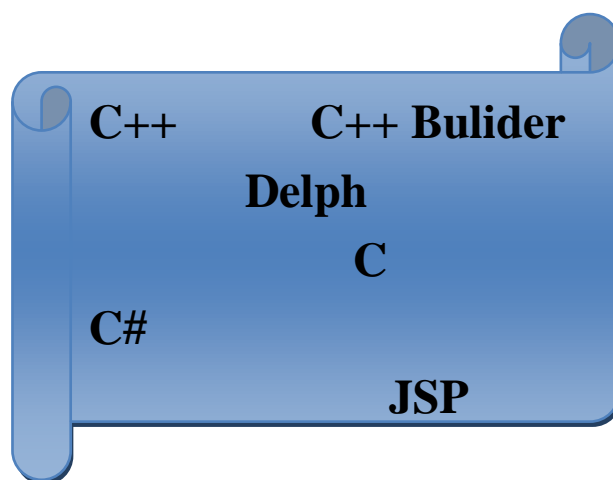


**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI**

Samarqand davlat universiteti

Jabbarov J.S.

PROGRAMMALASH ASOSLARI



O'quv-uslubiy majmua

5130200 – Amaliy matematika va informatika

Samarqand-2019

O'zbekiston respublikasi oliy va o'rta maxsus ta'lim vazirligi

Samarqand davlat universiteti

Amaliy matematika va informatika fakulteti

«Axborotlashtirish texnologiyalari» kafedrasida

Tasdiqlayman:

Ro'yxatga olindi	O'quv ishlari bo'yicha prorektor
№ _____	_____ prof. A. Soleev «__»
_____ 2019 y.	«__» _____ 2019 y.

”Programmash asoslari” fanidan o'quv uslubiy majmua (2-kurslar uchun)

Bilim sohasi:	100000 –Gumanitar soha
Ta'lim sohasi:	130 000 –Matematika
Ta'lim yo'nalishi:	5130200 – Amaliy matematika va informatika

Tuzuvchi: SamDU Amaliy matematika va informatika fakulteti
”Axborotlashtirish texnologiyalari” kafedrasida
assistenti, Jabbarov J.S.

Kafedra mudiri: prof. Jumanov I.I.

Fakultet dekani: dots.A.I.Babayarav

Samarqand – 2019

MUNDARIJA:

1. **Sillabus (yo‘nalishning namunaviy va ishchi o‘quv rejasi, fanning namunaviy va ishchi o‘quv dasturi**
2. **O‘tilayotgan fanning asosiy nazariy material (ma‘ruzalar matni).....**
3. **Glossariy**
4. **Foydalanilgan adabiyotlarning elektron shakli.....**
5. **Mavzular bo‘yicha taqdimotlar, mustaqil ta‘lim uchun materiallar (ilmiy maqolalarva boshqa manbalar).....**
6. **Amaliy va laboratoriya mashg‘ulotlar materiallari**
7. **Qo‘shimcha materiallar (videolar, keys-stadilar va boshqalar).....**

Annotatsiya

2002 yilda 30 mayda O'zbekiston Respublikasi Prezidenti tomonidan PF №3080 «Kompyuterlashtirish va axborot – kommunikasiya texnologiyalarini yanada rivojlantirish xaqidagi» qarori va «Kompyuterlashtirish va axborot – kommunikasiya texnologiyalarini 2002-2010 yillargacha rivojlantirish dasturi»ga asosan respublikamizda kompyuter va axborot texnologiyalarini rivojlantirish, ularni xalq xo'jaligida samarali qo'llash – dolzarb masalaga aylangan. Telekommunikasiya to'rlari, berilganlarni uzatish, Internet xizmatlariga kirish vositalari rivojlanib takomillashtirilmoqda.

Hozirgi kunda fan-texnikani jadal suratlar bilan rivojlanishi natijasida turli murakkab jarayonlarni, ularni matematik nuqtai nazardan tasavvur qilish, modellari tuzish, algoritmlar va programma ta'minotini yaratish nafaqat nazariy jihatdan, balki amaliy jihatdan ham dolzarb bo'lgan muammolardan biri hisoblanadi. Fan nazariy va amaliy qismlardan iborat bo'lib, C++ Bulider programmalash tilida matematika masalalarini yechish uchun mo'ljallangan programmlar tuzish bo'yicha boshlang'ich tushunchalarini o'z ichiga olgan bo'limlaridan tashkil topgan.

"Programmalash asoslari" fani tabiiy fundamental fanlar majmuasiga talluqli bo'lib, talabalar uni III, IV semestrlarda o'rganishadi.

"Programmalash asoslari" fanining bosh maqsadi talabalarga qo'yilgan masalani yechuvchi kompyuter programmasini tuzishga o'rgatishdir. Bu maqsadda Programmalash asoslari haqida umumiy tushunchalar berish va bu tillardan foydalanishga o'rgatish hamda ixtisoslik fanlarini o'rganishga tayyorlashdan iborat. Berilgan masalalarni yechish uchun algoritmlar qurish va ular asosida programma tuza olishbir qator fundamental bilimlarni muvaffaqiyatli o'zlashtirish uchun zarur bo'ladigan tayanch bilimlarni beradi.

«**Programmalash asoslari**» fanidan o'quv uslubiy majmua amaliy matematika va informatika fakultetida ta'lim olayotgan Amaliy matematika va informatika ta'lim yo'nalishi 2-kurs talabalari uchun mo'ljallangan.

Sillabus

“Programmash asoslari” fanining qisqacha tafsivi

OTM ning nomi va joylashgan manzili:	Samarqand davlat universiteti	Universitet hiyoboni, 15				
Kafedra:	Axborotlashtirish texnologiyalari	“Amaliy matematika va informatika” fakulteti				
Ta’lim sohasi va yo’nalishi:	110000 –Pedagogika 130 000 -Matematika	5130200 – Amaliy matematika va informatika				
Fanniolibboradigano’qituvchito’g’risidama’lumot:	ass.Jabbarov J.S.	e-mail:	jamoliddin.88@samdu.uz			
Dars vaqti va joyi:	Amaliy mat va informatika A1 o’quv auditoriyasi	Kursning davomiyligi:	02.09.2019-04.06.2020			
Fanga ajratilgan Soatlar	Auditoriya soatlari				Mustaqil ta’lim	120
	Ma’ruza	72	amaliyot	64		
Fanning boshqa fanlar bilan bog’liqligi (prerek-vizitlari):	“Algoritmlar”, “Matematika”, “Berilganlar bazasini boshqarish tizimlari”					

Fanning mazmuni

Fanning dolzarbligi va qisqacha mazmuni:	<p><u>Fanni o’qitishdan maqsad</u> talabalarga qo’yilgan masalani yechuvchi kompyuter programmasini tuzishga o’rgatishdir. Bu maqsadda Programmash asoslari haqida umumiy tushunchalar berish va bu tillardan foydalanishga o’rgatish hamda ixtisoslik fanlarini o’rganishga tayyorlashdan iborat.</p> <p><u>Fanning vazifasi:</u> berilgan masalalarni yechish uchun algoritmlar qurish va ular asosida programma tuza olish; operasion tizimlar; kompyuterlarning programma ta’minoti; programmalarining turlari; maxsus programma paketlaridan (mutaxassislik bo’yicha) foydalana olish; matematik analiz; amaliy masalalarni modellashtirish; zamonaviy kompyuter texnologiyalari vositalaridan foydalana olish hamda bilim va ko’nikmalariga ega bo’lish fanning asosiy vazifasiga kiradi.</p>
Talabalar uchun talablar	<ul style="list-style-type: none"> - o’qituvchiga va guruhdoshlarga nisbatan hurmat bilan munosabatda bo’lish; - universitet ichki tartib - intizom qoidalariga rioya qilish; - uyali telefonni dars davomida o’chirish; - berilgan uy vazifasi va mustaqil ish topshiriqlarini o’z vaqtida va sifatli bajarish; - ko’chirmachilik (plagiat) qat’iyan man etiladi; - darslarga qatnashish majburiy hisoblanadi, dars qoldirilgan holatda qoldirilgan darslar qayta o’zlashtirilishi shart; - darslarga oldindan tayyorlanib kelish va faol ishtirok etish; - talaba o’qituvchidan so’ng, dars xonasiga - mashg’ulotga kiritilmaydi; - talaba reyting ballidan norozi bo’lsa e’lon qilingan vaqtdan boshlab 1 kun mobaynida apellyasiya komissiyasiga murojat qilishi mumkin
Elektron pochta orqali	Professor-o’qituvchi va talaba o’rtasidagi aloqa elektron pochta

munosabatlar tartibi	orqali ham amalga oshirilishi mumkin, telefon orqali baho masalasi muhokama qilinmaydi, baholash faqatgina universitet hududida, ajratilgan xonalarda va dars davomida amalga oshiriladi. Elektron pochta ochish vaqti soat 14.00 dan 18.00 gacha
----------------------	---

Fan mavzulari va unga ajratilgan soatlar taqsimoti:

t/r	Mavzular nomi	Jami soat	Auditoriya soatlari			Mustaqil ta'lim soatlari
			Ma'ruza	Amaliy mashg'ulot	Laboratoriya	
1.	2.	3.	4.	5.		6.
III–semestr						
1.	Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi.	8	2	2		4
2.	Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari	12	2	2	2	6
3.	Statik metodlar va maydonlar	12	2	2	2	6
4.	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Abstraksiya	8	2	2		4
5.	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Inkapsulasiya	8	2	2		4
6.	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Vorislik	10	2	2	2	4
7.	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Polimorfizm	10	2	2	2	4
8.	Komponentli vizuval programmalash texnologiyasi.Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi.	12	2	2	2	6
9.	Borland C++ muhitida ko'p darchali va bir darchali ilovalar	12	2	2	2	6

10.	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	24	4	4	4	12
11.	Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari	12	2	2	2	6
12.	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	24	4	4	4	12
13.	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	24	4	4	4	12
14.	Borland C++ muhiti System komponentalari paneli ob'yektlari	24	4	4	4	12
III-semest bo'yicha jami		200	36	36	30	98

BAHOLASH MEZONLARI

“Programmash asoslari» fanidan talabalar bilimni baholash O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirining 2018 yil 9 avgustdagi 19-2018-son buyrug'i bilan tasdiqlangan **“Oliy ta'lim muassasalarida talabalar bilimni nazorat qilish va baholash tizimi to'g'risida”**gi nizom asosida amalga oshiriladi.

Mazkur fandan talabalar bilimni nazorat qilish **oraliq va yakuniy nazorat** turlarini o'tkazish orqali amalga oshiriladi.

Talabalar bilimni baholash **5 baholik tizimda** amalga oshiriladi.

Talabalarning bilimi quyidagi mezonlar asosida:

talaba mustaqil xulosa va qaror qabul qiladi, ijodiy fikrlay oladi, mustaqil mushohada yuritadi, olgan bilimni amalda qo'llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo'yicha tasavvurga ega deb topilganda — 5 (a'lo) baho;

talaba mustaqil mushohada yuritadi, olgan bilimni amalda qo'llay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo'yicha tasavvurga ega deb topilganda — 4 (yaxshi) baho;

talaba olgan bilimni amalda qo'llay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo'yicha tasavvurga ega deb topilganda — 3 (qoniqarli) baho;

talaba fan dasturini o'zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) bo'yicha tasavvurga ega emas deb topilganda — 2 (qoniqarsiz) baho bilan **baholanadi**.

**Baholashni 5 baholik shkaladan 100 ballik shkalaga o'tkazish
JADVALI**

5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala
5,00 — 4,96	100	4,30 — 4,26	86	3,60 — 3,56	72
4,95 — 4,91	99	4,25 — 4,21	85	3,55 — 3,51	71
4,90 — 4,86	98	4,20 — 4,16	84	3,50 — 3,46	70
4,85 — 4,81	97	4,15 — 4,11	83	3,45 — 3,41	69
4,80 — 4,76	96	4,10 — 4,06	82	3,40 — 3,36	68
4,75 — 4,71	95	4,05 — 4,01	81	3,35 — 3,31	67
4,70 — 4,66	94	4,00 — 3,96	80	3,30 — 3,26	66
4,65 — 4,61	93	3,95 — 3,91	79	3,25 — 3,21	65
4,60 — 4,56	92	3,90 — 3,86	78	3,20 — 3,16	64
4,55 — 4,51	91	3,85 — 3,81	77	3,15 — 3,11	63
4,50 — 4,46	90	3,80 — 3,76	76	3,10 — 3,06	62
4,45 — 4,41	89	3,75 — 3,71	75	3,05 — 3,01	61
4,40 — 4,36	88	3,70 — 3,66	74	3,00	60
4,35 — 4,31	87	3,65 — 3,61	73	3,0 dan kam	60 dan kam

Asosiy adabiyotlar:	<ol style="list-style-type: none"> 1. A. R.Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O`quv qo`llanma. T.:“Cho`lpon”, 2010 y. 2. A. R.Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O`quv qo`llanma. T.:“Cho`lpon”, 2013 y. 3. Sh.I.Razzoqov, M.J.Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o`quv qo`llanma. T.: “Ilim Ziyo”, 2011y. 4. T.X.Holmatov, N.I.Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta`minoti.O`quv qo`llanma. T.: “Mexnat”, 2000 y. 5. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T.:“O`qituvchi”, 2008 y. 6. M.Ashurov, M.Mirmaxmudov, SH.Sapaev.Zamonaviy Programmalash asoslari fanidan laboratoriya ishlari.T.:TDPU, 2008y. 7. Страуструп Б. Язык программирования C++. Специальное издание. – М.: Бинам – ПРЕСС, 2006, 1104 с 8. Павловская Т. C++ Программирование на языке высокого уровня. – СПб. : Питер, 2005. -461 с. 9. Мадрахимов Ш., Гайназаров С. C++ тилида программалаш асослари: услубий кулланма – Тошкент: УзМУ 2009 . – 196 б 10. Madraximov Sh. va boshq. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma. - Toshkent: O'ZMU, 2014.-160 b
Qo'shimcha	1. П.Дарахвелидзе, Э.Марков. Программирование в C++

<p>adabiyotlar:</p>	<p>Bulider7. Учебник.Санкт-Петербург, “БХВ-Петербург” 2003 г.</p> <p>2. В.М.Пестиков, А.Н.Маслобоев. Turbo PASCAL 7.0. Изучаем на примерах. Санкт-Петербург.: “БХВ-Петербург”, 2004 г.</p> <p>3. Фаронов В.В. Программирование на языке высокого уровня С++ Bulider. Учебник. М.: “Питер”, 2003 г.</p> <p>4. В.Т.Безручко. Практикум по курсу информатики. М.: «Финансы и статистика», 2004 г.</p> <p>5. Культин Н. С ++ Builderв задачах и примерах. – СПб. : ВХБ, 2005.-336с</p> <p>6. Ахатов А., Qarshiyev Н. С++ tilida program-malash(misollar va masalalar). 2016.SamDU nashri. 140b</p>
<p>Internet va Ziyonet saytlari</p>	<p>1. http://cppstudio.com – С++ тилида программалаш бўйича намуналар изохлари билан келтирилган</p> <p>2. http://cplusplus.com – С++ тилида мавжуд конструкциялар таърифи, ишлатиш намуналари билан келтирилган.</p> <p>3. http://www.compteacher.ru/programming - дастурлаш бўйича видео дарсликлар мавжуд.</p> <p>4. http://www.intuit.ru – интернет университет, дастурлаш бўйича ёзма ва видео маърузалар ўқиш, тест синовларидан ўтиш ва сертификат олиш имконияти мавжуд.</p> <p>5. http://www.ziyonet.uz – дастурлаш асослари бўйича рефератлар топиш мумкин.</p>

O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM
VAZIRLIGI

SAMARQAND DAVLAT UNIVERSITETI
AMALIY MATEMATIKA VA INFORMATIKA FAKULTETI
«AXBOROTLASHTIRISH TEXNOLOGIYALARI» KAFEDRASI

TASDIQLAYMAN:
O‘QUV ISHLARI BO‘YICHA
PROREKTOR
A.S. SOLEEV

«_____» _____ 2019 y.

**2-bosqich bakalavriatura talabalari
uchun PROGRAMMALASH
ASOSLARI**

fanidan

Bilim soxasi: 100000 – Gumanitar soha
Ta’lim sohasi: 130 000 – Matematika
Ta’lim yo‘nalishi: 5130200 – Amaliy matematika va informatika

SAMARQAND –2019

Fanning ishchi o‘quv dasturi Samarqand davlat universitetida o‘quv, ishchi o‘quv reja va o‘quv dasturiga muvofiq ishlab chiqildi

Tuzuvchi:

Meliyev F., Jabborov J. SamDU, «Axborotlashtirish texnologiyalari» kafedrasida assistentlari

Taqrizchilar:

Qobilov S. S. – SamDU «Axborotlashtirish texnologiyalari» kafedrasida dotsenti, t.f.n.

O‘rinboyev E. – SamDU «Matematik modellashtirish» kafedrasida dotsenti, t.f.n.

Fanning ishchi o‘quv dasturi SamDU «Axborotlashtirish texnologiyalari» kafedrasining 2019 yil “___” _____ dagi “___” - son yig‘ilishida muhokamadan o‘tgan va fakultet kengashida muhokama qilish uchun tavsiya etilgan.

Kafedra mudiri: _____ **I.I. Jumanov**

Fanning ishchi o‘quv dasturi Amaliy mexanika va informatika fakultet kengashida muhokama etilgan va foydalanishga tavsiya qilingan (2019 yil “___” _____ dagi “___” - sonli bayonnoma)

Fakultet kengashi raisi: _____ **A.Babayarov**

O‘quv-uslubiy boshqarma boshlig‘i: _____ **B.Aliqulov**

Kirish

“Programmash asoslari” fanining bosh maqsadi talabalarga qo‘yilgan masalani echuvchi kompyuter programmasini tuzishga o‘rgatishdir. Shu maqsadda programmash tillari va muhitlari haqida umumiy tushunchalar beriladi va bu tillardan foydalanishga o‘rgatiladi.

Fan nazariy va amaliy qismlardan iborat. Nazariy qism informatika va hisoblash texnikasi, algoritmlar, C/C++ programmash tili, Assembler tili, Borland C++ Ob‘yektga yo‘naltirilgan programmash muhiti boblaridan tashkil topgan.

Dasturda programmashga kirishning nazariy asosi bo‘lgan algoritmlarga alohida e‘tibor qaratilgan. Bu erda algoritmlarni tavsiflash va keyinchalik kompyuterda amalga oshirish uchun zarur bo‘lgan bir qator matematik tushunchalar - takrorlash, yordamchi algoritm, rekursiya, xotira, massiv, indeks, parametr va h.k. kiritilib, turli xil sinf masalalarining algoritmlari tuziladi.

Programmash tili - tuzilgan algoritmni kompyuter amalga oshirish uchun vositadir. Bu o‘rinda turli murakkablikdagi sintaksis va semantikaga ega bo‘lgan tillardan foydalanish mumkin. Fanda C++ tilining alfaviti, til qurilmalarining umumiy sintaksisi, berilganlar turlari, operatsiyalar va funktsiyalar qaraladi, oqim bilan ishlash, ko‘rsatgichlar, amallarni qayta yuklashlar, funktsiyalarni qayta yuklash, qoliplar va modulli programmash va grafika moduli bilan ishlash o‘rgatiladi.

Hozirda mavjud programmash tillari ikki toifaga bo‘linadi, birinichisiga Assembler tili kirsa, ikkinchisiga qolgan barcha tillar kiradi. Yuqori bosqich tillari u yoki bu tadbqiqiy masalalarni (iqtisodiy, matematik va hakoza) echishga moslashgan bo‘lib, ular bunday masalalarni echish algoritmini ifodalash vositalarini o‘z ichiga oladi. Assembler esa protsessor arxitekturasi aksini, ya‘ni uning mantiqiy qurilmalari, ish holatlari va amal qilish sxemalarini o‘zida akslantiradi. Assemblerda yozilgan programma yuqori samaradorligi, minimal hajmi va bajarilishining maksimal tezkorligi bilan ajralib turadi. Shuning uchun Assembler tezlik va xotiraga cheklovlar qo‘yilgan hollarda, qurilmalarga “qattiq” bog‘angan drayverlar yaratishda va kompyuterga nostandart apparat qurilmalarini ulash masalalarida qo‘l keladi. Assembler tilini o‘rganish metodik nuqtai-nazardan ham ahamiyatlidir. Assembler tilini o‘rganish – protsessor, va umuman kompyuter ishlashini to‘liq o‘rganish, C/C++ va boshqa tillarda programma tuzuvchi uchun yuqori bosqich tillar formalizmlari “ortida” qanday jarayonlar ro‘y berayotganligini angal imkoniyatini beradi.

Protsessorning himoyalangan rejimida (Windows OSda) amal qiluvchi 32-razryadli protsessorlar printsiplial jihatdan yangi programmash texnologiyalarini yuzaga kelishiga sabab bo‘ldi. Ob‘yektga yo‘naltirilgan programmashga asoslangan C++ ushbu talabalarga javob beruvchi programmash tilidir. C++ tili asosida Borland C++ vizual programmash muhiti yaratildi va dunyoda o‘zining salmoqli o‘rniga ega bo‘ldi. Borland C++ muhitida programmash Ob‘yektga

yoʻnaltirilgan programmalashning nazariy asoslarini tushuntirish, sinflar va ular shajarasini hosil qilish, vizual programmalash muhiti, shakl, komponentalar, ilovalar yaratish va shu kabi tayanch tushunchalar haqida maʼlumot berish va ular bilan ishlash saboqlarini berish orqali amalga oshiriladi.

Programmalashga oʻrgatish mos masalalarga tayangan holda olib boriladi. Fanning amaliy qismida algoritmlar, C++ va Assembler programmalash tili, Borland C++ muhitida ishlash boʻyicha masalalarni oʻz ichiga oladi.

Programmalash asoslari fani bevosita va bilvosita Tizimli programmalash, Berilganlar bazasini boshqarish tizimlari, Kompyuter grafikasi va Kompyuter tarmoqlari fanlari bilan uzviy bogʻliqdir va bu fan kompyuter texnologiyalari boʻyicha mutaxassis tayyorlashda umumiy asos rolini oʻynaydi.

Fanning maqsad va vazifalari

«Programmalash asoslari» fani 1-2 bosqichning I-IV semestrlarida oʻrganiladi.

Fanni oʻqitishdan maqsad – Amaliy matematika va informatika yoʻnalishining bakalavr bosqichi talabalariga programmalash asoslarini etarli darajada oʻqitish, shu bilimlarga tayangan holda kompyuter modellashtirishga keladigan tadbqiqiy masalalarning programma taʼminotini amalga oshirishga oʻrgatish va ixtisoslik fanlarini oʻzlashtirishda tayanch bilimlarga ega boʻlish.

Fanning vazifalari: Masala echishning algoritmik asoslarini oʻrganish, kompyuter ishlashining tamoili, programmalash tillari sinflash, kompyuterda berilganlar va buyruqlarni tasvirlanishi, C++ tilida programmalash, Assembler tili, Obʼyektga yoʻnaltirilgan programmalash texnologiyalari, vizual programmalash muhitida ishlash bu fanning asosiy vazifalari hisoblanadi.

Fan boʻyicha talabalarning malakasiga quyiladigan talablar

«Programmalash asoslari» fanini oʻrganish jarayonida bakalavr quyidagilarni bajara olish lozim:

- C++ programmalash tili. Til alfaviti, sintaksisi va semantikasi, programma tuzilishi, berilganlarning asosiy turlari, qiymat berish, tarmoqlanuvchi va takrorlash operatorlari, funktsiyalar, koʻrsatgichlar, oʻqish-yozish oqimlari, favqulotda holatlarni qayta ishlash, sinflar, Obʼyektga yoʻnaltirilgan programmalash texnologiyalari, konteynerlarni tushuniishi va ularni qoʻllagan holda tadbqiqiy masalalarni echa olishi kerak;
- Obʼyektga yoʻnaltirilgan va vizual programmalash. Sinflar, sinf maydonlari va metodlari, konstruktor va destruktorlar, vorislik, virtual metodlar, ilova yaratish ustasi. konsol va dialog ilovalari. Koʻpdarchali va bir darchali

ilovalar, xabarlar va buyruqlar, WIN32 tizimi xabarlar xaritasini ishlatish, grafik interfeys bilan ishlash, Windows ilovalarini yarata olishi kerak.

O‘quv rejadagi boshqa fanlar bilan bog‘liqligi

Mazkur fanni o‘zlashtirish Diskret matematika, Matematik mantiq va Algoritm nazariyasi fanlari bilan uzviy bog‘langan. Fan mazmuni Informatika sohasidagi Tizimli programmalash, Operatsion tizimlari, Web programmalash, Berilganlar bazasini boshqarish tizimlari, Tadbiqiy masalalarni matematik modelashtirish tizimlari (Mathcad, Maple, Statistika va boshqalar) o‘rganish uchun tayanch hisoblanadi.

Fanni o‘qitishda zamonaviy axborot va pedagogik texnologiyalar

Talabalarning fanni muvafaqiyatli o‘zlashtirishi uchun o‘qitishning ilg‘or va zamonaviy usullaridan foydalanish, yangi axborot-pedagogik texnologiyani tadbiq etish muhim ahamiyatga egadir. Fanni o‘zlashtirishda darslik, o‘quv va uslubiy qo‘llanmalar, ma‘ruza matnlari, tarqatma materiallar, elektron materiallar, plakatlardan foydalaniladi.

Ma‘ruza va amaliy mashg‘ulotlarda mos ravishda ilg‘or pedagogik va kompyuter texnologiyalardan foydalaniladi.

O‘quv jarayonida fanni o‘tish sifatini belgilovchi quyidagi holatlar e‘tiborga olinadi: yuqori ilmiy darajada dars berish, muammoli ma‘ruzalar o‘qish, darslarni savol-javob tarzda qiziqarli tashkil qilish, ilg‘or pedagogik texnologiyalardan va multimedia vositalardan foydalanish, tinglovchilarni undaydigan, o‘ylantiradigan muammolarni ular oldiga quyish, erkin muloqot yuritishga, ilmiy izlanishga jalb qilish.

«Programmalash asoslari» kursini loyihalashtirishda quyidagi asosiy konseptual yondoshuvlardan foydalaniladi:

Shaxsga yo‘naltirilgan ta‘lim. Bunda kelgusidagi mutaxassis faoliyati bilan bog‘liq o‘qitish, masalalar, mavzular ishchi dasturda ko‘rilishi kerakligi nazarda tutilgan.

Tizimli yondoshuv. “Amaliy matematika va informatika” ta‘lim yo‘nalishining barcha belgilari mujassam etilishi, barcha fanlarning o‘zaro bog‘langanligi va ta‘lim texnologiyasining yaxlitligi nazarda tutilgan.

Faoliyatga yo‘naltirilgan yondoshuv. Mazkur dasturda kelgusidagi mutaxassis sifatlarini shakllantirish, aktivlashtirish va uning barcha qobiliyati va tashabbuskorligini ochishga etibor berilgan.

Dialogik yondoshuv. Fanning amaliyot darslarida shaxsning o‘z-o‘zini faollashtirish, o‘zini ko‘rsata olish kabi ijodiy faoliyatlarini rivojlantirish nazarda tutilgan.

