasturni to'liq yangilash kerak emas, balki faqat bir DLL faylni yangilab uni foydalanuvchilarga tarqatish kifoya. DLL kutubxona bu aloxida fayl ekan demak, Delphi tizimida DLL fayllar aloxida loyiha sifatida yaratilishi kerak. Yangi dinamik kutubxona yaratish uchun **FILE->New->Other** menyusidan foydalanish mumkin. Bu bo'limdan DLL Wizard punktini tanlash kerak. Modul quyidagi kod namunasini saqlaydi:

Library Project1; Uses SysUtils,Classes;

[\$R *.res] Begin End. Misol tariqasida ikkita sonni qo'shish funktsiyasini DLL faylga kiritsak: Library Project1; uses SysUtils,Classes; Fucntion Summ(X,Y:integer):integer; StdCall; Begin Result:=X+Y; End; Exports Sum; End.

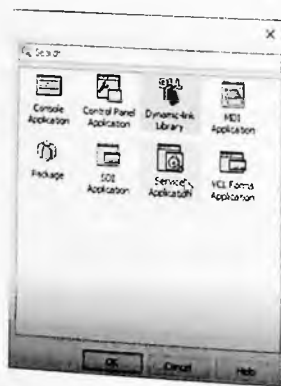
Amaliy mashg'ulot bo'yicha topshiriq:

Uchta sonni qo'shib, ularning kvadratini hisoblash funktsiyasini DLL faylga kiriting.

Ishni bajarish tartibi:

1.DLL alohida fayl bo'lganligi sababli, u Delphi tizimida aloxida loyiha sifatida yaratiladi. Yangi dinamik kutubxonani yaratish uchun **File | New | Other** ni tanlang. Yangi loyiha yaratish uchun oynada, **New** bo'limda **DLL Wizard** elementini tanlang va OK ni bosing.

5.9.1-rasm. Yangiu DLL fayl yaratish oynasi.



Modul quyidagi matnni o'z ichiga oladi: *Library Project1; Uses SysUtils,Classes; End.*

2.Kutubxona nomini Funksiyalar deb o'zgartiring va quyidagi dastur matnini yozing va dasturni xotirada saqlang.

library Funksiyalar;

(Important Note about DLL memory management: SharedWin must be the first unit in your library's USES clause AND your project's (select Project>View>Source) USES clause if your DLL exports any procedures or functions that have arguments or parameters or function results. This applies to all strings passed to and from your DLL-even those that are tagged as constant and classes. SharedWin is the interface unit to the LPMEMORY.dll shared memory manager, which must be deployed along with your DLL. To avoid using SHAREDWIN.dll, pass string information using PChar or ShortString parameters.)

uses

SysUtils,

Classes;

function Funksiya(a,b,c:Integer): Real; stdcall;

begin

Result:=1000(a+b+c);

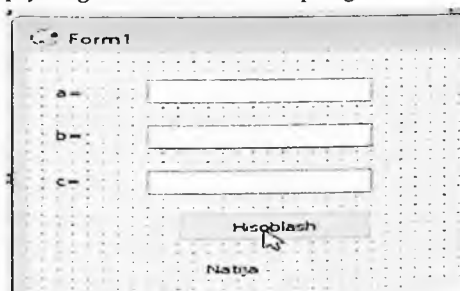
end;

exports Funksiya;

5.9.2-rasm. Amaliy mashg'ulotda yaratiladigan DLL fayl dastur kodi.

3.Yaratilgan kutubxona dasturini xotirada saqlang va *File-Close* buyrug'ini tanlang. Yangi loyiha yarating. *File- New- VCL Forms Application-Delphi.*

4. Formaga 3 ta Edit, 4 ta Label, 1 ta Button komponentini yuklang va quyidagi ko'rinishni hosil qiling.



5.9.3-rasm. Amaliy mashg'ulotda foydalaniladigan forma va komponentlar

Global o'zgaruvchilarni e'lon qilish bo'limiga quyidagini kiriting. *function Funksiya(a,b,c:integer):real; ctdcall;*

Implementation bo'limiga quyidagi kodni kiritamiz: *function* *Funksiya*; *ctdcall*; *external 'Funksiyalar.dll' name 'Funksiya'*;

5.Hisoblash tugmasining *onclick* hodisasiga quyidagi dastur matnini yozamiz:

```
procedure TForm1.Button1Click(Sender: TObject);
var
a,b,c:Integer;
d:Real;
begin
a:=StrToInt(Edit1.Text);
b:=StrToInt(Edit2.Text);
c:=StrToInt(Edit3.Text);
d:=Funksiya(a,b,c);
Label4.Caption:=FloatToStr(d);
end;

end.
```

5.9.4-rasm. Amaliy mashg'ulotda DLL fayldan foydalanish ilovasi dastur kodi.

6. Yaratilgan kutubxona faylini ayni paytda yaratilgan dastur fayllari qatoriga nusxalab qo'yishimiz kerak. Aks holda dasturda xatolik yuz beradi. Ya'ni kutubxonaga murojaat qila olmaydi.

Funksiyalar.dll	12.12.2017 18:01	Recursive DLL	297 KB
Project2.dpr	12.12.2017 18:02	Delphi Project File	1 KB
Project2.dproj	12.12.2017 18:02	Delphi Project File	5 KB
Project2.dproj.local	12.12.2017 18:02	Delphi Project File	1 KB
Project2.exe	12.12.2017 18:02	Delphi Project File	330 KB
Project2.res	12.12.2017 18:02	Compiled Resource	6 KB
Unit1.dcu	12.12.2017 18:02	Delphi Project File	6 KB
Unit1.dfm	12.12.2017 18:02	Delphi Form	2 KB
Unit1.pas	12.12.2017 18:02	Delphi Source File	1 KB

5.9.5-rasm. Amaliy mashg'ulotda loyiha zaruriy fayllari joylashuvi.

7. Dasturni ishlatib undan natija oling.

Bajarish uchun topshiriqlar

1.DLL kutubxonadan foydalangan holda ikkita son bilan qo'shish amalini bajarish funksiyasini DLL faylga kiriting.

2.Funksiya deb nomlangan DLL fayl yarating va sonlarning kubini hisoblashni DLL faylga kiriting.

3.DLL kutubxonadan foydalangan holda ikkita son bilan ayirish amalini bajarish funktsiyasini DLL faylga kiriting.

4.DLL kutubxonadan foydalangan holda ikkita son bilan ko'paytirish amalini bajarish funktsiyasini DLL faylga kiriting.

5.DLL kutubxonadan foydalangan holda ikkita son bilan bo'lish amalini bajarish funktsiyasini DLL faylga kiriting.

TAJRIBA MASHG'ULOTI №12

Obyektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan ishlashni o'rganish

Ishdan maqsad: Delphi tizimida mavjud ma'lumotlar bazasiga ulanish ko'nikmasini hosil qilish va ma'lumotlar bazasidagi ma'lumotlar bilan ishlashni o'rganish. Ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan komponentlar bilan tanishtirish.

Nazariy ma'lumotlar

Delphi tizimida ma'lumotlar bazasi bilan ishlashda ustun bo'yicha ma'lumotlarni saralab ko'rsatish uchun quyidagi misoldagidek koddan foydalanish mumkin:

```
ADOTable1.IndexFieldNames:= 'Фамилия';
```

```
Yoki
```

```
ADOTable1.IndexFieldNames:= 'Телефон';
```

Delphi tizimida ma'lumotlarni jadvaldan qidirish uchun ADOTable komponentining Filter xususiyatidan foydalanish mumkin. Unda ma'lumotlar aks etilishi mumkin bo'lgan shartlar kiritiladi. Masalan bu xususiyatda familiyasi "Saidov" bo'lgan talabalarni ko'rsatishni kiritishingiz mumkin. Filter kuchga kirishi uchun ADOTable komponentining Filtered xususiyatini true qiymatiga o'zgartirish zarur. Shundan keyin Filter xususiyatini o'zgartirishingiz bilan qidirish kuchga avtomatik kiradi.

Filter xususiyati bu matnli qator. Unda shart matnini kiritishimiz zarur. Filter xususiyatiga shartni kiritish tuzilishi:

Ustun [taqqoslash operatori] 'qiymat'

Misol:

ADOTable1.Filter:= 'Familiya="Saidov"';

Yoki ADOTable1.Filter:= 'Familiya='+QuotedStr(Saidov);

Yoki ADOTable1.Filter:= 'Familiya='+QuotedStr(Saidov) or
'Telefon='+ QuotedStr(+998901234567);

Tajriba mashg'uloti bo'yicha topshiriq:

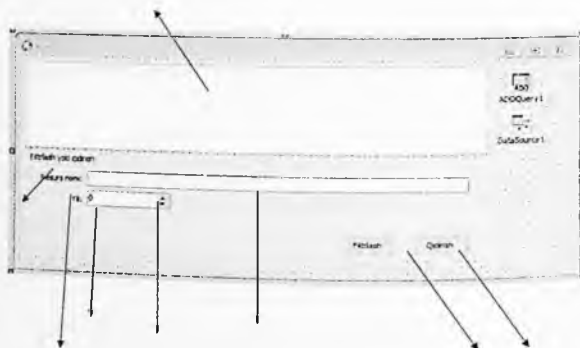
MB ga dinamik ulanib, AdoQuery komponentida sql so'rovlar tilida so'rovlar tashkil qilgan holda ma'lumotlarni qidirishni va filtrlashni amalga oshiring.

Ishni bajaish tartibi:

Baza2.mdb ma'lumotlar bazasi fayliga dinamik ulanishni tashkil eting.

1. Loyihada yangi forma yarating va formada TADOQuery, TDataSource va TDBGrid komponentlarni joylashtiring.

2. Formani quyidagi interfeysini tayyorlang: DBGrid
GroupB ox Label Edit



SpinEdit

Button

5.10.1-rasm. Tajriba mashg'ulotda foydalaniladigan forma va komponentlar

3. Yangi formaga F12 tugmasini tanglang va ushbu forma moduliga o'ting.

```
implementation
```

```
{CR *.def}
```

```
end.
```

5.10.2-rasm. Tajriba mashg'ulot moduli.

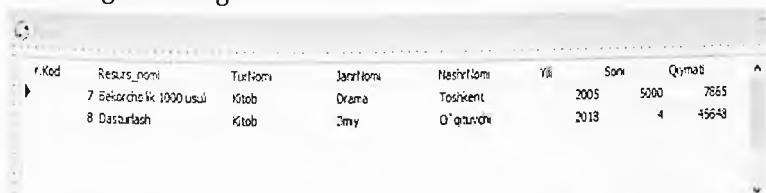
4. implementation so'zidan keying qatorda uses Unit1; so'zini kiriting.

5. Klaviaturadan F12 tugmasini tanlang va qaytib formaga o'ting.

6. ADOQuery komponentining Connection xususiyatiga Form1.ADOConnection1 qiymatini tanlang.

7. ADOQuery komponentining SQL xususiyatiga quyidagi SQL so'rovini kiriting: Select r.*, t.*,j.*,n.* from resurslar as r, turi as t, janrlar as j, nashrlar as n Where t.kod=r.turi and j.kod=r.janri and n.kod=r.nashriyot

8.ADOQuery komponentining Active xususiyatiga true qiymatiga o'zgartiring va DBGrid komponentasini quyidagi ko'rinishga keltiring:



r.Kod	Resurs_nomi	Tur_komi	Janr_komi	Nashr_komi	Yili	Soni	Qiymati
7	Bekorcha ko'rsatuvchi	Kitob	Drama	Toshkent	2005	5000	7865
8	Dasarlash	Kitob	Imy	O'qituvchi	2018	4	45648

5.10.3-rasm. Tajriba mashg'ulotda formasi ko'rinishi.

9. "Filtrlash" sarlavhali tugmaning "Click" hodisasiga "Resurs_nomi" va "Yili" ustunlari bo'yicha filtrlash kodini yozing.

10. "Qidirish" sarlavhali tugmaning "Click" hodisasiga "Resurs_nomi" va "Yili" ustunlari bo'yicha qidirish kodini yozing.

11. Dasturni ishga tushirib tekshiring.

Bajarish uchun topshiriqlar

1. "Filtrlash" sarlavhali tugmaning "Click" hodisasiga "Janr_nomi" ustuni bo'yicha filtrlash kodini yozing.

2. "Qidirish" sarlavhali tugmaning "Click" hodisasiga "Janr_nomi" va "Qiymati" ustunlari bo'yicha qidirish kodini yozing.

TAJIRIBA MASHG'ULOTI №13

Parallel dasturlash uchun oqimlarni qo'llash

Ishdan maqsad: Delphi tizimida kiritish chiqarish jarayonlarini oqimlar yordamida amalga oshirilishini, ma'lumotlarni qayta ishlash jarayonlari bilan bog'liq tahlillarni oqimlar va tizim tahlillariga bog'liqligini tushuntirish.

Nazariy ma'lumotlar

Dasturda yangi oqimlarni yaratish uchun File->New->Other bo'limidan Thread Object punktini tanlash kerak.

Delphi tizimi bizga yangi oqim yaratib beradi va yangi oqimda bajariladigan yoki hisoblanadigan vazifalar uchun Execute protsedurasi shablonini tayyorlab beradi. Oqimlarni qo'llashdan asosiy maqsad:

□ Biror-bir funktsiya asosiy jarayonga paralell ravisda bajarilishi kerak bo'lsa, aloxida oqim qo'llaniladi;

□ Biror-bir hisoblash ishlari juda uzoq davom etsa (bekor qilish imkonini ta'minlash uchun hisoblash kodi tarkibiga Application.ProcessMessages chaqiruvini joylashtirish mumkin);

□ Bajarilish vaqtida kod kiritish kerak bo'lganda.

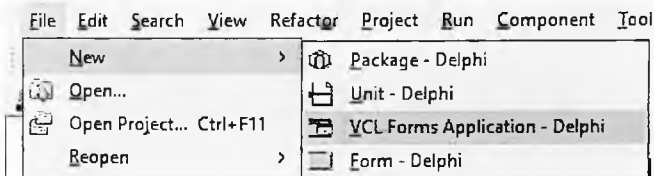
Tajriba mashg'uloti bo'yicha topshiriq:

Edit1da kiritilgan sonning o'rta arifmetik qiymatini hisoblash jarayoni amalga oshirilayotganda, parallel ravishda o'yin uchun dastui tuzish.

Ishni bajarish tartibi:

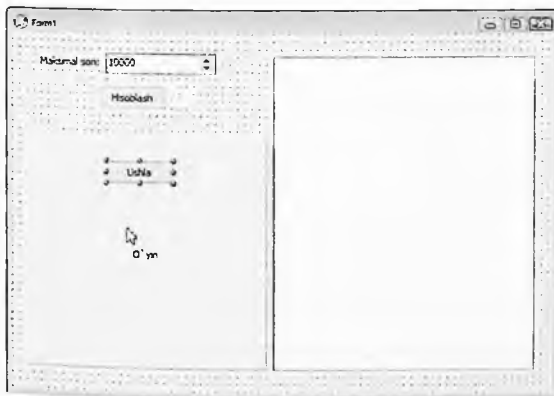
1. Delphi dasturini ishga tushiring va yangi loyiha oynasini yarating va darhol aloxida papkaga loyihani saqlang!

Delphi 2010



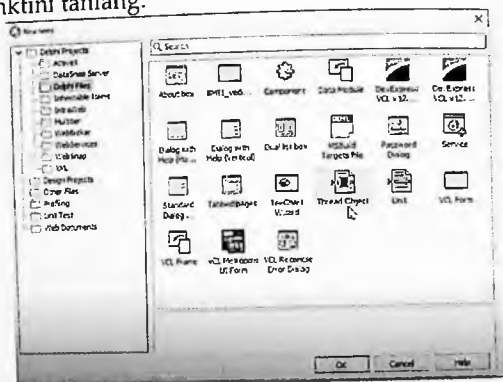
5.11.1-rasm. Delphi tizimida yangi loyiha yaratish.

2. Formaga Label, 2 ta Button, SpinEdit, ListBox, Panel komponentlarini yuklang. Quyidagi forma ko'rinishini hosil qiling.



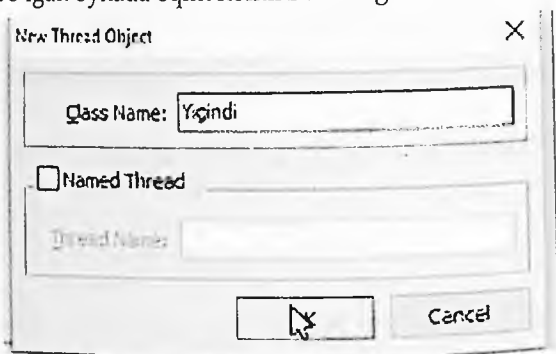
5.11.2-rasm. Tajriba mashg'ulotda foydalaniladigan forma va komponentlar.

3. Delphida yaratayotgan dasturingizni xotirada saqlang. Yangi oqim qo'shish uchun **File->New->Other** bo'limidan Thread Object punktini tanlang.



5.11.3-rasm. Tajriba mashg'ulotda yangi oqim yaratish oynasi.

4. Hosil bo'lgan oynada oqim nomini kiriting.



5.11.4-rasm. Yangi oqimga nom berish oynasi.

5. Hosil bo'lgan dastur matni muharriri oynasining *implementation* bo'limiga *Main* deb yozamiz. *Uses* bo'limiga *System.SysUtils* modulini kiritamiz.

6. *Hisoblash* tugmasining *onclick* hodisasiga quyidagi dastur kodini yozamiz.

```

procedure Yigindi.Executer;
var
  i,j:integer;
  F:double;
begin
  Form1.ListBox1.Clear;
  for i:=1 to Form1.SpanEdit1.Value do
  begin
    F:=0;
    for j:=1 to i do
      F:=F+j;
    F:=F/i;
    Form1.ListBox1.Items.Add(IntToStr(i)+' sonigacha yig andining o'rtacha qiymati: '+FloatToStr(F));
  end;
end;
end.

```

5.11.5-rasm. Tajriba mashg'ulotining dastur kodi.

7.Ushla tugmasining *onMouseEnter* hodisasiga quyidagi dastur matnini yozing;

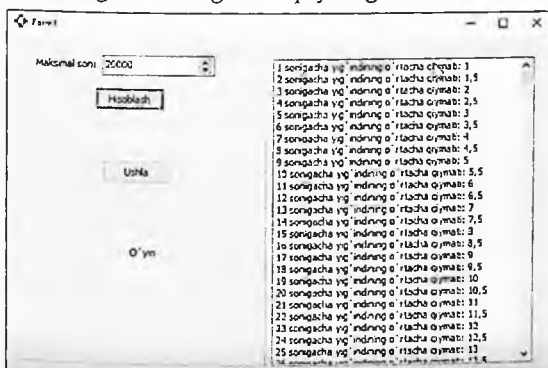
```

procedure TForm1.Button2.MouseEnter(Sender: TObject);
begin
  Button2.Top:=Random(Panel1.Height-Button2.Height);
  Button2.Left:=Random(Panel1.Width-Button2.Width);
end;
end.

```

5.11.6-rasm. Tajriba mashg'ulotining dastur kodi.

8.Dastur ishga tushirilganda quyidagi ko'rinishda bo'ladi.



5.11.7-rasm. Tajriba mashg'ulotilovaisning asosiy formasi ko'rinishi.

Bajarish uchun topshiriq:

1. Yuqorida keltirilgan misolni o'zingiz mustaqil bajaring.
2. Yangi oqim yarating va yangi oqimda bajariladigan yoki hisoblanadigan vazifalar uchun Execute protsedurasi shablonini tayyorlang.
3. Oqimni ishga tushirish kodini yozing.
4. Oqimni ishini to'xtatish uchun esa asosiy formada kod kiriting.