Xamkorlikdagi ta’limni tashkil qilish. Talabalarning quyilgan masala echimlarini olishda birgalikdagi ishlashni joriy etish zarurligi e’tiborga olingan.

Muammoli ta’lim. Ta’lim oluvchi faoliyatini aktivlashtirish uchun fan dasturi bilan bog‘liq qiziqarli mavzular muhokama qilinishligi, bunda ilmiy bilimning obektiv qarama-qarshiligi, uni hal etish usullari, amaliy faoliyatga ularni qo‘llash masalalarni muhokama qilish nazarda tutilgan.

Axborotni taqdim qilishning zamonaviy vositalari va usullarini qo‘llash – yangi kompyuter va axborot texnologiyalarni o‘quv jarayoniga qo‘llash.

O‘qitishning mavzulari va texnikasi. Ma’ruza, muammoli ta’lim, keys-texnologiya, pinbord, paradoks va loyihlash usullari, amaliy ishlar.

O‘qitishni tashkil etish shakllari. Dialog, muloqot, xamkorlik, o‘zaro o‘rganishga asoslangan frontal, kollektiv va guruh.

O‘qitish vositalari. Darslik, ma’ruza matni, elektron kitob, elektron o‘quv qo‘llanmalar, elektron o‘yinlar va shu bilan bir qatorda kompyuter va axborot texnologiyalari.

Kommunikasiya usullari. Tinglovchilar bilan operativ teskari aloqaga asoslangan bevosita o‘zaro munosabatlar.

Teskari aloqa usullari va vositalari: kuzatish, blis-so‘rov, oraliq, joriy, yakuniy nazorat tahlili.

Boshqarish usullari va vositalari: o‘quv mashg‘uloti bosqichlarini belgilab beruvchi texnologik xarita ko‘rinishidagi o‘quv mashg‘ulotlarini rejalashtirish, quyilgan maqsadga erishishda o‘qituvchi va tinglovchining birgalikdagi xarakati, auditoriya mashg‘ulotlari va mustaqil ishlar nazorati.

Monitoring va baholash. Kurs ohirida test topshiriqlari yoki yozma ish varinatlari bo‘yicha talabalar bilimlari baholanadi.

Ayrim mavzular bo‘yicha talabalar bilim baholash test asosida va kompyuter yordamida bajariladi. Internet tarmog‘idagi rasmiy iqtisodiy ko‘rsatkichlaridan foydalaniladi, tarqatma materiallar tayyorlanadi, tayanch so‘z va iboralar asosida oraliq va yakuniy nazoratlar o‘tkaziladi.

«Programmash asoslari» fanidan mashg'ulotlarning mavzular va soatlar bo'yicha taqsimlanishi

t/r	Mavzular nomi	Jami soat	Auditoriya soatlari			Mustaqil ta'lim soatlari
			Ma'ruza	Amaliy mashg'ulot	Laboratoriya	
1.	2.	3.	4.	5.		6.
III–semestr						
1.	Ob'yektga mo'ljallangan programmash asoslari. Ob'yekt va class tushunchasi.	8	2	2		4
2.	Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari	12	2	2	2	6
3.	Statik metodlar va maydonlar	12	2	2	2	6
4.	Ob'yektga mo'ljallangan programmashning asosiy prinsiplari. Abstraksiya	8	2	2		4
5.	Ob'yektga mo'ljallangan programmashning asosiy prinsiplari. Inkapsulatsiya	8	2	2		4
6.	Ob'yektga mo'ljallangan programmashning asosiy prinsiplari. Vorislik	10	2	2	2	4
7.	Ob'yektga mo'ljallangan programmashning asosiy prinsiplari. Polimorfizm	10	2	2	2	4
8.	Komponentli vizual programmash texnologiyasi. Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi.	12	2	2	2	6
9.	Borland C++ muhitida ko'p darchali va bir darchali ilovalar	12	2	2	2	6
10.	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	24	4	4	4	12

11.	Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari	12	2	2	2	6
12.	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	24	4	4	4	12
13.	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	24	4	4	4	12
14.	Borland C++ muhiti System komponentalari paneli ob'yektlari	24	4	4	4	12
	III-semest bo'yicha jami	200	36	36	30	98
IV - semestr						
1.	Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari	20	4	4	2	10
2.	Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari	20	4	4	2	10
3.	Borland C++ muhiti Samples komponentalari paneli ob'yektlari	18	4	4	2	8
4.	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	34	6	6	6	16
5.	Borland C++ muhiti ADO komponentalari paneli ob'yektlari	24	4	4	4	12
6.	Borland C++ muhiti ADO komponentalari paneli ob'yektlari. So'rovlar yaratish	24	4	4	4	12
7.	Borland C++ grafik muhitida ishlash	12	2	2	2	6
8.	Borland C++ grafik muhitida Chart komponentasi bilan ishlash	12	2	2	2	6
9.	Borland C++ grafik muhitida DBChart komponentasi bilan ishlash	12	2	2	2	6
10.	Borland C++ muhiti QReport komponentalari paneli ob'yektlari	24	4	4	4	12
	IV-semest bo'yicha jami	200	36	36	30	98
	Kurs bo'yicha jami	400	72	72	60	196

ASOSIY QISM

Fanning uslubiy jihatdan uzviy ketma-ketligi

Asosiy qismda (ma'ruza) fanni mavzulari mantiqiy ketma-ketlikda keltiriladi. Har bir mavzuning mohiyati asosiy tushunchalar va tezislar orqali ochib beriladi. Bunda mavzu bo'yicha talabalarga DTS asosida etkazilishi zarur bo'lgan bilim va ko'nikmalar to'la qamrab olinishi kerak.

Asosiy qism sifatiga qo'yiladigan talab mavzularning dolzarbligi, ularning ish beruvchilar talablari va ishlab chiqarish ehtiyojlariga mosligi, mamlakatimizda bo'layotgan ijtimoiy-siyosiy va demokratik o'zgarishlar, iqtisodiyotni erkinlashtirish, iqtisodiy-huquqiy va boshqa sohalardagi islohatlarning ustuvor masalalarini qamrab olishi hamda fan va texnologiyalarning so'ngi yutuqlari e'tiborga olinishi tavsiya etiladi.

Ma'ruza(nazariy) mashg'ulotlari mavzulari

Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi. Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Statik metodlar va maydonlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Abstraksiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektgamo'ljallangan programmalashning asosiy prinsiplari.Inkapsulasiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Vorislik

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Polimorfizm. Tiplarni C stiliga keltirish amali. `const_cast`, `dynamic_cast`, `static_cast`, `reinterpret_cast` amallari. Tipni dinamik aniqlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Komponentli vizual programmalash texnologiyasi.Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi. Berilganlarni formatlash. Oqimlar bilan almashish metodlari. Fayl va satr oqimlari. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar

Satrlar bilan ishlash. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Standart komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli,klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; Q1; Q3.

Borland C++ muhiti Additional komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti Win32 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti System komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Samples komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; Q1; Q3.

Borland C++ muhiti Office2k komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti ADO komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, guruhlarda ishlash metodi.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida Chart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ grafik muhitida DBChart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; Q1.

Borland C++ muhiti QReport komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Programmash asoslari fani bo'yicha ma'ruza mashg'ulotlarining kalendar tematik rejasi.

№	Ma'ruza mavzusi	Soat
1	2	3
I-semestr		
1	Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi.	2
2	Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari	2
3	Statik metodlar va maydonlar	2
4	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Abstraksiya	2
5	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Inkapsulatsiya	2
6	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Vorislik	2
7	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Polimorfizm	2
8	Komponentli vizual programmalash texnologiyasi. Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi.	2
9	Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar	2

10	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	4
11	Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari	2
12	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	4
13	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	4
14	Borland C++ muhiti System komponentalari paneli ob'yektlari	4
I-semestrjami		36
II-semestr		
1	Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari	4
2	Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari	4
3	Borland C++ muhiti Samples komponentalari paneli ob'yektlari	4
4	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	6
5	Borland C++ muhiti ADO komponentalari paneli ob'yektlari	4
6	Borland C++ muhiti ADO komponentalari paneli ob'yektlari. So'rovlar yaratish	4
7	Borland C++ grafik muhitida ishlash	2
8	Borland C++ grafik muhitida Chart komponentasi bilan ishlash	2
9	Borland C++ grafik muhitida DBChart komponentasi bilan ishlash	2
10	Borland C++ muhiti QReport komponentalari paneli ob'yektlari	4
II semester jami		36
Jami:		72

Amaliy mashg'ulotlarning mavzulari

Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi. Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Statik metodlar va maydonlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Abstraksiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Inkapsulatsiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Vorislik

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Polimorfizm. Tiplarni C stiliga keltirish amali. `const_cast`, `dynamic_cast`, `static_cast`, `reinterpret_cast` amallari. Tipni dinamik aniqlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Komponentli vizual programmalash texnologiyasi. Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi. Berilganlarni formatlash. Oqimlar bilan almashish metodlari. Fayl va satr oqimlari. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar

Satrlar bilan ishlash. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Standart komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; Q1; Q3.

Borland C++ muhiti Additional komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti Win32 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti System komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Samples komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; Q1; Q3.

Borland C++ muhiti Office2k komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti ADO komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, guruhlarda ishlash metodi.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida Chart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ grafik muhitida DBChart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; Q1.

Borland C++ muhiti QReport komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Programmalash asoslari fani bo'yicha amaliy mashg'ulotlarining kalendar tematik rejasi.

№	Amaliy mashg'ulotlar mavzusi	Soat
1	2	3
I-semestr		
1	Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi.	2
2	Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari	2

3	Statik metodlar va maydonlar	2
4	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Abstraksiya	2
5	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Inkapsulatsiya	2
6	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Vorislik	2
7	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Polimorfizm	2
8	Komponentli vizual programmalash texnologiyasi. Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi.	2
9	Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar	2
10	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	2
11	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	2
12	Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari	2
13	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	2
14	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	2
15	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	2
16	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	2
17	Borland C++ muhiti System komponentalari paneli ob'yektlari	2
18	Borland C++ muhiti System komponentalari paneli ob'yektlari	2
I-semestrjami		36
II-semestr		
1	Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari	2
2	Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari	2
3	Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari	2
4	Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari	2
5	Borland C++ muhiti Samples komponentalari paneli ob'yektlari	2
6	Borland C++ muhiti Samples komponentalari paneli ob'yektlari	2

7	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	2
8	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	2
9	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	2
10	Borland C++ muhiti ADO komponentalari paneli ob'yektlari	2
11	Borland C++ muhiti ADO komponentalari paneli ob'yektlari	2
12	Borland C++ muhiti ADO komponentalari paneli ob'yektlari. So'rovlar yaratish	2
13	Borland C++ muhiti ADO komponentalari paneli ob'yektlari. So'rovlar yaratish	
14	Borland C++ grafik muhitida ishlash	2
15	Borland C++ grafik muhitida Chart komponentasi bilan ishlash	2
16	Borland C++ grafik muhitida DBChart komponentasi bilan ishlash	2
17	Borland C++ muhiti QReport komponentalari paneli ob'yektlari	2
18	Borland C++ muhiti QReport komponentalari paneli ob'yektlari	2
		2
II semester jami		36
Jami:		72

Laboratoriya mashg'ulotlarning mavzulari

Ob'yektga mo'ljallangan programmalash asoslari. Ob'yekt va class tushunchasi. Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Statik metodlar va maydonlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Abstraksiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Ob'yektgamo'ljallangan programmalashning asosiy prinsiplari. Inkapsulatsiya.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Vorislik

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari. Polimorfizm. Tiplarni C stiliga keltirish amali. `const_cast`, `dynamic_cast`, `static_cast`, `reinterpret_cast` amallari. Tipni dinamik aniqlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Komponentli vizuval programmalash texnologiyasi. Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi. Berilganlarni formatlash. Oqimlar bilan almashish metodlari. Fayl va satr oqimlari. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar

Satrlar bilan ishlash. Foydalanuvchi tomonidan aniqlanadigan oqimlar va tiplar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Standart komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli, klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; Q1; Q3.

Borland C++ muhiti Additional komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti Win32 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Pog'ona, qadamba-qadam metodi, klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti System komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis, 4x 4 usuli,klaster, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; A4; Q1; Q2; Q3.

Borland C++ muhiti Samples komponentalari paneli ob'yektlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Bingo, blis, menyu, algoritm, munozara, o'z-o'zini nazorat.*

Adabiyotlar: A1;A2; A3; Q1; Q3.

Borland C++ muhiti Office2k komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Blis-so'rov, zigzag usuli, munozara, BBB, Insert, o'z -o'zini nazorat.*

Adabiyotlar: A2; A3; A4; Q1; Q3.

Borland C++ muhiti ADO komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, guruhlarda ishlash metodi.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A2; A3; A4; Q1; Q2.

Borland C++ grafik muhitida Chart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Borland C++ grafik muhitida DBChart komponentasi bilan ishlash

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; Q1.

Borland C++ muhiti QReport komponentalari paneli ob'yektlari

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; Q1; Q2; Q3.

Programmalash asoslari fani bo'yicha laboratoriya mashg'ulotlarining kalendar tematik rejasi.

№	Laboratoriya mashg'ulotlari mavzusi	Soat
1	2	3
I-semestr		
1	Class metodlari va maydonlari. O'zgaruvchilarni ko'rinish sohalari	2
2	Statik metodlar va maydonlar	2
3	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Vorislik	2
4	Ob'yektga mo'ljallangan programmalashning asosiy prinsiplari.Polimorfizm	2
5	Komponentli vizuval programmalash texnologiyasi.Borland C++ muhiti asosiy tushunchalari. Forma tushunchasi.	2
6	Borland C++ muhitida Ko'pdarchali va bir darchali ilovalar	2
7	Borland C++ muhiti Standart komponentalari paneli ob'yektlari	4
8	Borland C++ muhitida ma'lumotlar tiplarini o'zgartirish usullari	2

9	Borland C++ muhiti Additional komponentalari paneli ob'yektlari	4
10	Borland C++ muhiti Win32 komponentalari paneli ob'yektlari	4
11	Borland C++ muhiti System komponentalari paneli ob'yektlari	4
I-semestrjami		30
II-semestr		
1	Borland C++ muhiti Win3.1 komponentalari paneli ob'yektlari	2
2	Borland C++ muhiti Dialogs komponentalari paneli ob'yektlari	2
3	Borland C++ muhiti Samples komponentalari paneli ob'yektlari	2
4	Borland C++ muhiti Office2k komponentalari paneli ob'yektlari	6
5	Borland C++ muhiti ADO komponentalari paneli ob'yektlari	4
6	Borland C++ muhiti ADO komponentalari paneli ob'yektlari. So'rovlar yaratish	4
7	Borland C++ grafik muhitida ishlash	2
8	Borland C++ grafik muhitida Chart komponentasi bilan ishlash	2
9	Borland C++ grafik muhitida DBChart komponentasi bilan ishlash	2
10	Borland C++ muhiti QReport komponentalari paneli ob'yektlari	4
II semester jami		30
Jami:		60

Mustaqil ta'limni tashkil etishning shakli va mazmuni

«Programmash asoslari» fani bo'yicha talabani mustaqil ta'limi shu fanni o'rganish jarayonining tarkibiy qismidir.

Talabalar ayrim mavzularni kengroq o'rganish maqsadida qo'shimcha adabiyotlarni o'qib, referatlar tayyorlaydilar va mashg'ulot rejasi bo'yicha quyilgan masala echimini modellashtirish, algoritmlarini tuzish va dasturiy vositalarini qo'llash bilan bog'liq savollarni yoritadigan loyihalar tayyorlashadi.

Mustaqil ta'lim natijalari reyting tizimi asosida baholanadi. Buning uchun berilgan vazifalarni tekshirish va baholash amaliy mashg'ulot olib boruvchi o'qituvchi tomonidan amalga oshiriladi. Konspektlarni va mavzularni o'zlashtirish darajasini baholash esa, ma'ruza darslarini olib boruvchi o'qituvchi tomonidan bajariladi.

«Programmash asoslari» fani bo'yicha mustaqil ish majmuasi barcha mavzularni qamrab olgan va quyidagi mavzular ko'rinishida shakllantiriladi.

Mustaqil ta'limning mazmuni va hajmi

№	Mustaqil mashg'ulot mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi, soat
III – semestr				
1	Borland C++ muhitida satrlar va ular bilan ishlash amallari	Adabiyotlardan konspekt. Individual topshiriqlar	1,2,3 xaftalar	10
2	Borland C++ muhitida boshqa dasturlarni ishga tushirish	Adabiyotlardan konspekt. Individual topshiriqlar	4,5,6 xaftalar	14
3	Borland C++ muhitida multimedia ob'yektlari bilan ishlash	Adabiyotlardan konspekt. Individual topshiriqlar	7,8,9,10,11xaftalar	30
4	Borland C++ muhitida Office hujjatlari bilan ishlash	Adabiyotlardan konspekt. Individual topshiriqlar	12,13,14,15,16xaftalar	32
5	Borland C++ grafik muhitida Chart komponentasi	Adabiyotlardan konspekt. Individual topshiriqlar	17,18 xaftalar	12
Jami:			98	
IV – semestr				
6	Malumotlar bazasi va uning turlari	Adabiyotlardan konspekt. Individual topshiriqlar	19,20xaftalar	10
7	Malumotlar bazasi ob'yektlari	Adabiyotlardan konspekt. Individual topshiriqlar	21,22xaftalar	12
8	Malumotlar bazasida so'rovlar yaratish	Adabiyotlardan konspekt. Individual topshiriqlar	23,24 xaftalar	12

9	Malumotlar bazasida bir nechta jadvallar bilan ishlash	Adabiyotlardan konspekt. Individual topshiriqlar	25,26 xaftalar	12
10	Borland C++ grafik muhitida DBChart komponentasi	Adabiyotlardan konspekt. Individual topshiriqlar	27,28 xaftalar	10
11	Borland C++ muhiti QReport komponentalari paneli	Adabiyotlardan konspekt. Individual topshiriqlar	29,30,31 xaftalar	14
12	Tarmoq orqali ishlovchi loyihalar yaratish	Adabiyotlardan konspekt. Individual topshiriqlar	32,33,34 xaftalar	14
13	Loyihani web sahifalar bilan bog'lash	Adabiyotlardan konspekt. Individual topshiriqlar	35,36 xaftalar	14
Jami				98

Ishchi o'quv dasturining information uslubiy ta'minoti

Mazkur fanni o'qitish jarayonida ta'limning zamonaviy metodlari, pedagogik, axborot-kommunikatsiya texnologiyalarini qo'llash nazarda tutilgan.

- Nazariy mavzular (ma'ruzalar) zamonaviy kompyuter texnologiyalari yordamida jihozlangan elektron sinfda prezentatsiya va elektron–didaktik materiallardan foydalangan holda o'tkaziladi;

- Amaliy mashg'ulotlar auditoriyada "aqliy hujum" , "guruhli fikrlash" va "loyihani himoya qilish" pedagogik texnologiyalar asosida algoritmlar yaratish, kompyuter sinfida programmani loyihalash, taxlash va ishlash prinsiplarini namoyish etish orqali olib boriladi;

- Mustaqil ta'lim uchun ajratilgan mavzularni muhokamasi seminarlarda ma'ruza qilish, kichik ishchi guruh musobaqasini tashkil qilish va to'garaklarda dasturiy ta'minotni namoyish etish shaklida tashkil qilinadi.

Ushbu fandan talabalar bilimni reyting tizimi asosida baholash mezonlari

Fan bo'yicha reyting jadvallari, nazorat turi, shakli, soni, hamda har bir nazoratga ajratilgan maksimal ball, shuningdek joriy va oraliq nazoratlarning saralash ballari haqidagi ma'lumotlar birinchi mashg'ulotda talabalarga e'lon qilinadi.

Davlat ta'lim standartlariga muvofiq quyidagi nazorat turlari o'tkaziladi.

- **joriy nazorat (JN)** – talabanning fan mavzulari bo'yicha bilim va amaliy ko'nikma darajasini aniqlash va baholash usuli. Joriy nazorat fanning xususiyatidan kelib chiqqan holda amaliy mashg'ulotlarda og'zaki so'rov, test o'tkazish, suhbat, nazorat ishi, kollektivum, uy vazifalarini tekshirish va shu kabi boshqa shakllarda o'tkazilishi mumkin;

- **oraliq nazorat (ON)** – semestr davomida o'quv dasturining tegishli (fanlarning bir necha mavzularini o'z ichiga olgan) bo'limi tugallangandan keyin talabanning nazariy bilim va amaliy ko'nikma darajasini aniqlash va baholash usuli. Oraliq nazorat bir semestrda bir yoki ikki marta o'tkaziladi va shakli (yozma, og'zaki, test va hokazo) o'quv faniga ajratilgan umumiy soatlar hajmidan kelib chiqqan holda belgilanadi;

- **yakuniy nazorat (YaN)** – semestr yakunida muayyan fan bo'yicha nazariy bilim va amaliy ko'nikmalarni talabalar tomonidan o'zlashtirish darajasini baholash usuli. Yakuniy nazorat asosan tayanch tushuncha va iboralarga asoslangan "Yozma ish" shaklida o'tkaziladi.

ON o'tkazish jarayoni kafedra mudiri tomonidan tuzilgan komissiya ishtirokida muntazam ravishda o'rganib boriladi va uni o'tkazish tartiblari buzilgan hollarda, **ON** natijalari bekor qilinishi mumkin. Bunday hollarda **ON** qayta o'tkaziladi.

Oliy ta'lim muassasasi rahbarining buyrug'i bilan ichki nazorat va monitoring bo'limi rahbarligida tuzilgan komissiya ishtirokida YaN ni o'tkazish jarayoni muntazam ravishda o'rganib boriladi va uni o'tkazish tartiblari buzilgan hollarda, YaN natijalari bekor qilinishi mumkin. Bunday hollarda **YaN** qayta o'tkaziladi.

Talabanning bilim saviyasi, ko'nikma va malakalarini nazorat qilishning reyting tizimi asosida talabanning fan bo'yicha o'zlashtirish darajasi ballar orqali ifodalanadi.

Bilimlarni baholash usullari, mezonlari va tartibi	
Baholash usullari	Testlar, yozma ishlar, og'zaki so'rov, prezentasiyalar va h.z.
Fan bo'yicha talabalar bilimni nazorat qilish va baholash	<p style="text-align: center;">Nazorat shakllari</p> <p>Talabalar bilimni nazorat qilish oraliq va yakuniy baholashlar shaklida tashkil etiladi. Oraliq baholashlar "5" baholik tizimda o'tkaziladi. Yakuniy baholashlar esa 100 ballik tizimda o'tkazilib, "5" baholik tizimga o'tkaziladi.</p> <p>Tegishli fan bo'yicha auditoriya soatlari yig'indisi 34 soatdan 68 soatgacha bo'lsa, bitta, 69 soat va undan ortiq bo'lsa ikkita oraliq baholash o'tkazish belgilangan. Ushbu baholashlarning har biridan olingan baholarning o'rtachasi oraliq nazoratlarning umumiy natijasi hisoblanadi.</p> <p>Oraliq baholashlar fan xususiyatidan kelib chiqqan holda so'rovlar, og'zaki savol-javob, nazorat ishlari, yozma ish, test sinovlari</p>

	<p>yoki boshqa ko‘rinishda o‘tkazilishi mumkin. Bunda talabaning darsdagi faolligi, uy bazifalarini bajarishi, mustaqil ta‘lim mavzularini o‘zlashtirishi hisobga olinadi.</p> <p>Laboratoriyalari mavjud fanlardan talaba laboratoriya ishlari topshirib ijodiy baho olgandagina yakuniy baholashga qo‘yiladi.</p> <p>Kurs ishlari mavjud fanlardan talaba kurs ishini yakuniy nazoratga qadar topshirishi shart, aks holda yakuniy nazoratga qo‘yilmaydi.</p> <p>Yozma ish yoki test shaklidagi sinovlar fakultet dekani tomonidan tasdiqlanadigan jadval asosida darsdan tashqari paytlarda, boshqa ko‘rinishdagi sinovlar esa amaliy mashg‘ulot darslari paytida o‘tkaziladi.</p> <p>Talaba oraliq nazoratdan ijobiy (5 (a‘lo), 4 (yaxshi) va 3 (qoniqarli)) baho bilan baholansa yakuniy nazoratga qo‘yiladi. Talaba oraliq nazoratdan 2 (qoniqarsiz) baho olsa yoki oraliq nazoratni topshirmasa yakuniy nazoratga qo‘yilmaydi.</p> <p>Talabaning fan bo‘yicha bahosi yakuniy nazoratda olgan baho bo‘yicha aniqlanadi.</p> <p>Yakuniy nazorat o‘quv ishlari prorektori yoki o‘quv – uslubiy boshqarma boshlig‘i tomonidan tasdiqlagan jadval asosida yozma ish yoki test ko‘rinishida o‘tkaziladi.</p> <p>Takroriy baholashlar “OTM talabalari bilimini hazorat qilish va baholash tartibi to‘g‘risidagi nizom” talablari asosida tashkil etiladi.</p>
Baho	Fan bo‘yicha talabalar bilimni bahalash mezon
5 (a‘lo)	Talaba mustaqil xulosa va qaror qabul qiladi, ijodiy fikrlay oladi, mustaqil mushohada yuritadi, olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda;
4 (yaxshi)	Talaba mustaqil mushohada yuritadi, olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda;
3 (qoniqarli)	Talaba olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda;
2 (qoniqarsiz)	Talaba fan dasturini o‘zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda;

Darslik va o'quv qo'llanmalari ro'yxati

Asosiy adabiyotlar

1. А.Я.Архангельский. Программирование в С++Builder 6. Издательство.- М.:ООО «Бином», 2006.-1140 с.
2. С++Builder Задачи и решения – Издательство Томского политехнического университета. 2010.- 479 с.
3. Основы программирования в среде С++ Builder : Бусько В. Л. [и др.] . – Минск : БГУИР, 2007. –70 с..
4. Павловская Т.С. Щупак Ю.С. С/С++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с
5. Павловская Т.С. Щупак Ю.С. С++. Объектно- ориентированное программирование. Практикум.-СПб.: Питер,2005-265с
6. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования С++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.- 500с.
7. Юров В., Хорошенко С. Assembler: Учебный курс- СПб, “Питер”,2000.- 672с.

Qo'shimcha adabiyotlar

8. Финогенов К.Г. Основы языка Assemblera.-М.: Радио и связь, 2001. - 288 с.
9. Абель П. Assembler для IBM PC и программирования. 1991. М.: “Высшая школа”, 1992.- 447 с.
- 10.Скенлон Л. Персональный ЭВМ IBM PC и XT. Программирование на языке Assemblera. -М.: Радио и связь. 1991.- 336 с.
- 11.Гофман В. Э., Хомоненко А.Д. Delphi 5. - СПб.: БХВ-Санкт-Петербург, 2000. -800с.
- 12.Немнюгин С.А. Turbo pascal, учебник. Изд. Питер., 2001, -496 с.
- 13.Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо-паскаль. (версия 5.5).М.:МАИ,1992.-576с.
- 14.Абрамов С.А.,Гнезделова Капустина Е.Н.и др. Задачи по программированию. - М.: Наука, 1988.
- 15.Вирт Н. Алгоритмы + структуры данных = программа.-М.:Мир,1985.-405с.
- 16.Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург, 2001. под редакцией С.В.Симоновича.
- 17.Нортон П. Программно-аппаратная организация IBM PC.-М.:Мир,1991.- 327с.
- 18.Фигурнов В.Э. IBM PC для пользователя. М.: Финансы и статистика. Юнити. 1997.
- 19.Юров В. Assembler: практикум. -СПб.: Питер, 2002.- 400с.
- 20.Sh.F.Madraximov, U.E.Adamboev, O'zMU, 2005 yil, 145 bet.

Internet va Ziyonet saytlari

1. <http://www.intuit.ru>
2. www.udemy.com
3. www.ziyonet.uz
4. www.dasturchi.uz

Fanning ishchi o'quv dasturi

Kirish

Mazkur ishchi o'quv dasturi bakalavriat yo'nalishi: 5130200 - Amaliy matematika va informatikayo'nalishida taxsil olayotgan talabalarning o'zlashtirishi lozim bo'lgan bilimlari va unga qo'yiladigan talablar asosida tuzilgan bo'lib, bo'lajak fan o'qituvchisi egallashi kerak bo'lgan bilimlar va ko'nikmalar majmuini o'z ichiga oladi:

Programmash asoslari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buruqlar tizimi va operatorlarini, chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar tuzishni, obyektga yo'naltirilgan Programmash asoslari haqidagi, obyektga yo'naltirilgan loyihalash; obyektlarni loyihalash: obyektga yo'naltirilgan dasturlash tili va unda dastur tuzish.

Fanning maqsad va vazifalari

Programmash asoslari fani umumkasbiy fanlar biri bo'lib, III - IV semestrlarda o'qitiladi. Programmash asoslari fanini o'qitishdan **maqsad** – informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar va amalda qo'llash uchun kunikma va makalalarni shakllantirish va rivojlantirishdan iborat. Unda har bir kasb egasining faoliyati kerak bo'lgan tayanch nazariy va amaliy malumotlarni o'z ichiga oladi.

Programmash asoslari fanning **vazifasi**:

-Programmash asoslari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari bilan tanishtirish va amaliy masalalarga dasturlar tuzishni o'rgatish;

-obyektga yo'naltirilgan Programmash asoslari va ular ustida ishlashni o'rgatishdan iborat.

Fan bo'yicha talabaning malakasiga qo'yiladigan talablar

«Programmash asoslari» o'quv fanini o'zlashtirish jarayonida amalga oshiriladigan masalalar doirasida bakalavr:

- algoritm, va uning xossalari, algoritm ijrochilari, algoritmlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar, Programmash asoslari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buyuqlar tizimi va operatorlarini, chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar tuzishni, obyektga yo'naltirilgan Programmash asoslari haqidagi, obyektga yo'naltirilgan loyihalash; obyektlarni loyihalash: obyektga yo'naltirilgan dasturlash tili va unda dastur tuzishni **bilishlari kerak**;

-algoritmning asosiy hossalari, algoritm ijrochilari, algoritmlarni tasvirlash usullari, algoritm turlarini amaliyotga tadbiiq qilishi va ularga doir sodda masalalarni hal qila olishi, Programmalash asoslari va ularning turlari, dasturlash tilining buyuqlar tizimi va operatorlarini qo'llagan holda chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar tuza olish, dasturlashning obyektga yo'naltirilgan paradigmasi, obyektga yo'naltirilgan loyihalash haqidagi masalalarni algoritmini tuza olish **ko'nikmalariga ega bo'lishlari kerak**;

Kasbiy sohasida uchraydigan turli hil masalalarga algoritmlar tuza olishi, algoritmning turlarni farqlay olish, tasvirlash usullariga oid misollar keltira olish, rekursiya va iterasiya, algoritmning murakkabligi tushunchalarni ajrat olishi, samarali algoritmlar ishlab chiqishning asosiy usullari(balansirovka, dinamik dasturlash va boshqalar)ni amaliy qo'llay olishi, biror bir Programmalash asoslari va ularning turlarini farqlay olish, dasturlash tillalari yordamida amaliy masallalarga dasturlar tuza olish, massivlar, grafik operatorlar, satriy kattaliklar bilan ishlash, funksiyalar va proseduralar, yozuvlar, ro'yxatlar, fayllar, modulli dasturlar haqidagi bilimlarni amalda qullay olish, obyektga yo'naltirilgan Programmalash asoslaridan foydalana olish, boshqarish elementlari, oynalar, dialoglar; hodisalar va habarlar, obyektga yo'naltirilgan muhitlarda habarlarni uzatish va ularga ishlov berish, obyektlar iyerarxiyasi asosida dasturlarni loyihalash **malakalariga ega bo'lishlari kerak**.

O'quv rejadagi boshqa fanlar bilan bog'liqligi

Dasturni amalga oshirish o'quv rejasidagi rejalashtirilgan algoritmlar, oliy matematika, fizika, informatika, ma'lumotlar bazasi va ularni boshqarish tizimlari fanlaridan olingan nazariy va amaliy bilimlarga tayanadi.

Fanni o'qitishda zamonaviy axborot va pedagogik texnologiyalar

Talabalarning fanni muvafaqiyatli o'zlashtirishi uchun o'qitishning ilg'or va zamonaviy usullaridan foydalanish, yangi axborot-pedagogik texnologiyani tadbiiq etish muhim ahamiyatga egadir. Fanni o'zlashtirishda darslik, o'quv va uslubiy qo'llanmalar, maruza matnlari, tarqatma materiallar, elektron materiallar, plakatlardan foydalaniladi. Maruza va amaliy mashg'ulotlarda mos ravishda ilg'or pedagogik va kompyuter texnologiyalardan foydalaniladi.

O'quv jarayonida fanni o'tish sifatini belgilovchi quyidagi holatlar etiborga olinadi: yuqori ilmiy darajada dars berish, muammoli maruzalar o'qish, darslarni savol-javob tarzda qiziqarli tashkil qilish, ilg'or pedagogik texnologiyalardan va multimedia vositalardan foydalanish, tinglovchilarni undaydigan, o'ylantiradigan muammolarni ular oldiga quyish, erkin muloqot yuritishga, ilmiy izlanishga jalb qilish.

«Programmalash asoslari» kursini loyihalashtirishda quyidagi asosiy konseptual yondoshuvlardan foydalaniladi:

Shaxsga yo'naltirilgan talim. Bunda kelgusidagi mutaxassis faoliyati bilan bog'liq o'qitish, masalalar, mavzular ishchi dasturda ko'rilishi kerakligi nazarda tutilgan.

Tizimli yondoshuv. Ta'lim texnologiyasi tizimning barcha belgilarini o'zida mujassam etmog'i lozim: jarayonning mantiqiyiligi, uning barcha bo'g'inlarini o'zaro bog'langanligi, yaxlitligi.

Faoliyatga yo'naltirilgan yondoshuv. Mazkur dasturda kelgusidagi mutaxassis sifatlarini shakllantirish, aktivlashtirish va uning barcha qobiliyati va tashabbuskorligini ochishga etibor berilgan.

Dialogik yondoshuv. Fanning amaliyot darslarida shaxsning o'z-o'zini faollashtirish, o'zini ko'rsata olish kabi ijodiy faoliyatlarini rivojlantirish nazarda tutilgan.

Xamkorlikdagi ta'limni tashkil qilish. Talabalarning quyilgan masala yechimlarini olishda birgalikdagi ishlashni joriy etish zarurligi etiborga olingan.

Muammoli ta'lim. Talim oluvchi faoliyatini aktivlashtirish uchun fan dasturi bilan bog'liq qiziqarli mavzular muhokama qilinishligi, bunda ilmiy bilimning obektiv qarama-qarshiligi, uni hal etish usullari, amaliy faoliyatga ularni qo'llash masalalarni muhokama qilish ko'zda tutilgan.

Axborotni taqdim qilishning zamonaviy vositalari va usullarini qo'llash – yangi kompyuter va axborot texnologiyalarni o'quv jarayoniga qo'llash.

O'qitishning usullari va texnikasi. Ma'ruza (kirish, mavzuga oid, vizuallashtirish), muammoli ta'lim, keys-stadi, pinbord, paradoks va loyihalash usullari, amaliy ishlar.

O'qitishni tashkil etish shakllari. Dialog, muloqot, xamkorlik, o'zaro o'rganishga asoslangan frontal, kollektiv va guruh.

O'qitish vositalari. Darslik, maruza matni, elektron kitob, elektron o'quv qo'llanmalar, elektron o'yinlar va shu bilan bir qatorda kompyuter va axborot texnologiyalari.

Kommunikasiya usullari. Tinglovchilar bilan operativ teskari aloqaga asoslangan bevosita o'zaro munosabatlar.

Teskari aloqa usullari va vositalari: kuzatish, blis-so'rov, oraliq, joriy, yakuniy nazorat tahlili.

Boshqarish usullari va vositalari: o'quv mashg'uloti bosqichlarini belgilab beruvchi texnologik xarita ko'rinishidagi o'quv mashg'ulotlarini rejalashtirish, quyilgan maqsadga erishishda o'qituvchi va tinglovchining birgalikdagi xarakati, auditoriya mashg'ulotlari va mustaqil ishlar nazorati.

Monitoring va baholash. Kurs ohirida test topshiriqlari yoki yozma ish varinatlarini bo'yicha talabalar bilimlari baholanadi.

Ayrim mavzular bo'yicha talabalar bilim baholash test asosida va kompyuter yordamida bajariladi. Internet tarmog'idagi rasmiy iqtisodiy ko'rsatkichlaridan foydalaniladi, tarqatma materiallar tayyorlanadi, tayanch so'z va iboralar asosida oraliq va yakuniy baholashlar o'tkaziladi.

“Programmalash asoslari” fanidan mashg'ulotlarning mavzular va soatlar bo'yicha taqsimlanishi:

t/r	Mavzular nomi	Jami soat	Ma'ruza	Amaliy mashg'ulot	Laboratoriya	Mustaqil ta'lim
1.	Ob'yektga yo'naltirilgan	20	4	2	4	10

	Programmash asoslari					
2.	C++ Bulider ob'yektga yo'naltirilgan dasturlash tili	22	4	4	4	10
3.	C++ Bulider dasturlari strukturasi. Loyiha va modul	12	2	-	-	10
4.	C++ Bulider dasturlash tilining operatorlari	30	6	6	8	10
5.	Prosedura va funksiyalar	22	4	4	4	10
6.	C++ Bulider dasturlash tilining grafik vositalari	24	4	4	4	12
7.	C++ dasturlash tili	130	24	20	24	62
Jami		260	48	40	48	124

Asosiy qism: Fanning uslubiy jihatdan uzviy ketma-ketligi

Asosiy qismda (ma'ruza) fanni mavzulari mantiqiy ketma-ketlikda keltiriladi. Har bir mavzuning mohiyati asosiy tushunchalar va tezislar orqali ochib beriladi. Bunda mavzu bo'yicha talabalarga DTS asosida yetkazilishi zarur bo'lgan bilim va ko'nikmalar to'la qamrab olinishi kerak. Asosiy qism sifatiga qo'yiladigan talab mavzularning dolzarbligi, ularning ish beruvchilar talablari va ishlab chiqarish ehtiyojlariga mosligi, mamlakatimizda bo'layotgan ijtimoiy-siyosiy va demokratik o'zgarishlar, iqtisodiyotni erkinlashtirish, iqtisodiy-huquqiy va boshqa sohalaridagi islohatlarning ustuvor masalalarini qamrab olishi hamda fan va texnologiyalarning so'ngi yutuqlari e'tiborga olinishi tavsiya etiladi.

Ma'ruza mashg'ulotlari

Ob'yektga yo'naltirilgan dasturlashning asosiy tushunchalari. Sinf tavsifi. Ob'yektlar. Ob'yektlar sinfi iyerarxiyasi. Amallar va usullar. Maydonlar. Xossalar. Konstruktorlar, destrukturorlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Ma'ruza, namoyish etish, savol-javob, Charxpalak, metodlari.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ob'yektga yo'naltirilgan dasturlashning tamoyillari. ob'yektga yo'naltirilgan loyihalash. Inkapsulyatsiyalash. Merosxo'rlik. Polimorfizm.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, guruhlarda ishlash metodi.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlash tilining ishchi muhiti, undagi oynalar: Ob'ektlarning daraxtsimon ko'rinish oynasi, ob'ektlar inspektori oynasi, kod brauzeri oynasi, asosiy oyna, forma oynasi, dastur kodi oynasi, u o'rnatilishi zarur bo'lgan kompyuterga qo'yiladigan texnik talablar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Komponentlar palitrasi. Palitra bo'limlari va ayrim komponentlar xossalari bilan tanishish. Stanford bo'limi. Frame, MainMenu, PopupMenu, Label, Edit, Button, Memo, Panel komponentlar xossalari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Komponentlar palitrasi. Palitra bo'limlari va ayrim komponentlar xossalari bilan tanishish. Additional bo'limi. BitBtn, MaskEdit, StringGrid, Image komponentlar xossalari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlari strukturasi. Loyiha va modul. Xodisa tushunchasi, loyiha va modul strukturasi, dastur elementlari (alfavit, identifikatorlar, doimiyliklar, ifodalar va amallar.) Bo'sh forma va uning modifikasiyasi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ma'lumotlarning tiplari. Tiplarning ahamiyati, qatoriy (String) va simvulli (Char) tiplar, ularning berilishi va ular bilan bajariladigan amallar, butun tiplar (sodda: tartib va xaqiqiy tiplar, mantiqiy va simvulli tiplar, tip diapazon, vakt-sana tipi).

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Operatorlar va shartli operator. Tarkibiy va bo'sh operatorlar. If...Then...else shartli operatori. Tanlash(case) operatori. Goto o'tish operatori, label (belgilar) xizmatchi so'zidan foydalanish qoidalari bilan tanishish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlash tilida sikl operatorlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlash tilida massivlar, massivlarni tavsiflash. Massivlarni berilish usullari, Array, of, xizmatchi so'zlari, Random (max) funksiyasi bilan tanishtirish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Prosedura va funksiyalar C++ Bulider dasturlash tilining muhim instrumenti sifatida. Prosedura tarifi, uning nomi, undan foydalanish yullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Prosedura va funksiyalar C++ Bulider dasturlash tilining muhim instrumenti sifatida. Funksiya tarifi, uning nomlanishi, undan dasturda foydalanish va uning proseduradan farqi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Fayllar bilan ishlash. Fayllar bilan ishlashning ob'yektli yondashuvi.

Mantiqiy amallar va ifodalar. Shartli va shartsiz o'tish operatorlari. Tanlash operatori.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Buliderda muloqot oynalari. OpenFileDialog, SaveDialog, ColorDialog, FontDialog komponentalari bilan ishlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlash tilining grafik imkoniyatlari, C++ Buliderdagi maxsus TCanvas, TFont, TPen, Tbrush klasslari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

TFont, TPen TBrush klasslarining xossalari: Bitmap, Color, Style.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Ma'ruza, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Buliderda MBni boshqaradigan ilovalar tuzish. C++ Buliderda MBni boshqaradigan ilovalar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ dastrulashning tili haqida umumiy tushunca. C++ amallari, operatorlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ ko'rsatkichlari va ilovalari. Massivlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Funksiyalar. Standart funksiyalar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Murakkab ma'lumotlarning turlari. Yangi ma'lumotlarning turlarini kiritish.
Tuzilmalar va jamlanmalar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tuzilmalar va massivlar. Tuzilmalar va funksiyalar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Ma'ruza, namoyish etish, savol-javob, Charxpalak, metodlari.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ob'ekli Programmalash asoslari. Ob'ektga yo'naltirilgan dasturlash. Sinf va ob'ekt tushinchasi. Vorislik, inkapsulyasiya va polimorfizm.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Sinflar va ob'ektlar. Interfeyslar. Konstruktorlar va destruktorelar. Sinf strukturaning kengaytmasi sifatida.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Sinf va ko'rsatkichlar. Sinf komponentalariga ko'rsatkichlar. this ko'rsatkichi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z -o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

**“Programmash asoslari” fani bo'yicha
ma'ruza mashg'ulotlarining kalendar tematik rejasi**

№	<i>Maruza mavzusi</i>	Ajratilgan soat
1	2	3
	Ob'yektga yo'naltirilgan Programmash asoslari	
1.	“Dasturlash tillari” faniga kirish	2
2.	Ob'yektga yo'naltirilgan Programmash asoslari	2
	C++ Bulider ob'yektga yo'naltirilgan dasturlash tili	
3.	C++ Bulider dasturlash tilining ishchi muhiti	2
4.	Standart komponentlar palitrasi..	2
5.	Additional komponentlar palitrasi.	2
	C++ Bulider dasturlari strukturasi. Loyiha va modul.	
6.	C++ Bulider dasturlari strukturasi. Loyiha va modul.	2
	C++ Bulider dasturlash tilining operatorlari	
7.	C++ Buliderda tiplar, o'zgarimlar, o'zgaruvchilar va standart funksiyalar	2
8.	C++ Bulider dasturlash muhitida tarmoq operatorlari	2
9.	C++ Bulider dasturlash muhitida sikl operatorlari	2
10.	C++ Buliderda massivlar.	2
	Prosedura va funksiyalar	
11.	C++ Buliderda prosedura va funksiyalar	2
12.	C++ Buliderda fayllar bilan ishlash	2
	3 – SEMESTR JAMI:	24
	C++ dasturlash tili	
1.	C++tilining leksik asoslari	2
2.	O'zgaruvchi va o'zgarimas tipli kattaliklar	
3.	Dasturlashda operator tushunchasi. Oddiy operatorlar	2
4.	C++ dasturlash tilida tarmoqlanuchi jarayonlarni dasturlash	2
5.	C++ dasturlash tilida takrorlanuvchi jarayonlarni dasturlash	2
6.	C++ dasturlash tilida funksiyalar	2
7.	C++ dasturlash tilida tipsiz funksiyalar	2
8.	C++ dasturlash tilida bir o'lchovli massivlar .	2
9.	C++ dasturlash tilida ikki o'lchovli massivlar	2
10.	C++ da ko'rsatkichlar va satrlar.	2
11.	C++ da struktura va birlashmalar	2
12.	C++ tilida grafika	2
	4 – SEMESTR JAMI:	24
	JAMI	48

Amaliy mashg'ulotlarning tavsiya etiladigan mavzulari

Amaliyot ishlari mavzulari talabalarda berilgan amaliy masalalarni yechishda algortmlar va dasturlar tuzish amaliy ko'nikmalarni shakllantirishga qaratilgan.

Ob'jektga yo'naltirilgan dasturlashning asosiy tushunchalari. Sinf tavsifi. Ob'jektlar. Ob'jektlar sinfi iyerarxiyasi. Amallar va usullar. Maydonlar. Xos-salar. Konstruktorlar, destrukturorlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider dasturlash tilining ishchi muhiti bilan tanishish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Asosiy komponentlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Additional bo'limi: BitBtn, MaskEdit, StringGrid, Image komponentlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ma'lumotlarning tiplari bilan ishlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Shartli operatorlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Sikl operatorlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Massivlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Dasturlar tuzishda prosedura va funksiyalar foydalanish yullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Dasturlar tuzishda prosedura va funksiyalar foydalanish yullari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Fayllar bilan ishlash. Fayllar bilan ishlashning ob'jektli yondashuvi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
C++ Bulider muhitida grafik vositalardan foydalanib dasturlash.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
C++ Buliderda MBni boshqaradigan ilovalar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
C++ da oddiy dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Sikllarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Massivlar va ko'rsatkichlarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Funksiyalarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Tuzilmalarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Ob'ektga yo'naltirilgan dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Merosxo'rlikdan foydalanib dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*
Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

**“Programmash asoslari” fani bo’yicha
amaliy mashg’ulotlarning kalendar tematik rejasi**

№	<i>Mashg’ulot mavzusi</i>	Ajratilgan soat
1.	C++ Bulider dasturlash tilida forma tuzilishi va unda komponentlarni joylashtirish	2
2.	C++ Bulider dasturlash tilida tarkibiy operatorlar, shart operatori, tanlash opertori	2
3.	C++ Bulider dasturlash tilida sikl operatorlari	2
4.	C++ Bulider dasturlash tilida massivlar kattaliklar	2
5.	C++ Bulider dasturlash tilida satriy kattaliklar	2
6.	Dasturlar tuzishda funksiya va proseduradan foydalanish yo’llari.	2
7.	C++ Buliderda modullar va ulardan foydalanish	2
8.	C++ Bulider dasturlash tilining Office dasturlari bilan hamkorligi	2
9.	C++ Bulider dasturlash muxitida fayllar bilan ishlash.	2
10.	C++ Bulider muhitida grafik vositalardan foydalanib dasturlash	2
3 – SEMESTR JAMI:		20
1.	C++ tilida chiziqli dasturlash.	2
2.	C++ tilida shartli va shartsiz o’tish operatorlari, tanlash operatori	2
3.	C++ tilida takrorlanish operatorlari (while, do while, for)	2
4.	C++ tilida massivlarga doir dasturlar tuzush.	2
5.	C++ tilida funksiyalarga doir dasturlar tuzish	2
6.	C++ tilida ko’rsatkichlarga doir dasturlar tuzush.	2
7.	C++ tilida strukturalar va birlashmalar ga doir dasturlar tuzish.	2
8.	C++ tilida dinamik massivlar ga doir dasturlar tuzish.	2
9.	C++ tilida fayllar bilan ishlash	2
10.	C++ tilida sinflar.	2
4 – SEMESTR JAMI:		20
JAMI:		40

Laboratoriya mashg’ulotlari

C++ Bulider muhitida konsol ilovalarini yaratish.

Qo’llaniladigan ta’lim texnologiyalari: *dialogik yondoshuv, muammoli ta’lim. Klaster, Venn diagrammasi, T-sxemasi, o’z-o’zini nazorat.*

Adabiyotlar: A1; A2; A3; A5; Q1; Q2; Q3; Q4; Q5.

Foydalanuvchi modullarini yaratish va qo’llash.

Qo’llaniladigan ta’lim texnologiyalari: *dialogik yondoshuv, muammoli ta’lim. Klaster, Venn diagrammasi, T-sxemasi, o’z-o’zini nazorat.*

Adabiyotlar: A1; A2; A4; A5; Q1; Q2; Q3; Q4; Q5.

Asosiy komponentlarni qo’llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q2; Q3; Q4; Q5.

Additional bo'limi: BitBtn, MaskEdit, StringGrid, Image komponentlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1;A2; A3; A4;A5; Q1; Q2; Q3;.

Shartli operatorlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Sikl operatorlarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Massivlarga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z -o'zini nazorat, klaster.*

Adabiyotlar: A1;A2; A3; A4;A5; Q1; Q2; Q3; Q4; Q5.

Prosedura va funksiyalarni yaratish va qo'llash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Fayllar bilan ishlashga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ Bulider muhitida grafik vositalarni qo'llab dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

C++ da oddiy, tarmoqlanuvchi va sikllarga doir dasturlar tuzis.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1;A2; A3; A4;A5; Q1; Q2; Q3; Q4; Q5.

Massivlar va ko'rsatkichlarga doir dasturlar tuzush.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1;A2; A3; A4;A5; Q1; Q2; Q3; Q4; Q5.

Funksiyalarga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z -o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tuzilmalarga doir dasturlar tuzush.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tuzilmalar va funksiyalarga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ob'ektga yo'naltirilgan dasturlar tuzush.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Integrativ, munozara, o'z-o'zini nazorat, klaster.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

**“Programmash asoslari” fani bo'yicha
laboratoriyamashg'ulotlarining kalendartematik rejasi**

№	<i>Laboratoriya mashg'ulotlarining mavzusi</i>	Ajratilgan soat
1	2	3
1.	C++ Bulider dasturlash muhitiga kirish	2
2.	Sodda dasturlarni tuzish(Standart va Additional bo'limi komponentlari yordamida)	2
3.	Shartli o'tish operatori yordamida dasturlar tuzish	2
4.	Sikl operatorlarni qo'llab dasturlar tuzish	2
5.	C++ Bulider dasturlash muhitida massivlarga doir dasturlar tuzish	2
6.	Satriy kattaliklar bilan ishash	2
7.	C++ Bulider dasturlash muhitida funksiya va protseduralardan foydalanish	2
8.	C++ Bulider dasturlash muhitida modul tuzilishi va undan foydalanish	2
9.	C++ Bulider dasturlash tilining Office dasturlari bilan hamkorligi	2
10.	C++ Bulider dasturlash muxitida fayllar bilan ishlash	2
11.	C++ Bulider dasturlash mixitining grafik imkoniyatlari	2
12.	C++ Buliderda spravka tizimini yaratish	2
	3-semestr jami:	24
1.	C++ tili asosidagi dasturlash muhiti bilan tanishish	2
2.	Oddiy tiplar, arifmetik amallar va standart funksiyalarni qo'llash	2
3.	Ta'minlash operatorini asosiy va qo'shimcha formalarini ishlatish	2
4.	If operatorining to'liq va qisqa formalaridan foydalanish. Switch tanlash operatorini qo'llash	2
5.	Old shartli (while) va so'ng shartli sikl (do while) operatoridan foydalanish	2
6.	Parametrli sikl operatori (for) qulayliklarini namoyish etish	2
7.	Ichma-ich joylashgan sikllarni tashkil qilish	2
8.	Bir va ikki o'lchovli massivlarni qo'llash	2
9.	Simvollar va satrlar bilan ishlash	2

10.	Strukturalar va birlashmalar ustida amallar	2
11.	Funksiyalarni e'lon qilish va aniqlash	2
12.	Faylni tasvirlash va kutubxonani ulash	2
	4-semestr jami:	24
	JAMI	48

Mustaqil ta'lim tashkil etishning shakli va mazmuni

“Programmash asoslari” bo'yicha talabaning mustaqil ta'limi shu fanni o'rganish jarayonining tarkibiy qismi bo'lib, uslubiy va axborot resurslari bilan to'la ta'minlangan. Talabalar auditoriya mashg'ulotlarida professor-o'qituvchilarning ma'ruzasini tinglaydilar, misol va masalalar yechadilar. Auditoriyadan tashqarida talaba darslarga tayyorlanadi, adabiyotlarni konspekt qiladi, uy vazifa sifatida berilgan masalalarni hal qiladi. Bundan tashqari ayrim mavzularni kengroq o'rganish maqsadida qo'shimcha adabiyotlarni o'qib referatlar tayyorlaydi hamda ayrim mavzular bo'yicha testlar yechadi. Mustaqil ta'lim natijalari reyting tizimi asosida baholanadi. Uyga vazifalarni bajarish, qo'shimcha darslik va adabiyotlardan yangi bilimlarni mustaqil o'rganish, kerakli ma'lumotlarni izlash va ularni topish yo'llarini aniqlash, internet tarmoqlaridan foydalanib ma'lumotlar to'plash va ilmiy izlanishlar olib borish, ilmiy to'garak doirasida yoki mustaqil ravishda ilmiy manbalardan foydalanib ilmiy maqola va ma'ruzalar tayyorlash kabilar talabalarning darsda olgan bilimlarini chuqurlashtiradi, ularning mustaqil fikrlash va ijodiy qobiliyatini rivojlantiradi. Shuning uchun ham mustaqil ta'limsiz o'quv faoliyati samarali bo'lishi mumkin emas. Uy vazifalarini tekshirish va baholash amaliy mashg'ulot olib boruvchi o'qituvchi tomonidan, konspektlarni va mavzuni o'zlashtirish darajasini tekshirish va baholash esa ma'ruza darslarini olib boruvchi o'qituvchi tomonidan har darsda amalga oshiriladi. “ Programmash asoslari” fanidan mustaqil ish majmuasi fanning barcha mavzularini qamrab olgan va quyidagi 18 ta katta mavzu ko'rinishida shakllantirilgan.