TAJRIBA MASHG'ULOTI №14

Delphi tizimida buffer xotiradan foydalanish

Ishdan maqsad: Delphi tizimida buffer xotiradan keng foydalanish ko'nikmasiga ega bo'lish.

Nazariy ma'lumotlar

Delphi tizimining ko'pgina komponentlari buffer xotira bilan ishlash uchun tayyor holatda bo'ladi. Bu asosan shunday komponentlarga ta'alluqliki, ularda ma'lum bir ma'lumotlar saqlanadi va foydalanuvchi ularni buffer xotiraga joylashtirib biladi.

Agar komponentda kiritilgan ma'lumotni buffer xotirada olish imkoniyati bo'lsa demak, unda quyidagi metodlar mavjud:

- ☐ CutToClipboard-buffer xotiraga qirqib olish;
- ☐ CopyToClipboard-buffer xotiraga nusxalash;
- ☐ PasteFromClipboard-buffer xotiradan joylashtirish.

Delphi tizimida buffer xotira bilan ishlash uchun **Clipboard** ob'yekti mavjud. Bu ob'yekt bo'lishiga qaramay, uni initsializatsiyalash shart emas (TApplication yoki TPrinter singari), balkim, uni hech qanday tayyorgarliksiz ishlatish mumkin. Bu uchun **uses** bo'limiga **clipbrd** so'zini yozish kifoya.

SetTextBuf metodi matnini buffer xotiraga nusxalaydi.
Clipboard.SetTextBuf(Pchar(Memo1.SelText));

Bu yerda matnni PChar tipiga keltirish zarur chunki, bu metod Windows API funktsiyasini chaqirmoqda.

AsText metodi matn ko'rinishida bufferdagi matnni ko'rsatadi.

```
Memo1.SelText:=Clipboard.AsText;
```

TImage komponentidagi tasvirni buffer xotiraga joylashtirishimiz uchun quyidagi koddan foydalanishimiz mumkin:

```
Clipboard.Assign(Image1.Picture);
```

Buffer xotiradagi tasvirni TImage komponentasiga yuklash uchun quyidagi koddan foydalanish mumkin:

```
Image1.Picture.Assign(Clipboard);
```

Tajriba mashg'uloti bo'yicha topshiriq:

Nusxalash va Joylashtirish buyruqlari mavjud bo'lgan asosiy menyu yarating va Image komponentida nusxalash va joylashtirishni amalga oshirish dasturini tuzing

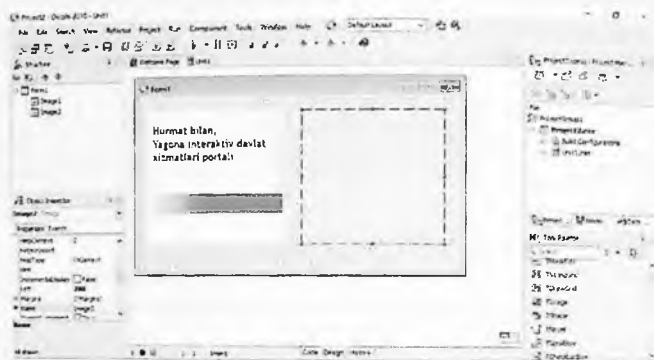
Ishni bajarish tartibi:

1. Delphi tizimini ishga tushuring.
2. Yangi loyiha yarating.
3. Loyihani aloxida papkaga saqlang.
4. Formada ikkita TImage komponentini yonma-yon joylashtiring.



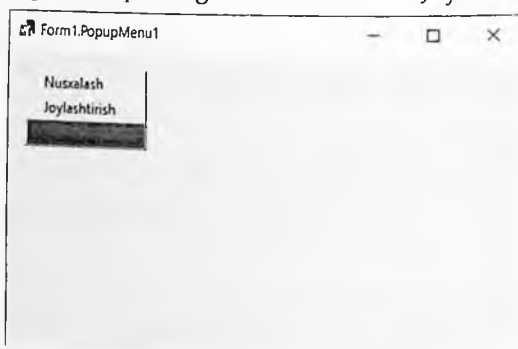
5.12.1-rasm. Tajriba mashg'ulotida zarur bo'ladigan komponentalari.

5. Image1 komponentasining "Picture" xususiyatidan foydalanib ixtiyoriy tasvirni kompyuter xotirasidan yuklang.



5.12.2-rasm. Tajriba mashg'ulotida komponentalardan foydalanish natijasi.

6. Formaga TPopupMenu komponentasini joylashtiring va unga ikkita punktini qo'shing: "Nusxalash" va "Joylashtirish".



5.12.3-rasm. Tajriba mashg'ulotida yordamchi menyuni hosil qilish.

7. Forma moduli kodida (odatda bu Unit1) uses bo'limida Clipbrd kalitli so'zini kiriting, ya'ni Clipboard kutubxonasini loyihamizga qo'shdik.



5.12.4-rasm. Tajriba mashg'ulotida modulida Clipbrd modulini qo'shish.

8.PopupMenu komponentasining "Nusxalash" Click hodisasiga quyidagi kodni yozing: *Clipboard.Assign(Image1.Picture);*

9.PopupMenu komponentasining "Joylashtirish" click hodisasiga quyidagi kodni kiriting:

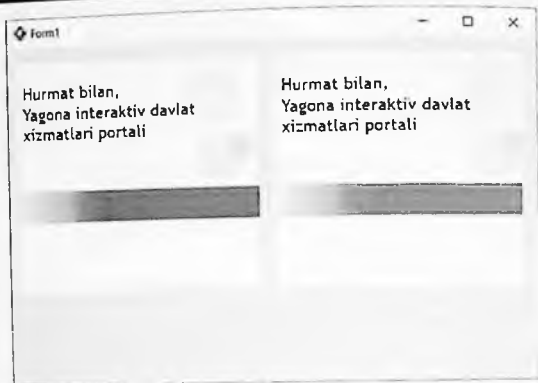
Image2.Picture.Assign(Clipboard);

10.Image1 va Image2 komponentalari "PopupMenu" xususiyatiga PopupMenu1 komponentasini tanlab qo'ying.

11.Loyihani saqlang, kompilyatsiyalang va ishga tushuring.

12.Ilovani ishga tushurgandan so'ng, tasvir ko'rinib turadigan component ustiga sichqonchani o'ng tugmasini tanlang va paydo bo'lgan menyudan "Nusxalash" punktini tanlang.

13.Hech qanday tasvir ko'rinmaydigan Image komponentasi ustiga sichqonchani o'ng tugmasini tanlang va paydo bo'lgan menyudan "Joylashtirish" punktini tanlang. Ushbu holatda tasvir ikkinchi rasm komponentiga nusxalanadi:



5.12.5-rasm. Tajriba mashg'uloti ilovasi formasining yakuniy natijasi.

Bajarish uchun topshiriqlar:

1. Formada 3 ta tasvir komponentini joylashtirng va ular o'rtasida buffer almashinuvini tashkil eting.
2. Formada 4 ta tasvir komponentini joylashtirng va ular o'rtasida buffer almashinuvini tashkil eting.
3. Formada 5 ta tasvir komponentini joylashtirng va ular o'rtasida buffer almashinuvini tashkil eting.
4. Formada 6 ta tasvir komponentini joylashtirng va ular o'rtasida buffer almashinuvini tashkil eting.
5. Formada 7 ta tasvir komponentini joylashtirng va ular o'rtasida buffer almashinuvini tashkil eting.

V-MODUL BO'YICHA TEST SAVOLLARI

1. Korxonada ma'lumotlar bazasidan foydalanuvchilar soni kichik, ma'lumotlarni butunligi va xavfsizligi yuqori darajada ta'minlash shart bo'lmagan, ma'lumotlar bazasi uchun aloxida serverni zarur bo'lmagan hollarda qanday tipdagi ma'lumotlar bazasidan foydalanish mumkin?

- A. Fayl-server
- B. Mijoz-server
- C. Tarmoq-server
- D. Local-server

2. Korxonada ma'lumotlar bazasidan foydalanuvchilar soni katta, ma'lumotlarni butunligi va xavfsizligi yuqori darajada ta'minlash zarur bo'lgan, ma'lumotlar bazasi uchun aloxida serverni zarur bo'lgan hollarda qanday tipdagi ma'lumotlar bazasidan foydalanish mumkin?

- A. Mijoz-server
- B. Fayl-server
- C. Tarmoq-server
- D. Local-server

3. MSAccess, Paradox, dBase, ForPro ma'lumotlar bazasini boshqarish tizimlari qaysi ma'lumotlar bazasi tipiga kiradi?

- A. Fayl-server
- B. Mijoz-server
- C. Tarmoq-server
- D. Local-server

4. SQLServer, Oracle, InterBase ma'lumotlarni bazasini boshqarish tizimlari qaysi ma'lumotlar bazasi tipiga kiradi?

- A. Mijoz-server
- B. Fayl-server
- C. Tarmoq-server
- D. Local-server

5. Relatsion ma'lumotlar bazasidagi kalitli maydon nima uchun kerak?

- A. Qatorni identifikatsiyalash uchun
- B. Ustunni identifikatsiyalash uchun
- C. Jadvalni identifikatsiyalash uchun
- D. Indeksflashgan ustunni identifikatsiyalash uchun

6. Relatsion ma'lumotlar bazasini yaratishda autoincrement tipi nima uchun ishlatiladi?

- A. Kalitli maydonlar tipi sifatida
- B. Sonli ustunlar tipi sifatida
- C. Matnli ustunlar tipi sifatida
- D. Sanali ustunlar tipi sifatida

7. Odatda qanday ustunlar ma'lumotlar bazasida indekslashgan bo'ladi?

- A. Ko'p qidiriladigan
- B. Qidirilmaydigan
- C. O'zgarmaydigan
- D. Sonli

8. Quyidagi komponentlarning qaysi biri yordamida Delphi tizimida ma'lumotlar bazasiga ulanish uchun foydalanish mumkin?

- A. ADOConnection
- B. ADOTable
- C. DataSource
- D. DBGrid

9. Ulangan ma'lumotlar bazasining ma'lum bir jadvalini ochish uchun quyidagi komponentlardan qaysi birini ishlatish mumkin?

- A. ADOTable
- B. DataSource
- C. DBGrid
- D. ADOConnection

10. Ulangan ma'lumotlar bazasida so'rovlarni amalga oshirish uchun quyidagi komponentlardan qaysi birini ishlatish mumkin?

- A. ADOQuery
- B. ADOTable
- C. DataSource
- D. DBGrid

11. Ma'lumotlar bazasi jadvalidan ma'lumotlarni ajratib olish uchun ADOTable komponentining qaysi xususiyatidan foydalanish kerak?

- A. Filter
- B. IndexFieldNames
- C. RecordCount
- D. Fields

12. Ma'lumotlar bazasi jadvalida nechta qator mavjudligini bilish uchun ADOTable komponentining qaysi xususiyatidan foydalanish kerak?

- A.RecordCount
- B.Filter
- C.IndexFieldNames
- D.Fields

13. Ma'lumotlar bazasi jadvalida joriy qator raqamini bilish uchun ADOTable komponentining qaysi xususiyatidan foydalanish kerak?

- A.RecNo
- B.IndexFieldNames
- C.RecordCount
- D.Fields

14. Ma'lumotlar bazasi jadvalida nechta ustun mavjudligini bilish uchun ADOTable komponentining qaysi xususiyatidan foydalanish kerak?

- A.FieldCount
- B.IndexFieldNames
- C.RecordCount
- D.Fields

15. Ma'lumotlar bazasi jadvalining ixtiyoriy ustuniga murojaat etish uchun ADOTable komponentining qaysi xususiyatidan foydalanish kerak?

- A.Fields
- B.FieldCount
- C.IndexFieldNames
- D.Field

16. Ma'lumotlar bazasi jadvalining oxiriga yangi qator qo'shish uchun ADOTable komponentining qaysi metodidan foydalanish kerak?

- A.Append
- B.Insert
- C.Edit
- D.Last

17. Ma'lumotlar bazasi jadvalining joriy pozitsiyasiga yangi qator qo'shish uchun ADOTable komponentining qaysi metodidan foydalanish kerak?

- A.Insert

B.Append

C.Edit

D.Last

18. Ma'lumotlar bazasi jadvalining joriy qatorini o'chirish uchun ADOTable komponentining qaysi metodidan foydalanish kerak?

A.Delete

B.Append

C.Insert

D.Edit

19. Ma'lumotlar bazasi jadvalining joriy qatorini tahrirlash uchun ADOTable komponentining qaysi metodidan foydalanish kerak?

A.Edit

B.Append

C.Insert

D.Delete

20. Ma'lumotlar bazasi jadvalining oxirgi qatoriga o'tish uchun ADOTable komponentining qaysi metodidan foydalanish kerak?

A.Last

B.Append

C.Edit

D.Delete

21. TADOTable komponentining LOCATE buyrug'i parametrlari va ularning ketma-ketligi to'g'ri ko'rsatilgan qatorni toping.

A.Qidirish ustunlari, qidirish qiymatlari, qidirish parametri

B.Qidirish qiymatlari, qidirish ustunlari, qidirish parametric

C.Qidirish parametri, qidirish ustunlari, qidirish qiymatlari

D.Qidirish ustunlari, qidirish parametri, qidirish qiymatlari

22. MS Office paketi dasturlari bilan ma'lumot almashganda (ishlaganda) qanday o'zgaruvchi tipidan foydalanish zarur?

A.Variant

B.Integer

C.File

D.Office

23. XLSApp:=CreateOLEObject('Excel.Application'); ushbu kod nima ishni bajaradi?

A.Excel dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtiradi

B.Word dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtiradi

C.Excel dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtirmaydi

D.Excel dasturi ishga tushuradi

24. XLSApp.visible:=true; //XLSApp o'zgaruvchi Excel dasturi bilan ishlash o'zgaruvchisi ushbu kod nima ishni bajaradi?

A.Excel dasturini ekranga ko'rsatadi

B.Excel dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtiradi

C.Word dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtiradi

D.Excel dasturi bilan ishlash ob'yektini yaratib, uni XLSApp o'zgaruvchiga o'zlashtiradi

25.XLSApp.WorkBooks[1].WorkSheets[2].Name:="Fevral";

//XLSApp o'zgaruvchi Excel dasturi bilan ishlash o'zgaruvchisi buyrug'i nima ishni bajaradi?

A.Birinchi kitobdagi ikkinchi listni nomini "Fevral" ga almashtiradi

B.Birinchi kitobdagi birinchi listni nomini "Fevral" ga almashtiradi

C.Ikkinchi kitobdagi birinchi listni nomini "Fevral" ga almashtiradi

D.Ikkinchi kitobdagi ikkinchi listni nomini "Fevral" ga almashtiradi

26.XLSApp.WorkBooks[1].WorkSheets[2].Cells[4,3]:="Fevral";

//XLSApp o'zgaruvchi Excel dasturi bilan ishlash o'zgaruvchisi buyrug'i nima ishni bajaradi?

A.Excel birinchi kitobidagi ikkinchi listning C4 yacheykasiga 'Fevral' qiymatini yozadi

B.Excel birinchi kitobidagi ikkinchi listning D3 yacheykasiga 'Fevral' qiymatini yozadi

C.Excel birinchi kitobidagi birinchi listning C4 yacheykasiga 'Fevral' qiymatini yozadi

D.Excel birinchi kitobidagi birinchi listning D3 yacheykasiga 'Fevral' qiymatini yozadi

27. Delphi dasturlash tilida Word matn muharriri bilan ishlashda kursorni pozitsiyalanishi muhim hisoblanadi! Ushbu pozitsiyalanishni to'g'ri amalga oshirish uchun WORD matn muharririning qaysi imkoniyatidan foydalanish zarur?

A."Вставка" -> "Закладка"

B."Вставка" -> "Перекрестная ссылка"

C. "Вставка" -> "Надпись"

D. "Вставка" -> "Таблица"

28. WordApp:=CreateOLEObject('Word.Application');

doc:=WordApp.documents.Open(ExtractFilePath(Application.ExeName)+'
Sh ablon.doc'); WordApp.documents.SaveAs
(ExtractFilePath(Application.ExeName)+'Fevral.doc'); //ushbu kodda
"WordApp" va "doc" o'zgaruvchilari qanday tipga ega?

A. Variant, Variant

B. Variant, Integer

C. Integer, Variant

D. Integer, Integer

29. WordApp:=CreateOLEObject('Word.Application');

doc:=WordApp.documents.Open(ExtractFilePath(Application.ExeName)+'
Sh ablon.doc'); WordApp.documents.SaveAs
(ExtractFilePath(Application.ExeName)+'Fevral.doc');
WordApp.Selection.GoTo(-1,unsigned,unsigned,'Oy');
WordApp.Selection.TypeText('Fevral'); //ushbu kod nima ishni bajaradi?

A. Fevral.doc hujjatini ochadi, "Oy" zakladkasiga o'tadi va kursor pozitsiyasida "Fevral" so'zini kiritadi

B. Fevral.doc hujjatini ochadi, "Oy" zakladkasiga o'tadi va kursor pozitsiyasidan chap tomon yo'nalishda "Fevral" so'zini kiritadi

C. Fevral.doc hujjatini ochadi, "Oy" zakladkasiga o'tadi va kursor pozitsiyasidan bir qator pastda "Fevral" so'zini kiritadi

D. Fevral.doc hujjatini ochadi, "Oy" zakladkasiga o'tadi va kursor pozitsiyasidan bir qator oldingi qatorda "Fevral" so'zini kiritadi

30. Delphi dasturlash tili orqali WORD matn muharriri bilan ishlash jarayonida tayyorlangan namuna faylingizni buzilish holatiga keltirmaslik uchun qanday imkoniyatlardan foydalanish mumkin?

A. Faylga faqat o'qish uchun flagini o'rnatish yoki hujjatni yangi nom bilan saqlash

B. Faylga faqat o'qish uchun flagini o'rnatish yoki hujjatni nomini o'zgartirish

C. Faylga faqat o'qish uchun flagini o'rnatish yoki hujjatni yangidan yaratish

D. Faylga tahrirlash flagini o'rnatish yoki hujjatni yangi nom bilan saqlash

31. Oqimlar nima uchun texnik ilovalarda qo'llaniladi?

A. Dasturda parallel hisoblashlarni amalga oshirish uchun

B. Dastur hajmini qisqartirish uchun

- C.Dastur sifatini oshirish uchun
- D.Dasturni tezroq yaratish uchun

32.Oqimlar texnik ilovalarda qo'llashda qaysi tamoyildan foydalaniladi?

- A.Doimiy qo'llash tamoyili
- B.Hech qachon qo'llamaslik tamoyili
- C.Yetarlicha keraklilik tamoyili
- D.Tezkorlik tamoyili

33. Oqimlarni Delphi tizimida yaratishda qaysi tipdagi ob'yektdan foydalaniladi?

- A.Thread Object
- B.DLL Object
- C.VCL application
- D.Data Module

34.Delphi tizimida oqim ob'yektining qaysi nomdagi funksiyasi asosiy hisoblanadi?

- A.initialization
- B.create
- C.execute
- D.destroy

35.Delphi tizimida oqim ob'yektini asosiy dasturdan ishga tushurish uchun qaysi buyruqdan foydalaniladi?

- A.create
- B.resume
- C.terminate
- D.destroy

36.Delphi tizimida oqim ob'yektini asosiy dasturdan to'xtatish uchun qaysi buyruqdan foydalaniladi?

- A.create
- B.resume
- C.terminate
- D.destroy

37.DLL kutubxonalarni Delphi tizimida yaratishda qaysi tipdagi ob'yektdan foydalaniladi?

- A.Thread Object
- B.DLL Wizard
- C.VCL application
- D.Data Module

38.DLL kutubxonalar kengaytmasi qanaday bo'ladi?

- A.thr
- B.dll
- C.vcl
- D.exe

39. DLL kutubxonalar kodida qanday kalitli so'z bilan boshlanadi?

- A.thread
- B.library
- C.project
- D.unit

40.Delphi tizimida buffer xotira bilan ishlash uchun qanday moduldan foydalanish zarur?