Talabalar mustaqil ta'limining mazmuni va hajmi

No	Mustaqil ta'lim mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi (soatda)
III semestr				
1.	Ob'yektga yo'naltirilgan dasturlashning asosiy tamoyillari	Adabiyotlardan konspekt. Individual topshiriqlar	1,2 - haftalar	8
2.	C++ Bulider dasturlash tilining ishchi muhiti, undagi oynalar: Ob'ektlarning daraxtsimon ko'rinish oynasi, ob'ektlar inspektori oynasi, kod brauzeri oynasi, asosiy oyna, forma oynasi, dastur kodi oynasi, u o'rnatilishi zarur bo'lgan kompyuterga qo'yiladigan texnik talablar	Adabiyotlardan konspekt. Individual topshiriqlar	3,4 - haftalar	8
3.	Komponentlar palitrasi. Palitra bo'limlari	Adabiyotlardan	5,6 -	8

	va ayrim komponentlar xossalari bilan tanishish. Stanfard bo'limi. Frame, Main-Menu, PopupMenu, Label, Edit, Button, Memo, Panel komponentlar xossalari	konspekt. Individual topshiriqlar	haftalar	
4.	C++ Bulider dasturlari strukturasi. Loyiha va modul. Xodisa tushunchasi, loyiha va modul strukturasi, dastur elementlari (alfavit, identifikatorlar, doimiyliklar, ifodalar va amallar.) Bo'sh forma va uning modifikatsiyasi	Adabiyotlardan konspekt. Individual topshiriqlar	7,8 - haftalar	6
5.	C++ Bulider dasturlash tilining operatorlari	Adabiyotlardan konspekt. Individual topshiriqlar	9,10 - haftalar	6
6.	C++ Bulider dasturlash tilida massivlar, massivlarni tavsiflash. Massivlarni berilish usullari, Array, of, xizmatchi so'zlari, Random (max) funksiyasi bilan tanishtirish	Adabiyotlardan konspekt. Individual topshiriqlar	11,12 - haftalar	6
7.	Prosedura va funksiyalar C++ Bulider dasturlash tilining muhim instrumenti sifatida. Prosedura tarifi, uning nomi, undan foydalanish yullari. Funksiya tarifi, uning nomlanishi, undan dasturda foydalanish va uning proseduradan farqi.	Adabiyotlardan konspekt. Individual topshiriqlar	13,14 - haftalar	6
8.	Fayllar bilan ishlash. Fayllar bilan ishlashning ob'yektlilik yondashuvi.	Adabiyotlardan konspekt. Individual topshiriqlar	15,16 - haftalar	6
9.	C++ Bulider dasturlash tilining grafik imkoniyatlari, C++ Buliderdagi maxsus TCanvas, TFont, TPen, Tbrush klasslari. TFont klassi xossalari: Color, Name, Size, Style. TPen klassi xossalari: Color, Mode, Width, Style. TBrush klassi xossalari: Bitmap, Color, Style.	Adabiyotlardan konspekt. Individual topshiriqlar	17,18 - haftalar	8
III semestrjami			62	
IV semestr				
10.	C++ dastrulashning tili haqida umumiy tushunca. C++ amallari, operatorlari.	Adabiyotlardan konspekt. Individual topshiriqlar	1, 2- haftalar	8
11.	Funksiyalar. Standart funksiyalar	Adabiyotlardan konspekt. Individual	3,4-haftalar	8

		topshiriqlar		
12.	C++ da ko'rsatkichlari	Adabiyotlardan konspekt. Individual topshiriqlar	5,6-haftalar	8
13.	C++ da massivlar.	Adabiyotlardan konspekt. Individual topshiriqlar	7, 8,9 - haftalar	8
14.	Murakkab ma'lumotlarning turlari. Yangi ma'lumotlarning turlarini kiritish. Struktura va birlashmalar.	Adabiyotlardan konspekt. Individual topshiriqlar	10-11-haftalar	8
15.	Ob'ektli Programmalash asoslari. Ob'ektga yo'naltirilgan dasturlash. Sinf va ob'ekt tushinchasi. Vorislik, inkapsulyasiya va polimorfizm	Adabiyotlardan konspekt. Individual topshiriqlar	12,13 - haftalar	8
16.	Sinflar va ob'ektlar. Interfeyslar. Konstruktorlar va destruktorelar. Sinf strukturaning kengaytmasi sifatida.	Adabiyotlardan konspekt. Individual topshiriqlar	14,15,16-haftalar	8
17.	Sinf va ko'rsatkichlar. Sinf komponentalariga ko'rsatkichlar. this ko'rsatkichi.	Adabiyotlardan konspekt. Individual topshiriqlar	17,18 - haftalar	6
IV semestr jami				62
JAMI				124

Dasturning informasion -uslubiy ta'minoti

Mazkur fanni o'qitish jarayonida ta'limning zamonaviy metodlari, pedagogik va axborot-kommunikasiya texnologiyalarini qo'llash nazardatutilgan.

- ma'ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentasiya va elektron-didaktik texnologiyalaridan;
- amaliy mashg'ulotlarda zamonaviy pedagogik va inovasion texnologiyalardan;
- laboratoriya mashg'ulotlarida zamonaviy kompyuter sinflaridan foydalanish ko'zda tutiladi. Shuningdek, bugungi kun talabiga javob beradigan Programmalash asoslari Pascal, C++ Bulider tillarini o'rnatuvchi disklar ham bo'lishi lozim..

BAHOLASH MEZONLARI

“Programmalash asoslari» fanidan talabalar bilimni baholash O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirining 2018 yil 9 avgustdagi 19-2018-son buyrug'i bilan tasdiqlangan “**Oliy ta'lim muassasalarida talabalar bilimni nazorat qilish va baholash tizimi to'g'risida**”gi Nizom asosida amalga oshiriladi.

Mazkur fandan talabalar bilimni nazorat qilish **oraliq va yakuniy nazorat** turlarini o'tkazish orqali amalga oshiriladi.

Talabalar bilimini baholash **5 baholik tizimda** amalga oshiriladi.

Talabalarning bilimi quyidagi mezonlar asosida:

talaba mustaqil xulosa va qaror qabul qiladi, ijodiy fikrlay oladi, mustaqil mushohada yuritadi, olgan bilimini amalda qoʻllay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) boʻyicha tasavvurga ega deb topilganda — 5 (aʻlo) baho;

talaba mustaqil mushohada yuritadi, olgan bilimini amalda qoʻllay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) boʻyicha tasavvurga ega deb topilganda — 4 (yaxshi) baho;

talaba olgan bilimini amalda qoʻllay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) boʻyicha tasavvurga ega deb topilganda — 3 (qoniqarli) baho;

talaba fan dasturini oʻzlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) boʻyicha tasavvurga ega emas deb topilganda — 2 (qoniqarsiz) baho bilan **baholanadi**.

Baholashni 5 baholik shkaladan 100 ballik shkalaga oʻtkazish

JADVALI

5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala
5,00 — 4,96	100	4,30 — 4,26	86	3,60 — 3,56	72
4,95 — 4,91	99	4,25 — 4,21	85	3,55 — 3,51	71
4,90 — 4,86	98	4,20 — 4,16	84	3,50 — 3,46	70
4,85 — 4,81	97	4,15 — 4,11	83	3,45 — 3,41	69
4,80 — 4,76	96	4,10 — 4,06	82	3,40 — 3,36	68
4,75 — 4,71	95	4,05 — 4,01	81	3,35 — 3,31	67
4,70 — 4,66	94	4,00 — 3,96	80	3,30 — 3,26	66
4,65 — 4,61	93	3,95 — 3,91	79	3,25 — 3,21	65
4,60 — 4,56	92	3,90 — 3,86	78	3,20 — 3,16	64
4,55 — 4,51	91	3,85 — 3,81	77	3,15 — 3,11	63
4,50 — 4,46	90	3,80 — 3,76	76	3,10 — 3,06	62
4,45 — 4,41	89	3,75 — 3,71	75	3,05 — 3,01	61
4,40 — 4,36	88	3,70 — 3,66	74	3,00	60
4,35 — 4,31	87	3,65 — 3,61	73	3,0 dan kam	60 dan kam

Tavsiya etilgan adabiyotlar ro'yxati
Asosiy darsliklar va o'quv qo'llanmalar

1. A. R.Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O`quv qo`llanma. T.:“Cho`lpon”, 2010 y.
2. A. R.Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O`quv qo`llanma. T.:“Cho`lpon”, 2013 y.
3. П.Дарахвелидзе, Э.Марков. Программирование в С++ Bulider7. Учебник.Санкт-Петербург, “БХВ-Петербург” 2003 г.
4. Sh.I.Razzoqov, M.J.Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o`quv qo`llanma. T.: “Ilm Ziyo”, 2011y.
5. M.Ashurov, M.Mirmaxmudov, Sh.Sapaev.Zamonaviy Programmalash asoslari fanidan laboratoriya ishlari.T.: TDPU, 2008 y.
6. Axatov A., Qarshiyev H. C++ tilida programmalash(misollar va masalalar). 2016.SamDU nashri. 140b
7. Страуструп Б. Язык программирования С++. Специальное издание. – М.: Бинам – ПРЕСС, 2006, 1104 с
8. Павловская Т. С++ Программирование на языке высокого уровня. – СПб. : Питер, 2005. -461 с.
9. Мадрахимов Ш., Гайназаров С. С++ тилида программалаш асослари: услубий кулланма – Тошкент: УзМУ 2009 . – 196 б
- 10.Madrahimov Sh. va boshq. С++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma. - Toshkent: O'ZMU, 2014.-160 b

Qo'shimcha adabiyotlar

1. В.М.Пестиков, А.Н.Маслобоев. Turbo PASCAL 7.0. Изучаем на примерах. Санкт-Петербург.: “БХВ-Петербург”, 2004 г.
2. Фаронов В.В. Программирование на языке высокого уровня С++ Bulider. Учебник. М.: “Питер”, 2003 г.
3. В.Т.Безручко. Практикум по курсу информатики. М.: «Финансы и статистика», 2004 г.
4. Культин Н. С ++ Builderв задачах и примерах. – СПб. : ВХБ, 2005.-336с
5. Nazarov F.M С++ tilida dasturlash asoslari 2017.SamDU nashri. 160 b

Elektron ta'lim resurslari

1. Программирование в С++ Bulider. Учебник. А.Я.Архангелский ([www.ozon.ru /context /detail /id /2705337/](http://www.ozon.ru/context/detail/id/2705337/))
2. www.pedagog.uz
3. www.Ziyonet.uz
4. www.edu.uz
5. <http://www.borlpasc.narod.ru/>.
6. [http:// www.intuit.ru](http://www.intuit.ru)

O'quv materiallari

Ma'ruza matni

Mavzu: Ob'jektga yo'naltirilgan dasturlashning asosiy tushunchalari. Sinf tavsifi. Ob'yektlar. Ob'yektlar sinfi iyerarxiyasi.

Ma'ruza rejasi:

1. Ob'ekli dasturlashning asosiy tushunchalari
2. Ob'ektga yo'naltirilgan dasturlar yaratish
3. Ob'ektga yo'naltirilgan Programmalash asoslari

1. Ob'ekli dasturlashning asosiy tushunchalari

Ob'ektga yo'naltirilgan yondoshuv dasturiy tizimlarni dasturlash tiliga bog'liq bo'lmagan holda yaratish jarayonida modellardan sistematik foydalanishga asoslangan. Har bir model, uning o'zi aks ettirayotgan predmetning barcha xususiyatlarini ifodalay olmaydi, u faqat ba'zi juda muhim belgilarni ifodalaydi. SHunday qilib, model o'zi aks ettirayotgan predmetga nisbatan ancha sodda bo'ladi. Bizga shu narsa muhimki, model endi formal konstruktsiya hisoblanadi: modellarning formalligi esa ular orasida formal bog'lanishlarni aniqlashni va ular orasida formal operatsiyalar bajarishni ta'minlay. Bu ish modellarni ishlab chiqishni va o'rganishni hamda kompyuterda amalga oshirishni osonlashtiradi. Xususan esa, modellarning formal xarakteri yaratilayotgan dasturning formal modelini olishni ta'minlaydi.

Shunday qilib, ob'ektga yo'naltirilgan yondoshuv quyidagi murakkab muammolarni hal etishda foydalaniladi:

- dasturiy ta'minotning murakkabligini pasaytiradi;
- dasturiy ta'minotning ishonchliligini oshiradi;
- dasturiy ta'minotning alohida komponentalarini modifikatsiya qilishini osonlashtiradi;
- alohida komponentalardan qayta foydalanishni ta'minlaydi.

Ob'ektga yo'naltirilgan yondoshuvning sistemali qo'llanilishi yaxshi tuzilmalangan, ishlatishda barqaror bo'lgan, oson modifikatsiya qilinuvchi dasturiy tizimlarni yaratish imkoniyatini beradi. Aynan ana shu imkoniyatlar dasturchilarni ob'ektga yo'naltirilgan yondoshuvdan foydalanishga juda ham qiziqirmoqda. Ob'ektga yo'naltirilgan yondoshuvli programmalash hozirgi vaqtda eng tez rivojlanayotgan dastur yozish texnologiyasi hisoblanadi. Ob'ektga yo'naltirilgan yondoshuv ikkita qismga bo'linadi:

- ob'ektga yo'naltirilgan dasturlar yaratish;
- ob'ektga yo'naltirilgan tillari.

2. Ob'ektga mo'ljallangan dasturlar yaratish

Ob'ektga yo'naltirilgan dasturlar yaratish, dastur yaratishda ob'ektga yo'naltirilgan modellarni yaratishga asoslangan.

Ob'ektga yo'naltirilgan dasturlar yaratish deganda biz quyidagilarni tushunamiz:

- dasturiy tizimlarni yaratishdagi ob'ektga yo'naltirilgan metodologiyani;
- bu texnologiyani qo'llovchi instrumental vositalarni.

Ob'ektga yo'naltirilgan dasturlar yaratish dasturiy vositalarni yaratishning hayotiy siklining birinchi bosqichidayoq qo'llanilishi mumkin va u dasturlash tiliga bo'liq emas. Yaratish jarayonida ob'ektlar – bu formal konstruksiyalar bo'lib (masalan, burchaklari yoydan tashkil topgan to'rtburchaklar), ular ob'ektlarni aks ettiradi. Ob'ektga yo'naltirilgan yaratish, shunday qilib, ob'ektga yo'naltirilgan metodologiyani (texnologiyani) qo'llashga asoslangan.

3. Ob'ektga mo'ljallangan Programmalash asoslari

Ob'ektga yo'naltirilgan tillari oxirgi vaqtlarda juda ommaviylashgan tillarga kiradi. Bunday tillarga quyidagilar kiradi: Object Paskal, C++ Bulider, C++, Visual C++, Visual Basic, Java va boshqalar. C++ Bulider, C++ eng ko'p tarqalgan ob'ektga yo'naltirilgan Programmalash asoslariga kiradi.

Ob'ektga yo'naltirilgan dasturlashda dastur ob'ektlarini va uning xususiyatlarini (atributlarni) va ularni birlashtiruvchi sinflarni tavsiflashga olib kelinadi. SHu jumladan, ob'ektlar ustida operatsiyalar (usullar) ni aniqlashga olib kelinadi.

Atributlar va usullarni tadqiq qilish asosida bazaviy sinflar va ularning hosilalarini yaratish imkoniyati tug'iladi.

Ob'ektga yo'naltirilgan dasturlashning yana bir nazariy jihatdan juda muhim va zarur xususiyatlaridan biri hodisalarni ishlash mexanizmi hisoblanadi, ular yordamida ob'ektlar atributlari qiymatlari o'zgartiriladi. Ob'ektga yo'naltirilgan dasturlashda avval yaratilgan ob'ektlar kutubxonasi va usullaridan foydalanish hisobiga ob'ektga yo'naltirilgan dasturlashda ancha mehnat sarfi tejiladi.

Ob'ektlar, sinflar va usullar polimorfizm bo'lishlari mumkin, bu esa dasturiy vositaning qo'lay foydalanishliligi va universalligini ta'minlaydi.

Adabiyotlar ro'yxati

1. ГрадиБуч. Объектно –ориентированной анализ и проектирование с примерами приложений на C++. Невский диалект, 560 стр, 2001 г.
2. Грехем И. Объектно ориентированные методы. Принципы и практика. Вильямс. 879 стр, 2004 г.
3. Иванова Г.С. Объектно ориентированное программирование. Учебник. МГТУ им Баумана. 320 стр, 2003 г.
4. Страуструп Б. Язык программирования C++, "ДиаСофт", Киев 1993г.

Mavzu: Ob'yecktga yo'naltirilgan dasturlashning tamoyillari. ob'yecktga yo'naltirilgan loyihalash. Inkapsulyatsiyalash. Merosxo'rlik. Polimorfizm.

OYDning asosiy tamoyillari

1. Vorislik
2. Inkapsulyasiya (usullar va xususiyatlarni ob'ekt ichida saqlash-yashirish).
3. Polimorfizm, berilmalarni ishlash funksiyalarining mavjudligi.
4. Abstraksiya. Abstraksiya – bu identifikatorlardan farqli bo'lgan istalgan dasturlash tili ifodasi hisoblanadi.

Ob'ektga yo'naltirilgan dasturlashda har bir ob'ekt prinsipial dinamik mohiyatga ega, ya'ni vaqtga bog'liq holda va unga nisbatan tashqi omillar ta'sirida o'zgaradi. Boshqa aytganda, ob'ekt ma'lum bir darajada o'zini tutishiga ega. Ob'ektga yo'naltirilgan dasturlashda abstraksiya OYD ning modeli hisoblanadi. Sinf umumiy xususiyatlar va hulq-atvoriga ega ob'ektlarni birlashtiradi. Bitta sinfga mansub ob'ektlar bir xil xususiyatlarga ega bo'lib, bir xil xatti – harakat namoyon etadi.

Sinflar shablon (qolip)ga o'xshaydi: ular ob'ektlarning ekzemplarlarini tayyorlash uchun qo'llaniladi. Belgilar – sinfning tashqaridan ko'rinib turgan xususiyatlari. Ob'ekt ichki o'zgaruchiga bevosita kirishni taqdim etganda yoki usul yordamida qiymatni qaytargandagina, o'z belgilarini namoyon qilishi mumkin.

Hulk-atvor - xabarga yoki holatning o'zgarishiga javoban ob'ekt tomonidan bajariladigan xatti-xarakatlar. U ob'ekt nima qilayotganini bildiradi.

Bir ob'ekt ikkinchi ob'ekt ustida xatti-harakatlar bajarib, uning hulq-atvoriga ta'sir ko'rsatishi mumkin. "Xatti-harakat" atamasi o'rniga "usulni chaqirish", "funksiyasini chaqirish" yoki "xabarni uzatish" atamalari qo'llaniladi. Muhimi bu atamalarning qaysi biri qo'llanilayotganida emas, albatta, muhimi bu xatti-harakatlar ob'ektlar hulq-atvorini namoyon qilishga etishidadir.

Ob'ektlar o'rtasida aloqa ob'ektga yo'naltirilgan dasturlashning muhim tarkibiy qismidir. Ob'ektlar o'zaro aloqasining ikkita asosiy usuli mavjuddir.

Birinchi usul – ob'ektlar biri ikkinchisidan mustaqil ravishda mavjud bo'ladi. Agar alohida ob'ektlarga o'zaro aloqa kerak bo'lib qolsa, ular bir-birlariga xabar jo'natadi

Ob'ektlar bir-birlari bilan xabarlar yordamida aloqa qiladi. Xabar olgan ob'ekt ma'lum xatti-xarakatlarni bajaradi.

Xabar uzatish bu ob'ekt xolatini o'zgartirish maqsadida ululni chaqirib olish yoki xulq-atvor modellaridan birini qo'llashning o'zginasidir.

Ikkinchi usul – ob'ekt tarkibida boshqa ob'ektlar bo'lishi mumkin. Xuddi OYD da bo'lganidek, dastur ob'ektlardan tashkil topganidek, ob'ektlar ham, o'z navbatida, agregatsiya yordamida boshqa ob'ektlardan jamlanishi mumkin. Ushbu ob'ektlarning har bittasida usul va belgilarga ega bo'lgan interfeys mavjud bo'ladi.

Xabar – ob’ektga yo‘naltirilgan yondoshuvning muhim tushunchasi. Xabarlar mexanizmi tufayli ob’ektlar mustaqilligini saqlab qolishi mumkin. Boshqa biror ob’ektga xabar jo‘natayotgan ob’ekt uchun xabar olgan ob’ekt talabdagi xatti-xarakatni qanday bajarishi unchalik muhim emas. Unga xatti-harakat bajarilganligining o‘zi muhimdir.

Ma’lumki, C++ Bulider dasturiy vositasi Object Pascal ob’ektga yo‘naltirilgan dasturlash tilini ishlatadi. Ob’ektga yo‘naltirilgan dasturlash (OYD) – bu dastur ishlab chiqish usulikasi bo‘lib, uning asosida real dunyo ob’ekti va uning holatini ifodalovchi ma’lum strukturaga ega ob’ekt tushunchasi yotadi. Object Pascalda ob’ekt modelining qo‘llanilish natijasi bu komponentalarni qo‘llash va yaratishdir. Object Pascal ob’ekt modelining asosi sinf va ob’ekt tushunchalaridir.

Sinf

Sinf - bu Object Pascalda maxsus turlar bo‘lib, o‘zida maydon, usullar va xossalarni mujassamlashtiradi.

Pascal sinfik tili dasturchiga o‘zining murakkab ma’lumotlar turlari - yozuvlar (records) kiritishga imkon beradi. Ob’ektli dasturlash konsepsiyasiga asoslangan C++ Bulider tili, sinflar kiritishga imkon beradi. Sinf murakkab struktura bo‘lib, ma’lumotlar ta’riflaridan tashqari, protsedura va funksiyalar ta’riflarini o‘z ichiga oladi.

Sodda sinf ta’rifiga misol:

```
TPerson = class
private
fname: string[15]; faddress: string[35];
public
procedure Show;
end;
```

Sinf ma’lumotlari maydonlar, protsedura va funksiyalar usullar deb ataladi.

Keltirilgan misolda TPerson - sinf nomi, fname va faddress – maydonlar nomlari, show - usul nomi.

Maydon - bu sinfga birlashtirilgan ma’lumotlardir. Sinfga qarashli maydonlar oddiy yozuv maydoni kabi bo‘lib, ularning farqi har xil turda bo‘lishidir. Masalan,

```
Type
TchildClass=Class
    Fore: Integer;
    Ftwo: String;
    Fthree: Tobject;
End;
```

Maydonlarga murojaat qilish sinf xossalari va usullari yordamida amalga oshiriladi. Maydonga murojaat qilish uchun oldin sinf nomi yozilib, keyin ajratuvchi nuqta qo‘yilib maydon nomi yoziladi. Masalan,

```

Var
  MyObject: TchildClass;
  Begin
MyObject.Fone:=16;
MyObject.Ftwo:='qator qiymati';
  End;

```

Maydon nomi unga mos xossa nomining birinchi harfi “F” bo‘lishi bilan farqlanadi.

C++ Bulider da qabul qilingan kelishuv bo‘yicha maydonlar nomlari f (field — maydon so‘zidan) harfidan boshlanishi lozim.

Sinf ta’rifi dasturda turlar ta’rifi bulimiga joylashtiriladi (type).

Usullar. Sifga birlashtirilgan protsedura va funksiyalarga usullar deyiladi.

Masalan:

```

Type
  TchildClass=Class
    Fore: Integer;
    Ftwo: String;
    Fthree: Tobject;
    Function FirstFunc(x:Real):Real;
    Procedure SecondProc;
  End;

```

Sinf usullari (sinf ta’rifiga kiritilgan protsedura va funksiyalar) sinf ob’ektlari us-tida amal bajaradi. Usul bajarilishi uchun ob’ekt nomi va nuqtadan sung usul nomi ko‘rsatilishi lozim. Masalan:

```
professor. Show;
```

Sinf usuli ta’riflanganda sinf nomi va usul nomi ko‘rsatiladi. Masalan:

```
// TPerson sinfi Show usuli
```

```
procedure TPerson.Show;
```

```
begin
```

```
Write ( 'Nom:' + fname + #13+ 'Adres:' + faddress );
```

```
end;
```

Usul tanasida ob’ekt maydonlariga murojaat kilinganda ob’ekt nomi ko‘rsatilmaydi.

Usulga murojaat qilish dasturda uning nomini ko‘rsatish bilan bajariladi.

Masalan:

```

Var
  MyObject: TchildClass;
y: Real;

```

```

    Begin
    .....
    MyObject. SecondProc;
    y:=MyObject.FirstFunc(3.14);
    End;

```

Sinfda aniqlangan usullarni statistik, virtual (Virtual), dinamik (Dynamic) yoki abstrakt turlarga bo‘lish mumkin. Agar usul turi ko‘rsatilmasa, u avtomatik ravishda statistik turni oladi. Masalan:

```

Type
    TBase=Class
        Procedure MyJoy; Virtual;
    End;
    Tdescedant=Class(TBase)
        Procedure MyJoy; Override;
    End;
Var
    FirstObject: TBase;
        SecondObject: TDescedant;
Begin
    .....
    FirstObject.MyJoy;
    SecondObject.MyJoy;
    .....
End;

```

Sinf davomchisida ishlatiladigan usul uchun Override kalit so‘zi ko‘rsatilishi lozim.

Agar Tbase sinfida MyJoy usuli dinamik bo‘lsa, Virtual so‘zi Dynamic so‘ziga almashtiriladi. Ularning asosiy farqi murojaat qilinganda Virtual usul vaqt jihatdan ancha effektiv bo‘lsa, Dynamic usul esa operativ xotiradan ratsional foydalanish imkonini beradi.

Ob’ekt

Ob’ekt - bu sinfning real nusxasi bo‘lib, ma’lumotlar va funksiyalardan tashkil topadi. U dasturning Var bo‘limida e’lon qilinadi.

Ob’ektlar sinflar vakillari sifatida dasturda var bo‘limida ta’riflanadi, masalan:

var

student: TPerson; professor: TPerson;

C++ Bulider da ob'ekt - bu dinamik strukturadir. O'zgaruvchi-ob'ekt ma'lumotlarni emas, ob'ekt ma'lumotlariga ilovani o'z ichiga oladi. SHuning uchun dasturchi bu ma'lumotlarga xotiradan joy ajratishni ko'zda tutishi lozim.

Joy ajratish sinf maxsus usuli - konstruktor yordamida amalga oshiriladi. Bu usul odatda Create (yaratish) nomiga ega bo'ladi. Sinf ta'rifida konstruktor uchun procedure so'zi o'rniga constructor so'zi ishlatiladi.

Quyida tarkibida konstruktor qatnashgan TPerson sinfi ta'rifi keltirilgan:

```
TPerson = class private
fname: string [ 15 ];
faddress: string[35];
constructor Create; // konstruktor
public
procedure show; // usul
end;
```

Xotiradan joy ajratish konstruktor sinfga qo'llash natijasini qiymat sifatida berish orqali amalga oshiriladi. Misol uchun

```
professor := TPerson.Create;
```

instruksiyasi bajarilishi natijasida professor ob'ektga xotiradan joy ajratiladi. Xotiradan joy ajratishdan tashqari konstruktor, odatda ob'ekt maydonlariga boshlang'ich qiymatlar berish ya'ni ob'ekt initsializatsiyasi vazifasini ham bajaradi. Quyida TPerson ob'ekti uchun konstruktor misoli keltirilgan:

```
constructor TPerson.Create;
begin
fname := "";
faddress := "";
end;
```

Ob'ekt maydoniga murojaat qilish uchun ob'ekt nomi va nuqtadan so'ng maydon nomi kursatiladi. Masalan:

```
professor.fname
```

Ob'ektga ajratilgan xotira qismini bo'shatish uchun maxsus usul destruktur Free. ishlatiladi. Masalan, professor.Free;

Inkapsulyasiya va ob'ekt xossalari

Inkapsulyasiya - bu sinfga birlashtirilgan qayta ishlash uchun mo'ljallangan ma'lumotlar va qism dasturlardir. Sinf maydonlari ma'lumotlarni o'z ichiga oladi. Bu ma'lumotlarni qayta ishlashda qo'llaniladigan protsedura va funksiyalarga usullar deyiladi. Ob'ektga yo'naltirilgan dasturlashda sinf maydonlariga to'g'ridan to'g'ri murojaat ruxsat etilmaydi. SHu sabab Object Pascalda "xossa" (svoystva) deb ataluvchi maxsus konstruksiya qaraladi. U mos usulni chaqirish yordamida maydondan o'qish yoki yozishni amalga oshiradi.

Misol tariqasida C++ Buliderda aniqlangan standart sinflarni keltirish mumkin.

TEdit - formaga qatorlarni kiritish ishlarini boshqarish va tashkil qilishni bajaradi.

TLabel - formaga belgilarni kiritish ishlarini boshqarish va tashkil qilishni bajaradi.

TButton - formaga joylashtirilgan tugmacha yordamida dasturchi tomonidan kiritilgan dastur kodlarining bajarilishini ta'minlaydi.

Sinf nomlarining bosh harfi "T" harfi bilan boshlanish qabul qilingan.

Inkapsulyasiya deyilganda ob'ekt maydonlariga to'g'ridan to'g'ri emas, faqat sinf usullari orqali murojaat qilishga aytiladi.

C++ Bulider tilida ob'ekt maydonlariga murojaat ob'ekt xossalari orqali amalga oshiriladi. Ob'ekt xossasiga murojaat kilish uchun ikki usuldan foydalaniladi. Xossa tashqaridan sinf maydonini anglatsa, ayni holda u maydondan foydalanish imkonini beruvchi boshqarish mexanizmdir. Xossa qandaydir sinf maydonlari bilan bog'liq bo'lgan o'qishda va yozishda ishlatilishi kerak bo'lgan sinf usullarini ko'rsatadi. O'qish uchun ishlatiladigan funksiya nomi Get bo'lib, unga mos xossa nomi qo'shib yoziladi. YOzish uchun ishlatiladigan usul bitta parametrli Set nomi qism dastur bo'lib, uning nomiga ham mos xossa nomi qo'shib yoziladi. O'qish va yozish usullari va uning parametri ham bir xil xossaga ega bo'lishi lozim. Xossani e'lon qilish uchun Property, Read va Write so'zlari ishlatiladi. Read va Write usul nomlari bo'lib, ular mos ravishda o'qish va yozish uchun mo'ljallangan. Masalan,

Type

```
TStudent=Class
```

```
    Fage: Integer;
```

```
    Function GetAge: Integer;
```

```
    Procedure GetAge(Value:Integer);
```

```
    Property Age: Integer Read GetAge Write SetAge;
```

```
End;
```

Bu erda Age - FAge maydoni, hamda GetAge va SetAge usullari bilan bog'liq xossa bo'lib, Fage maydonidan o'qish yoki yozish uchun xizmat qiladi.

Dastur matnida xossaga murojaat qilish usul va maydon uchun qanday bo'lsa xuddi shunday ob'ekt nomi, nuqta va xossa nomi yozilishi bilan ishlatiladi. Masalan,

Var

```
GoodStudent: TStudent;
```

```
HisAge: Integer;
```

Begin

```
    GoodStudent:=Tstudent.Create;
```

```
    GoodStudent.Age:=19;
```

```

.....
HisAge:=GoodStudent.Age;
.....
GoodStudent.Free;
    End;

```

Xossa qiymatini oʻrnatish - xossani yozish (write) usuli, xossa qiymatini olish - xossani oʻqish (read) usuli deb ataladi.

Sinf taʼrifida xossa nomidan oldin property (xossa) soʻzi yoziladi. Xossa nomidan soʻng uning turi koʻrsatiladi, read soʻzidan soʻng xossani oʻqishni taʼminlovchi usul, write - soʻzidan soʻng xossani yozishni taʼminlovchi usul nomi yoziladi.

Quyida ikki Name va Address xossalarini oʻz ichiga oluvchi TPerson sinfi taʼrifi keltirilgan:

```

type
TName = string[15]; TAddress = string[35];
TPerson = class
private
FName: TName;
FAddress: TAddress;
Constructor Create(Name:Tname);
Procedure Show;
Function GetName: TName;
Function GetAddress: TAddress;
Procedure SetAddress(NewAddress:TAddress);
public
Property Name: Tname
read GetName;
Property Address: TAddress
read GetAddress
write SetAddress; end;

```

Dasturda student obʼektining, Address xossasiga qiymat berishini quyidagicha yozish mumkin

```

student.Address := 'Toshkent, YUnusobod 21, kv.3';

```

Vorislik - har qanday sinf boshqa sinf asosida yaratilishi mumkinligini bildiradi.

Bosh sinfdan yangi bir sinf yaratish quyidagi dastur kodi yordamida bajariladi.

```

TnewClass=Class(TotolClass);

```


Bu erda TotalClass -bosh sinf, TnewClass - esa yangi sinf nomlari. Yangi sinf bosh sinfning barcha xossa va usullarini qabul qiladi.

Vorislik bu mavjud sinflarga yangi maydonlar, xossalar va usullar qo'shish yordamida yangi sinflar hosil qilish imkoniyatini beradi. Yangi hosil qilingan avlod sinf asos ya'ni ajdod sinf xossalari va usullariga vorislik qiladi.

Avlod sinf ta'rifida ajdod sinf nomi ko'rsatiladi. Misol uchun TEmployee (xodim) sinfi TPerson sinfidan FDepartment (bo'lim) maydonini qo'shish yordamida hosil qilinishi mumkin. TEmployee sinfining ta'rifi quyidagicha bo'ladi:

```
TEmployee = class(TPerson)
```

```
FDepartment: integer;
```

```
constructor Create(Name:TName; Dep:integer);
```

```
end;
```

Bu misolda TEmployee sinfi TPerson sinfining vorisidir.

TEmployee o'z konstruktoriga ega bo'lishi lozim. TEmployee sinfi konstruktori quyidagicha berilishi mumkin:

```
constructor TEmployee.Create(Name:Tname;Dep:integer);
```

```
begin
```

```
inherited Create(Name);
```

```
FDepartment:=Dep;
```

```
end;
```

Bu misolda inherited direktivasi bilan ajdod sinf konstruktori chaqiriladi va avlod sinf maydoniga qiymat beriladi

protected i private direktivalari

Sinf elementlariga murojaatni boshqarish uchun protected (himoyalangan) va private (xususiy), direktivalaridan foydalaniladi.

Himoyalangan, ya'ni protected sinf elementlariga sinfdan tashqari faqat voris sinflarda murojaat qilish mumkin. Odatda protected seksiyasiga sinf usullari ta'rifi joylashtiriladi.

YOpiq, ya'ni private sinf elementlari faqat modul ichida murojaat qilish mumkin. Odatda bu seksiyaga sinf maydonlari ta'riflari joylash-tiriladi.

Quyida murojaatni boshqarish direktivalaridan foydalanilgan TPerson sinfi ta'rifi keltirilgan.

```
TPerson = class private
```

```
FName: TName;
```

```
FAddress: TAddress;
```

```
protected
```

```
Constructor Create(Name:TName);
```

```
Function GetName: TName;
```

```
Function GetAddress: TAddress;
```

```
Procedure SetAddress(NewAddress:TAddress);
```

```
Property Name: TName
```

```
read GetName;
```

```
Property Address: TAddress
```

```
read GetAddress  
write SetAddress;  
end;
```

Polimorfizm - bu har xil sinfrga kiruvchi usular uchun bir xil nomlarni ishlatish imkoniyatini yaratishdir. Polimorfizm prinsipi shundan iboratki sinf ob'ektiga mos bo'lgan biror ishning bajarilishida bir xil usulga murojaat qilish mumkinligini ta'minlab beradi. Misol uchun biz yangi sinf tashkil qilishga qaror qildik. Bu sinf bosh sinfdan farqi uning usulida algoritm o'zgartirilgan deylik. Natijada biz bir xil nomli usulga ega bo'lgan ikkita sinfni tashkil qilgan bo'lamiz. U holda tashkil qilingan yangi sinf "polimorfizm" xossasiga ega bo'ladi. Sinflar dasturning Type bo'limida umumiy holda quyidagicha e'lon qilinadi.

Type

<yangi sinf nomi>=Class(<bosh sinf nomi>)

Public

<umumiy bo'lgan elementlar e'lon qilinadi>

Published

<inspektor ob'ektiga ta'luqli bo'lgan elementlarni e'lon qilish>

Projected

<yangi qaram sinfga ta'luqli elementlarni e'lon qilish>

Private

<faqat modulga ta'luqli elementlarni e'lon qilish>

End;

Har bir bo'lim ichida maydonlar, usullar, xossalar va hodisalarni e'lon qilish mumkin.

Polimorfizm - bu har xil sinflarga kiruvchi usullar uchun bir xil nomlardan foydalanish imkoniyatidir.

Uchta sinf ta'rifi berilgan bo'lib bulardan biri qolgan ikki sinf uchun asos sinf bo'lsin:

type

fname: string;

constructor Create(name:string);

function info: string;

virtual;

end;

fgr:integer;

constructor Create(name:string;gr:integer);

function info: string; override; end;

fdep:string;

constructor Create(name:string;dep:string);

```
function info: string;  
override;  
end;
```

Bu sinflarning har birida info usuli ta'riflangan. Asos sinfda virtual direktivasi yordamida info usuli virtual deb e'lon qilingan. Usulning virtual deb e'lon qilinishi avlod sinfda bu usulni shaxsiy usul bilan almashtirishga imkon beradi. Hosil qilingan sinfda virtual usulni almashtiruvchi usul override direktivasi bilan ta'riflanadi. Quyida har bir avlod sinfda info usulining ta'rifi keltirilgan.

```
function TPerson.info:string;  
begin  
result := " "  
end;  
function TStud.info:string;  
begin  
result := fname + ' gp.' + IntToStr(fgr);  
end;  
function TProf.info:string;  
begin  
result := fname + ' kaf.' + fdep;  
end;
```

Ikkala sinf bitta asos sinfdan hosil qilingani uchun talabalar va domlalar ro'yxatini quyidagicha ta'riflash mumkin:

```
list: array[1..SZL] of TPerson;
```

Talabalar va domlalar ro'yxatini info usulini massiv elementlariga qo'llab chiqarish mumkin. Masalan:

```
st := " "  
for i:=1 to SZL do  
if list[i] o NIL  
then st := st + list[i].Info+ #13;  
writeln (st);
```

Adabiyotlar ro'yxati

1. ГрадиБуч. Объектно –ориентированной анализ и проектирование с примерами приложений на C++. Невский диалект, 560 стр, 2001 г.
2. Грехем И. Объектно ориентированные методы. Принципы и практика. Вильямс. 879 стр, 2004 г.
3. Иванова Г.С. Объектно ориентированное программирование. Учебник. МГТУ им Баумана. 320 стр, 2003 г.
4. Страуструп Б. Язык программирования C++, "ДиаСофт", Киев 1993г.

Mavzu: C++ Bulider dasturlash tilining ishchi muhiti, undagi oynalar: Ob'ektlarning daraxtsimon ko'rinish oynasi, ob'ektlar inspektori oynasi, kod brauzeri oynasi, asosiy oyna, forma oynasi, dastur kodi oynasi, u o'rnatilishi zarur bo'lgan kompyuterga qo'yiladigan texnik talablar

C++ Bulider -Windows operatsion tizimida dastur yaratishga yo'naltirilgan dasturlash muxitidir. C++ Buliderda dastur tuzish zamonaviy vizual loyihalash texnologiyalariga asoslangan bo'lib, unda dasturlashning ob'ektga yo'naltirilgan g'oyasi mujassamlashgan. C++ Buliderda dastur Turbo Pascal dasturlash tilining rivoji bo'lgan Object Pascal tilida yoziladi.

C++ Bulider -bir necha muhim ahamiyatga ega bo'lgan texnologiyalar kombinatsiyasini o'zida mujassam etgan:

- yuqori darajadagi mashinali kodda tuzilgan komplyator;
- ob'ektga yo'naltirilgan komponentalar modellari;
- dastur ilovalarini vizual tuzish;
- ma'lumotlar bazasini tuzish uchun yuqori masshtabli vosita.

C++ Bulider - Windows muhitida ishlaydigan dastur tuzish uchun qulay bo'lgan vosita bo'lib, kompyuterda dastur yaratish ishlarini avtomatlashtiradi, xatoliklarni kamaytiradi va dastur tuzuvchi mehnatini engillashtiradi. Delhida dastur zamonaviy vizual loyihalash texnologiyasi asosida ob'ektga yo'naltirilgan dasturlash nazariyasini hisobga olgan holda tuziladi. C++ Bulider sistemasi Turbo Pascal 7.0. tilining rivoji bo'lgan ob'ektga yo'naltirilgan Object Pascal dasturlash tilini ishlatadi.

Ma'lumki dastur tuzish sermashaqqat jarayon, lekin C++ Bulider tizimi bu ishni sezilarli darajada soddalashtiradi va masala turiga qarab dastur tuzuvchi ishining 50-80%ni tizimgayuklaydi. C++ Bulider tizimi dasturni loyihalash va yaratish vaqtini kamaytiradi, hamda Windows muhitida ishlovchi dastur ilovalarini tuzish jarayonini osonlashtiradi.

C++ Bulider o'zida bir qancha zamonaviy ma'lumotlar bazasini boshqarish tizimlari dasturlash texnologiyalarini ham ma'lumotlar bazasini yaratishda ishlata-di.

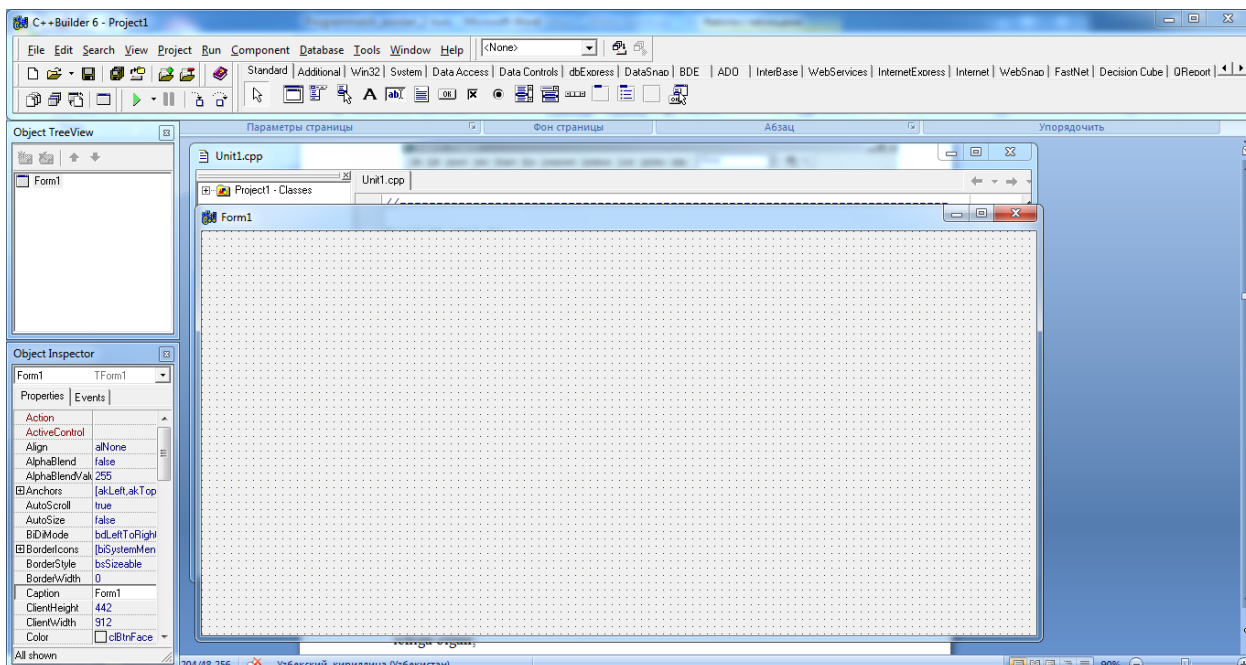
C++ Bulider tizimi oynasi va uning elementlari

C++ Bulider tizimida ishni boshlash uchun uni dasturlar menyusidan topib ishga tushiramiz.

Pusk=>Программы=>Borland C++ Bulider=>C++ Bulider

C++ Bulider oynasi ko'rinishi odatdagidan ancha boshqacharoq bo'lib, u o'z ichiga beshta oynani oladi:

- boshoyna - C++ Bulider Project1;
- forma oynasi - Form1;
- ob'ekt xossalarini taxrirlash oynasi-Object Inspector;
- ob'ektlar ro'yxatini ko'rish oynasi - Object tree View;
- dastur kodlarini tahrirlash oynasi - Unit.pas.



Bosh oyna ekranning yuqori qismida joylashgan bo‘lib, uning birinchi qatorida sarlovha, ya’ni proektning nomi joylashgan. Ikkinchi qatorda buyruqlar menyusi gorizantal ko‘rinishda joylashgan. Keyingi qatorning chap tarafida uskunalar paneli va o‘ng tarafida komponentalar politrasi joylashgan.

Buyruqlar menyusi quyidagilarni o‘z ichiga olgan:

- File (fayl) bo‘limi fayllar ustida ish bajarish uchun kerakli buyruqlarni o‘z ichiga olgan;

- Edit (taxrir) bo‘limi fayl ichidagi ma’lumotlarni taxrirlash uchun kerakli buyruqlarni o‘z ichiga olgan;

- Seerch

- View

- Compile

- Runformaniishgatushirish.

- Options

- Tolsservisxizmatidanfoydalanish.

- Help yordam chaqirish.

Forma oynasida ilovalar yaratiladi. Object Inspector oynasi ob’ekt xossalari taxrirlash uchun xizmat qiladi. Ob’ekt xossalari bu - ob’ektga berilgan xarakteristika bo‘lib, uning ko‘rinish, joylashishi va holatidir. Masalan, Width va Height xossalari forma o‘lchamini, top va Lift esa formaning ekrandagi holati, Caption - sarlovha matnini aniqlaydi.

Vizual dasturlash texnologiyasida ob’ekt deganda muloqat oynasi va boshqarish elementlari (kiritish va chiqarish maydoni, buyruq tugmalari, pereklyuchatellar va boshqa) tushuniladi.

C++ Buliderda dasturlash ikkita o‘zaro ta’sir etuvchi bir-biri bilan bog‘liq jarayon asosida tashkil qilinadi:

- dasturni vizual loyihalash jarayoni;
- dastur kodlarini kiritish (yozish) jarayoni.

Kodlarni yozish uchun maxsus kod oynasi mavjud bo‘lib, u dastur matnini kiritish va taxrirlash uchun mo‘ljallangandir. Bu kodlarni yozish oynasida dasturlash Pascal tilining rivoji bo‘lgan va kengaytirilgan Object Pascal tilida tuziladi.

Kodlarni yozish oynasi boshlanishda o‘z ichiga hali bo‘sh formani akslantiruvchi dastur matnini yozib chiqaradi. Dastur loyihasini ishlashi mobaynida dasturchi kerakli dastur operatorlarini kiritib, formani loyiha bo‘yicha akslantiradi. C++ Buliderda dasturlash forma oynasini tashkil etishdan boshlanadi.

Oddiy dastur ilovasini yaratish ketma-ket File=> New=> Applisation buyrug‘ini berish bilan boshlanadi. Bu buyruqni berishdan oldin ikkita asosiy ishni bajarish lozim:

- papka tashkil etish;
- tizimni to‘g‘rilash.

Papka tuzing, masalan, **My_C++** Buliderda nomli. **My_C++** Bulider papkasi ichida yana o‘z dasturnigizni saqlash uchun papka ochish, masalan Pgm_1.

C++ Bulider muhitining standart nastroykasiga o‘zgartirish kiritish uchun Tols=>Environment Options menyu buyrug‘ini berish va muloqat darchasidan krakli o‘zgarishlarni bajarish lozim.

C++ Bulider dasturlash muhitida ishlash jarayonida quyidagi kengaytmali fayllar ishlatiladi:

- loyiha fayli, kengaytmasi **.dpr**;
- paskal moduli fayli, kengaytmasi **.pas**;
- komponentalar joylashgan fayl, kengaytmasi **.dcu**;
- formalar joylashgan fayl, kengaytmasi **.dfm**;
- ma’lumotlar bazasi fayli, kengaytmasi **.dbf**.

Tayyorlanadigan C++ Bulider dastur uchta asosiy etapdan o‘tadi:

- kompilyasiya;
- komponovka;
- bajarish.

Kompilyasiya etapida tayyorlangan dastur matni Object Pascal tiliga o‘tkaziladi. Kompanovka etapida esa kerakli qo‘shimcha yordamchi dasturlar va ostdasturlar unga birlashtiriladi. F9 tugmasini bosish bilan Save UnitAs dialog oynasi paydo bo‘ladi va sizdan Unit.pas moduli uchun fayl nomini va joylashadigan papkani ko‘rsatishingizni so‘raydi. Agar joy ko‘rsatilmasa C++ Bulider avtomatik ravishda dasturnigizni Bin papkasiga joylashtiradi. YAxshisi siz bu papkani o‘z ishchi papkangiz nomiga almashtiring, masalan My_Delph. Dastur kompilyasiya qilinishi paytida C++ Bulider sistemasi pas, dfm va dcu kengaytmali mod-

ullar tuzadi. .pas kengaytmali fayl kodlarni yozish oynasida kiritilgan dastur matnini, .dfm forma oynasi tashkil etuvchilarini, .dcu kengaytmali fayl esa .pas va .dfm kengaytmali fayllarning birgalikdagi mashina kodiga o'tkazilgan variantini saqlaydi. Bu .dcu kengaytmali fayl komplyator tamonidan tashkil qilinadi va yagona ishchi (bajariluvchi) .exe kengaytmali fayl tashkil qilishga baza yaratadi.

Adabiyotlar ro'yxati

1. ГрадиБуч. Объектно –ориентированной анализ и проектирование с примерами приложений на C++. Невский диалект, 560 стр, 2001 г.
2. Грехем И. Объектно ориентированные методы. Принципы и практика. Вильямс. 879 стр, 2004 г.
3. Иванова Г.С. Объектно ориентированное программирование. Учебник. МГТУ им Баумана. 320 стр, 2003 г.
4. Страуструп Б. Язык программирования C++, "ДиаСофт", Киев 1993г.

Mavzu: Komponentlar palitrasi. Palitra bo'limlari va ayrim komponentlar xossalari bilan tanishish. Stanford bo'limi. Frame, MainMenu, PopupMenu, Label, Edit, Button, Memo, Panel komponentlar xossalari.

Komponentlar palitrasi bu dasturni boshqarish uchun maxsus tugmachalar bo'lib uni formaga joylashtirishdan oldin bosh oynadan kerakli komponentalar palitrasi tanlanadi. C++ Bulider 6 dasturlash muhitida Standart, Additional, DataAccess, DataControls, Diologs, System kabi komponentlar palitrasi bo'limlari mavjud. Masalan, Standart (Standart) komponentalar palitrasida quyidagi piktogrammalar (tugmachalar) majmuasi mavjud:



MainMenu - dastur bosh menyusi. Komponenta murakkab ierarxik strukturali menyu yaratish uchun xizmat qiladi.

PopupMenu - yordamchi yoki lokal menyusi. Bu menyu oynada sichqoncha o'ng tugmasini bosish bilan chiqadi.

Label - metka (belgi). Bu komponenta forma oynasiga uncha uzun bo'lmagan bir qatorli yozuvni chiqarishda ishlatiladi va uning piktogrammasi panelda "A" ko'rinishda berilgan.

Edit - kiritish qatori. Forma oynasida matnli qator kiritish va taxrirlashda ishlatiladi.

Memo - ko'pqatorli matn muxarriri. Kupqatorli matnlarni kiritish yoki chiqarishda ishlatiladi.

Button - buyruq tugmasi (Obrabotchik sobitiya OnClick). Bu komponenta dasturchi tamonidan berilgan bir necha buyruqlarni bajarishda ishlatiladi.

CheckBox - bog'liq bo'lmagan tanlash tugmasi (pereklyuchatel). Dasturda bu komponenta asosiy mantiqiy xossasi (Checked) o'zgartiriladi.

RadioButton - bog'liq bo'lgan tanlash tugmasi (pereklyuchatel). Yangi tanlash tugmasi bosilganda, oldin tanlangan tugma avtomatik ravishda ozod etadi.

ListBox - ro'yxatdan tanlash. Ro'yxat variantlarini taqdim etadi va tanlash imkonini yaratadi.

ComboBox – kiritish qatoriga ega (kombinirovanny) ro'yxatdan tanlash. Ro'yxatdan kombinatsiya qilib tanlash

ScrollBar - yo'lchali boshqarish. Windows oynasi chetlarida gorizontal yoki vertikal yo'lcha tashkil etadi.

GroupBox - elementlar guruhi. Ma'no bo'yicha bir necha bog'lik komponentalarni gruhlashda ishlatiladi.

RadioGroup - bog'liq guruhlangan tanlash tugmalari (o'chirib yoquvchi tugmalar). Bir necha bog'liq tanlash tugmalari xossalari saqlaydi.

Panel - panel. Bu komponenta, xuddi GroupBoxga o‘xshab bir necha komponentalarni birlashtirish uchun xizmat qiladi.

Actionlist - ta’sir qilish ro‘yxatlari. Foydalanuvchi dasturga markazlashgan holda ta’sir qilishi uchun ishlatiladi.

Label, Edit, Memo matn komponentlari va Button tugmachasi bilan ishlash texnologiyasi

Label belgisi. Belgi tushuntirishlar, nomlar, mavzular va boshqa har xil turdagi matnli ma’lumotlarni ekranga joylashtirish uchun ishlatiladi. Belgi uchun **Caption** asosiy xossalardan biri bo‘lib, unda ekranga chiqariladigan matn joylashadi.