- A. Clibboard
- B. DLL
- C. Thread
- D. Variant

41.Delphi kodlashtirish tilida buffer xotira ob'yektdan foydalanish uchun qaysi kalitli so'zni uses bo'limiga yozish kerak?

- A.Clibrd
- B.DLL
- C.Thread
- D.ComObj

42.Clipboard ob'yektining SetTextBuf funksiyasini vazifasi qanday?

- A.Matnni buffer xotiraga nusxalash
- B.Matnni buffer xotiradan joylashtirish
- C.Oqimlarni ishga tushurish
- D.Kutubxona yuklash

43.Clipboard ob'yektining SetTextBuf funksiyasini qo'llashda matnni qanday funktsiya orqali OT tushunadigan tipga o'tkazish kerak?

- A.Pchar
- B.Inttostr
- C.FloattoStr
- D.DateToStr

44.Clipboard ob'yektining AsText funksiyasini vazifasi qanday?

- A.Matnni buffer xotiraga nusxalash
- B.Matnni buffer xotiradan joylashtirish
- C.Oqimlarni ishga tushurish
- D.Kutubxona yuklash

45.Clipboard ob'yektining Assign funksiyasini vazifasi qanday?

A.Matnni buffer xotiraga nusxalash

B.Matnni buffer xotiradan joylashtirish

C.TPersistent tipiga mos ob'yektni buffer xotiraga belgilash

D.Kutubxona yuklash

46.Clipboard ob'yektining Clear funksiyasini vazifasi qanday?

A.Matnni buffer xotiraga nusxalash

B.Matnni buffer xotiradan joylashtirish

C.TPersistent tipiga mos ob'yektni buffer xotiraga belgilash

D.Buffer xotira tarkibini tozalash

47.Delphi tizmida to'xtatish nuqtasini joylashtirish uchun qaysi funktsional tugmadan foydalaniladi?

A. F5

B. F6

C. F7

D. F8

48. Delphi tizmida kodni qadamba-qadam bajartirish uchun qaysi funktsional tugmadan foydalaniladi?

A. F5

B. F6

C. F7

D. F9

49.Delphi tizmida to'xtatish nuqtasini qo'llashda qo'shimcha Condition parametric nimani anglatadi?

A.To'xtash shartini kiritish

B.To'xtashgacha o'tishlar soni

C.Oqimni kiritish

D.Kutubxonani kiritish

50.Delphi tizmida to'xtatish nuqtasini qo'llashda qo'shimcha PassCount parametri nimani anglatadi?

A.To'xtash shartini kiritish

B.To'xtashgacha o'tishlar soni

C.Oqimni kiritish

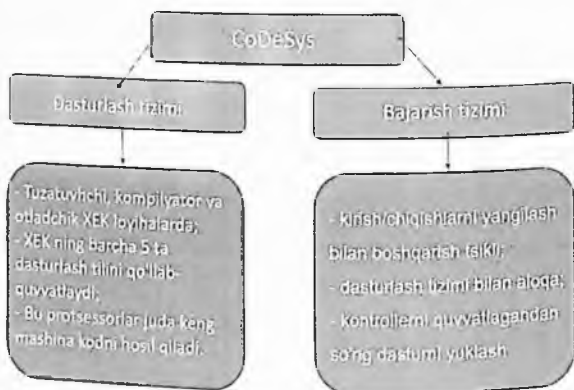
D.Kutubxonani kiritish

VI-MODUL. DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALARNI YARATISHNING BOSHQA TIZIMLARI.

§6.1. CodeSys tizimi bilan tanishish. CodeSys tizimida signallarni kiritish va chiqarish

CODESYS-CONtroller DEVELOPMENT SYStem so'zlaridan olingan bo'lib, "Sanoatni avtomatlashtirishda instrumental dasturiy ta'minot to'plami" degan ma'noni anglatadi.

Ushbu dastur dastlab 3S-Smart Software Solutions GmbH, 1.0 versiyasi 1994 yilda ishlab chiqilgan. 2012 yil noyabr oyida esa CodeSys nomiga o'zgartirilgan. Dasturiy ta'minotni Windows XP va undan yuqori versiyadagi OTlarga o'rnatilishi mumkin. 2019 yil 18 aprelda esa CodeSys ning oxirgi versiyasi V3.SP14 Patch dasturiy ta'minoti ishlab chiqildi. Ushbu dastur haqida codesys.com saytidan barcha ma'lumotni olishingiz mumkin

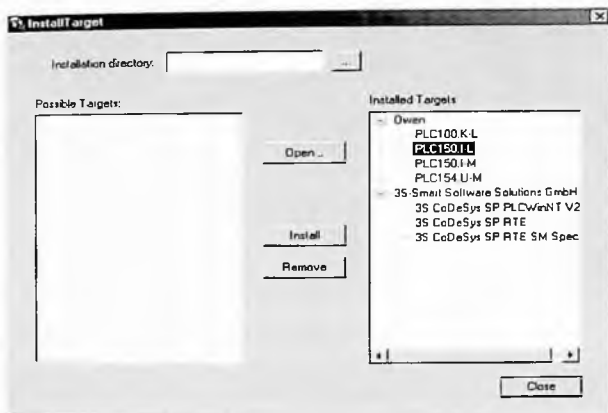


XEK-Xalqaro elektrotexnika komissiyasi

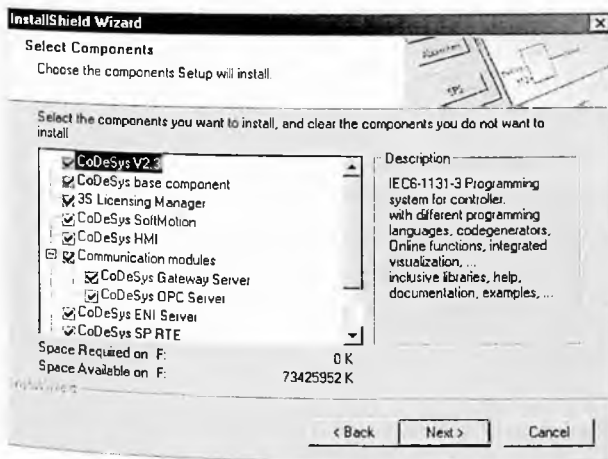
6.1.1-rasm. CodeSys tizimining umumiy tuzilishi.

CoDeSys ni o'rnatish

- CoDeSysni kompakt disk yoki rasmiy sayti orqali (www.owen.ru) o'rnatish mumkin.
- Maqsadli platforma fayllarini o'rnatish (Target Files)

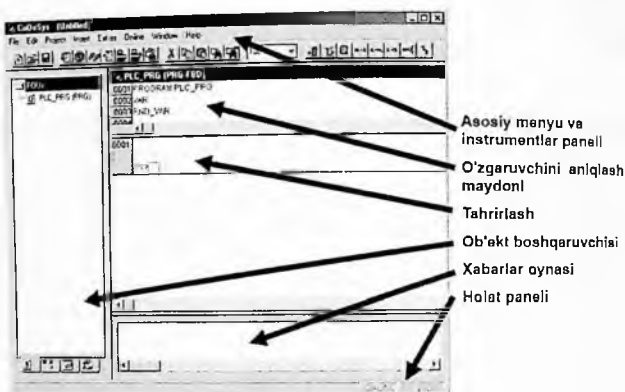


6.1.2-rasm. CodeSys tizimida target fayllarini o'rnatish.



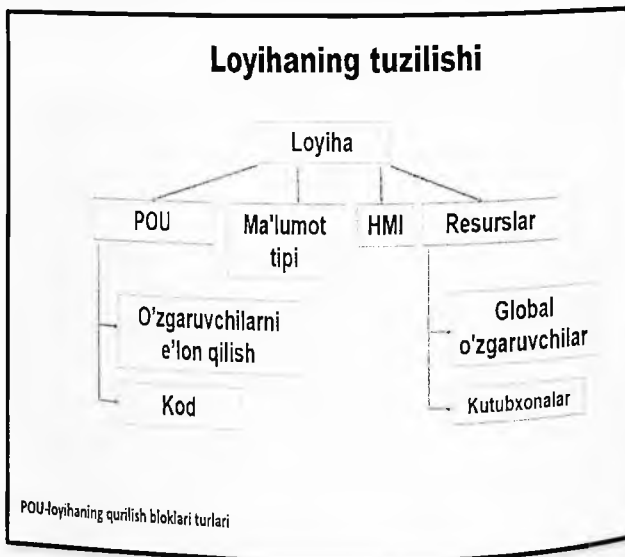
6.1.3-rasm. CodeSys tizimini o'rnatish oynasi.

Birinchi marta CoDeSys ishga tushirish



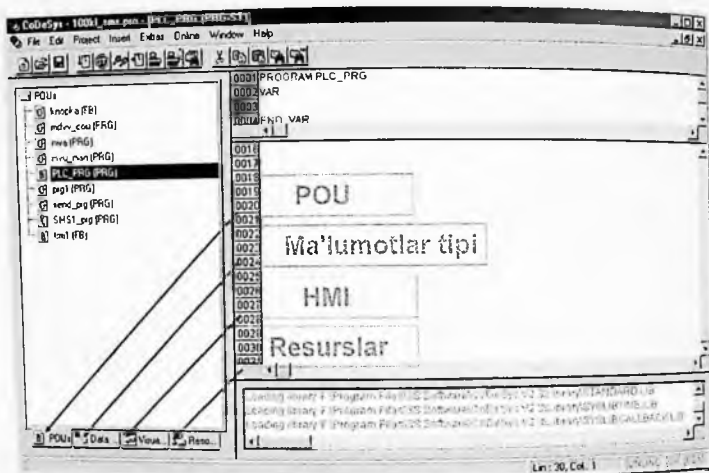
6.1.4-rasm. CodeSys tizimining asosiy oynasi.

CoDeSysda ko'p ishlatiladigan variantlar



6.1.5-rasm. CodeSys tizimini sozlash oynasi.

Loyihaning tuzilishi



6.1.8-rasm. CodeSys tizimida yangi loyiha oynasi.

CoDeSysda loyiha qanday?

- bitta faylda saqlanadi (name.pro);
- dasturiy ta'minot komponentlarini o'z ichiga oladi (POU), vizuallashtirish, resurslar va boshqalar;
- dasturni bajarish POU bilan boshlanadi PLC_PRG(funktsiya analogi main);
- davriy ravishda bajarilgan (tsikli).

POU nima?

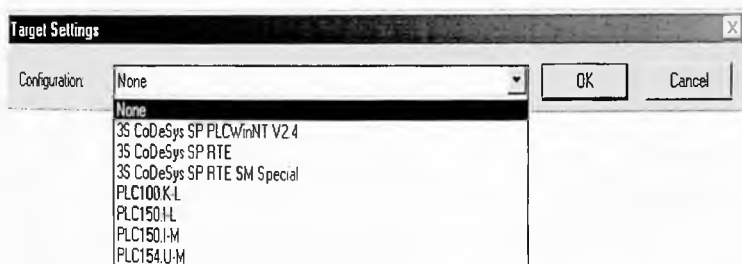
POU (Program organisation unit) –bu dasturiy modul yoki dasturni tashkil qilish bo'limi POU PLC_PRG tizim bajarilishi sababli bilvosita ishlatiladi XEK standarti – POU ning 3 turini belgilaydi

- Dasturlar <PROGRAM>
- Funktsional blok <FUNCTION_BLOCK>
- Funktsiyalar <FUNCTION>

Birinci loyiha

(Инкремент-oshirish o'zgaruvchini)

- <File / New>
- Target Settings
- Создание главной программы PLC_PRG
- Автоматическое объявление
- <Online / Simulation>
- <Online / Login>
- <Online / Start>



6.1.9-rasm. CodeSys tizimida loyihani qaysi mikrokotrollerga mosligini tanlash oynasi.

Standart ma'lumotlar tipini

6.1.1-jadval

Kalit so'z	Diapazon	Misol
BOOL	0,1	FALSE, TRUE, 0,1
SINT, INT, DINT	-128 .. 127, -32768 .. 32767,- 2147483648 .. 2147483647	0, 24453-38099887
USINT, UINT, UDINT	0 .. 255, 0 .. 65535, 0 .. 4294967295	200, 47453 138099887
BYTE, WORD, DWORD	0 .. 255, 0 .. 65535, 0 .. 4294967295	8450 16#2102
REAL, LREAL	-1.2×10^{-38} .. 3.4×10^{38} -2.3x 10^{-308} .. 1.7×10^{308}	1.34996 2.8377E-15
TIME, TOD,	0 ms .. 1193h2m47s295ms	T#1d8h12m8s125ms

DATE, DT	00:00:00 .. 23:59:59 01.01.1970 до. 06.02.2106	TOD#12:34:17 D#2001-03-15 DT#2001-03-15- 12:17:03
STRING	1 .. 255 СИМВОЛОВ	`Emergency Stop`

O'zgaruvchilarni e'lon qilishning 3 ta metodi mavjud.

1. matnli, jadvalli va avtomatik
2. Lokal (1-FB uchun) yoki Global (barcha FB) (FB-Funksional blok)
3. Saqlanuvchi va doimiy o'zgaruvchilar

Identifikator sintaksisi

Harflar va raqamlar;

- Harf bilan boshlash kerak;
- Faqatgina pastki chiziqlar;
- Bo'sh joy qo'yish mumkin emas;
- Siz XEK zahiralangan so'zlar va operatorlardan foydalana

olmaysiz;

- Registrning farqi yo'q;

Misollar

- Otto, otto, OTTO;
- Valve1;
- a_long_name;

Online rejimining asosiy buyruqlari

- <Online / Simulation Mode >
- <Online / Login [Alt+F8] / Logout [Ctrl+F8]>
- <Online / Start [F5]>
- <Online / Stop [Shift+F8]>
- <Online / Single Cycle>
- <Online / Breakpoint [F9]>
- <Online / Write Values [Ctrl+F7]>
- <Online / Force Values [F7]>
- <Online / Release Force [Shift+F7]>

6.1.10-rasm. CodeSys tizimida online rejimi buyruqlari. Dasturiy platformani ishga tushirish (OWEN da DMK)

- Tizimni ishga tushirish vaqti;
- Simulyatsiya rejimini o'chiring <Online / Simulation Mode>;
- Aloqa parametrlarini sozlash <Online / Communication

Parameter...>;

DMK bu - dasturlanadigan mantiqiy controller yoki qurilmalar!

XEK tillari

- Ko'rsatmalar ro'yxati / Список инструкций (Instruction List);
- Tuzilmaviy matn / Структурированный текст (Structured

Text);

• Funktsiyalashgan diagramma bloki tili / Язык функциональных блокковых диаграмм (Function Block Diagram);

• Releyli diagramma tili / Язык релейных диаграмм (Ladder Diagram);

• Ketma-ket funktsiyalanadigan sxema tili (Sequential Function Chart).

(IL) ko'rsatmalar ro'yxati tilining xususiyatlari

- Matnli til;
- Assembler tiliga o'xshash;
- Katta bo'lmagan dasturlarni osongina o'qish
- Tarkibiy dasturlashni qo'llab-quvvatlamaydi.

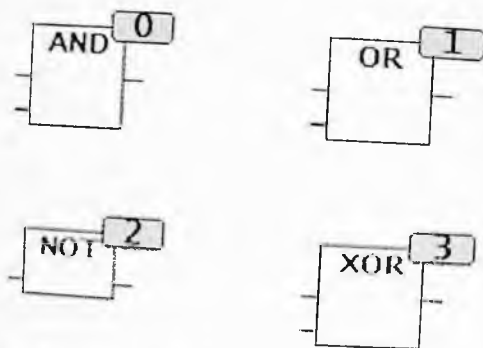
Tuzilmali matn (ST) tilining xususiyatlari

- Matnli til;
- Yuqori darajali til;
- Paskal tiliga o'xshash;
- Tsikl va shartni ishlatish uchun qo'llaniladigan eng yaxshi til (IF, WHILE, FOR, CASE).

Funktsiyalashgan diagramma bloki tili (FBD) xususiyatlari

- Grafik til;
- Dastur bir nechta sxemalardan tashkil topadi;
- Oson o'qish imkoniyati;

- Har bir sxema bloklardan va operandlardan tashkil topadi.
 - Uzluksiz funktsional sxema (CFC) tili xususiyatlari
 - FBD ga o'xshash
 - Bloklar va ulagichlar ochiq joylashtiriladi;
 - Tsikllarga ruxsat etiladi va bog'lanishlar juda oson;
- CodeSys tizimida ko'pgina loyihalar quyidagi mantiqiy elementlardan keng foydalaniladi.



6.1.11-rasm. CodeSys tizimida online rejimi buyruqlari

Ular har qanday diskret algoritmlarning asosini tashkil qiladi. Ushbu mantiqiy elementlardan foydalangan holda ko'plab sxemalarni avtomatlashtirish mumkin. Bunday yechimlar sanoatning muhim sohalarida ommaviy ravishda ishlatilmoqda. Tasavvur qiling katta hajmga ega avtomatlashtirilgan shkaflar o'rnini bir yoki bir nechta kichik DMK bilan almashtirishingiz mumkin.

Siz "Fayl" menyusi orqali yangi loyiha yaratasiz. Shundan so'ng kerakli elementini tanlash imkonini yaratiladi. Foydalanmoqchi bo'lgan DMK ni tanlab unda so'ng "Barcha platforma sozlamalari" oynasiga murojaat etamiz.

Qo'shimcha ma'lumot yuklamoqchi bo'lsangiz kerakli faylni oching va ma'lumotlarni xotiradan yangilang. Ushbu holatda **PLK150-I-M** ni tanlaymiz. Boshqa biron bir kontrollerni tanlash uchun **OBEH PLC1xx**, masalan **PLC110** yoki

160 tanlash imkoni mavjud. CFC funktsiya bloki tili bilan ishlashmiz. Asosan shu tilni tanlab **PLC_PRG** da asosiy dastur kodini yozamiz.

Shundan so'ng "DMK configuration" ni tanlang. Dasturda biz **bitta tsikl, bitta takrorlash** harakatlarni tasvirlaymiz. Loyihani DMK ga yuklagandan so'ng, doimiy ravishda nazoratni siz yaratgan algoritm amalga oshiradi. Buning uchun "Konfiguratsiyalar" oynasida "PLC_150_I" tanlash mumkin. Sizning DMK modelingizga boshqa bo'lsa, siz "Module Settings" oynasiga o'tib kerakli modelni tanlashingiz mumkin(6.1.12-rasm).

PLC 150 I

- ☐ Discrete input 6 bit[FIX]
- ☐ Discrete output 4 bit[FIX]
- ☐ Special output[FIX]
- ☐ Unifed signal sensor[SLOT]
- ☐ Unifed signal sensor[SLOT]
- ☐ Unifed signal sensor[SLOT]
- ☐ Unifed signal sensor[SLOT]
- ☐ Analog output[FIX]
- ☐ Analog output[FIX]

Настройка **Параметры модуля**

Индекс	Имя	Значение
1	MinCycleLength ms	10
2	MaxCycleLength ms	1000

6.1.12-rasm. Tanlangan controller modeliga mos kiritish – chiqarish portlarini sozlash oynasi.

Odatda **PLC1xx** tsikl vaqti **1ms**, ya'ni DMK ning har millisekundida algoritm qayta yuklanadi va bajariladi. Shu daqiqada ishni yakunlashi ham kerak. Buni bajarish uchun vaqtzni to'g'ri tashkil etish mumkin. Yanada jiddiy vazifalarda ba'zi bir chegaralarni belgilash uchun **DMK** ishlantiladi. Ya'ni, tsikl vaqtini aytaylik 10 ms qilib belgilang. **PLC1xx** da qo'yilgan vazifalarning 80% uchun bu yetarli bo'ladi. Loyihani yaratganingizdan so'ng "Qiymat" maydoniga **1ms** o'rniga **tsikl** vaqtini 10 ms kiriting. Bu bizning dasturimizni qayta ishlaydigan tsikl vaqti.