Matnni ekranga joylash uchun C++ Buliderning **Standart** palitrasidan (uskunalar panelidan) “A” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn joylashtirilishi lozim bo‘lgan joy ajratiladi. Natijada **Label1** matn maydoni hosil qilinadi va **Caption** xossasiga kirilib kerakli matn teriladi.

Matnga ishlov berish uchun (masalan, kattalashtirish yoki kichiklaytirish; kursiv yoki qalin qilish va boshqa) ya’ni unga o‘zgartirish kiritish uchun kerakli xossa tanlanib ular o‘zgartiriladi. Maslan, kiritilgan matnni kattalashtirish yoki kichirtish uchun oldin matn maydoni ajratilib, keyin **Font** xossasiga kiriladi va muloqot darchasidan shrift, uning o‘lchami va rangi tanlanib Ok tugmasi bosiladi.

Label komponentasi nafaqat ma’lumotlarni ekranga joylash-tirish uchun xizmat qiladi, balki dastur natijalarini chiqarishda ham ishlatish mumkin. Buning uchun dasturda **Label5.caption:='Dastur natijasi'**; buyrug‘i berilishi kerak. Misol, **Label5.caption:='Echim='+s**; bu erda **s:String** o‘zgaruvchisi.

Edit kiritish qatori. Edit kiritish qatori matnni bir qatordan kiritish va uni tahrirlash uchun ishlatiladi.

Matn kiritish qatorini ekranga joylash uchun C++ Buliderning **Standart** palitrası (uskunalar paneli) dan “ab” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn kiritilishi lozim bo‘lgan joy ajratiladi. Natijada **Edit1** matn kiritish maydoni hosil qilinadi. Matnni kiritish dastur ishchi holatiga o‘tilganda bajariladi.

Matn qatoriga kiritilgan ma’lumot faqat matn, ya’ni String (qator) bo‘lib hisoblanadi. Edit kiritish qatorida kiritilgan ma’lumotni dasturda o‘qib va uni raqamga o‘tkazish uchun ko‘p hollarda Val funksiyasidan foydalaniladi. Bu funksiya Turbo Paskalda quyilagicha yoziladi. **Val(Edit1.Text,a,cod)** - bu erda a: Real; - o‘zgaruvchisi bo‘lib, Edit1.Text maydonidagi ma’lumotni raqam qilib o‘zlashtiradi. **cod: Integer**; deb e’lon qilinadi.

Memo matn chiqarish qatori. Memo matnlarni bir necha qator qilib chiqarish uchun ishlatiladi.

Memo matn chiqarish qatorini ekranga joylash uchun C++ Buliderning **Standart** palitrası (uskunalar paneli) dan “ab” piktogrammasi yonidagi Memo tugmasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn chiqarilishi lozim bo‘lgan joy ajratiladi. Natijada **Memo1** matn chiqarish maydoni hosil qilinadi. Bu matn chiqarish maydoni dasturda natijalarni chiqarishda

qo‘l keladi. Natijani chiqoishda u dastur ichida quyidagicha ishlatiladi.

```
Memo1.Lines.add('Echim='+S);
```

Memo maydonini tozalash esa natijani chiqarishdan oldin modulda **Memo1.Clear;** buyrug‘ini berish bilan amalga oshiriladi.

Button tugmachasi. Button tugmachasi bosilishi natijasida kutilishi lozim bo‘lgan jarayonlar (masalan, hisoblashlar yoki bajarilishi lozim bo‘lgan operatsiyalar) bajarilishga tushiriladi.

Button tugmachasini ekranga joylash uchun C++ Buliderning **Standart** palitrasi (uskunalar paneli) dan “**Ok**” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugmacha qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada **Bottom1** tugmachasi hosil qilinadi. Tugmacha nomini o‘zgartirish **Caption** xosasiga kirilib o‘zgartiriladi.

Dasturdagi hisoblash jarayonlari, kiritish va chiqarish operatsiyalari hosil qilingan tugmachani ikki marta tez-tez bosish bilan “sobitiyani qiyta ishlash” darchasiga o‘tilib, u yerdan modul ichiga kerakli operatorlarni yozish bilan amalga oshiriladi.

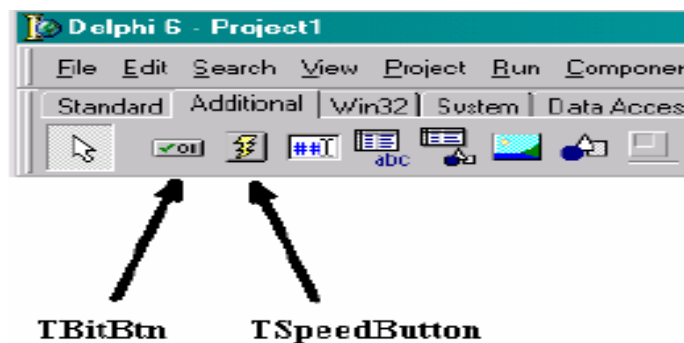
Adabiyotlar ro‘yxati

1. ГрадиБуч. Объектно –ориентированной анализ и проектирование с примерами приложений на C++. Невский диалект, 560 стр, 2001 г.
2. Грехем И. Объектно ориентированные методы. Принципы и практика. Вильямс. 879 стр, 2004 г.
3. Иванова Г.С. Объектно ориентированное программирование. Учебник. МГТУ им Баумана. 320 стр, 2003 г.
4. Страуструп Б. Язык программирования C++, "ДиаСофт", Киев 1993г.

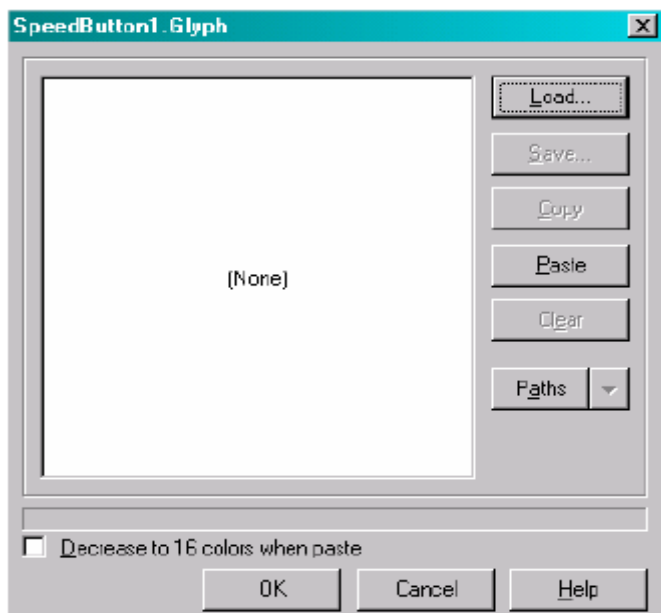
Mavzu: Komponentlar palitrasi. Palitra bo'limlari va ayrim komponentlar xossalari bilan tanishish. Additional bo'limi. BitBtn, MaskEdit, StringGrid, Image komponentlar xossalari.

TSpeedButton va TBitBtn tugmalari.

Bu tugmalar *TButton* vazifalarini bajaradi. YAgona farqi matndan tashqari racmlarni ham aks ettiradi. *TSpeedButton* tugmasi fokus olmaydi. Bushuni bildiradiki, agar matn qatorida satr terib, bu tugma bosilsa, shu hodisa qayta ishlangandan so'ng fokus yana matn qatoriga qaytib keladi. TAB tugmasi bilan butugmanija ratib bo'lmaydi.



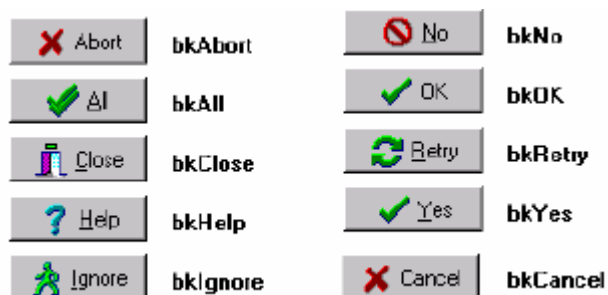
Tugmaga rasm o'rnatish uchun ikki marta *Glyph* xossasi qatorida chertish lozim. Natijada rasm paydo bo'lgan yuklash oynasida *Load* tugmasini bosish lozim. Ko'p rasmlar ProgramFiles\Common Files\Borland Shared\Images \Buttons katalogida joylashgandir.



TBitBtn va *TSpeedButton* tugmalari deyarli bir xil xossalarga egadir. Ular uchun umumiy *Layout*, xossasi rasm va matn o'zaro joylashuvini o'zgartirishga imkon beradi. Quyidagi rasmda har xil qiymatlar mos variantlari ko'rsatilgan



TBitBtn tugmasining yana bir xossasi *Kind* bo‘lib oldindan tayyorlangan standart tugmalarni tanlash imkonini beradi. Quyidagi rasmda standart tugmalar va ularga mos qiymatlarni ko‘rish mumkin.

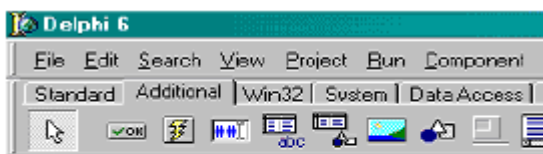


Yana bir xossa *ModalResult* – dialog oynasi uchun tugma qaytaradigan natijani tanlashga imkon beradi.

TSpeedButton tugmasining *GroupIndex* xossasi tugmalarni guruhlashga imkon beradi. Buning uchun bir guruhga tegishli tugmalarning *GroupIndex* xossasi bir xil qiymatga masalan 1 ga teng bo‘lishi kerak. Guruhlangan tugmalarning biri bosilsa, qolganlaridan ajralib qoladi. Buning uchun *Down* xossasi qiymati *true* ga teng bo‘lishi kerak.

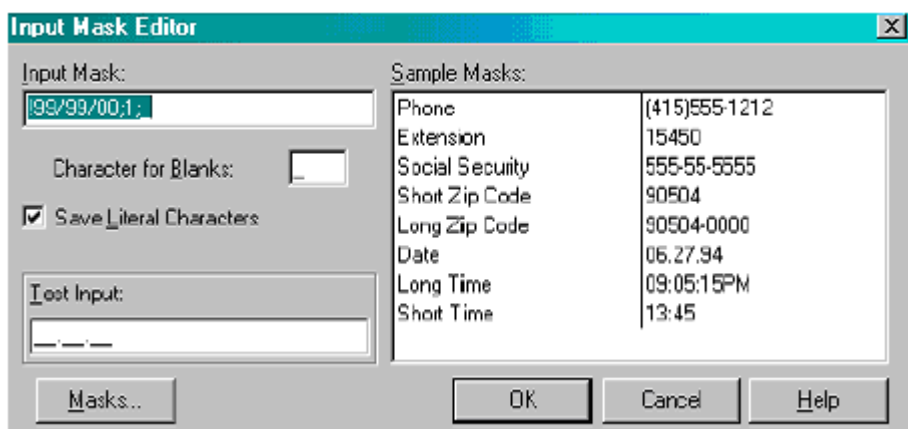
Maskalangan kiritish qatori (*TMaskEdit*)

Bu komponent ma’lum formatga mos satr kiritishga imkon beradi.



↑
TMaskEdit

Asosiy xossasi *EditMask* bo‘lib, shu xossa qatorida ikki marta chertilsa kiritish muharriri ochiladi.



Input Mask qatorida maska kiritish mumkin. *Test Input*, qatorida maskani testlash mumkin.

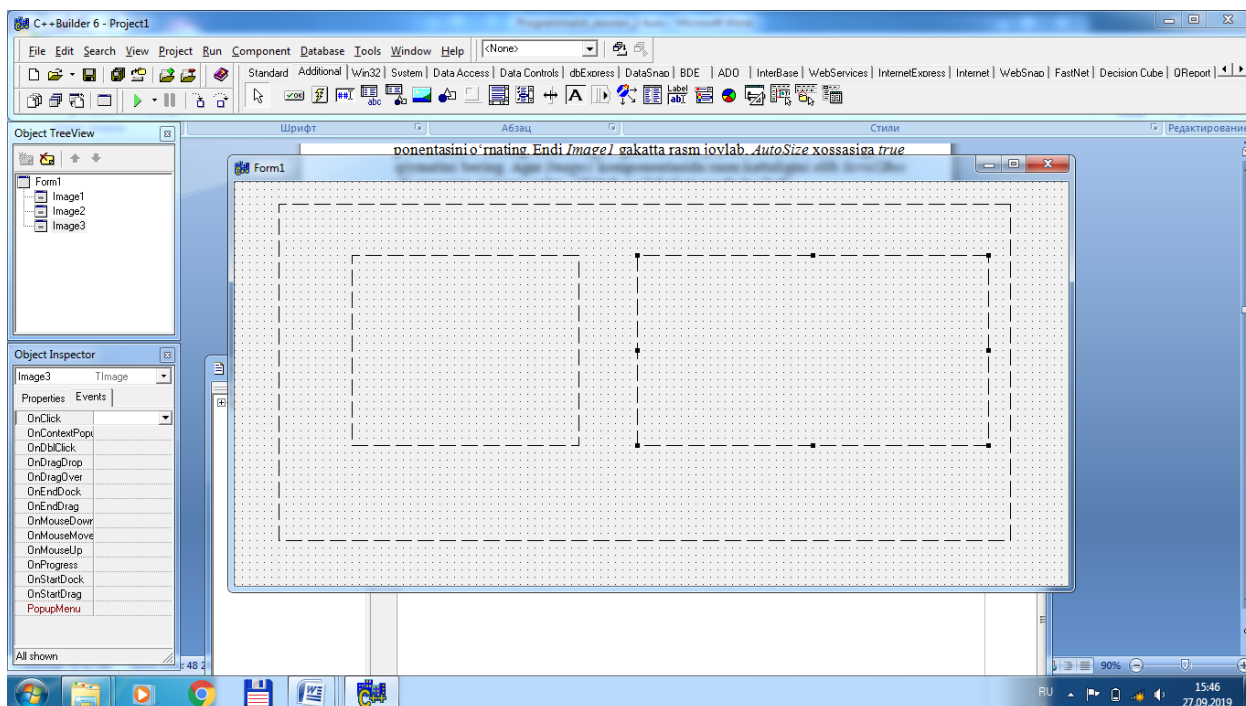
Maska terish osondir. Agar qator to'rt raqamli son, tere va uch raqamli sondan iborat bo'lishi kerak bo'lsa, *Input Mask* qatoriga 9999-999 kiritish mumkin.

Siljitish yo'lchasiga ega panel (*TScrollBox*)

TScrollBox komponentasining oddiy Panel komponentasidan farqi siljitish yo'lchasiga ega bo'lishi mumkinligidir.

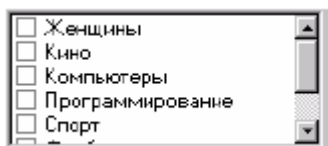


Formaga *TScrollBox* komponentasini o'rnatib, uning ichiga (*TImage*) komponentasini o'rnatib. Endi *Image1* gakatga rasm joylab, *AutoSize* xossasiga *true* qiymatini bering. Agar *Image1* komponentasida rasm kattaligini olib *ScrollBox* chegarasiga sig'may qolsa, siljitish yo'lchalari paydo bo'ladi.

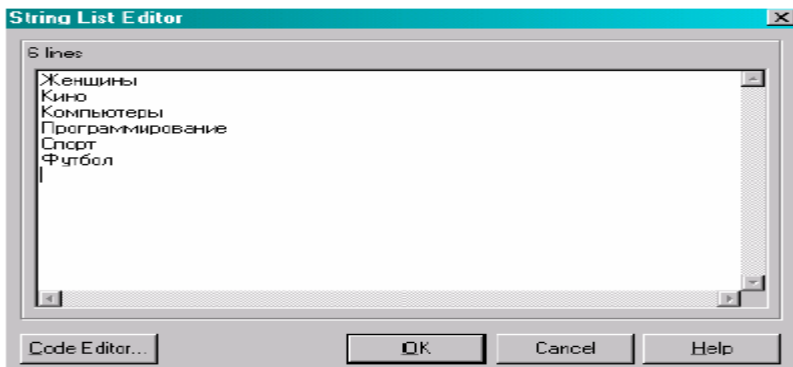


Markirovka qilingan ro'yxat (*TCheckBoxList*)

TCheckBoxList komponentasi *TListBox*, komponentasiga juda o'xshash, faqat har bir ro'yxat yonida *TCheckBox* komponentasidagi kabi ajratish to'rtburchagi mavjuddir.

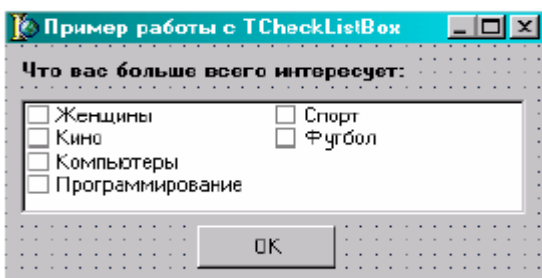


Ro'yxat kiritish uchun *Items* xossasi qatorida ikki marta chertish lozim.



TCheckBox yana bir xossasi – *columns*, ya’ni ustunlar sonidir. Agar bu xossa qiymati birdan katta bo’lsa va ro‘yxat bir ustunga sig‘masa, ko‘rsatilgan sonli ustunlarga ajratiladi.

Quyida shu elementdan foydalanilgan dastur formasini keltiramiz.



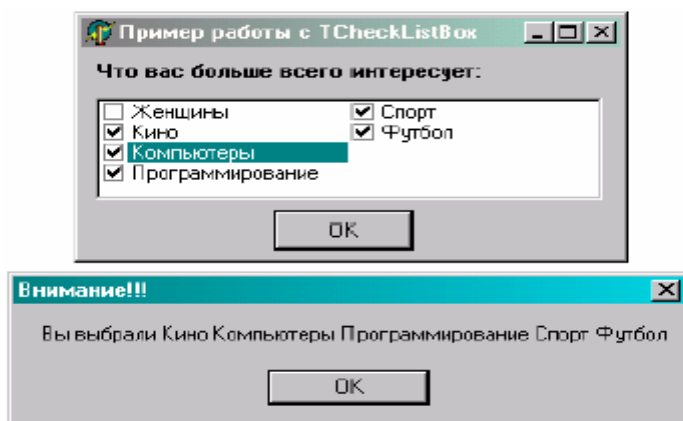
OK tugmasining *OnClick* hodisasi uchun quyidagi protsedurani kiritamiz:

```

procedure TForm1.OKButtonClick(Sender: TObject);
var
i:Integer;
Str:String;
begin
Str:='Siz tanladingiz';
for i:=0 to CheckListBox1.Items.Count-1 do
if CheckListBox1.Checked[i] then
Str:=Str+CheckListBox1.Items[i]+' ';
Application.MessageBox(PChar(Str), 'Dikkat!!!');
end;

```

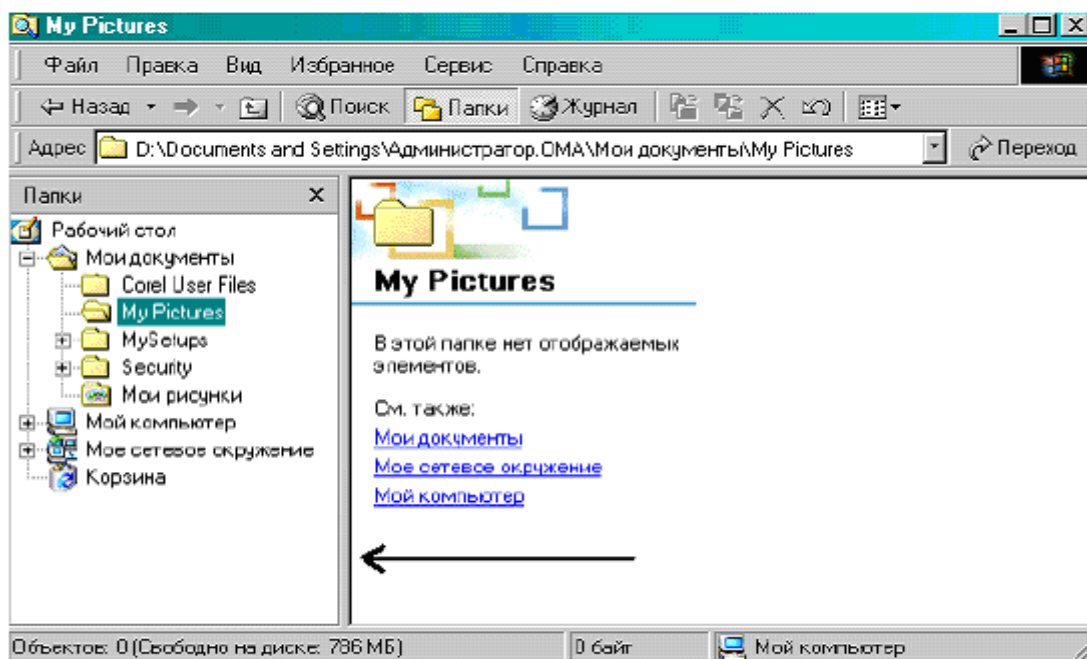
Dastur ishlash natijasiga misol.



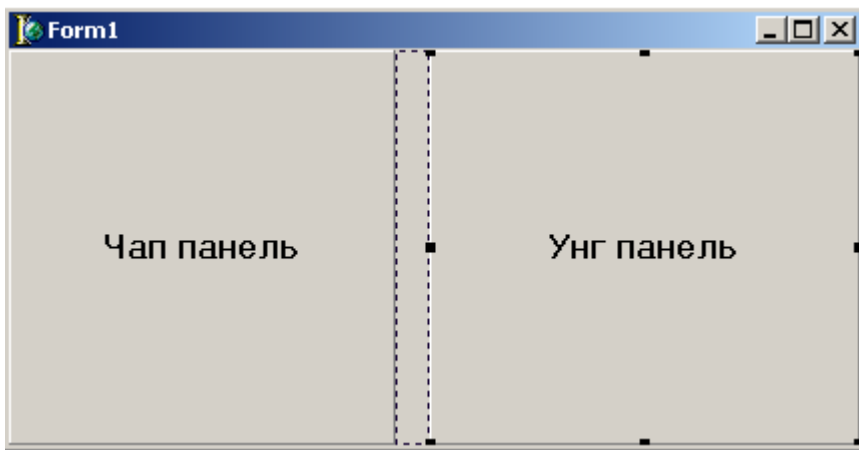
Ajratish yo'lchasi (TSplitter)



Agar Windows Explorerni ochib ko'rsak oynasi ikkiga ajralgan bo'lib, o'rtasida siljitish mumkin bo'lgan yo'lchani ko'ramiz. Mana shu effektini *TSplitter* komponentasi yaratishga imkon beradi.



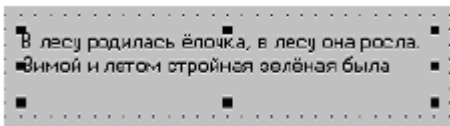
TSplitter komponentasidan foydalanishga misol. Formaga panel (*TPanel*) komponentasini joylashtirib *Align* xossasiga *v allLeft* qiymat beramiz va *Caption* xossasiga «СНар panel » qatorini kiritamiz. Formaga *TSplitter* joylashtirib *Align* xossasiga *v allLeft* qiymat beramiz. Yana bir panel joylashtirib *Align* xossasiga *alClient* qiymat beramiz va *Caption* xossasiga «Ung panel » qatorini kiritamiz. Natija rasmda ko'rsatilgan



Dasturni ishga tushirib, ajratish yo‘lchasi sichqoncha bilan harakatlantirilsa panellar kattaligi o‘zgaradi.

Ko‘p qatorli matn (TStaticText)

Ko‘pincha dasturda bir necha qatorli matn chiqarishga to‘g‘ri keladi. Buning uchun formaga bir necha **TLabel** komponentasini o‘rnatish mumkin. Lekin osonroq **TStaticText** komponentasini o‘rnatib *AutoSize* xossasiga *false* qiymatini berishdir. Rasmda shu komponentadan foydalanishga misol keltirilgan.



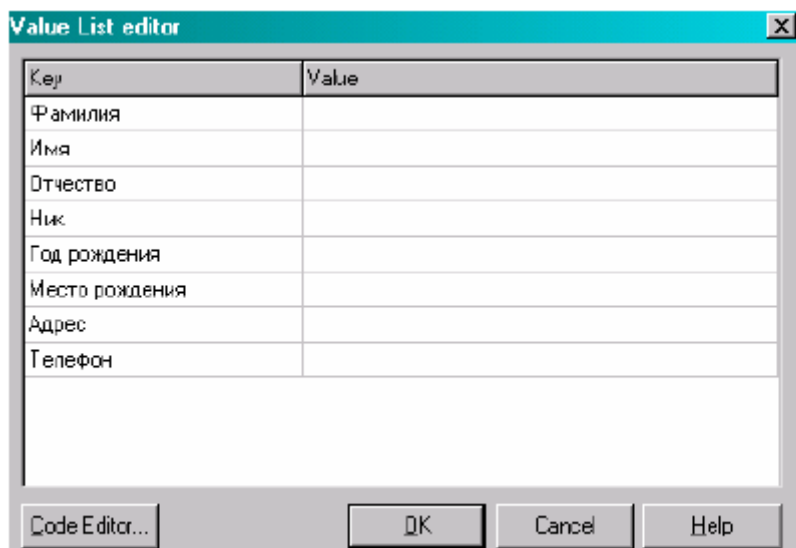
Parametrlar muharriri (TValueListEditor)

Bu komponenta ob’ektlar inspektoridagi kabi xossalar muharririni yaratishga imkon beradi.



Asosiy xossalari:

- *DefaultColimnWidth* – ustunlar ko‘zda tutilgan kengligi;
- *DefaultColimnHeight* – ustunlar ko‘zda tutilgan kengligi;
- *DisplayOption* – komponentani akslantirish opsiyalari;
- *TitleCaptions* – sarlavhalar nomlari. Agar ikki marta chertilsa oddiy matn muharriri chiqadi.
- *FixedColor* – fiksirlangan ustun rangi.
- *FixedCols* – fiksirlangan ustun indeksi.
- *KeyOption* – kalit maydon opsiyalari
- *Strings* – xossalar nomlari. Shu qatorga ikki marta chertilsa xossalar muharriri chiqadi



Formaning OnShow hodisasi uchun quyidagi protsedura yaratamiz:

```

procedure TForm1.FormShow(Sender: TObject);
begin
ValueListEditor1.ItemProps[6].EditStyle:=esPickList;
ValueListEditor1.ItemProps[6].PickList.Add('Moskva');
ValueListEditor1.ItemProps[6].PickList.Add('Piter');
ValueListEditor1.ItemProps[6].PickList.Add('Rostov-na-Donu');
ValueListEditor1.ItemProps[4].EditMask:='99/99/9999';
end;

```

ItemProps. Xossasida ro‘yxat elementlari xossasi joylashgan. Agar 3- element xossasini o‘zgartirish lozim bo‘lsa *ValueListEditor1.ItemProps[2]* yozish kerak.

EditStyle – xossasi tahrirlash uslubini o‘rnatadi. (*ValueListEditor1.ItemProps[6].EditStyle*):= *esPickList* instruksiyasi qator tugma bosilganda chiquvchi qatorga aylantiradi.

ValueListEditor1.ItemProps[6].PickList.Add(tekst elementa) buyrug‘i oltinchi qatorga satr qo‘shadi.