Diskret mantiq bilan shug'ullanish uchun bizga **DMK** ning bir nechta **kiritish** va **chiqarish** portlari zarur bo'ladi. Buning uchun siz DMK konfiguratsiya elementlarini ochasiz va oltitasini

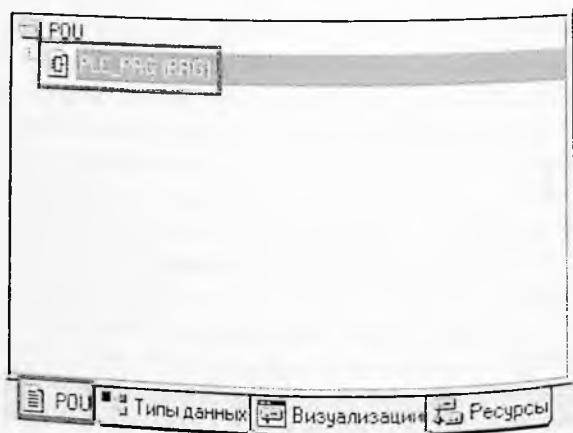
diskret kirishlar sifatida belgilaysiz. Bu x_1 , x_2 , x_3 va boshqa o'zgaruvchilar bo'lsin. Eslatib o'taman, ma'lum bir kiritish yoki chiqarish uchun o'zgaruvchining nomini aniqlash uchun siz kerakli satrda "AT" yozuvidagi DMK ni ikki marta sichqoncha bilan tanlash kerak. Shunda sizda 6 ta diskret kiritish hosil bo'ladi. Analog chiqish bo'yicha biz to'rtta diskret chiqish o'zgaruvchilarni belgilaymiz.

Bu y_1 , y_2 , y_3 va y_4 o'zgaruvchilar bo'lsin.

Biz uchun quyidagilar yetarli darajada aniq bo'lishi kerak.

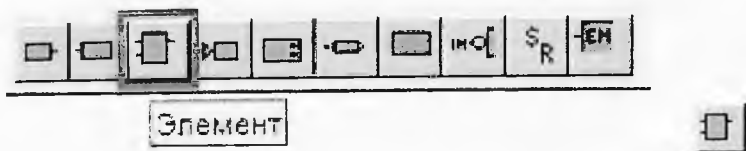
- Target fayl haqida ma'lumot;
- Asosiy dasturni yaratish, loyihani texnik topshirig'i va dasturlash tili tanlanadi;
- DMK siklining vaqtini belgilash;
- Zaruriy kiritish va chiqarish o'zgaruvchilarini e'lon qilish yoki ularni tanlash;
- Loyihani kompyuterda saqlash.

Keling, asosiy dasturimizga qaytaylik. Eslatib o'taman, u "POU" oynasida joylashadi. DMK ni tanlab unga sichqoncha chap tugmasi bilan ikki marotaba tez murojaat eting va "PLC_PRG" dasturini yarating (6.1.13-rasm).



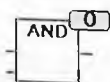
6.1.13-rasm. Asosiy loyiha dastur kodiga oynasi.

Algoritmimizda u yoki boshqa mantiqiy operatsiyani o'rnatish uchun biz davom etamiz. Ekranning yuqori qismida tezkor kirish panelida "Element" tugmachasi mavjud () (6.1.14-rasm). Biz qo'shgan blok "VA" mantiqiy elementi bo'ladi.



6.1.14-rasm. Loyiha kodiga yangi mantiqiy element qo'shish.

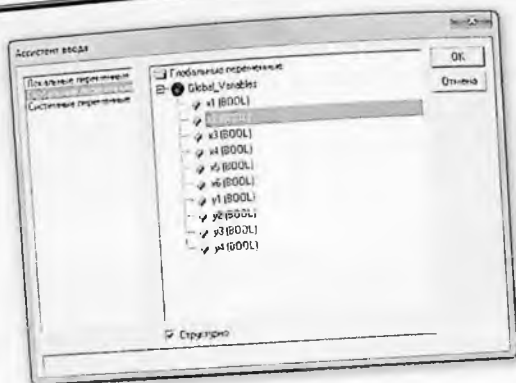
Bu kalit "AND" so'zida ko'rsatilgan (6.1.15-rasm). Shunga o'xshash elementlar blokini kontekst menyusi orqali ish maydonida ishlatish imkoni mavjud.



Элемент	Ctrl+B
Вход	Ctrl+I
Выход	Ctrl+U
Переход	Ctrl+J
Метка	Ctrl+L
Возврат	Ctrl+R
Комментарий	Ctrl+K
Выход блока	Ctrl+A

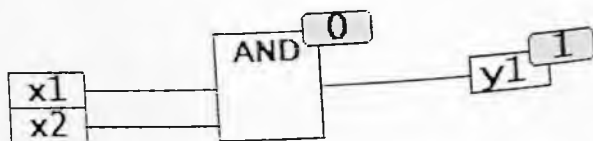
6.1.15-rasm. Yangi elementni diagramma blokida joylashtirish.

"BOOL" turidagi o'zgaruvchilar "AND" elementi kiritiladi. Faqat bunday o'zgaruvchilar diskret kiritishda ishlatiladi. "Kirish-Bxo4" tugmasi yordamida biz dasturimizda o'zgaruvchining qiymatini chaqiramiz. Siz buni klaviaturangizdagi "F2" tugmachasi orqali amalga oshirishingiz mumkin. Paydo bo'lgan oynada global ro'zgaruvchilar ro'yxatini tanlang va unda kerakli o'zgaruvchuni topamiz x1 va x2 (6.1.16-rasm).



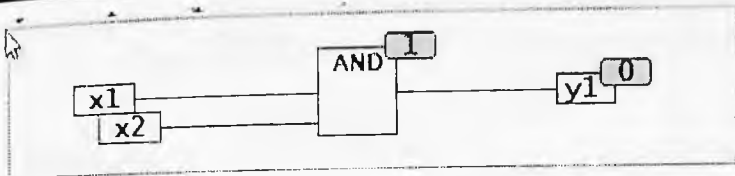
6.1.16-rasm. Yangi elementning kirish qiymatlariga mos o'zgaruvchilarni belgilash oynasi.

Mantiqiy chiqarish operatsiyada (blok AND) TRUE yoki FALSE signallari paydo bo'ladi, ya'ni, BOOL tipining qiymati bo'ladi. Keyin AND elementi va chiqarish y1 o'zgaruvchisining aloqa liniyasini bog'laymiz.



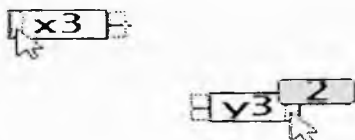
6.1.17-rasm. Yangi elementning kiritish yoki chiqish qiymatlariga mos o'zgaruvchilarni ulanganlik ko'rinishi.

CoDeSys bilan birinchi marotaba ishlashda sizda ba'zi qiyinchiliklar bo'lishi mumkin. Ishlash davrida ba'zi dastur elementlarining pozitsiyasi o'zgaradi. Keling, bu holatni batafsilroq ko'rib chiqamiz. Agar siz DMK ning biron-bir elementini dastur blokidan ikkinchi qismiga ko'chirmoqchi bo'lsangiz kerakli elementlarni tanlab joylashtiring (6.1.17-rasm).



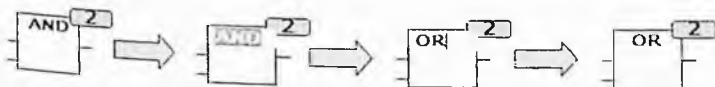
6.1.18-rasm. Loyihada mantiqiy element joylashuvini o'zgartirish.

Agar siz shu tarzda bitta o'zgaruvchining o'rnini o'zgartirishingiz kerak bo'lsa, unda quyidagilarni e'tiborga olish kerak (6.1.18-rasm): kirishlar elementning chap tomoniga suriladi, chunki u x3 sifatida ko'rsatilgan. Siz elementlarni o'ng tomonidagi DMK ga "birlashtirib" olib borasiz, masalan, y11 o'zgaruvchisi uchun 6.1.19-rasmda ko'rsatilgandek.



6.1.19-rasm. Kirish elementini joylashtirish.

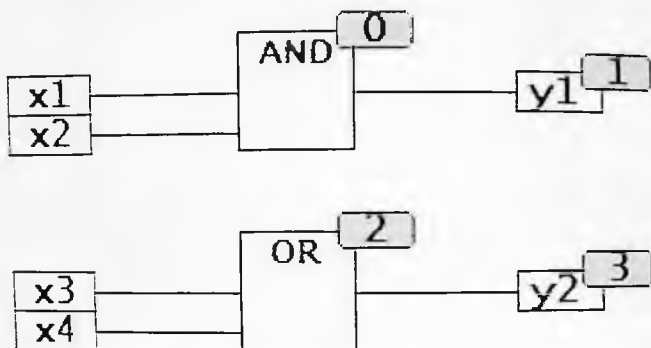
Keling, bizning loyihamizga yana qanday mantiqiy operatsiyalar qo'shilishi mumkinligini ko'rib chiqamiz. Agar siz "AND" o'rniga "OR" bilan ishlashingiz kerak bo'lsa buni siz quyidagicha amalga oshirasiz. Siz yangi element qo'shasiz, DMK sarlavhasini tanlang va yangi "OR" nomni kiritib so'ng "Enter" tumasini bosib (6.1.20-rasm).



6.1.20-rasm. AND mantiqiy elementni Or mantiqiy elementga aylantirish jarayoni.

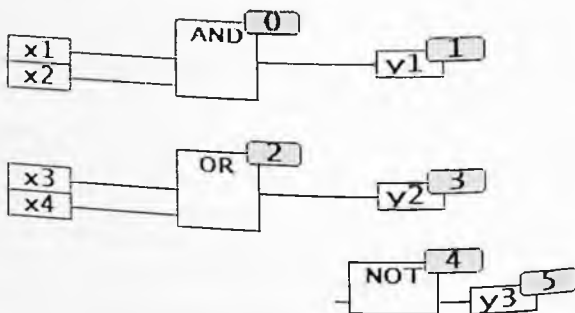
"AND" mantiqiy elementini "OR" elementiga almashtirish tamoyili aynan bir xil. Biz elementning kirish nuqtalarida x3 va x4 o'zgaruvchilarini belgilaymiz. Keyin biz aloqa liniyalarini "OR"

blokiga birlashtiramiz. Ushbu blok natijasini biz ikkinchi diskret chiqarishga uzatamiz, ya'ni y2 o'zgaruvchisiga.

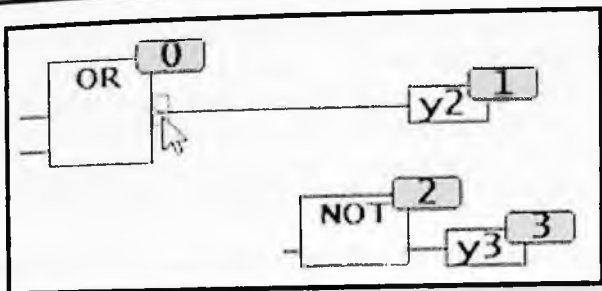


6.1.21-rasm. "OR" mantiqiy elementiga kirish va chiqish o'zgaruvchilarini ulanish sxemasi

Endi mantiqiy "YO'Q-NOT" elementi bilan ishlashni ko'rib o'tsak. "Element" tugmachasi orqali kontekst menyusiga o'ting. Bu orqali yangi blok yaratiladi va uning o'rniga yozuvlar va unda NOT so'zining kaliti ko'rsatiladi. NOT operatsiyasining chiqish qismini o'zgaruvch y3 ulang (6.1.22-rasm).

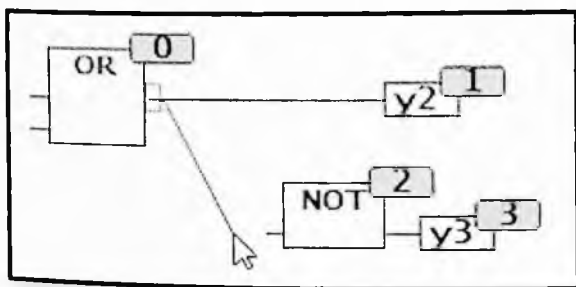


6.1.21-rasm. NOT elementi chiqish qismiga o'zgaruvchiga ulash sxemasi



6.1.23-rasm. OR mantiqiy elementi chiqish qiymati sifatida o'zgaruvchini ulash.

Shu blokning chiqishidan yoki bir xil o'zgaruvchidan, siz shu signalni bir yoki bir nechta qabul qiluvchilariga olib kelishingiz mumkin. Mantiqiy elementni tanlab, ikkinchi aloqa liniyasini "NOT" blokining kiritishiga ulang (6.1.23-rasm). Xuddi shu tarzda uchinchi, to'rtinchi va undan ham ko'proq elementlarni qo'shish imkoni mavjud.



6.1.24-rasm. OR mantiqiy elementi chiqish qiymati sifatida o'zgaruvchini ulash.

Asosiy tayanch so'z va iboralar

CodeSys, target fayl, loyiha, DMK, POU, ma'lumotlar tipi, identifikator, OWEN, Instruction List, Structured Text, Function Block Diagram, Ladder Diagram, Sequential Function Chart, mantiqiy element.

Takrorlash va mustaqil ishlash uchun savollar

1. CodeSys o'zi nima?
2. CodeSys tizmini ishlash sxemasini nimaledan iborat?
3. CodeSys tizimida loyiha odatda nimalardan iborat bo'ladi?
4. CodeSys standart ma'lumotlar tipini nimalardan iborat?
5. CodeSys tizimida o'zgaruvchilarni qanday e'lon qilinadi?

§6.2. CodeSys tizimida texnik ilovalarda mantiqiy amallarni qo'llash.

Axborotni qayta ishlash, raqamli aloqa va avtomatik boshqaruv tizimlarining ishlash tamoyili bir necha bazaviy operatsiyalarning takrorlanishi yoki ularning kombinasiyalariga asoslangan bo'ladi. Buni amalga oshirish uchun "VA", "EMAS", "YOKI" mantiqiy elementlari, shuningdek, triggerlardan foydalaniladi.

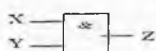
Mantiqiy element (ME) deb eng sodda mantiqiy operatsiyalarni bajaruvchi elektron qurilmaga aytiladi.

Mantiqiy elementlar tarkibiy tuzilishi hamda yasash texnologiyasiga qarab farqlanadi. Shu bilan birga ular **asosiy parametrlari**: qayta ulanish tezligi, signalning kechikish vaqti, ishchi harorat oralig'i, kirishni birlashtirish koeffitsiyenti va chiqishdagi yuklama qobiliyati bilan ham bir-biridan farq qiladi.

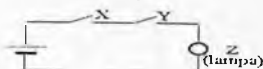
Mantiqiy elementlar mantiqiy ifodalarni bajarishga mo'ljallangan bo'lib, barcha **arifmetik va mantiqiy amallarni** ular asosidagi qurilmalar yordamida amalga oshiriladi. Quyida hisoblash mashinalarida qo'llaniladigan asosiy mantiqiy elementlar va ularning ishlash tamoyillari keltirilgan.

«VA» - mantiqiy ko'paytirish, «konyunksiya» elementi.

Sxematik belgilanishi



Ishlash prinsipi



Mantiqiy ko'paytirish

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

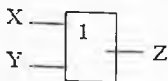
6.2.1-rasm. VA mantiqiy elementning sxematik, zanjirdagi va sonlardagi ifodasi.

X va Y kirishlarga bir vaqtda "1" signali berilsa (ya'ni ulagichlar bir vaqtda ulansa), Z chiqishda "1" signali hosil bo'ladi (ya'ni lampa yorishadi). Kirishlardan birortasiga yoki bir vaqtda ikkalasiga «0» signali berilsa (ya'ni ulagichlardan biri yoki bir vaqtda ikkalasi ulanmagan holda bo'lsa), chiqishda «0» signali hosil bo'ladi (ya'ni lampa o'chgan holda bo'ladi). «VA» elementi mantiqiy funksiya sifatida $Z = X \& Y$, hamda $Z = X*Y$ yoki

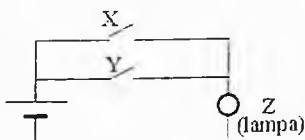
$Z = X \odot Y$ ko'rinishlardan birortasida tasvirlanishi mumkin.

«YOKI» - mantiqiy qo'shish, «dizyunksiya» elementi

Sxematik belgilanishi



Ishlash prinsipi



Mantiqiy qo'shish

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

6.2.2-rasm. YOKI mantiqiy elementning sxematik, zanjirdagi va sonlardagi ifodasi

X va Y kirishlarga bir vaqtda "0" signali berilsa (ya'ni ulagichlar bir vaqtda ulanmagan holda bo'lsa), Z chiqishda "0" signali hosil bo'ladi (ya'ni lampa o'chgan holda bo'ladi). Kirishlardan birortasiga yoki bir vaqtda ikkalasiga «1» signali berilsa (ya'ni ulagichlardan biri yoki bir vaqtda ikkalasi ulansa), chiqishda «1» signali hosil bo'ladi (ya'ni lampa yorishadi).

«YOKI» elementi mantiqiy funksiya sifatida $Z = X+Y$ hamda $Z = X \vee Y$ ko'rinishlarida tasvirlanadi.

«INKOR» - mantiqiy inkor qilish («EMAS») elementi «INKOR» elementining chiqishidagi son uning kirishidagi songa nisbatan teskari qiymatga ega bo'ladi.

«INKOR» elementi mantiqiy funksiya sifatida $Y=\bar{X}$ ko'rinishda tasvirlanadi.

Sxematik belgilanishi



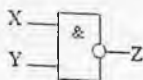
Mantiqiy inkor

X	Y
0	1
1	0

6.2.4-rasm. INKOR mantiqiy elementning sxematik, zanjirdagi va sonlardagi ifodasi

«VA - INKOR» - mantiqiy ko'paytirishning inkori elementi

Sxematik belgilanishi



Mantiqiy funksiyasi

$$Z = \overline{X \& Y}, \quad Z = \overline{X * Y}$$

Ishlash jadvali

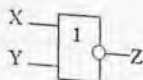
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

6.2.3-rasm. VA-INKOR mantiqiy elementning sxematik, zanjirdagi va sonlardagi ifodasi

X va Y kirishlarga bir vaqtda "1" signali berilsa, Z chiqishda "0" signali hosil bo'ladi. Kirishlardan birortasiga yoki bir vaqtda ikkalasiga «0» signali berilsa, chiqishda «1» signali hosil bo'ladi.

«YOKI - INKOR» - mantiqiy qo'shishning inkori elementi

Sxematik belgilanishi



Mantiqiy funksiyasi

$$Z = \overline{X \vee Y},$$

Ishlash jadvali

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

6.2.5-rasm. YOKI-INKOR mantiqiy elementning sxematik, zanjirdagi va sonlardagi ifodasi.

X va Y kirishlar bir vaqtda "0" signali berilsa, Z chiqishda "1" signali hosil bo'ladi. Kirishlardan birortasiga yoki bir vaqtda ikkalasiga «1» signali berilsa, chiqishda «0» signali hosil bo'ladi.

Asosiy tayanch so'z va iboralar

Mantiqiy elementlar, konyunksiya, dizyunktisya, VA mantiqiy elementi, YOKI mantiqiy elementi, inversiya, INKOR mantiqiy elementi.

Takrorlash va mustaqil ishlash uchun savollar

- 1.CodeSys o'zi nima?
- 2.CodeSys tizmini ishlash sxemasini nimalaadan iborat?
- 3.CodeSys tizimida loyiha odatda nimalardan iborat bo'ladi?
- 4.CodeSys standart ma'lumotlar tipini nimalardan iborat?
- 5.CodeSys tizimida o'zgaruvchilarni qanday e'lon qilinadi?