EditMask – xossasi kiritish maskasini yaratishga imkon beradi.

Adabiyotlar ro‘yxati

1. ГрадиБуч. Объектно –ориентированной анализ и проектирование с примерами приложений на С++. Невский диалект, 560 стр, 2001 г.
2. Грехем И. Объектно ориентированные методы. Принципы и практика. Вильямс. 879 стр, 2004 г.

3. Иванова Г.С. Объектно ориентированное программирование. Учебник. МГТУ им Баумана. 320 стр, 2003 г.

4. Страуструп Б. Язык программирования C++, "ДиаСофт", Киев 1993г.

Mavzu: C++ Bulider dasturlari strukturasi. Loyiha va modul. Xodisa tushunchasi, loyiha va modul strukturasi, dastur elementlari (alfavit, identifikatorlar, doimiyliklar, ifodalar va amallar.) Bo'sh forma va uning modifikasiyasi

C++ Bulider loyihasi strukturasi

C++ Bulider dasturi - bu bir necha bir biri bilan bog'liq fayllardir. Har qanday dastur .dpr kengaytmali loyiha fayli va bir yoki bir necha .pas kengaytmali modullardan tashkil topadi. Loyiha fayli dasturchi tomonidan kiritilmaydi, u foydalanuvchining ko'rsatmalari asosida avtomatik ravishda C++ Bulider sistemali dasturi tomonidan tuziladi. Loyiha fayli matnini ko'rish uchun Project/View Source buyrug'ini berish zarur. Loyiha matni umumiy holda quyidagicha bo'lishi mumkin.

Program Project1;

Uses

Forms,

Unit1 in 'Unit1.pas' {Form1}

{ \$R *.res }

Begin

Application.Initialize;

Application.CreateForm(Tform,Form1);

Application.Run;

End.

Loyiha nomi dasturchi tomonidan loyiha faylini saqlash vaqtida beriladi, va u C++ Bulider muhitida bajariluvchi fayl, ya'ni kengaytmasi .exe bo'lgan faylni tashkil qilishni aniqlaydi. Loyiha faylidan keyin ishlatiladigan modullar: standart modullar Forms va Unit1 joylashadi. { \$R *.res } direktivasi kompilyatorga ishlatilishi kerak bo'lgan rusurs fayllari, masalan dasturlarni elon qilish kerakligini bildiradi. Yulduzcha belgisi rusurs faylining kengaytmasi .res ekanligini bildiradi. Bosh modulning bajariluvchi qismi Begin .. End operatorlari orasiga joylashadi.

Modul - bu biror bir dastur. Modullar standart konstruksiyaga ega. Object Pascalda modul strukturasi umumiy holda quyidagi ko'rinishda bo'ladi:

Unit <Modul nomi>

Interfase

.....

Implementation

.....

Initialization

.....

Finalization

End.

C++ Bulider tizimini ishga tushirgandan keyin modul strukturasi quyidagi ko‘rinishda bo‘ladi.

Unit unit1;

Interface

Uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs;

Type

TForm1 = class(TForm)

Private

{ Private declarations }

Public

{ Public declarations }

end;

Var

Form1: TForm1;

Implementation

{ \$R *.dfm }

End.

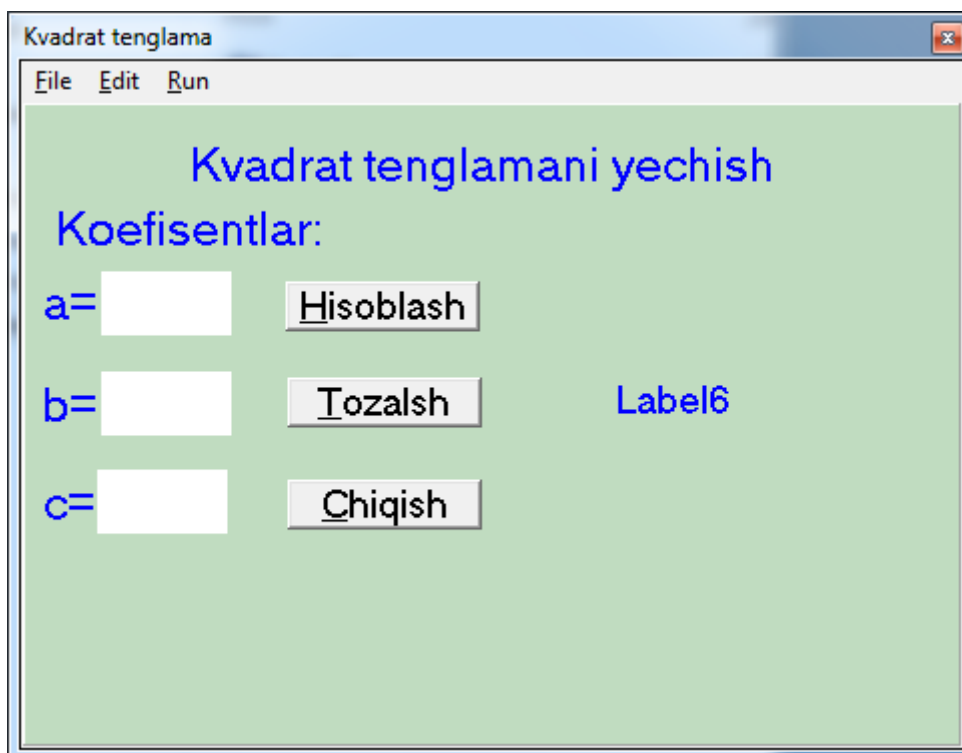
C++ Buliderning imkoniyatlarini va uning vizual loyihalash vositasi texnologiyasini namoyish etish uchun misol tariqasida kvadrat tenglama echimlarini topish dasturini yaratishni qaraylik. Buning uchun loyihaning boshlang‘ich elementlarini yaratishdan boshlaylik. C++ Bulider foydalanuvchiga Form1 nomli standart formani taklif etadi. Foydalanuvchi forma xossalarini Object Inspector oynasidan o‘zgartirish imkoniga ega. Forma xossalari uning ekrandagi ko‘rinishini aniqlaydi. Xossalar ro‘yxatini ob‘ektlar inspektori (Object Inspector) oynasining pastki qatoridagi Propertiesni (xossalarni) aktivlashtirish bilan ko‘rish mumkin. Oynaning chap ustunida xossalar nomlari, o‘ng ustunida xossalarning joriy qiymatlari berilgan. Xossa qiymatini o‘zgartirish uchun, mos xossa yozilgan maydanni sichqonchada chiqillatib, natijada hosil bo‘lgan o‘ng tamondagi unga mos xossa qiymati aniqlanadi, ya’ni o‘zgartiriladi. Masalan, caption (sarlavha) xossasi qiymatini o‘zgartirish uchun, oldin caption sichqonchada chiqillatilib keyin “form1” xossa qiymati klaviaturadan Backspace tugmasini bosish bilan o‘chirilib, o‘rniga “Kvadrat tenglamani echishdasturi” matni kiritiladi.

Masalan, kvadrat tenglamani echish dastursi uchun birinchi uchta piktogramma kerak bo‘ladi. Bu piktogrammalarni formaga joylashtirish uchun komponentalar palitrasidagi kerakli piktogramma ikki marta sichqonchada chiqillatiladi

va keyin forma maydonida hosil bo'lgan kiritish belgisi yoki tugmacha kerakli joyga joylashtiriladi.

Berilgan misol uchun formaga ishlanadigan ilovaga oltitametka qo'yish kerak bo'ladi. Birinchi Label1 tenglama echimlarini chiqarish uchun, ikkinchi Label2 forma boshida ma'lumot berish uchun (masalan, tenglama koeffitsientlari:) va qolgan uchasi Label3, Label4, Label5 taxrirlash maydaniga tushuntirish berish uchun (masalan, koeff. a) formaga qo'yiladi.

Formaga yangi, hisob va chiqish tugmachalarini joylashtirish uchun standart komponentalar palitrasidan Ok piktogrammasi uch marta ikki martadan chiqillatilib, formaning kerakli joylariga qo'yiladi va keyin ular nomlari, ya'ni qiymatlari xossadan aniqlanadi. Natijada quyidagi formaga ega bo'linadi.



Hodisa va uni qayta ishlash. Yaratilgan forma ilovaning qay tarzda ishlashini ko'rsatib beradi. Formadagi buyruq tugmachalari biror ish bajarishi uchun ular sichqonchada ko'rsatilib chiqillatiladi. Sichqonchada tugmachani chiqillatish (bosish) xodisaga misol bo'lib, u ilovaning ishlash jarayonida hosil bo'ladi. Bu erda hodisa so'zini yuz beradigan jarayon deb tushinish kerak.

Hodisalarga javob C++ Buliderda ularning qayta ishlovchi protseduralar ko'rinishida tashkil qilinadi. Turbo Pascal tilida yoziladigan bu protseduralar hodi-sa qayta ishlovchisi ("obrabotchik") deb ataladi.

C++ Bulider avtomatik ravishda qayta ishlovchiga ikkita qismdan iborat nom beradi. Birinchi qism nom formani, ob'ektga kiruvchilarni o'z ichiga olib, ikkinchi qism nom esa aynan ob'ekt o'zini va qayta ishlovchini aks ettiradi. Bizning misolimizda forma nomi - Form1, birinchi buyruq tugmasi nomi "hisob" - Button1, qayta ishlovchi nomi esa - Click. Endi Begin va End orasiga qayta ishlovchi bajaruvchi Paskal tilidagi operatorlarni quyidagi protsedurada kiritish mum-

kin. Bu protsedura “hisob” tugmasini ikki marta tez-tez chiqillatish bilan ekranga chaqiriladi.

Procedure TForm1.Button1click(Sender:Tobject);

Var

A,B,C:Real;

D:Real;

X1,X2:Real;

S1,S2:String[7];

Code:Integer;

Begin

Val(Edit1.Text,a,Code);

Val(Edit2.Text,b,Code);

Val(Edit3.Text,c,Code);

If a=0

Then

Label6.Caption:=’Xato! ’+Chr(13)

+’No’malum ikkinchi darajasi koeffitsenti’

+Chr(13)+’nolga teng’

Else Begin

d:=b*b-4*a*c;

x1:=(-b+Sqrt(d))/(2*a);

x2:=(b+Sqrt(d))/(2*a);

Str(x1:7:3,S1);

Str(x2:7:3,S2);

Label6.Caption:=’Tenglama ildizlari:’

+Chr(13)+’x1=’+S1

+Chr(13)+’x2=’+S2;

End;

End;

Keltirilgan dastur matnida Turbo Pascalning oddiy Read va Write (Kiritish va chiqarish) operatorlari ishlatilmagan. O‘zgaruvchilar qiymatini kiritish taxrirlash maydonidan Text xossasiga murojaat qilish bilan amalga oshiriladi. Kiritilgan o‘zgaruvchilar qiymati matn bo‘lgani uchun ular Val funksiyasi yordamida

raqamga o'tkaziladi. Kvadrat tenglamaning ildizlari x_1 va x_2 lar qiymatlari Str funksiyasi orqali mos ravishda s1 va s2 o'zgaruvchilarga matnli qilib o'zatiladi. Natijani ekranga matn ko'rinishida berish uchun Label6. Caption metkasiga qiymat qilib yuboriladi.

Xuddi shunday "yangi" va "chiqish" tugmachalari uchun ham qayta ishlovchi protseduralarini tashkil qilish kerak. Ular matnlari quyidagi ko'rinishga ega.

Procedure TForm1.Button2Click(Sender:Tobject);

Begin

Edit1.Text:=' ';

Edit2.Text:=' ';

Edit3.Text:=' ';

Label2.Caption:=' ';

Edit1.SetFocus;

End;

Procedure TForm1.Button3click(Sender: Tobject);

Begin

Form1.Close;

End;

Loyihanisaqlash. Ilovani kompilyasiya qilish va ishga tushirish.

Loyihani saqlashda C++ Bulider bir necha fayl tashkil qiladi. Ayrimlari loyiha butun loyihani tavsiflashni, boshqalari forma va dastur modulini tavsiflashni o'z ichiga oladi. Agar hali saqlanmagan loyiha bo'lsa Fayl (File) menyusidan Sox-ranit proekt (Save Project) buyrug'i beriladi va keyin dastur moduli va proekt nomi beriladi.

Loyihani bog'lab bo'lgandan so'ng Compile menyusidan compile (Kompilirovat) buyrug'i beriladi. Agar dasturda sintaksik xato bo'lmasa ekranda kompilyasiya to'g'ri o'tganligi haqida xabar beriladi. Agar kompilyasiya dasturda qandaydir xatoni topsa xato haqida ekranga ma'lumot beradi. Kompilyasiyadan to'g'ri o'tgan dastur uchun maxsus - .exe kengaytmali fayl tuzib beradi va u faylni C++ Bulider tizimisiz ishlatish mumkin.

C++ Bulider tizimidan chiqmasdan turib ilovani ishga tushirish mumkin, buning uchun Run menyusining Run buyrug'ini yoki F9 tugmachasini bosish kifoya bo'ladi. YUqoridagi misol uchun ilova ishga tushirilib a, b va c qiymatlari kiritilib "xisob" tugmasi bosilsa dastur quyidagi natijani ekranga chiqadi.



Protsedura TForm1.Button2Click “yangi” tugmachasini sichqonchada chiqillatish bilan ishlaydi va taxirlash maydoniga kursorni koeffitsient qiymatlarini kiritish uchun olib kelib qo‘yadi.

Protsedura TForm1.Button3Click “tamom” tugmachasini sichqonchada chiqillatish bilan ishlaydi va formani yopadi.

Dasturlash til elementlari

C++ Bulider tili ham boshqa Programmalash asoslari kabi o‘z alfavitiga va belgilariga ega. U 26 bosh lotin harflarini, 0 dan 9 gacha bo‘lgan arab raqamlarini va quyidagi belgilarni ishlatadi: bo‘shliq belgisi; 4 ta arifmetik amallar + , - , * , / ; mantiqiy amallarni bajarish uchun < , > , <= , >= , <> , = belgilarini ishlatadi. Bulardan tashqari vergul, nuqta, ikki nuqta, kichik qavs, katta va o‘rta qavslar. Dasturda izohlar istalgan joyda berilishi mumkin. Ular katta qavs ichida yoziladi.

Masalan. Program ad; { Bu dastur nomi }

O‘zgarmlar, o‘zgaruvchilar va standart funksiyalar

Haqiqiy turdagi sonlar umumiy holda quyidagi ko‘rinishda bo‘ladi:

$$s a_1 a_2 \dots a_n . b_1 b_2 \dots b_k$$

Bu erda s ishora (+ yoki -) yoki bush joy; $a_1 a_2 \dots a_n$ butun qism; $b_1 b_2 \dots b_k$ kasr qism.

Masalan: +3,147 soni +3.147 yoki 3.147

-143,03 soni -143.03

57,0 soni 57.0

0,493 soni 0.493 yoki .493

Haqiqiy sonlarning o‘zgarish diapazoni kompyuterning turiga karab turlicha bo‘ladi. $10^{-38} < x < 10^{+38}$ x-ixtiyoriy son. Ular eksponensial (darajali) ko‘rinishda ifodalanishi ham mumkin, ya’ni $\pm m10^{\pm n}$. Bunday sonlar quyidagicha yoziladi

$\pm mE \pm n$. Masalan:

$$0,43 \cdot 10^{-6} \quad .43E-6$$

0,0003 3E-4

Butun sonlar umumiy holda quyidagicha yoziladi $s a_1 a_2 \dots a_n$.

Masalan: +345 soni +345 yoki 345
-106 soni -106

Butun sonlar o'zgarish diapozoni -32768 dan +32767 gacha. Agar butun son qiymati bu dipazondan chiqsa, u haqiqiy son shaklida ifodalanadi yoki kompyuter turiga qarab, u o'noltilik sanoq sistemasida ifodalanishi ham mumkin. Belgilar qo'shtirnoq ichida yoziladi. YOzilish diapazoni 0 dan 255 tagachadir. Misol. "Paskal", "405.5"

Paskal tilida identifikator tushunchasi majud bo'lib, dasturda ob'ektlarni nomlashda ishlatiladi. O'zgarmaslarni, o'zgaruvchilarni, belgi(metka), protsedura va funksiyalarni belgilashda ishlatilgan nom **identifikatorlar** deyiladi. Identifikatorlar lotin alfaviti harflaridan boshlanib qolgan harflari belgi yoki raqam ketma-ketligidan tashkil topgan bo'lishi mumkin. Masalan: xx, xx1, alfa&.

C++ Bulider tilida dastur ishlashi mobaynida qiymati o'zgarmaydigan identifikatorlar **o'zgarmaslar** deyiladi va ular dasturning bosh qismida **Const** so'zi bilan e'lon qilinib, unga aniq qiymat tenglashtiriladi.

Misol. Const aa1=2.27;
 Pi=3.14;
 radius=14;

Dastur ishlashi mobaynida qiymatlari o'zgarishi mumkin bo'lgan identifikatorga **o'zgaruvchilar** deyiladi va ular dastur bosh qismida **Var** so'zi bilan e'lon qilinadi. O'zgaruvchilar nomi keltirilib, ularning turlari beriladi. O'zgaruvchilarning eng ko'p ishlatiladigan turlari **butun, haqiqiy, belgili, qator** va **mantiqiy**dir. Ular mos ravishda butun - **Integer**, haqiqiy - **Real**, belgili - **Char**, qator (matn) - **String** va mantiqiy - **Boolean** deb yoziladi.

Masalan: Var a, d1, alfa : Integer;
 c121, df : Real;
 Etx, xx : Char;
 St,Sw: String;
 fl : Boolean;

Mantiqiy o'zgaruvchilar faqat ikkita qiymat qabul qiladi: "True" (chin) va "False" (yolg'on).

Standart va nostandart matematik funksiyalar

Funksiya nomi	Tilda yozilishi	Ma'nosi
Sinx	SIN(x)	x ning sinusi
Cosx	COS(x)	x ning kosinusi
Ln x	Ln(X)	x ning natural logarifmi
e^x	EXP(x)	Eksponenta
\sqrt{x}	SQRT(x)	Kvadrat ildiz
Arctgx	ARCTAN(x)	x ning arktangensi
x	ABS(x)	x ning moduli

x^2	SQR(x)	x ning kvadrati
a^b	EXP(b*LN(a))	a ning b chi darajasi

Nostandart matematik funksiyalar.

$$1. \text{Sec}x = \frac{1}{\text{Sin}x}; \quad 2. \text{Co sec}x = \frac{1}{\text{Cos}x}; \quad 3. \text{Tgx} = \frac{\text{Sin}x}{\text{Cos}x}; \quad 4. \text{Arcctgx} = \text{Arctg} \frac{1}{x};$$

$$5. \text{Arcsin}x = \text{Arctg} \frac{x}{\sqrt{1-x^2}}; \quad 6. \text{Arc cos}x = \text{Arctg} \frac{\sqrt{1-x^2}}{x}; \quad 7. \text{Arc sec}x = \text{Arctg} \frac{1}{\sqrt{1-x^2}};$$

$$8. \text{Arc cosec}x = \text{Arctg} \sqrt{1-x^2}; \quad 9. \text{Log}_a b = \frac{\text{Lnb}}{\text{Lna}}; \quad 10. \text{Padian} = \frac{\text{Gradius} \cdot \pi}{180}$$

O'zgartirish funksiyalari

Funksiya	Qiymati
Chr(n)	Kodi n ga teng simvol
IntToStr(k)	Butun k ni tasvirlovchi satr
FloatToStr (n)	Haqiqiy n tasvirlovi satr
FloatToStrF(n, f , k,m)	Haqiqiy n tasvirlovi satr. Bunda: f - format; k - aniqlik; m - kasr qismidagi raqamlar soni
StrToInt (s)	Satrnı butun songa o'tkazish
StrToFloat (s)	Satrnı haqiqiy songa o'tkazish
Round (n)	Haqiqiy sonni yaxlitlash
Trunc (n)	Haqiqiy son kasr qismini olib tashlash
Frac(n)	Kasrli sonning kasr qismi
Int (n)	Kasr sonning butun qismi

Dasturda arifmetik va mantiqiy ifodalar o'zgaruvchi, o'zgarmas, standart funksiyalar, qavslar va amal belgilari orqali tashkil qilinadi.

Ifodalarda hisoblashlar tartibi qavslar ichidagi ifodalar bajarilgandan keyin quyidagi tartibda bajariladi:

1. NOT amali; 2. *, /, DIV, MOD, AND;

3. +, -, OR; 4. taqqoslash belgilari: <, >, <=, >=, <>, =, IN.

Ifodadagi amal natijasi qanday turda bo'lishi amallarda qatnashayotgan o'zgaruvchilarning turlariga bog'liq. Agar ikkita o'zgaruvchining turi Integer yoki Real bo'lsa, amal natijasi ham Integer yoki Real bo'ladi. Agar biri Integer ikkinchi-

si Real bo'lsa natija Real bo'ladi. NOT, OR, AND va taqqoslash amallarining natijalari esa Boolean turida bo'ladi.

Kompyuter foydalanuvchi tomonidan qo'yilgan masalani aniq va tushunarli ko'rsatmalar berilgandagina bajara oladi. Bu ko'rsatmalar ma'lum bir ma'noni anglatuvchi so'zlardan iborat bo'lib, kompyuterga qanday operatsiyani bajarish lozimligini bildiradi va bu ko'rsatmalarga **operatorlar** deyiladi. Operatorlar dastur ishlaganda ketma-ket ravishda bajariladi. C++ Bulider tilida bir satrga bir necha operatorlarni yozish umkin.

C++ Bulider tilida dastur matni bosh va asosiy bo'limdan tashkil topadi. Bosh bo'lim dastur nomi va o'zgaruvchilar, o'zgarmaslar, massivlar, belgilar(metkalar), protseduralar va funksiyalarni tavsiflashdan iborat bo'ladi. Asosiy bo'lim dastur tanasi deyilib, unda dasturda bajariladigan hamma operatorlar ketma-ketligi beriladi va u Begin (boshlamoq) so'zi bilan boshlanib End (tugash) so'zi bilan tugaydi. Umumiy holda dastur strukturasi quyidagi ko'rinishga ega:

Program <dastur nomi>;

Uses <Foydalanadigan bibliotekalar (modullar) ro'yxati>;

Label <Ishlatiladigan belgilar(metkalar) ro'yxati>;

Const <Ishlatiladigan o'zgarmaslarni aniqlash>;

Type <YAngi turlarni aniqlash>;

Var <O'zgaruvchilarni e'lon kilish>;

<Protsedura va funksiyalarni aniqlash>

Begin

<Bajariladigan operatorlar ketma ketligi>

End.

Ma'lumotlar turlari

Ma'lumotlar turlarini C++ Bulider tilida umumiy holda ikkiga ajratish mumkin:

- standart turlar. Bu turlar oldindan C++ Bulider tili tomonidan aniqlangan bo'ladi;
- dasturchi tomonidan kiritiladigan (aniqlanadigan) turlar.

Standart turlar tarkibiga quyidagilar kiradi: butun, haqiqiy, belgili (simvol), qator (strok), mantiqiy, ko'rsatgichli va variant.

Dasturchi turlarni dasturning **Var** bo'limida o'zgaruvchilarni tavsiflashda aniqlaydi yoki maxsus turlarni aniqlash uchun bo'lim bo'lgan -turlarni tavsiflash **Type** bo'limida aniqlaydi.

Bu bo'lim umumiy holda quyidagicha bo'ladi.

Type

<tur nomi>=<turlarning tavsifi>;

Misol:

Type

TColor=(Red,Blue,Black);

Var Color1,Color2,Color3: TColor;

Type bo'limida dasturchi tomonidan yangi Tcolor nomli tur kiritilmoqda va u Red,Blue,Black mumkin bo'lgan qiymatlarni qabul qilishi mumkin.

Var bo‘limida dasturchi tomonidan turi aniqlangan uchta Color1,Color2,Color3 o‘zgaruvchilar tavsiflanmoqda.

Bu o‘zgaruvchilarni to‘g‘ridan to‘g‘ri quyidagicha ham tavsiflash mumkin.

Var Color1,Color2,Color3: (Red,Blue,Black);

Standart turlarni Type bo‘limida tavsiflash shart emas, ularni to‘g‘ridan to‘g‘ri Var bo‘limida tavsiflash mumkin.

C++ Buliderda standart turlarni quyidagicha klassifikatsiya qilish mumkin.

❖ Oddiy

➤ Tartibli

- Butun
- Belgi
- Mantiqiy
- Sanoqli (Perechislyaemy)
- CHegaralangan

➤ Haqiqiy

❖ Qator

❖ Struktura

- To‘plam
- Massiv
- YOzuv
- Fayl
- Klass
- Interfeys

❖ Ko‘rsatgichli

❖ Protsedurali

❖ Variant

Oddiy turlarga tartiblashgan va haqiqiy turlar kiradi. Tartiblashgan turlar shu bilan xarakterlanadiki uning har bir qiymati o‘zining tartiblangan nomeriga ega. Haqiqiy tur qiymatlari kasr qismidan iborat bo‘lgan sonlardan iboratdir.

Tartiblashgan turlarga butun, belgili, mantiqiy, sanoqli va chegaralangan turlar kiradi.

Butun turlar. Butun turlar butun sonlarni tasvirlash uchun ishlatiladi. Jadvalda C++ Bulider 7 da ishlatiladigan butun turlar ro‘yxati keltirilgan.

Tur	O‘zgarish diapazoni	O‘lcham (baytda)
Integer	-2147483648..2147483647	4
Cardinal	0..4294967295	4
Shjrtint	-128..127	1
Smallint	-32768..32767	2
Longint	-2147483648..2147483647	4
Int64	$-2^{63}..2^{63}-1$	8
Byte	0..255	1
Word	0..65535	2

LongWord 0..4294967295 4

Haqiqiy turlar. Haqiqiy turlar haqiqiy sonlarni tasvirlash uchun ishlatiladi. Jadvalda C++ Bulider 7 da ishlatiladigan haqiqiy turlar ro'yxati keltirilgan.

Tur	O'zgrish diapazoni	O'lcham (baytda)
Real	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$	8
Real48	$2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$	6
Single	$1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$	4
Double	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$	8
Extended	$3.6 \cdot 10^{-4951} \dots 1.1 \cdot 10^{4932}$	10
Comp	$-2^{63} + 1 \dots 2^{63} - 1$	8

Belgili turlar. Ma'lumotlarning belgili turlari faqat bitta belgini saqlash uchun xizmat qiladi. Jadvalda C++ Bulider 7 da ishlatiladigan belgili turlar ro'yxati keltirilgan.

Tur	O'lcham (baytda)
Char	1
ANChar	1
WideChar	2

Mantiqiy turlar. Mantiqiy turlar chin (True) yoki yolg'on (False) qiymatning birini qabul qiladi. Jadvalda C++ Bulider 7 da ishlatiladigan mantiqiy turlar ro'yxati keltirilgan.

Tur	O'lcham (baytda)
Boolean	1
ByteBool	1
WordBool	2
LongBool	4

Dasturchi tomonidan kiritiluvchi turlar

C++ Bulider tili dasturchiga uzining turlarini kiritishga imkon beradi. Bu turlar standart turlarga eki avval kiritilgan turlarga asoslangan bulib quyidagi turlarga tegishli bulishi mumkin:

- sanovchi;
- interval;
- murakkab tur (ezuv).