§6.3. Java dasturlash tili bilan tanishish

Java dasturlash tili – eng ommalashgan dasturlash tillaridan biri bo'lib, unda korporativ darajadagi mahsulotlarni(dasturlarni) yaratish mumkin. Bu dasturlash tili Oak dasturlash tili asosida paydo bo'ldi. Oak dasturlash tili 90-yillarning boshida Sun Microsystems tomonidan platformaga (Operatsion tizimga) bog'liq bo'lmagan holda ishlovchi yangi avlod aqlli qurilmalarini yaratishni maqsad qilib harakat

boshlagan edi. Bunga erishish uchun Sun hodimlari C++ ni ishlatishni rejalashtirdilar, lekin ba'zi sabablarga ko'ra bu fikridan voz kechishdi.

Oak muvofaqiyatsiz chiqdi va 1995-yilda Sun uning nomini Java ga almashtirdi va uni WWW rivojlanishiga hizmat qilishi uchun ma'lum o'zgarishlar qilishdi.

Java Ob'yektga Yo'naltirilgan Dasturlash (OOP – object oriented programming) tili va u C++ ga ancha o'xshash. Eng ko'p yo'l qo'yiladigan xatolarga sabab bo'luvchi qismlari olib tashlanib, Java dasturlash tili ancha soddalashtirildi.

Yuqoridagi maxsus atamalarining har biri James Gosling va Henry McGilton tomonidan yozilgan "Java tili muhiti" (The Java Language Environment) texnik tavsifnomasida yoritib berilgan.

Java Dasturlash tili

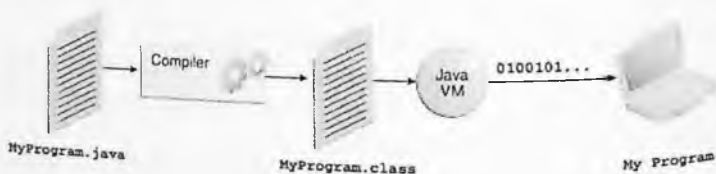
Java dasturlash tili yuqori darajadagi til bo'lib, quyidagi maxsus atamalarining barchasi bilan harakterlanadi:

Sodda	Arxitekturasi neytral
Obyektga yo'naltirilgan	Portativ, ixcham (portable)
Tarqalgan	Yuqori samaradorlik
Ko'pyo'nalishli (multithreaded)	Barqaror
Dinamik	Ishonchli

6.3.1-rasm. Java dasturlash tili umumiy tavsifi.

Java Dasturlash tili

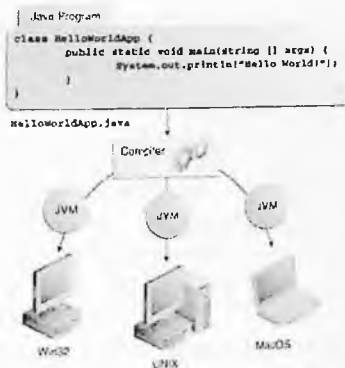
Java dasturlash tilida, barcha dastlabki kodlar avval .java kengaytma bilan tugaydigan ochiq tekst faylda yoziladi. Anashu asosiy kod fayllar javac kompilyator orqali .class faylga o'tkaziladi. .class file sizning protsessoringizga tabiiy bo'lgan kodni o'z ichiga olmaydi; aksincha, u Java Virtual Mashinasining¹ (Java VM) mashina tili **bytecode** (baytkod) lardan iborat bo'ladi. Keyin javani ishga tushiruvchi qurilma sizning ilova-dasturingizni Java virtual mashinasida ishga tushiradi.



6.3.2-rasm. Dasturiy ta'minot ishlab chiqish jarayonining umumiy ko'rinishi

Java VM si ko'p operatsion tizimlarda mavjud, shuningdek .class fayllari ham Microsoft Windows, Solaris operatsion tizimi (the Solaris™ Operating System), Linux yoki Mac operatsion tizimlarida ishlay oladi. "Java SE HotSpot at a Glance" kabi ba'zi virtual mashinalar sizning ilova-dasturingizni unumdorligini oshirish uchun ishlash vaqtida qo'shimcha faoliyatlarni amalga oshiradi. Bu o'z ichiga, unumdorlikka halaqit beruvchi kodlarni topish va kodning tez-tez ishlatiladigan qismini mashina kodiga kompilyatsiya qilish kabilarni o'z ichiga oladi.

"Java Virtual Mashinasi" va "JVM" atamaları Java platformasi uchun Virtual Mashina ma'nosini anglatadi.



6.3.3-rasm. Java VM orqali, bir xil dastur ko'plab platformalarda ishlay olish sxemasi.

Java Platformasi

Platform bu – uskunaviy yoki dasturiy ta'minot muhiti bo'lib, unda dastur ishga tushiriladi. Biz yuqorida Microsoft Windows, Linux, Solaris OS va Mac OS kabi eng mashhur platformalardan ba'zilarini eslab o'tdik.

Ko'pchilik platformalar operatsion tizim va foydalaniladigan uskunaning kombinatsiyasi sifatida ta'riflanadi. Java platformasi

boshqa ko'pchilik platformalardan o'zining faqat dasturiy ta'minotdan iborat platformaligi bilan ajralib turadi, shuningdek boshqa qurilmaviy platformalarda ishlay oladi.

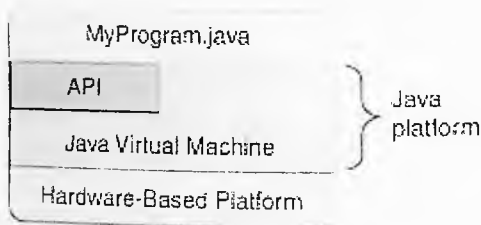
Java platformasi ikkita tarkibiy qismdan iborat:

✓ Java Virtual mashinasi

✓ Java - Dasturlash Ilova Interfeysi (Application Programming Interface (API))

Siz yuqorida Java Virtual Mashinasi bilan tanishib chiqdingiz; u Java platformasi uchun asos va turli qurilmaviy platformalarga ko'chib o'ta oladi.

"API" ko'plab qulayliklarni ta'minlab beradigan tayyor dasturiy komponentlar to'plami. U bir-biriga bog'liq sinf (classes) va interfeyslar kutubxonasiga birlashtirilgan va bu kutubxonalar paketlar (to'plamlar) sifatida tanilgan.



6.3.4-rasm. "API" va Java Virtual Mashinasi dasturni asos qurilmadan ajratilgan sxemasi.

Platformaga bog'liq bo'lmagan muhit sifatida, Java platformasi hususiy kodga nisbatan biroz sekinroq bo'lishi mumkin. Biroq, kompilyator va virtual mashina texnologiyasidagi yuksalishlar, ko'chirishga xavf solmagan holda unumdorlikni hususiy kodga yaqinlashtiradi. Java yuqori darajali, mustahkam, xavfsiz va ob'yektga yo'naltirilgan til. Platforma: dastur bajarila oladigan ixtiyoriy apparat yoki dasturiy muhit platformadir. Javaning ham o'zini mahsus bajarilish muhiti - platformasi mavjud (JRE - Java Runtime Environment).

Javadan qayerda foydalaniladi?

"Sun" firmasi ma'lumotiga ko'ra, 3 mlrd atrofidagi qurilmalar Javani ishlatadi (eski ma'lumot, hozir undan ko'p). Mana ulardan ba'zilari: -Shaxsiy kompyuter dasturlari (Desktop Applications) – acrobat reader, media-pleyer, antiviruslar va h.k. - Web-dasturlar;

-Korxona-tashkilotlar dasturlari (Enterprise Applications) – bank yoki ishlab chiqarishga oid dasturlar;

-Mobil dasturlar;

-Smart kartalar;

-Robotlar;

-O'yinlar.

Asosiy konsepsiyalari:

Java quyidagi 5 maqsad uchun qurilgan, u shunday til bo'lishi kerakki:

✓Oddiy, ob'yektga mo'ljallangan, taqsimlangan va o'rganishga oson bo'lsin;

✓Mustahkam va xavsiz bo'lsin;

✓Qaysidir qurilma platformasidan yoki uning arxitekturasidan mustaqil bo'lsin(ya'ni qaysidir platformaga tobe bo'lmasin);

✓Juda samarali bo'lsin;

✓Dasturlash tili uchun tarjimon(interpreter) yozish mumkin bo'lsin.

Javada qanaqa dasturlar yoziladi:

Asosan 4 tipdagi:

Standalone applications – Linux, Mac yoki Windowsga o'rnatib, kundalik ishlatadigan dasturlarimiz: mp3 pleyer, ofis, antivirus kabilar. Ular AWT, Swing yoki JavaFX texnologiyalari orqali tuziladi.

Web Applications – tarmoq orqali ishlovchi ixtiyoriy dasturlar. Eslatma: web dasturlar ikki qismdan, server tomon hamda mijoz tomondan iborat. Javada faqat server tomoni uchun

yoziladi. Bunda servlet, jsp, jsf kabi fundamental texnologiyalardan boshlab Spring, Play kabi freymworklar qo'l keladi. Umuman olganda brauzer uchun HTML, CSS hamda Java Scriptdan boshqa tilda yozib bo'lmaydi.

3) Enterprise Applications – bu dasturlar yirik salmoqqa ega bo'lib, odatda katta jabhalarda ishlatiladi. Banklar, tashkilotlar yoki astronomiya ishlarida foydalaniladi. Ular yuqori havfsizlik, yuklamani serverlarga teng taqsimlash (load balancing) yoki klasterlash (clustering – katta tizimdan xuddi yagona ob'yekt sifatida foydalanish) kabi sifatlarni talab qiladi. Javada bular albatta bor.

4) Mobile Applications – Mobil qurilmalarga mos dasturlarni ham Javada yozish mumkin. Androiddan boshlab, Java ME (JME – Java Micro Edition) gacha. JME ga misol qilib, Nokia telefonlarimiz uchun ishlangan JAR o'yinlarni misol keltirish mumkin.

Javaning sintaksisi C++ ga asoslangan

C++ tilini biladiganlar Javani oson o'rganishadi. Lekin undagi ko'pchilik xususiyatlar olib tashlangan. Masalan: Pointer(ko'rsatkich)lar bilan to'g'ri ishlash, ya'ni Javada alohida ko'rsatkich tushunchasi yo'q. Operatorlarni qayta yuklash ham olib tashlangan. Yana eng muhimi, ishlatilmaydigan xotira (unreferenced objects) avtomatik tozalanadi. Buni Javadagi Garbage Collector (GC – chiqindi yig'ishtirgich) amalga oshiradi. C++ da bu destruktorlar orqali qo'lda (manual – ruchnoy) qilingan. Bundan tashqari Java har bir yangi versiyada bundan qulay imkoniyatlarni qo'shib kelmoqda. Ayni paytda oxirgi versiya 8-sidir. 2014-yil 18- martda ommaga e'lon qilindi.

JRE va JDK ni nima farqi bor?

JRE – Java Runtime Environment JDK – Java Development Kit
JRE bu faqat dastur bajarilishi uchun kerak bo'lgan muhit, xolos. Dasturchi bo'lmagan oddiy foydalanuvchilarga Java dasturlari bajarilishi uchun JRE yetarli. JDK = JRE + dasturlashga oid

qo'shimcha instrumentlar. Bu esa dasturchilarga kerak. Formuladan ko'rinib turibdiki, JDK ning hajmi JREdan katta bo'ladi. U bir marta ko'chirib o'rnatiladi xolos.

Dunyoda qancha Java dasturchilar bor?

Oracle kompaniyasi tahminan 9 mln desa, Wikipedia 10 mln atrofida demoqda. NumberOf.net komandasi esa 9.007.346 sonini keltirishgan. Albatta bu statistika qaysidir vaqt uchun mos keladi. Ayni paytda yana o'sib bormoqda. Javada kod yozishga mo'ljallangan Eclipse (bepul) muhiti bir yilni ichida 7 mln. marta ko'chirilgan. Eng mashhur web-konteynerlardan biri bo'lgan Tomcat esa oyiga o'rtacha 550 ming, yiliga esa 6.6 mln. marta ko'chirilar ekan.

O'zgaruvchilar va konstantalar

Malumotlar tipi o'zgaruvchi nomi; `int x; int x; x = 10; System.out.println(x); // 10`, dan keyin o'zgaruvchilarni e'lon qilish mumkin: `int x, y; x = 10; y = 25; System.out.println(x); // 10 System.out.println(y); // 25`

Bir vaqtda bir nechta o'zgaruvchini qiymati bilan e'lon qilish mumkin:

```
int x=10, y=25;
```

```
x = 10; y = 25;
```

```
System.out.println(x); // 10
```

```
System.out.println(y); // 25
```

final so'zi konstantani bildiradi *final int LIMIT = 5;*
System.out.println(LIMIT); // 5

Ma'lumotlar tipi

boolean: true yoki false qiymatini qabul qiladi: **boolean** `isActive = false;` **boolean** `isActive = true;` **byte:** 1 bayt uzunlikka ega. Butun son qabul qiladi -128 dan 127 gacha: **byte** `a = 3;` **byte** `b = 8;` **short:** 2 bayt uzunlikka ega. Butun son qabul qiladi -32768 dan 32767 gacha: **short** `a = 3;` **short** `b = 8;` **int:** 4 bayt uzunlikka ega. Butun son qabul qiladi -2147483648 dan 2147483647 gacha: **int** `a =`

4; int b = 9; long: 8 bayt uzunlikka ega. Butun son qabul qiladi -9
223 372 036 854 775

808 dan 9 223 372 036 854 775 807 gacha: long a = 5; long b =
10; float: 4 bayt uzunlikka ega. Haqiqiy son qabul qiladi -9 223 372
036 854 775 808 dan 9 223 372 036 854 775 807 gacha: float x = 8.5;
float y = 2.7; char: 2 bayt uzunlikka ega. Simvol qabul qiladi.
Uzunlik o'lchovi 0 dan 65535: char ch=102; // simvol - 'f'
System.out.println(ch); String hello = "Hello...";
System.out.println(hello);

Java dasturlash tilida eng ko'p ishlatiladigan operatorlar:

- solishtirish operatorlari;
- mantiqiy operatorlar;
- o'zlashtirish operatori.

Solishtirish operatori

Ikki operatorni bir-biri bilan solishtirishda ishlatiladi. Odatda, solishtirish operatorlar, shart berish operatori (if) va tsikl (while, for) operatorlari bilan ishlatiladi.

Bu operatorlar ikki xil natija qaytarishi mumkin: **true** yoki **false** Java dasturlash tilida quyidagi solishtirish operatorlari mavjud:

= teng != teng emas > kata < kichik >= katta
yoki <= kichik yoki teng
public class CompareOperators
{ public static void main(String[] args) { int a = 3; int b = 6; boolean
c; c = (a==b); System.out.println(c); } // false

Mantiqiy operatorlar

Mantiqiy operatorlar natijasi **true** yoki **false** bo'lgan operandlar ustida amalga oshiriladi. Bu operatorlar quyidagilardan iborat:

- & – mantiqiy VA(AND)
- | – mantiqiy YOKI(OR)
- ^ – mantiqiy XOR(YOKI inkori)
- ! – mantiqiy YO'Q(NOT)

Yuqoridagi operatorlar orqali ikki operand qiymatni solishtiramiz:

a	b	a&b	a b	a^b	~a or !a
true(1)	true(1)	true	true	false	false
true(1)	false(0)	false	true	true	false
false(0)	true(1)	false	true	true	true
false(0)	false(0)	false	false	false	true

6.3.5-rasm. Mantiqiy operandlar ustid amallar va ularning natijalari.

```
public class CompareOperators {public static void
main(String[] args) {
    boolean a = true; boolean b = false; boolean c, d, e, f; c = a & b;
    d = a | b; e = a ^ b; f = !a; System.out.println("a&b = " + c);
    System.out.println("a|b = " + d); System.out.println("a^b = " + e);
    System.out.println("!a = " + c);}} //a&b = false //a|b = true //a^b =
true //!a = false
```

O'zlashtirish operatori

O'zlashtirish operatori tenglik (=) bilan ifodalanadi. Yuqorida ko'rgan misolimizga qaraydigan bo'lsak, $c = (a=b)$ ifoda a va b ni tekshirganimizdan chiqqan qiymatni c ga o'zlashtiryapmiz.

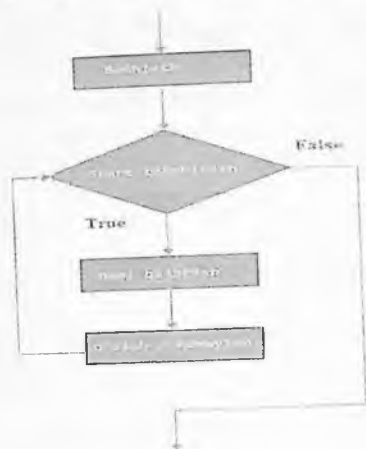
Xulosa qilib shuni aytish mumkunki, o'zlashtirish bilan solishtirish operatorlari farqi: (==) bo'lsa solishtirish; (=) bo'lsa o'zlashtirish bo'ladi.

Javada For Loop (Takrorlanuvchi tsikl operatori)

O'z nomi bilan ma'lum bo'lib turibdi takrorlanuvchi ya'ni qandaydir jarayonni qayta va qayta takrorlanishidir.

Yerning quyosh atrofida aylanishi yil fasllarning almashinishi va h.k.larni misol qilib keltirish mumkin. Takrorlanuvchi ish harakatlar qandaydir shartlar asosida bajariladi. Ularning

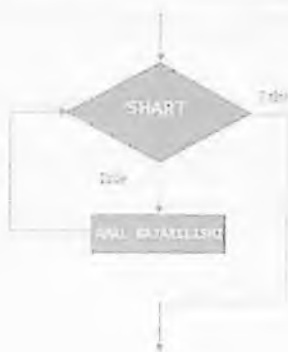
boshlang'ich, oxirgi nuqtalari va bajarilish davriyligi mavjud bo'ladi.



6.3.6-rasm. Takrorlash tsikllarining umumiy algoritmik blok sxemasi.

Java While Loop (Takrorlanuvchi operator)

«While» operatori dastur tuzishda ko'p ishlatiladigan tsikl operatori hisoblanadi. Bu operator bir yoki bir necha operatorlar guruhiga qo'yilgan shart **yolg'on** (**false**) bo'lguncha bajaradi. Qachonki shart rost bo'lsa, tsikl o'z ishini boshlaydi va shartdagi qiymatlar tsikl ichida o'zgartirib boriladi.



6.3.7-rasm. Takrorlash tsikllarining umumiy algoritmik blok sxemasi.

Shart har doim mantiqiy qiymat qabul qiladi: **rost(true)** yoki **yolg'on(false)**. Blok ichiga istalgancha operatorlar yozish mumkin, yoki umuman yozmaslik ham mumkin.

□ «**While**» operatorida avval shart tekshirilib keyin amal bajariladi;

□ «**While**» operatoridan raqamlar ketma-ketligi doimiy bo'lmagan hollarda ishlatiladi.

□ «**While**» operatori takrorlanish davri har xil bo'lgan tsikllar ustida ammalar bajarishda qo'l keladi.

Javada massivlar

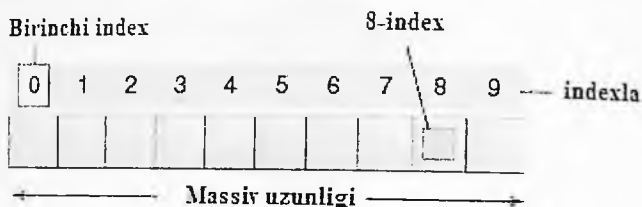
Massivlar bizga qanday imkoniyatlarni beradi:

✓ **Optimal kodlar:** kodlarni optimal darajaga keltirishimiz, qayta yozish yoki saralashlarni juda oson amalga oshirishimiz mumkin.

✓ **Ixtiyoriy kirish:** har qanday indeks(tartib)dagi qiymatni olish imkoniyati, qidirish imkoniyati.

✓ **Kamchiligi:** Massivning o'lchami oldindan beriladi shuning uchun uning o'lchami statik bo'ladi. Xoxlaguncha qiymat qabul qila olmaydi. Bunday holatlarda to'plamlardan foydalaniladi(collection).

Massiv tuzilishi:

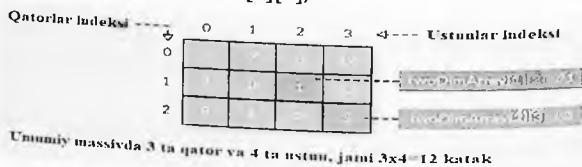


6.3.8-rasm. Massivlarning tuzilishini ifodalovchi sxemasi.

Massivlar 2 xil turda bo'lishi mumkin: **bir o'lchamli** va **ko'p o'lchamli**.

Bir o'lchamli massivlar bitta tipdagi o'zgaruvchilar ro'yxatidan iborat bo'ladi. Bir o'lchamli massivni e'lon qilish uchun, dastlab, massiv tipi, so'ng massiv nomi ko'rsatiladi(undan so'ng qavslar).

```
int a[]=new int[5]; //massivni e'lon qilish a[0]=10; a[1]=20;
a[2]=70; a[3]=40; a[4]=50; class Test Array{
    public static void main(String args []){
        int a[]=new int [5]; //massiv e'lon qilinyapti a[0]=10; a[1]=20;
        a[2]=70; a[3]=40; a[4]=50; //massivni ekrada chiqarish for(int
        i=0; i<a.length; i++) //length massivning uzunligi out.println(a[i]);}
        int[][] twoDimArray = new int[3][4];
```



6.3.9-rasm. Massiv elementlari soni va ularning indekslari.

Asosiy tayanch so'z va iboralar

Java, Oak, platforma, bytecode, JVM, API, Virtual mashina, JRE, JDK, o'zgaruvchi, ma'lumto tiplari, solishtirish operatorlari, mantiqiy

operatorlar, o'zlashtirish operatorlari, takrorlanuvchi operatorlar, massivlar.

Takrorlash va mustaqil ishlash uchun savollar

1. Java dasturlash tili boshqa OYD tillari bilan nimasi bilan farq qiladi?
2. Java dasturlash tilida yaratilgan ilovalar odatda qanday fayllardan iborat bo'ladi?
3. Java dasturlash tilida kross platformalilikka erishish uchun qanday kompilyatsiyalanadi?
4. JVM nima va u nima uchun kerak?
5. JDK nima va u nima uchun qo'llaniladi?

§6.4. Android tizimi uchun dasturlarni yaratish

Android (yunoncha so'z bo'lib—odamsifat robot ma'nosini ifoda etadi). Balki mana shuning uchun Android operatsion tizimi logotipida robot tasvirlangandir. Android operatsion tizimining yaratilishi tarixi 2002-yillardan boshlangan. Mana shu davrda Google korporatsiyasi yaratuvchilari E. Rubinning dasturiy ishlanmalari to'plami bilan qiziqib qoladilar. Dastlab mobil qurilmalar uchun yangi operatsion tizimni yaratish loyihasi bilan katta maxfiylik ostida Android Inc. kompaniyasi shug'ullangan, ushbu kompaniyani keyinchalik Google sotib oladi.

Android—Linux yadrosiga asoslangan kommunikatorlar, planshetli kompyuterlar, elektron kitoblar, raqamli musiqa uskunalari, qo'l soatlari, netbuklar va smartbuklar uchun portativ (tarmoqli) operatsion tizimdir. Keyinchalik Google Open Handset Alliance (OHA) alyansini tashkil qildi, u hozirda ham platformani qo'llab-quvvatlash va yanada rivojlantirish bilan shug'ullanadi.

Android Google tomonidan ishlab chiqilgan kutubxona orqali qurilmani boshqaruvchi Java-ilovasini yaratishga imkon beradi. Android Native Development Kit Si va boshqa tillarda yozilgan

illovalarni yaratadi. 2012-yilning uchinchi choragida sotilgan smartfonlarning 75 foizida Android operatsion tizimi o'rnatilgan.

Android operatsion tizimi uchun telefonlar ishlab chiqarish:

Hozirda bir qancha kompaniyalar Android uchun o'z telefonini sotuvga chiqarmoqda, masalan HTC kompaniyasining Dream, Magic, Droid hamda Motorola kompaniyasining Motorola Droid telefonlaridir.

Android operatsion tizimi uchun yaratilgan dasturlardagi qo'shimcha qulayliklar:

Oddiy foydalanuvchi uchun yana ko'plab dasturiy ta'minotlar yaratilgan bo'lib, bular sizga bir qator qulayliklar yaratishga xizmat qiladi. Masalan, Quickpedia – Wikipedia bilan ishlash uchun yaratilgan. Bunda dastur Wikipedia saxifasini sizning telefon ekrani hajmiga moslashtirib ko'rsatadi, hamda siz ko'rgan saxifaning matnini xotiraga saqlab qo'yish imkoniyati mavjud.

Dasturlashtirish tilini tanlash va qo'yilgan masalani dasturiy joriy qilish:

Android dasturlari Java dasturiy tizimida tuziladi. Android uchun Windows, MacOS, Linux operatsion tizimlarida dastur tuzish mumkin.

Dasturlash muhiti sifatida Eclipse yoki Android Studio dasturidan foydalanish tavsiya etiladi. Android operatsion tizimiga dastur tuzish uchun kerakli vositalarni keltirib o'tamiz:

Bular:

1. JRE (Java Runtime Environment);
2. JDK (Java Development Kit);
3. Eclipse yoki Android Studio dasturlash muhiti;
4. Android SDK (Android Software Development Kit);
5. ADT (Android Development Tools).

JRE (Java Runtime Environment)

JRE. Java tilida yoziladigan dasturlarni o'qish va ishlatish uchun kompyuterlarda JRE(Java Runtime Environment), ya'ni Java ishlash muhitini o'rnatish kerak.

JDK (Java Development Kit)

JDK. Java tilida dasturlar tuzish uchun JDK (Java Development Kit) ya'ni Java dasturlash jamlamasini o'rnatish lozim. JDKning ichida Java kompilyatori, Java kutubxonalari, har xil utilitalar, misollar va hujjatlar bo'ladi. Lekin JDKni ichiga Java dasturlash muhiti kirmaydi.

Eclipse yoki Android Studio dasturlash muhiti

Eclipse yoki Android Studio. Javada dastur tuzish uchun Eclipse yoki Android Studioni ishlatamiz. Android uchun aynan Eclipse yoki Android Studio dasturlash muhitini tavsiya etilishining sabablari:

- 1) Eclipse yoki Android Studio bepul hamda ularda ishlashni o'rganib olish juda oson;
- 2) Google kompaniyasi aynan Eclipse yoki Android Studio uchun ADT plaginini ishlab chiqargan.

ADT (Android Development Tools)

ADT yordamida Eclipse yoki Android Studioda Android loyihalarini tuzish, kompilyatsiya qilish va Android qurilmalarining emulyatorlaridan foydalanish mumkin. ADT plaginini kompyuterga o'rnatishimiz bilan Android loyihalari uchun avtomatik tarzda sozlab qo'yiladi.

ASDK (Android Software Development Kit)

Android SDK. Androidda dastur tuzish uchun Android SDK dastur tuzish jamlanmasi kerak bo'ladi. Android SDK ning ichida Android qurilmasining emulyatori ham mavjud. Shuning uchun Androidda dastur tuzmoqchi bo'lsak, bizda haqiqiy kommutator

bo'lishi shart emas, dasturni bema'lol virtual qurilmada sinab ko'rish mumkin.

Google Androidda dastur yozish uchun yuqorida keltiriladigan dasturlarni yuklab olish va ularni bir-biriga moslashtirish kerak. Quyida kerakli dasturlarni yuklab olish va ularni o'rnatib, bir-biriga moslashtirish keltirib o'tiladi:

1-bosqich: Eclipse yoki Android Studio

Dasturni yozish uchun avvalambor sizga Integrated Development Environment (IDE) kerak bo'ladi. Hozirda Eclipse yoki Android Studio IDE keng tarqalgan, lekin siz Eclipse dan foydalanmagan bo'lsangiz, NetBeans dan foydalanib ko'rishingiz ham mumkin.

2-bosqich: Java JDK

Agar sizning kompyuteringizda Java JDK o'rnatilmagan bo'lsa, Java JDK ko'chirib olinadi (operatsion tizim va versiyasiga mos):

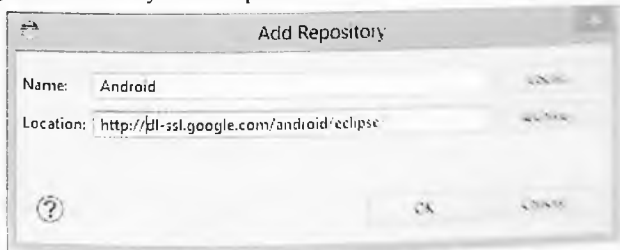
<http://www.oracle.com/technetwork/java/javase/downloads/index-136632.html>

3-bosqich: Android SDK Tool

Sizga Android SDK Tool kerak bo'ladi. Uni to'g'ridan to'g'ri Googledan olishingiz mumkin (operatsion tizim va versiyasiga mos): <http://developer.android.com/sdk/index.html>

4-bosqich: Eclipse yoki Android Studioni Android uchun moslashtirish

Eclipse yoki Android Studio ishga tushiriladi va Help->Install New Software (yangi dasturiy vosita o'rnatish menyusini) tanlanadi. Android Studioda tizimida yoki Eclipse yangi versiyalarida menyu boshqa manzilda bo'lishi mumkin.



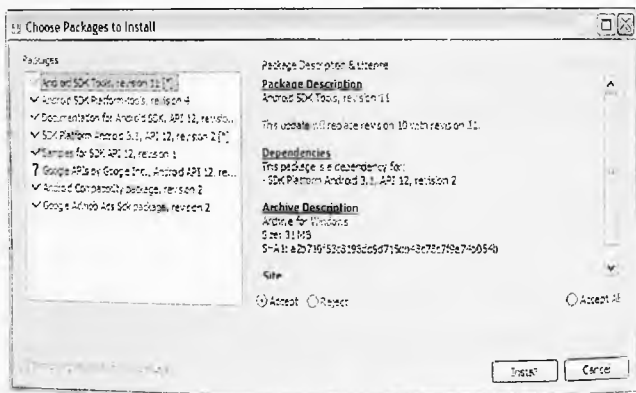
6.4.1-rasm. Yangi repository qo'shish o'rnatish

Yuqoridagi rasmda ko'rsatilganidek, dasturiy vosita uchun yangi repository qo'shish o'rnatish —Android deb, linkka <http://dl-ssl.google.com/android/eclipse/> deb belgilash kerak. Agar bu link ishlamayotgan bo'lsa, —https ni o'rniga —http deb yozish kerak bo'ladi).

Barcha paketlarni tanlab, —Next ni bosganimizdan so'ng, —Development Tools lar ko'chirila boshlaydi. Ko'chirib bo'lganidan so'ng, Eclipse yoki Android Studioni yopib, qaytadan ishga tushiramiz.

5-bosqich: Android SDK ni moslashtirish (configure).

Android SDK ni ko'chirilib, arxiv ochilgan papkadan—SDK Manager.exe degan faylni topib, uni ishga tushiramiz. Quyidagiga o'xshash dialog yoqiladi:



6.4.2-rasm. IDE tizimga SDK paketlarni o'rnatish oynasi.

Asosiy tayanch so'z va iboralar

Android, google, Java, Android Studio, Eclipse, JRE, JDK, SDK, ADT, ASDK, IDE, NetBeans, repository, http, https.

Takrorlash va mustaqil ishlash uchun savollar

1. Android tizimini nima va undan qaysi soxada keng foydalaniladi?
2. Android tarixini bilasizmi?
3. Android tizimi qanday qulayliklarga ega?
4. ADT (Android Development Tools) nima?
5. Emulyator nima va undan dasturchilar nima maqsadda foydalanishadi?

VI-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI

AMALIY MASHG'ULOT №15

CodeSys tizimini o'rnatish va sozlash

Ishdan maqsad: CodeSys dasturini kompyuterga o'rnatish ketma-ketligi, dastur sozlash, dasturning oynasini tashkil etuvchilarning vazifasini tushuntirish.

Nazariy ma'lumotlar

Sanoatni avtomatlashtirishda instrumental dasturiy ta'minot to'plami 3S- Smart Software Solutions GmbH, 1.0 ver. 1994 yilda ishlab chiqilgan. 2012 yil noyabr oyida esa CodeSys nomi o'zgartirilgan. Dasturni Windows XP va undan yuqori versiyali OSLarga o'rnatilishi mumkin.

CodeSysda dasturlash tizimi quyidagilarni o'z ichiga oladi:

- Tuzatuvchi, kompilyator va otladchik XEK loyihalarida;
- XEK ning barcha 5 ta dasturlash tilini qo'llab-quvvatlaydi;
- Bu protsessorlar juda keng mashina kodni hosil qiladi.

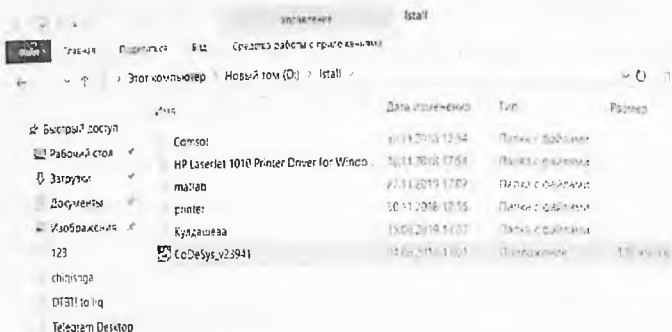
CodeSysda bajarish tizimi quyidagilarni o'z ichiga oladi:

- kirish/chiqishlarni yangilash bilan boshqarish sikli;
- dasturlash tizimi bilan muloqot;
- kontrollerni quvvatlagandan so'ng dasturni yuklash.

CoDeSysni kompakt disk yoki quyidagi sayt orqali www.owen.ru tortib olib o'rnatish mumkin.

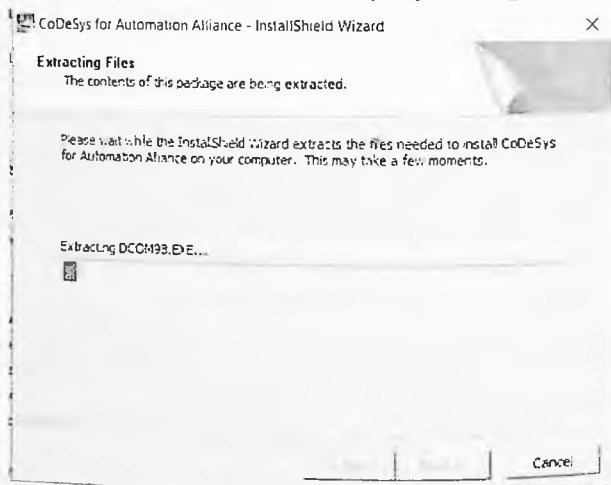
Ishni bajarish tartibi:

1.CodeSys dasturining kompyuterga o'rnatiladigan faylini tanlang.



6.5.1-rasm. CodeSys tizimini o'rnatish faylini tanlash.

exe kengaytmali fayl ustiga sichqoncha tugmasini 2 marta tez tanlang. Natijada dasturni o'rnatish jarayoni ishga tushadi.



6.5.2-rasm. CodeSys tizimini o'rnatish oynasi.

3.Jarayon yakunlangandan so'ng, quyidagi hosil bo'lgan oynadan OK tugmasini tanlang.

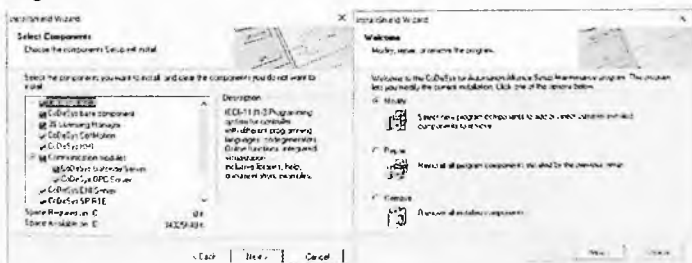


Please close all running applications
before continuing the installation !

OK

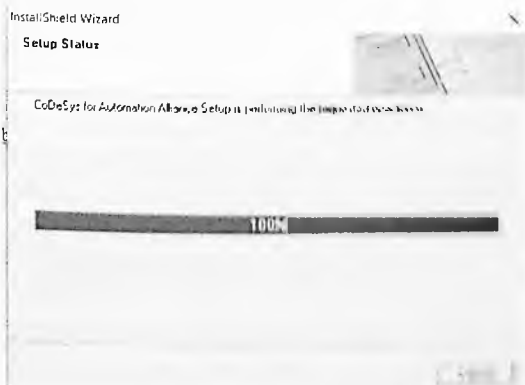
6.5.3-rasm. CodeSys tizimini o'rnatishni boshlash jarayoni

4.Hosil bo'lgan oynalarda ketama-ket Next tugmasini tanlang.



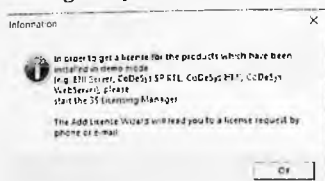
6.5.4-rasm. CodeSys tizimini o'rnatish oynalari

5.Natijada o'rnatishning keying bosqichi ketayotganligini guvohi bo'lamiz.



6.5.5-rasm. CodeSys tizimini o'rnatish jarayoni

6. Keyingi hosil bo'lgan oynadan OK tugmasini tanlang.



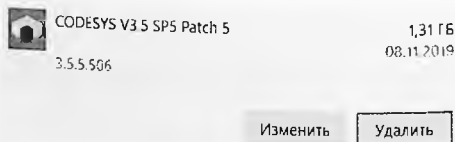
6.5.6-rasm. CodeSys tizimini o'rnatish jarayoni yakuni.

7. *Finish* tumasini tanlaymiz. Amallar bajarilgandan so'ng dastur o'rnatilib bo'linganini ko'rishimiz mumkin.



6.5.7-rasm. CodeSys tizimini o'rnatish jarayoni yakuniy oynasi.

8. Kompyuterdan dasturni *deinstallyatsiya* qilish jarayoni quyidagi ketma-ketlikda bajariladi. *Пуск-параметры-приложение*. Paydo bo'lgan oynadan dastur nomini tanlaymiz va *Удалить* buyrug'ini tanlaymiz.



6.5.8-rasm. CodeSys tizimini operatsion tizimdan o'chirish.

9. Natijada dasturni o'chirish jarayoni ketayotganligini guvohi bo'lamiz.



6.5.9-rasm. CodeSys tizimini operatsion tizimdan o'chirish jarayoni.

Bajarish uchun topshiriqlar:

1. CodeSys dasturida AND elementidan foydalanib quyidagi mantiqiy ifodani bajaring: $A \& B \& A$.
2. CodeSys dasturida NOT elementidan foydalanib quyidagi mantiqiy ifodani bajaring: $\neg A \& \neg B \& \neg C$.
3. CodeSys dasturida OR elementidan foydalanib quyidagi mantiqiy ifodani bajaring: $\neg A \vee \neg B \& C \vee A$.

TAJRIBA MASHG'ULOTI №15

**CodeSys tizimida signallarni kiritish va chiqarishni o'rganish
hamda mantiqiy amallarni qo'llash.**

Ishdan maqsad: CodeSys tizimida loyihalarni yaratishni o'rganish va dasturlash muhitini ishlash ko'nikmalariga ega bo'lish.

Nazariy ma'lumotlar

Har qanday ob'yektni boshqarish uchun u to'g'risidagi ma'lumotlarni kontrollerga jo'natish kerak. Bu uchun dasturiy ta'minot orqali fizik kiritish portiga signal jo'natiladi, shundan so'ng mantiqiy elementning chiqish porti orqali ob'yektni boshqarish imkoni yaratiladi.

Avvalo, siz Target platformaning texnik xususiyatlari haqidagi ma'lumotni, tizimga qanday o'rnatishni aniqlab olishingiz va albatta boshqaruvni bilishiz zarur. Agar OBEH PLK110 aniqrog'i PLK110-30.R-L modifikatsiyasi haqida

gapiradigan bo'lsak, unda biz uchun muhim bo'lgan quyidagi xususiyatlarni ajratib ko'rsatish kerak.

- 18 diskret kirishlar;
- 12 diskret chiqishlar (rele);
- 3 mb– foydalanish xotirasi;
- RS-232 interfeysi 2 ta;
- RS-485 interfeysi 2 ta;
- Ethernet interfeysi;
- USB-device dasturlash interfeysi
- Tizimda real vaqt soati o'rnatilgan.

Dasturlanadigan mantiqiy kontroller uchun **tizim protsessor** turini, **operativ xotiradan** qancha joy olishini hamda dastur qaysi tilda yozilishini dasturchi bilishi kerak.

Bir so'z bilan aytganda, juda ko'p ma'lumotlar mavjud. Bularning barchasini qo'lda kiritish yoki boshqarish qiyin. CoDeSysda quyidagilar qabul qilingan: OBEH kompaniyasi kontrollerlar yaratadi, mantiqiy element haqida kerakli ma'lumotlarini o'z ichiga olgan **Target platforma fayllarini** saqlaydi. Ushbu fayllar shuningdek qisqa (**target** ingliz tilidan) **Target fayllar** deb ataladi.

Tajriba mashg'uloti bo'yicha topshiriq:

CodeSys dasturi uchun Target faylni o'rnatish.

Ishni bajarish tartibi:

Biz target fayllarni bir marotabaga barchasini o'rnatishimiz mumkin, lekin bu noqrulaylik yaratadi chunki odatda dasturchilar bir yoki bir nechtagina mikrokontrollerlar bilan ishlaydilar. Shuning uchun biz konkret mikrokontroller modeli uchun target faylni o'rnatishni ko'rib o'tamiz.

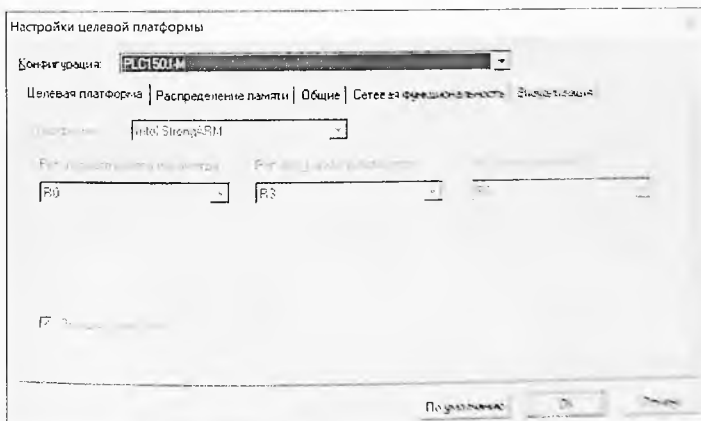
1.Target-files.rar arxivini oching va unda PLC150.I-M papksini toping. Ushbu papkani topganizdan so'ng, InstallTarget.bat faylini ishga tushuring.

2. Ishga tushurgandan so'ng, mikrokontrollerni usbu modelni target fayli o'rnatiladi.

CodeSys tizimida loyiha yaratish va uni emulyatsiya rejimida ishga tushurish.

1. CodeSys tizimini ishga tushuring. Asosiy oyna ishga tushgandan so'ng, yangi loyiha yarating (Файл >> Создать).

2. Ekranda quyidagi oyna paydo bo'ladi «Настройка целевой платформы». Ushbu oynada ro'yxatdan «PLC150.I-M» ni tanlang. OK tugmasini tanlagandan so'ng, tizim yangi bo'sh loyiha yaratadi.



6.6.1-rasm. CodeSys tizimida yangi loyiha uchun controller tanini tanlash oynasi.

3. Yangi loyihani yaratishdan oldin tizim siz loyihani qaysi dasturlash tilida yaratishizni so'raydi.

Новый программный компонент (POU)

Имя нового POU:

PLC_PRG

OK

Отмена

Тип POU

- ☒ Программа
- ☐ Функциональный блок
- ☐ Функция

Тип результата:

BOOL

Язык реализации

- ☐ IL
- ☐ LD
- ☐ FBD
- ☐ SFC
- ☐ SI
- ☒ CFC

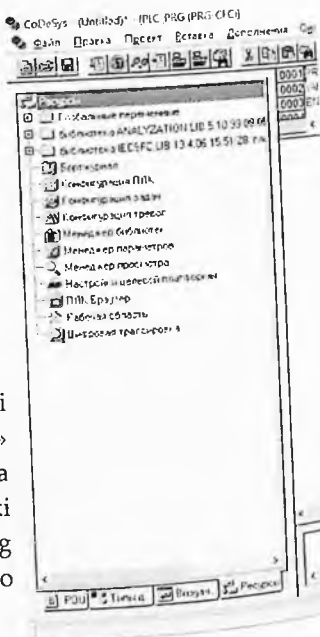
6.6.2-rasm. CodeSys tizimida yangi loyiha uchun dasturlash tilini tanlash oynasi.

4.Hozircha biz ro'yxatdan CFC tilini tanlaymiz va OK tugmasini bosamiz.

5.Loyiha yaratilgach oynaning quyi qismida «Ресурсы» bo'limiga o'ting.

6.6.3-rasm. CodeSys tizimida yangi loyiha resurslari oynasi.

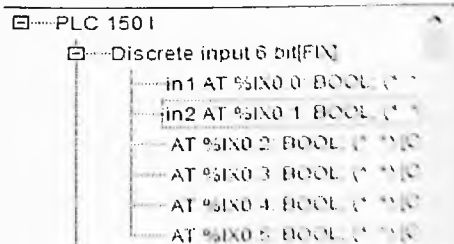
6.Oynaning chap qismidagi daraxtdan «Конфигурация ПЛК» punktini tanlang va unga sichqonchani chap tugmasini ikki marotaba tanlang. Ushbu oynaning o'ng qismida quyidagi interfeys paydo bo'ladi.





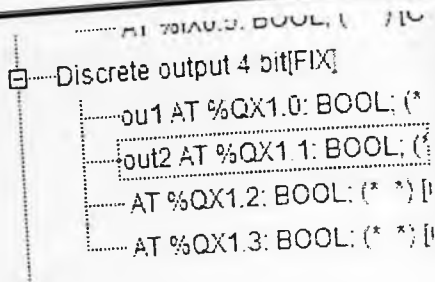
6.6.4-rasm. CodeSys tizimida yangi loyiha uchun controller kirish va chiqish portlarini sozlash oynasi.

7. Mikrokontroller nomiga + belgisini tanlang va ochilgan daraxtda Discrete input shoxasini toping va unda ikkita o'zgaruvchi yarating. O'zgaruvchilarni yaratish uchun ro'yxatdan istalgan o'ringa sichqonchani chap tugmasi bilan ikki marta murojaat eting va o'zgaruvchi nomini kiriting, masalan «in1» va hokazo.



6.6.5-rasm. CodeSys tizimida kirish portlariga o'zgaruvchilarni belgilash oynasi

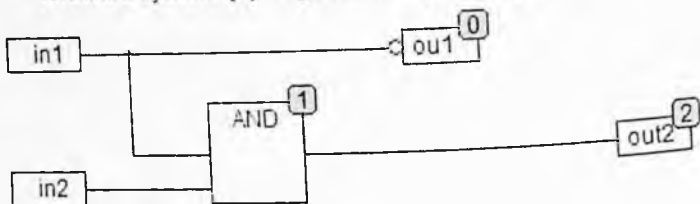
8. Xuddi shunga o'xshash Discrete output bo'limida qolqori qiymatlari uchun 2 ta o'zgaruvchi yarating.



6.6.6-rasm. CodeSys tizimida chiqish portlariga o'zgaruvchilarni belgilash oynasi.

9.O'zgaruvchilarni e'lon qilingandan so'ng, POU bo'limiga o'ting. Ushbu bo'limda daraxtdan chapda PLC_PRG(PRG) so'ziga sichqonchani chap tugmasini ikki marotaba tanlang. Shundan so'ng o'ng tomonda kodni yozish uchun oyna paydo bo'ladi.

10.Ushbu oynada quyidagi sxemani yarating.



6.6.7-rasm. CodeSys tizimida dastur sxemasi ko'rinishi.

11.Dasturni emulyatsiya rejimida ishga tushurish uchun «Онлайн» menyusiga o'tib, «Режим эмуляции» punkti tanlanadi. So'ngra «Подключение» punkti tanlanadi va nihoyat «Старт» punkti tanlanadi.

12.Kirish signallari qiymatlarini o'zgartirish uchun "in1" o'zgaruvchisiga sichqoncha chap tugmasi bilan ikki marotaba bosing va kirish signallari qiymatlarini o'zgartiring. CTRL+F7 issiq klavishasi bosilgach, siz chiqish qiymati natijasini ko'rishingiz mumkin.

Bajarish uchun topshiriqlar:

1.Ushbu loyihada "in1" kirish signali o'zgaruvchisiga false va "in2" kirish signali o'zgaruvchisiga esa false qiymatlarini belgilab, natijani olib, so'ngra loyihani saqlab o'qituvchiga uzating.

2.Ushbu loyihada "in1" kirish signali o'zgaruvchisiga true va "in2" kirish signali o'zgaruvchisiga esa false qiymatlarini belgilab, natijani olib, so'ngra loyihani saqlab o'qituvchiga uzating.

3.Ushbu loyihada "in1" kirish signali o'zgaruvchisiga false va "in2" kirish signali o'zgaruvchisiga esa true qiymatlarini belgilab, natijani olib, so'ngra loyihani saqlab o'qituvchiga uzating.

4.Ushbu loyihada "in1" kirish signali o'zgaruvchisiga true va "in2" kirish signali o'zgaruvchisiga esa true qiymatlarini belgilab, natijani olib, so'ngra loyihani saqlab o'qituvchiga uzating.

TAJIRIBA MASHG'ULOTI №16

Android tizimida ilovalarni yaratish jarayoni bilan tanishish

Ishdan maqsad: Android tizimi uchun ilovalarni yaratishni o'rganishdan iborat.

Nazariy ma'lumot

Androidda dastur tuzish uchun android SDK dastur tuzish jamlanmasi kerak bo'ladi. Android SDK ning ichida Android qurilmasining emulyatori ham mavjud. Shuning uchun Androidda dastur tuzmoqchi bo'lsak, bizda haqiqiy kommutator bo'lishi shart emas, dasturni bimalol virtual qurilmada sinab ko'rish mumkin.

Google Androidda dastur yozish uchun yuqorida keltiriladigan dasturlarni yuklab olish va ularni bir-biriga moslashtirish kerak. Quyida kerakli dasturlarni yuklab olish va ularni o'rnatib, bir-biriga moslashtirish keltirib o'tiladi:

1. Eclipse yoki Android Studio. Dasturni yozish uchun avvalambor sizga Integrated Development Environment (IDE) kerak bo'ladi. Hozirda Eclipse yoki Android Studio IDE keng tarqalgan, lekin siz Eclipse dan foydalanmagan bo'lsangiz, NetBeans dan foydalanib ko'rishingiz mumkin.

2. Java JDK. Agar sizning kompyuteringizda Java JDK o'rnatilmagan bo'lsa, Java JDK ko'chirib olinadi.

3. Android SDK Tool. Sizga Android SDK Tool kerak bo'ladi.

4. Eclipse yoki Android Studioni Android uchun moslashtirish. Eclipse ishga tushiriladi va Help -> Install New Software (yangi dasturiy vosita o'rnatish) menyusini tanlanadi.



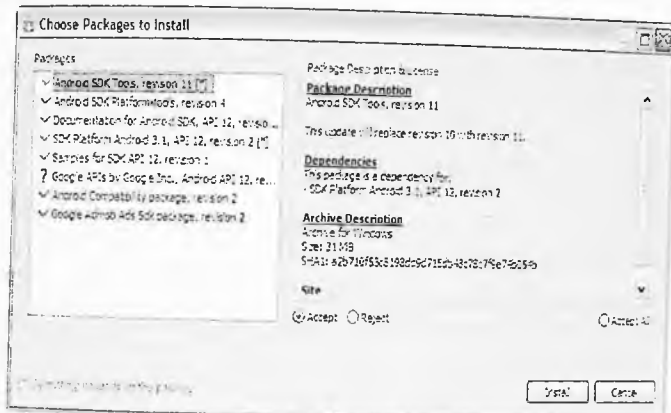
6.7.1-rasm. Yangi repository qo'shish oynasi.

Yuqoridagi rasmda ko'rsatilganidek, dasturiy vosita nomiga —Android deb, linkka esa "https://dl-ssl.google.com/android/eclipse/" deb yoziladi (agar link ishlamayotgan bo'lsa, —https ni o'rniga —http deb yozsa ham bo'ladi).

Barcha paketlarni tanlab, —Next ni bosganimizdan so'ng, —Development Tools lar ko'chirishni boshlaydi. Ko'chirib bo'lganidan so'ng, Eclipse yoki Android Studio ni o'chirib, qaytadan ishga tushiramiz.

5. Android SDK ni moslashtirish (configure). Android SDK ni ko'chirilib, arxiv ochilgan papkadan—SDK Manager.exe degan

faylni topib, uni ishga tushiramiz. Quyidagiga o'xshash dialog yoqiladi:



6.7.2-rasm. IDE tizimga SDK paketlarni o'rnatish oynasi. Tajriba mashg'ulotini bajarish uchun topshiriq

Ikkita sonli qiymatlarni kirituvchi va ularning yig'indisini chiqaruvchi Java dasturlash tilidagi mobil ilovasini yarating. Dasturni kompilyatsiyalang va krossplatformali ilova hosil qiling.

Ishni bajarish tartibi:

1. Android Studio dasturlash muhitini ishga tushuring.
2. Android ilova yangi loyihasini yarating.
3. activity_main.xml fayli tarkibiga quyidagi kodni joylashtiring:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout
```

```
xmlns:android=http://schemas.android.com/apk/res/android
```

```
xmlns:app=http://schemas.android.com/apk/res-auto
```

```
xmlns:tools=http://schemas.android.com/tools
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```



```

tools:context=".MainActivity"><EditText
    android:id="@+id/number1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="120dp"
    android:autofillHints="no"
    android:ems="10" android:hint="@string/Chislo1"
    android:inputType="number|numberSigned|numberDecimal"
/> <EditText android:id="@+id/number2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="190dp"
    android:autofillHints="no"
    android:ems="10"
    android:hint="@string/Chislo2"
    android:inputType="number|numberSigned|numberDecimal"/> <Button
    android:id="@+id/button" android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="260dp" android:padding="20dp"
    android:text="@string/Summa" android:textSize="24sp" />
</RelativeLayout>

```

4.MainActivity.java fayliga quyidagi dasturiy kodni yozing:

```

package uz.beti.uilll.myproject;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import android.view.View; import
android.widget.Button; import android.widget.EditText; public
class MainActivity extends AppCompatActivity implements

```

```
View.OnClickListener {
```

```
  @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Button button=findViewById(R.id.button);
```

```
        button.setOnClickListener(this);
```

```
    } @Override public void onClick(View view) {EditText  
et1=findViewById(R.id.number1);
```

```
    EditText et2 = findViewById(R.id.number2); String  
number1=et1.getText().toString(); String  
number2=et2.getText().toString(); System.out.println (number1);  
System.out.println (number2);
```

```
    Intent intent = new Intent (this, BackActivity.class); intent.  
putExtra ("NUMBER_1",number1); intent. Put Extra  
("NUMBER_2",number2); startActivity (intent);  
}}5.activity_back.xml fayliga quyidagi tuzilma kodini kiriting:
```

```
<?xml version="1.0" encoding="utf-8"?> <Relative Layout
```

```
xmlns:android=http://schemas.android.com/apk/res/android
```

```
xmlns:app=http://schemas.android.com/apk/res-auto
```

```
xmlns:tools=http://schemas.android.com/tools
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
tools:context=".BackActivity"><TextView
```

```
android:id="@+id/summa"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_alignParentStart="true"
```

```
android:layout_alignParentLeft="true"
```

```
android:layout_alignParentTop="true"
```

```
android:layout_marginStart="151dp"
```

```
android:layout_marginLeft="151dp" android:layout_margin
```

```
Top="131dp" android: text ="Text View" android:text
```

Color="@color/color Accent" android:text Size="24sp" /><Button
 android: id="@+id/button" android: layout_width = "wrap_content"
 android:layout_height="wrap_content" android:layout_align
 Start="@+id/summa" android: layout_align Parent Top="true"
 android: layout_margin Top="197dp" android:on Click="click
 Button" android:text= "Haqda" /></RelativeLayout>

5.BackActivity.java fayliga quyidagi dasturiy kodni yozing:
 package uz.beti.uill. myproject; import android.content.Intent;
 import android.support.v7.app. App Compat Activity; import
 android.os. Bundle; import android.view. View; import
 android.widget.TextView; public class BackActivity extends
 AppCompatActivity {@Override protected void onCreate(Bundle
 savedInstanceState State) {super.on Create(savedInstanceState);
 setContentView (R.layout.activity_back); Intent intent=getIntent();
 String number1=intent.get String Extra ("NUMBER_1"); String
 number2=intent.get String Extra ("NUMBER_2"); double result=
 Double. parse Double(number1)+Double.parse Double(number2);
 TextView summa=find View ById (R.id.summa); summa. set Text
 (number1+ "+" +number2+" =" +result);}public void click Button
 (View view) { Intent intent=new Intent (this, Main Activity. class);
 startActivity(intent); }}

6.Dasturni kompilyatsiyalang va ishga tushuring hamda
 ishlashini tekshiring.

Bajarish uchun topshiriqlar:

Andriod tizimi uchun Java dasturlash tilidan foydalanib
 quyidagi ifodani hisoblovchi mobil krossplatformali ilova
 yarating: $S = \sum_{i=4}^{0,4} i(i+1) + i$

2.Andriod tizimi uchun Java dasturlash tilidan foydalanib
 quyidagin 2ifodani hisoblovchi mobil krossplatformali
 ilova yarating: $S = \sum_{j=1}^{5+j} 6j + 2,7$

3. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{k=1}^n k + 1, 2$

4. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=1}^{35} i + i^5$.

Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=1}^n \cos i$

6. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=5}^n \cos i + \sin i$

7. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{k=1}^n k + 0,4 k^3 + k^8$. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=1}^n 2^k + 2^k + 9$. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=1}^n 0,4i + \cos i + 3i$

10. Andriod tizimi uchun Java dasturlash tilidan foydalanib quyidagi ifodani hisoblovchi mobil krossplatformali ilova yarating: $S = \sum_{i=1}^n i + 2$

VI-MODUL BO'YICHA TEST SAVOLLARI

1.CodeSys tizmda loyiha tuzilmasining birinchi bo'limi qanday nomlanadi?

- A.POU
- B.Ma'lumot tipi
- C.HMI
- D.Resurslar

2.CodeSys tizmda loyiha tuzilmasining ikkinchi bo'limi qanday nomlanadi?

- A.POU
- B.Ma'lumot tipi
- C.HMI
- D.Resurslar

3. CodeSys tizmda loyiha tuzilmasining uchinchi bo'limi qanday nomlanadi?

- A.POU
- B.Ma'lumot tipi
- C.HMI
- D.Resurslar

4.CodeSys tizmda loyiha tuzilmasining to'rtinchi bo'limi qanday nomlanadi?

- A.POU
- B.Ma'lumot tipi
- C.HMI
- D.Resurslar

5.CodeSys tizmda yaratilgan loyiha qanday kengaytmali faylda saqlanadi?

- A.Pro
- B.Ma'lumot tipi
- C.HMI
- D.Resurslar

6.CodeSys tizmda dasturni bajarilishi qaysi funktsiyadan boshlanadi?

- A.PLC_PRG
- B.Ma'lumot tipi
- C.HMI

D.Resurslar

7.CodeSys tizmida o'zgaruvchilarni e'lon qilishning necha turi mavjud?

A.2

B.3

C.4

D.5

8.CodeSys tizmida o'zgaruvchilarni e'lon qilishning birinchi turi qanday?

A.Matnli, jadvalli va avtomatik

B.Lokal (1-FB uchun) yoki Global (barcha FB) (FB- Funktsional blok)

C.Saqlanuvchi va doimiy o'zgaruvchilar

D.Let o'zgaruvchilar

9.CodeSys tizmida o'zgaruvchilarni e'lon qilishning ikkinchi turi qanday?

A.Matnli, jadvalli va avtomatik

B.Lokal (1-FB uchun) yoki Global (barcha FB) (FB- Funktsional blok)

C.Saqlanuvchi va doimiy o'zgaruvchilar

D.Let o'zgaruvchilar

10.CodeSys tizmida o'zgaruvchilarni e'lon qilishning uchinchi turi qanday?

A.Matnli, jadvalli va avtomatik

B.Lokal (1-FB uchun) yoki Global (barcha FB) (FB- Funktsional blok)

C.Saqlanuvchi va doimiy o'zgaruvchilar

D.Let o'zgaruvchilar

11.Java dasturlash tili qaysi dasturlash tili asosida paydo bo'lgan?

A.OAK

B.C

C.Delphi

D.Python

12.java kengaytmali faylida nima saqlanadi?

A.java kod

B.bytecode

C.forma tuzilishi

D.xml kod

13.class kengaytmali faylida nima saqlanadi?

A.java kod

B.bytecode

C.forma tuzilishi

D.xml kod

14.Java dasturlash tilini biladigan dasturchilar odatda qaysi kengaytmali fayl bilan ishlaydi?

A.java

B.class

C.exe

D.dfm

15.Bytecode formatiga o'girishdan maqsad nima?

A.bajariluvchi fayl hosil qilish

B.dastur hajmini qisqartirish

C.dastur bajarilishini tezlashtirish

D.krossplatformalikka erishish

16.JVM qisqartmasi to'g'ri ko'rsatilgan variantni tanlang.

A.Java Virtual Machine;

B.Java Virtual Module;

C.Java Virutal Models;

D.Java Virtual Moments;

17. JDK qisqartmasi to'g'ri ko'rsatilgan variantni tanlang.

A.Java Development Kit;

B.Java Development Kotlin;

C. Java Dynamic Kit;

D.Java Design Kit;

18.Java dasturlash tilida qaysi quyidagi IDE lardan foydalanish mumkin?

A.NetBeans

B.Delphi

C.Visual J#

D.DevC++

19.JRE qisqartmasi to'g'ri ko'rsatilgan variantni tanlang.

A.Java Runtime Environment; B. Java Runtime Engine;

C.Java Runtime Execute

D.Java Runtime Exit;

FAN BO'YICHA GLOSSARIY

Atamaning o'zbekcha nomlanishi	Atamaning ruscha nomlanishi	Atamaning inglizcha nomlanishi	Atamaning ma'nosi
Labview	Labview	Labview	virtual laboratoriya uskunalarini yaratuvchi dasturlash muhiti hisoblanib, siz ushbu muhit yordamida ilovalar ishlab chiqishingiz mumkin
yuzadagi panel	Лицевая панель	Front panel	virtual uskunaning interfaol foydalanadigan interfeysi deb aytiladi
diagrammalar-bloki	Блок диаграмм	Diagram bloke	LabVIEW grafik dasturlash tilida yaratilgan dastur kodi deb aytiladi
boshqarish elementi	Элемент управления	Control element	foydalanuvchi tomonidan kiritadigan ma'lumot ar to'plami
Indicator	Индикатор	Indicator	foydalanuvchiga yoki ekranga chiqariladigadigan ma'lumotlar
ma'lumotlar bog'lami	Узлы данных	Node	bu dasturning ixtiyoriy bajariluvchi elementining umumiy nomi
VIU(virtual uskuna)	ВВП(внутренней виртуальный прибор)	SubVI	bu VU(virtual uskuna) bo'lib, uni siz boshqa VU dan chaqirasiz
Belgi	Название	Label	yuzadagi panel va diagrammalar blokidagi komponentlarni nomlovchi matn bloki

Sarlavha	Заголовок	Caption	matn hisoblanib, u boshqaruvchi element yoki indikatorni izohlaydi
tunnel	Тунель	tunnel	LabVIEW tuzilishlarda faqatgina o'tkazgich va tuzilish chegarasi kesishgan nuqtada yaratiladi
For Loop	For Loop	For Loop	iteratsiyalar sonini qayd etish sikli
While Loop	While Loop	While Loop	shart bo'yicha tsikl
Avtoindekslash	Авто индексирование	Auto index	LabVIEW da For Loop siklini takroran bajarilishi tsikl tarkibiga kirgan massiv elementlarini avtomatik tartib bo'yicha kiritish
siljuvchan regist	Сдвиговой регистр	Shift registr	Siljuvchan registrlar shart bo'yicha tsiklda va aniq sonli iteratsiyali tsiklarda qo'llanilib, bir tsikl iteratsiyasidan keyingisiga qiymatni uzatish uchun ishlatiladigan aloxida boshqaruvchi tipi hisoblanadi
Variant tuzilishi	Структура варианта	Case Structure	shart mavjud bo'lgan matnni bajarish usuli bo'lib, ya'ni implikatsiya operatoriga o'xshaydi (if-then-else)
massiv	Массив	array	boshqa dasturlash tillaridagi kabi bir xil tipdagi elementlar to'plamini tashkil etad
Klaster	Кластер	cluster	Massiv kabi

			<i>klaster (cluster)</i> lar ham elementlar to'plamidan iborat bo'ladi. Ammo massivlardan farqli ravishda, klasterlar har xil tipdagi elementlarni birlashtira oladi (sonli, mantiqiy va hokazo).
NI-DAQmx	NI-DAQmx	NI-DAQmx	NI-DAQmx – NI kompaniyasining ma'lumotlarni yig'ish uskunalarini uchun drayverlar to'plamini hisoblanib, quyidagi yutuqlarga ega: □ yaxshilangan holat modeli; □ ko'poqimli drayver; □ to'xtab qolishda yuqori barqarorlik; □ juda oddiy sozlash; □ LabVIEW da VIU kamligi; □ ...
Mahalliy o'zgaruvchilar	Локальные переменные	Local variable	diagrammalar blokining turli nuqtalaridan o'tkazuvchi ma'lumotlarini ob'yekt terminaliga ulash imkoniyatingiz bo'lmagan holatda yuzadagi panel ob'yektiga murojaat etish imkonini beradi;
Global o'zgaruvchilar	Глобальные переменные	Global variable	ixtiyoriy tipdagi qiymatlarga (yoki bir necha tipga bir vaqtda) bir necha VU orasida bir vaqtning o'zida bir necha VU baravar

			ishlaganda yoki VIU lar orasida signallarni uzatish imkoniyatingiz bo'lmaganda ularga murojaat etish imkonini beradi;
Umumiy(tarmoq) o'zgaruvchilari	Сетевые переменные	Network variable	xuddi global o'zgaruvchilarga o'xshaydi, lekin ular turli mahalliy va tarmoq ob'yektlari xususiyatlarini o'zgartirishingiz mumkin.
Metdolar bog'lami	Узел методов	Invoke node	Metodlar bog'lami xususiyatlar bog'lamiga o'xshaydi. Metod bog'lamini yuzadagi panelda ishlatish, ma'no jihatidan VIU yoki biror- bir metod yoki funktsiyani ishlatishga o'xshaydi. Xususiyatlar bog'lami bilan metdolar bog'lami o'rtasidagi farq shundaki: metod bog'lami "biror- bir ish bajaradi", xususiyatlar bog'lami singari biror- bir o'zgaruvchi qiymatini almashtirmaydi. Metodlar bog'lamini yaratish uchun diagramma bloki terminali

			ega VIU ni yuzadagi panelga joylashtirganda, siz uning to'plamdan faqat biz a'zosini ko'rasiz.
.Net va ActiveX	.Net и ActiveX	.Net & ActiveX	Microsoft tomonidan yaratilgan ob'ekt tuzilmalari bo'lib, bu tuzilmalar Windows dasturlari o'rtasida muloqot qilish va bir dastur interfeysini boshqa dastur interfeysiga imkonini beradi. Masalan, ActiveX yordamida Microsoft Excel jadvalini Microsoft Word hujjatiga joylashtirishingiz mumkin(bu ko'pincha OLE texnologiya deb ham aytiladi).
Server	Сервер	Server	bu dastur bo'lib, LabVIEW dan tashqarida ishlaydi, boshqarish elementi- bu komponent bo'lib, VU yuzadagi paneliga biz joylashtiramiz
AppleEvents	AppleEvents	AppleEvents	LabVIEW da MacOS X uchun bir qator VU lar mavjud bo'lib, ular Apple (AppleEvents) hodisasi orqali dasturlar orasida aloqani ta'minlaydi. Ushbu VU lar Data Communication »

			Protocols» AppleEvent palitrasida joylashgan.
Kanallar	Каналы	Pipes	Kanallar – bu juda qulay instrument hisoblanib, u odatda mutaxassislar tomonidan dasturlar oʻrtasida maʼlumotlarni uzatish uchun ishlatiladi. Kanal oddiy soʻz bilan aytganda – bu FIFO (First In First Out) navbati boʻlib, u jarayonlar oʻrtasidagi maʼlumotlarni oʻtkazgichi sifatida xizmat qiladi (IPC). Kanllarning ikki turi mavjud: nomlangan va buyruqli.

O'QUV-USLUBIY VA AXBOROT TA'MINOTI

Asosiy adabiyotlar

1. Ibragimov U.M. Fayziev Sh.I. va Soliyeva O.K. "Dasturlash tillari bazasida texnik ilovalar" fanidan darslik. Buxoro nashriyoti. Buxoro. 2019. 172 bet.

2. Douglas S. Learn LabVIEW 2013/2014 Fast. – KS.: SDC Publications, 2015. – 314 p.

3. Trevis Dj, Kring Dj. (PrenticeHall) LabVIEW for everyone. - NJ:Prentice Hall, 2011 – 1236 p.

4. Yik Yang. LabVIEW Graphical Programming Cookbook. Birmingham B3 2PB, UK. Packt Publishing Ltd. 2014. 252 p.

5. Marco Schwartz, Oliver Manickum. Programming Arduino with LabVIEW. Birmingham B3 2PB, UK. Published by Packt Publishing Ltd. 2015. 102 p.

6. Магда Ю. С. LabVIEW: практический курс для инженеров и разработчиков. – М.: ДМК. Пресс, 2012. – 208 с.

7. Цифровая обработка сигналов в LabVIEW: учеб. пособие / под ред. В. П. Федосова. – М.: ДМК Пресс, 2007. – 456 с.

8. Григорьев А. Б. О чем не пишут в книгах по Delphi. – СПб.: БХВ-Петербург, 2008. – 576 е.: ил. + CD-ROM

9. Фленов М.В. «Библия Delphi» БХВ-Петербург, 865 стр, 2004 г.

Qo'shimcha adabiyotlar

1. Васильев А.С. Основы программирования в среде LabVIEW(2015). СПб.: Унив. ИТМО, 2015-386 стр.

2. Баран Е.Д. LabVIEW FPGA. Реконфигурируемые измерительные и управляющие системы. – М.: ДМК, 2010. – 448 стр.

3. Кехтарнаваз Н. Ким Н. Цифровая обработка сигналов на системном уровне с использованием LabVIEW. М.: "Додэка-XXI". 2007 г. 304 стр.

4. Бугырина П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7. М.: ДМК Пресс., 2005. 264 с.

5. Л. Батоврин В.К., Бессонов А. С., Мошкин В. В., LabVIEW: практикум по основам измерительных технологий / под ред. В. К. Батоврина. 2-е изд, переработ. и доп. - М.: Д М К Пресс. - 232 с.

6. Галисеев Г.В. «Компоненты в Delphi 7: Профессиональная работа». Диалектика, 619 стр, 2004 г.

Internet saytlari

1. <http://www.ziyonet.uz>
2. www.ni.com
3. <https://habrahabr.ru>
4. www.intuit.ru
5. bankreferatov.ru
6. <https://labview.izmeril.ru/>
7. <https://cxem.net/software/labview.php>
8. <https://nitech.nstu.ru/labview/>
9. <https://oit.utk.edu/research/research-software/labview/>
10. www.owen.ru
11. <https://www.codesys.com/>
12. <https://www.helpme-codesys.com/>
13. <https://gitconnected.com/learn/android-development>
14. www.udemy.com

MUNDARIJA

KIRISH	3
I MODUL. "DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALAR"	
§1.1. "Dasturlash tillari bazasida texnik ilovalar" fanining maqsadi va vazifalari. labview instrumental vositasi va uning imkoniyatlari	6
§1.2. LabVIEW muhiti bilan tanishish.....	11
§1.3. Elementlarni bog'lash. dasturni otladkalash metodikasi	16
I-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI	38
AMALIY MASHG'ULOT №1.	38
AMALIY MASHG'ULOT №2	44
AMALIY MASHG'ULOT №3	53
AMALIY MASHG'ULOTNI BAJARISH TARTIBI.....	56
I-MODUL BO'YICHA TEST SAVOLLARI.....	63

II-MODUL. TUZILISHLAR YORDAMIDA DASTURLAR BAJARILISHINI BOSHQARISH

§2.1. Tsikllar tuzilmasining ikki turi.....	69
§2.2. Siljuvchan registrlar	74
§2.3. Variant tuzilmasi. Shart bo'yicha tsikl va variant tuzilmasi.....	81
§2.4. Massivlar.....	90
§2.5. Klasterlar. O'zaro almashinuvchan massivlar va klasterlar	100
§2.6. LabVIEW vizual aks etish vositalari.....	109
§2.7. Qatorlarni o'rganish va kiritish/chiqarish ichki uskunalari.....	118
II-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI	125
AMALIY MASHG'ULOT №4	125
AMALIY MASHG'ULOT №5	130
AMALIY MASHG'ULOT №6	132
TAJRIBA MASHG'ULOTI №3	135
TAJRIBA MASHG'ULOTI №4	139
TAJRIBA MASHG'ULOTI №5	141
II-MODUL BO'YICHA TEST SAVOLLARI.....	144

III-MODUL. KOMPYUTERGA MA'LUMOTLARNI KIRITISH CHIQUARISH. MA'LUMOTLARNI YIG'ISH VA LABVIEW USKUNALARINI BOSHQARISH.

§ 3.1. Kompyuterga ma'lumotlarni	153
§3.2. Ma'lumotlarni yig'ish va LabVIEW uskunalari boshqarish.....	162
§3.3. Mahalliy, global va tarmoq o'zgaruvchilari	172
§3.4. Hodisaga asoslangan dasturlash.....	183
§3.5. Diagramma-bloklari fragmentlaridan virtual ichki uskunalarni avtomatik yaratish. Kodlarni olib tashlash tuzilmalari.	188
III. MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI	197
AMALIY MASHG'ULOT №7.....	197
AMALIY MASHG'ULOT №8	203
TAJRIBA MASHG'ULOTI №8.....	210
TAJRIBA MASHG'ULOTI №9.....	212
III-MODUL BO'YICHA TEST SAVOLLARI	216

IV-MODUL. DELPHI DASTURLASH TIZIMI BILAN TANISHISH

§4.1. Delphi tizimi haqida ma'lumot. Delphi visual modeli.....	223
§4.2. Delphi dasturlash tizimi asoslari. Komponentlar bilan ishlash.	230
§4.3. Komponentlar bilan ishlash. Standart komponentlar palitrasi.	241
§4.4. Ishchi ilovalarni yaratish	247
§4.5. Ob'yektga yo'naltirilgan dasturlash asoslari usuli	252
§4.6. Qo'shimcha komponentlar tahlili	259
IV-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI	270
AMALIY MASHG'ULOT №9	270
AMALIY MASHG'ULOT №10	275
AMALIY MASHG'ULOT №11	279
TAJRIBA MASHG'ULOTI №10	290
TAJRIBA MASHG'ULOTI №11	296
IV-MODUL BO'YICHA TEST SAVOLLARI	301

V-MODUL. OB'EKTGA YO'NALTIRILGAN DASTURLASHDA KENGAYTIRILGAN TEXNOLOGIYALARIDAN FOYDALANISH

§ 5.1. Ob'yektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan ishlash.....	309
§5.2. Ob'yektga yo'naltirilgan dasturlashda ma'lumotlar bazasi bilan ishlash	318
§5.3. Hisobotlarni tayyorlash.....	324

§5.4. Oqimlar bilan ishlash.....	331
§5.5. Dinamik ulanadigan kutubxonalar.....	334
§5.6. Ob'yektga yo'naltirilgan dasturlashda buffer xotiradan foydalanish	338
V-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI.....	343
AMALIY MASHG'ULOT №12	343
AMALIY MASHG'ULOT №13	348
AMALIY MASHG'ULOT №14	353
TAJRIBA MASHG'ULOTI №12.....	357
TAJRIBA MASHG'ULOTI №13.....	360
TAJRIBA MASHG'ULOTI №14.....	364
V-MODUL BO'YICHA TEST SAVOLLARI	369

VI-MODUL. DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALARNI YARATISHNING BOSHQA TIZIMLARI

§6.1. CodeSys tizimi bilan tanishish. CodeSys tizimida signallarni kiritish va chiqarish.....	378
§6.2. CodeSys tizimida texnik ilovalarda mantiqiy amallarni qo'llash.....	394
§6.3. Java dasturlash tili bilan tanishish	397
§6.4. Android tizimi uchun dasturlarni yaratish	409
VI-MODUL BO'YICHA AMALIY VA TAJRIBA MASHG'ULOTLARI	415
AMALIY MASHG'ULOT №15	415
TAJRIBA MASHG'ULOTI №15	419
TAJRIBA MASHG'ULOTI №16	425
VI-MODUL BO'YICHA TEST SAVOLLARI	432
TEST JAVOBLARI.....	437
O'QUV-USLUBIY VA AXBOROT TA'MINOTI.....	446

U.M.Ibragimov

DASTURLASH TILLARI BAZASIDA TEXNIK ILOVALAR

DARSLIK

Texnik muharrir:
Musahhih:
Sahifalovchi:

G.Samiyeva
M.Raximov
M.Arslonov



Nashriyot litsenziyasi AI № 178. 08.12.2010. Original –
maketdan bosishga ruxsat etildi: 26.02.2024. Bichimi 60x84.
Kegli 16 shponli. «Palatino Linotype» garn. Ofset bosma
usulida. Ofset bosma qog'ozi. Bosma tabog'i 28,25 Adadi 50.
Buyurtma № 10.



«Sharq-Buxoro» MCHJ bosmaxonasida chop etildi.
Buxoro shahar O'zbekiston Mustaqilligi ko'chasi, 70/2 uy.

Tel: 0(365) 222-46-46



ISBN 978-9910-04-302-4



9 789910 043024