Sanoqli turlar. Sanoqli turlar tartiblangan qiymatlar to'plamini ishlatadi.

Tur =(1 Qiymat, 2 Qiymat, ... ,I Qiymat)

Masalan:

Type

Color=(black,green,yellow blue,red,white);

Fam=(Petrov,Sidorov,Raximov,Sobirov);

DayOfWeek=(mon,tue,wed,thu,fri,sat,sun);

Bu erda

Color sanoq turi beshta ranglar ketma ketligini aniqlaydi.

Fam sanoq turi to'rtta familiyani aniqlaydi.

DayOfWeek sanoq turi hafta nomlarini aniqlaydi.

Odatda C++ Bulider tilida turlar nomlari T xarfidan boshlanadi (Ture — tip so'zidan).

Yangi tur ta'riflangandan sung shu turga tegishli o'zgaruvchini ta'riflash mumkin, masalan:

```
type
```

```
TDayOfWeek = (MON,TUE,WED,THU, FRI,SAT,SUN) ;
```

```
var
```

```
ThisDay, LastDay: TDayOfWeek;
```

Sanovchi tur ta'rifi qiymatlar o'zaro munosabatini ko'rsatadi. Eng chap element minimal, eng ung element maksimal hisoblanadi. YUqorida kiritilgan DayOfWeek turi elementlari uchun quyidagi munosabat o'rinli:

```
MON < TUE < WED < THU < FRI < SAT < SUN
```

Sanovchi tur elementlari orasidagi munosabat o'zgaruvchilarni boshqaruvchi instruksiyalarda qo'llashga imkon beradi, masalan:

```
if (Day = SAT) OR (Day = SUN) then
```

```
begin
```

```
{ agar kun shanba yoki yakshanba bo'lsa bajarilsin }
```

```
end;
```

Bu instruksiyani quyidagicha yozish mumkin:

```
if Day > FRI then begin
```

```
{ agar kun shanba eki yakshanba bo'lsa bajarilsin }
```

```
end;
```

Sanovchi tur ta'rifi nomlangan konstantalar kiritishning qisqartirilgan shakli deb qarash mumkin. Misol uchun TDayOfWeek turining ta'rifi quyidagi ta'riflarga tengdir:

```
const
```

```
MON=0; TUE=1; WED=2; THU=3; FRI=4; SAT=5; SUN=6;
```

Interval (diapazon) turi.Interval (diapazon) turiberiladigan qiymatga chegara qo'yadi.

```
Type
```

```
<tur nomi>=<minimal>..<maksimal>;
```

Masalan:

```
Type
```

```
Color=red..green; // ranga chegara
```

```
Digit=0..9; //butun sonlarga chegara
```

```
Symb='A'..'Z'; // harflarga chegara
```

Haqiqiy turlarga chegara qo'yilmaydi.

Interval tur ta'rifida nomlangan konstantalardan foydalanish mumkin. Quyidagi misolda interval tur TIndex ta'rifida HBOUND nomlangan konstantadan foydalanilgan:

```
const
```

```
HBOUND=100;
```

```
type
TIndex=1..HBOUND;
Interval turdan massivlarni ta'riflashda qulaydir:
```

```
type
TIndex =1 .. 100;
var
tab1 : array[TIndex] of integer; i:TIndex;
```

Butun son turidan tashqari asos tur sifatida sanovchi turdan foydalanish mumkin. Quyidagi dastur qismida TMonth sanovchi tur asosida interval tur TSammer ta'riflangan:

```
type
TMonth = (Jan, Feb, Mar, Apr, May, Jun,
Jul, Aug, Sep, Oct, Nov, Dec);
TSammer = Jun.. Aug;
```

Yozuv

Dasturlash amaliyotida standart ma'lumotlardan tashkil topgan murakkab ma'lumotlar bilan ishlashga to'g'ri keladi. Misol uchun talaba to'g'risidagi ma'lumot ismi sharifi, tugilgan yili, adresi, kursi, guruhi va hokazolardan iborat bo'lishi mumkin. Bunday ma'lumotlarni ta'riflash uchun C++ Bulider da yozuv (record) lardan foydalaniladi.

Yozuv bu - alohida nomlangan har xil turli komponentalardan iborat murakkab turdir.

Har qanday tur kabi, "yozuv" type bo'limida ta'riflanishi lozim. Bu ta'rif umumiy ko'rinishi:

```
Nom = record
1_ Maydon: 1_Tip; 2_ Maydon: 2_Tip;...; K_ Maydon: K_Tip; end;
```

Ta'riflarga misollar:

```
type
TPerson = record
f_name: string[20];
l_name: string[20];
day: integer;
month: integer;
year: integer;
address: string[50]; end;
```

```
TDate = record
day: integer; month: integer; year: integer;
end;
```

Yozuv turidagi o'zgaruvchini quyidagicha ta'riflash mumkin:

```
var
student : TPerson; birthday : TDate;
```

YOzuv elementiga (maydoniga) murojaat qilish uchun yozuv nomi va nuqtadan so'ng maydon nomini ko'rsatish kerak. Masalan:

```
Writeln('Imya: ', student.f_name + #13 + 'Adres: ', student.address);
```

Instruksiya ekranga student o'zgaruvchi-yozuvning f_name (nom) va address (adres) maydonlarini chiqaradi.

Ba'zida o'zgaruvchi-yozuv turi o'zgaruvchilar e'lon qilish bo'limida e'lon qilinadi. Bu holda yozuv turi o'zgaruvchi nomidan so'ng ko'rsatiladi. Misol uchun student yozuvi var bo'limida quyidagicha ta'riflanishi mumkin:

```
student: record
f_name:string[20];
l_name:string[20];
day:integer;
month:integer;
year:integer;
address:string[50];
end;
```

With instruksiyasi

With instruksiyasi dasturda maydonlar nomlarini o'zgaruvchi – yozuv nomini ko'rsatmasdan ishlatishga imkon beradi. Umumiy holda with instruksiyasi quyidagi ko'rinishga ega:

```
with nom do
```

```
begin
```

```
( dastur instruksiyasi } end;
```

Misol uchun dasturda quyidagi yozuv ta'riflangan bo'lsin

```
student:record
f_name: string[30];
l_name: string[20];
address: string[50];
end;
```

va studentlar to'g'risidagi ma'lumotlar E1, E2 va E3 o'zgaruvchilarda joylashgan bo'lsin. U holda

```
student.f_name := E1;
```

```
student.l_name := E2;
```

```
student.address := E3;
```

instruksiyalar o'rniga quyidagi instruksiyani yozish mumkin:

```
with student do begin
```

```
f_name := E1; l_name := E2; address := E3; end;
```

Mavzu:Operatorlar va shartli operator. Tarkibiy va bo'sh operatorlar. If...Then...else shartli operatori. Tanlash(case) operatori. Goto o'tish operatori, label (belgilar) xizmatchi so'zidan foydalanish qoidalari bilan tanishish

Tarmoqlanuvchialgoritmlarnidasturlash

C++ Bulider dasturlash tilining if va case operatorlari

Paskalda tarmoqlanuvchi algoritmlarni dasturlash uchun faqat *true* va *false* qiymatlarni qabul qiladigan *boolean* turidagi maxsus o'zgaruvchilar, shuningdek *if* va *case* operatorlari ishlatiladi. *If* operatori mantiqiy turdagi qiymatni yoki

boolean turidagi o'zgaruvchini tekshiradi va hisoblash jarayonining tarmoqlanishini tashkil etadi.

Masalan, agar *bl: boolean, x,y,u:integer* bo'lsa, *if* operatori qo'llanilgan dastur fragmenti quyidagicha bo'ladi:

```
bl:=x>y;  
if bl then u:=x-y else u:=x+y;
```

Case – tanlash operatori ko'rsatilgan turdagi ayrim o'zgaruvchilarning qiymatlaridan kelib chiqib tarmoqlanishni tashkil etadi.

Masalan, agar *in: integer* bo'lsa,

```
case in of  
0: u:=x+y;  
1: u:=x-y;  
2: u:=x*y;  
else u=0;  
end;
```

operatorlar bajarilgandan keyin *in* ning qiymati gamosuhisoblanadi. Agar *in=0* bo'lsa, u holda $u=x+y$, agar *in=1*, bo'lsa $u=x-y$, agar *in=2*, bo'lsa $u=x*y$, *in* ning 0, 1 va 2 dan boshqa ixtiyoriy qiymatlari uchun $u=0$.

C++ Bulider da almashtiruvchi tugmalar tashkil qilish

C++ Bulider da dastur tuzishda tarmoqlanishni tashkil etish uchun almashtiruvchi tugmalar (knopok-pereklyuchateley) komponentlaridan foydalaniladi. Bu tugmaning holati (yoqish - o'chirish) formada ko'rinib turadi. Formada (1-rasm) ikki turdagi almashtiruvchi tugmalar (*TCheckBox*, *TRadioGroup*) keltirilgan.


TCheckBox komponenti yordamida foydalanuvchi ha/yo'q kabi o'zining natijasini ko'rsatishi mumkin. Dasturda tugmalar holatlari *if* operatori yordamida tekshiriladigan mantiqiy o'zgaruvchilarning qiymatlari bilan bog'liq.

TRadiogroup komponenti almashtiruvchi tugmalarga bog'liq tugmalar guruhini tashkil etadi. Guruhdagi bitta tugmaning bosilishi bilan boshqa tugmalar o'chiriladi. *Case* operatori yordamida tahlil qilinadigan, bosilgan tugmalarning (0,1,2,...) nomeri dasturga uzatiladi.


Masala: x, y, z sonlarni kiriting. $u=\sin(x)$ yoki $u=\cos(x)$, yoki $u=\operatorname{tg}(x)$ lardan birini hisoblang. Xohishga qarab uchta sonning kattasini toping: $\max(u, y, z)$ yoki $\max(|u|, |y|, |z|)$.

1 - rasmda ko'rsatilgandek forma yarating va dasturni tuzing.

TSheckBox komponenti bilan ishlash

Standard komponentlar menyusidan  piktogrammasini tanlang va uni formaning kerakli joyiga joylashtiring. Ob'ektlar inspektori yordamida sarlavha (*Caption*) ni "max abs" ga o'zgartiring. Dastur matnida *TSheckBox* turdagi *SheckBox* o'zgaruvchi paydo bo'ladi. Endi bu tugma bosilgan yoki yo'qligiga qarab **CheckBox1.Checked** mantiqiy o'zgaruvchi *true* yoki *false* qiymatlarni qabul qiladi.

TRadioGroup komponenti bilan ishlash

Standard komponentlar menyusidan  piktogrammasini tanlang va formaning kerakli joyiga joylashtiring. Formada *RadioGroup1* sarlavhali to'rtburchak hosil bo'ladi. Sarlavha (*Caption*) nomini $U(x)$ bilan o'zgartiring. *TRadioGroup* komponentida tugmalarni o'rnatish uchun *Columns* ga 1 ni yozish kerak. SHundan keyin *Items* ning o'ng tomonini sichqoncha bilan ikki marta bosning. Tugmalar sarlavhalarining ro'yxati muharriri paydo bo'ladi. Birinchi qatorga $\cos(x)$, ikkinchi qatorga $\sin(x)$, uchinchi qatorga esa $\text{tg}(x)$ ni yozing va *OK* ni bosning.

Dastur matnida *TRadioGroup* turidagi *RadioGroup* o'zgaruvchisi paydo bo'lganiga e'tibor bering. Endi guruhdagi *RadioGroup1.ItemIndex* turidagi butun o'zgaruvchilardan biri bosilganda, (0 dan boshlab hisoblanadi) bosilgan tugmaning nomeri joylashadi.

FormSreate va Botton1Slick hodisalarni qayta ishlash

FormSreate va *Botton1Slick* protseduralar birinchi mavzuda ko'rsatilgandek tuziladi. Protseduralar matni quyida keltirilgan.

Dasturni ishga tushiring va algoritm to'g'ri ishlayotganiga ishonch hosil qiling. Forma.1-rasmda keltirilgan.



1 – rasm

Dastur matni quyida keltirilgan.

```
unitUnit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics,  
Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```

Edit1: TEdit;
Label2: TLabel;
Edit2: TEdit;
Label3: TLabel;
Edit3: TEdit;
CheckBox1: TCheckBox;
RadioGroup1: TRadioGroup;
Memo1: TMemo;
Button1: TButton;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
Edit1.Text:='0,1';
Edit2.Text:='0,356';
Edit3.Text:='0';
Memo1.Clear;
RadioGroup1.ItemIndex:=0;
end;

procedure TForm1.Button1Click(Sender: TObject);
var x, y, z, u, max : extended;
begin
Memo1.Clear;
x:=StrToFloat(Edit1.Text);
Memo1.Lines.Add('x='+Edit1.Text);
y:=StrToFloat(Edit2.Text);
Memo1.Lines.Add('y='+Edit2.Text);
z:=StrToFloat(Edit3.Text);
Memo1.Lines.Add('z='+Edit3.Text);
case RadioGroup1.ItemIndex of
0: begin
u:=cos(x);
Memo1.Lines.Add('cos(x)='+FloatToStr(u));
end;
1: begin

```

```

        u:=sin(x);
        Memo1.Lines.Add('sin(x)='+FloatToStr(u));
    end;
2: begin
        u:=sin(x)/cos(x);
        Memo1.Lines.Add('tg(x)='+FloatToStr(u));
    end;
end;
if CheckBox1.Checked then
begin
if (x>=y) and (x>=z) then max:=x;
if (y>=x) and (y>=z) then max:=y;
if (z>=x) and (z>=y) then max:=z;
Memo1.Lines.Add('max(x, y, z)='+FloatToStr(max));
end;
end;
end.

```

Mavzu: C++ Bulider dasturlash tilida sikl operatorlari.

Takrorlanish deganda, bir xil operatorlarni har xil qiymatlar uchun bir necha marta bajarilishi tushuniladi. Takrorlanishlar soni aniq yoki noaniq formada berilgan bo‘lishi mumkin.

C++ *Bulider* da tilida uch xil takrorlanish operatorlari mavjud.

```

repeat
<operatorlar>
until <shart>;

```

operatori *repeat* va *until* kalit so‘zlar orasidagi operatorlarning takrorlanishini *<shart >=true* bajarilgunicha tashkil etadi, shundan keyin boshqarish takrorlanish operatoridan keyingi operatorga uzatiladi.

```

While<shart>do begin
<operatorlar>
end;

```

operatori *while* *<shart>=false* bajarguncha *begin* va *end* orasidagi operatorlarning takrorlanishini tashkil etadi. *Repeat* dan farqliravishda *While* operatorida agar birinchi tekshirish-dayoq *<shart>=false* bajarilsa, *<operatorlar>* birmarta ham takrorlanmaydi.

```

for i:=i1 to i2 do begin
<operatorlar>
end;

```

operatori *for* *i* o‘zgaruvchini *i1* boshlang‘ich qiymatdani *i2* oxirgi qiymatgacha «bir» qadam bilan *i1* dan *i2* gacha tartibda operatorlarning qaytarilishini tashkil etadi.

```

for i:=i2 downto i1 do begin <operatorlar> end;

```

operatorini qo‘llasak, *i* qiymati birga kamayib boradi.

C++ Bulider dasturlarini sozlash usullari

Deyarli barcha yangi kiritilgan dasturlar ishga tushirilganda xatolar topiladi. Birinchi darajadagi xatoliklar operatorlarning noto‘g‘ri yozilishi (orfografik, sin-

taksik) bilan bog‘liq. C++ *Bulider* kompilyatori xatoni aniqlaganda xato joylashgan qatorning boshini ko‘rsatadi. Ekraning pastki qismida loyihada topilgan xatolar haqida ma’lumot yoziladi. Oynaning har bir qatori xato topilgan faylning nomi, qator nomeri va xato xarakterinio‘z ichiga oladi. Xato haqida to‘liqroq ma’lumot olish uchun *F1* klavishi yordamida *HELP* ga murojaat qilish kerak. SHunga e’tibor berish kerak-ki, bitta xato o‘z navbatida boshqa xatoliklarning bo‘lishiga olib kelishi mumkin va bu xato tuzatilganda boshqalari ham yo‘qoladi. SHuning uchun xatolarni ketma-ket yuqoridan-pastga qarab to‘g‘irlab va har bir to‘g‘irlangandan keyin dasturni kompilyasiya qilish kerak bo‘ladi.

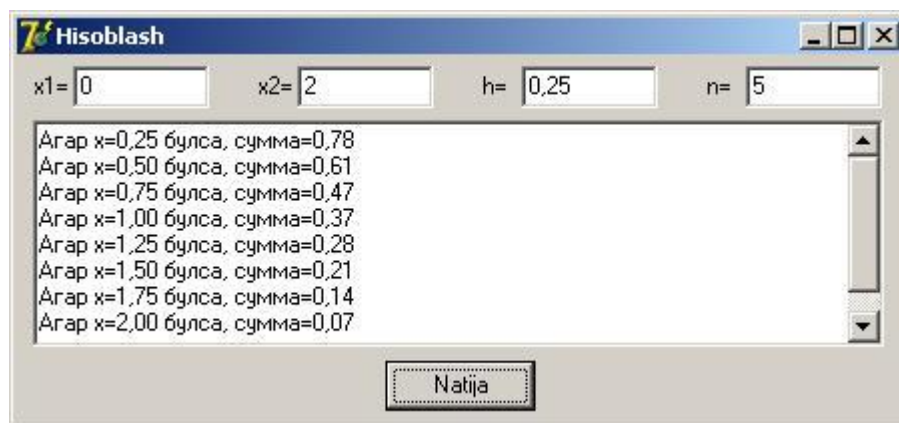
Ikkinchi darajadagi xatoliklar echim algoritmini tanlash yoki algoritmni noto‘g‘ri realizatsiya qilish bilan bog‘liq. Bu xatolar hisob natijalari noto‘g‘ri yoki nolga bo‘lish va h.k. holatlarda paydo bo‘ladi. SHuning uchun dasturni ishlatishdan oldin uni testdan o‘tkazish kerak, ya’ni oldindan natijasi ma’lum bo‘lgan boshlang‘ich qiymatlar kombinatsiyasini hisoblab ko‘rish kerak. Agar test natijalari xatoni ko‘rsatsa, u holda xatoni topish uchun C++ *Bulider* muhitida o‘rnatilgan sozlash vositalarini ishlatish kerak.

Oddiy holatlarda xato topmoqchi bo‘linsa, quyidagicha yo‘l tutish tavsiya etiladi. Matnni tahrirlash oynasida xato bor deb shubha qilingan joydan oldingi satrga kursorni keltirib qo‘ying va *F4* ni bosing. Dastur kursor turgan joygacha bajariladi. Endi kerakli o‘zgaruvchining qiymati qanday o‘zgarganini ko‘rish mumkin. Buning uchun kerakli o‘zgaruvchiga kursorni keltirib (ekranda uning qiymati ko‘rinadi) yoki *Ctrl-F7* ni bosib paydo bo‘lgan muloqot oynasida kerakli o‘zgaruvchini ko‘rsatib bilish mumkin. *F7* (qadamma-qadam tekshirish) ni bosib dasturni qadamma-qadam tekshirib borish mumkin. Agar kursor sikl ichida bo‘lsa, *F4* sikl tanasini bir marta bajarilgandan so‘ng hisob to‘xtaydi. Hisobni davom ettirish uchun *Run* menyusidan *<Run>* ni bosish kerak.

Masala: x o‘zgaruvchi uchun $S(x)$ funksiyaning $X1$ dan to $X2$ gacha intervalda, h qadam bilan qiymatlarini jadvalga chiqarish dasturini tuzing.

$$S(x) = \sum_{k=0}^N (-1)^k \frac{x^k}{k!}$$

Dialoglar paneli 1-rasmda keltirilgan.



1-rasm.

Dastur matni quyida keltirilgan.

```

unit Unit1;

interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms, Dialogs, StdCtrls;
type
    TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Edit4: TEdit;
    Memo1: TMemo;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    private
    { Private declarations }
    public
    { Public declarations }
    end;
    var Form1: TForm1;
implementation
    {$R *.dfm}
    procedure TForm1.FormCreate(Sender: TObject);
    begin
    Edit1.text:='0';
    Edit2.text:='2';
    Edit3.text:='5';
    Edit4.text:='0,25';
    Memo1.Clear;
    end;
    procedure TForm1.Button1Click(Sender: TObject);
    var x1, x2, x, h, a, s : real;
    N, k, c : integer;
    begin
    Memo1.Clear;
    x1:=StrToFloat(Edit4.Text);
    x2:=StrToFloat(Edit2.Text);
    n:=StrToInt(Edit3.Text);
    h:=StrToFloat(Edit4.Text);
    c:=-1;

```

```

x:=x1;
repeat
a:=1;
  S:=1;
for k:=1 to n do
begin
a:=c*a*x/k;
s:=s+a;
end;
Memo1.Lines.Add('Agar x='+FloatToStrF(x,ffFixed,6,2)
+'bo'lsa, summa='+FloatToStrF(s,ffFixed,6,2));
x:=x+h;
until x>x2;
end;
end.

```

Dastur sozlangandan keyin test tuzing ($n=2$, $x1=0$, $x2=1$, $h=3$), kursorni birinchi operator ($n=$) ga olib keling, *F4* tugmasini bosning. Shundankeyin *F7* tugmasini bosib turib dasturni qadamma –qadam tekshiring va dastur bajarilishi jarayonida o'zgaruvchilar qiymatlariga e'tibor bering.

Mavzu: C++ Bulider dasturlash tilida massivlar, massivlarni tavsiflash. Massivlarni berilish usullari.

Massiv bitta nom bilan birlashtirilgan bir xil tipli elementlarning tartiblangan majmuasidan iborat. Massivlarning har bir elementi nom bilan belgilanadi, ularda orqasidan kvadrat qavslarda vergullar bilan ajratilgan bir yoki bir nechta indeksar keladi, masalan: $a[1]$, $bb[I]$, $c12[I,j*2]$, $q[1,1,I*j-1]$. Indeks sifatida LongIntdan tashqari ixtiyoriy tartiblangan tiplardan foydalanish mumkin.

Massiv tipi yoki massivning o'zi mos ravishda tiplar (type) yoki o'zgaruvchilar (Var) bo'limida **Array** kalit so'zi yordamida quyidagicha aniqlanadi:

Array [indekslar tavsifi] **of** <massiv elementlari tipi>

Massivlarni tavsiflashga misollar:

```

Const N=20; // indeksning maksimal qiymatini berish;
Type TVector=array[1..N] of real; // bir o'lchovli massiv tipini tavsiflash;
Var a:TVector; // A –Tvector tipli massiv;
  Ss:array[1..10] of integer; // Ss – o`nta butun sonli massiv;
  Y:array[1..5,1..10] of char; // Y – simvol tipidagi ikki o'lchovli massiv.

```

Massiv elementlaridan ifodalarda oddiy o'zgaruvchilar kabi foydalanishi mumkin, masalan:

```

F:=2*a[3]+a[ss[I]+1]*3;
A[n]:=1+sqrt(abs(a[n-1]));

```

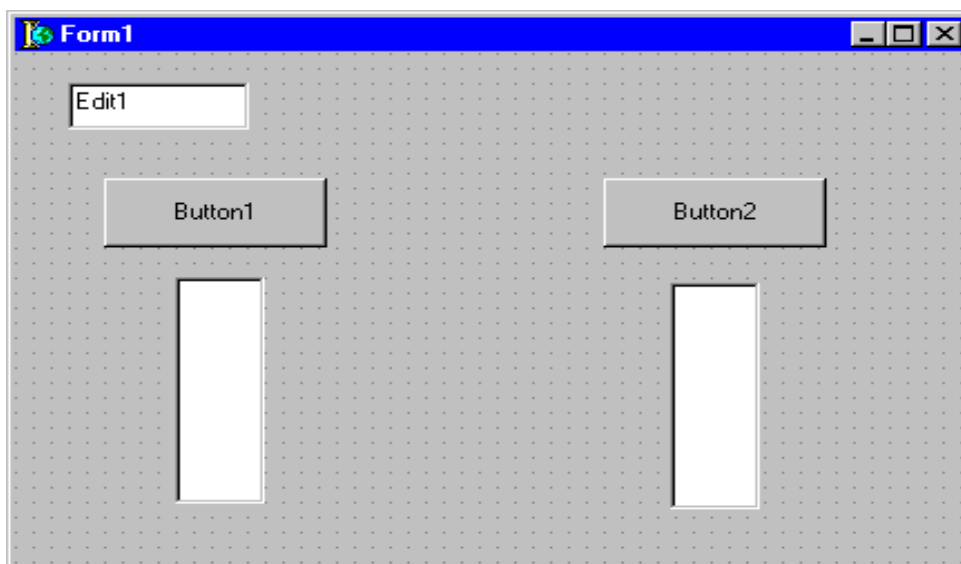
TStringGrid komponenti

Massivlar bilan ishlashda ekranga axborotni kiritish va chiqarishni jadval shaklida tashkil etish qulay. TStringGrid komponent axborotni ikki o'lchovli jadval shaklida aks ettirishga mo'ljallangan. Uning har bir yacheykasi bir satrli redaktorli oynasidan (Tedit oynasiga o'xshash) iborat. Axborotga kirish Cells[ACol, ARow: Integer]:string xossasi yordamida amalga oshiriladi, bu erda ACol, Arow – ikki o'lchovli massiv elementi indeksi. ColCount va RowCount xossalari jadvaldagi satr va ustunlar sonini o'rnatadi, FixedCols va FixedRows xossalari tayinlangan sohadagi satr va ustunlar sonini beradi. Tayinlangan soha boshqa rang bilan belgilangan, unga klavaiaturadan axborot kiritish man qilinadi.

Misol.A(10) butun sonli massivning maksimal va minimal elementlarini topish va ularning joyini almashtiradigan ilova yaratamiz.

Yechish:

1. Yangi loyiha uchun papka yaratamiz, masalan, **C:\MyProject\Lbox**.
2. Bosh menyudan **File | New | Applicaton** buyrugi yordamida yangi loyiha ochamiz.
3. **Form1**ga **Standard** sahifasidan quyidagi elementlarni joylashtiramiz
4. Ob'yektlar Inspektori yordamida **Edit1** elementning **Text** xossasiga ki-



1-pacm. **Edit, Button, ListBox**larning shaklda joylashishi

ymat sifatida bo'sh satrni beramiz. Knopkalarining chapdaxisiga **Caption** xossasiga 'Kiritish', ungdaxisiga esa 'Echish' kiyimatlarini o'rnatamiz.

5. **Interface** bo'limida **A** massiv va ishlatilayotgan uzgaruvchilarni tavsiflaymiz.

A: array [1..10] of integer;

I, min, max, imin, imax: integer;

6. **Form1** shaklda sichqonchani ikki bosish qilib **OnCreate** hodisasi uchun operatorlar yozamiz. Bu erda yozilgan operatorlar ko'pgina hollarda modulning **initializaton** seksiyasida joylashtirilgan operatorlar bilan ekvivalent. U erga quyidagi satrlarni joylashtiramiz:

i:=0;

ListBox1. Clear;

ListBox2. Clear;

7. ‘**Kiritish**’ tugmasida sichqonchani ikki bosish qilib faollashtiramiz va **OnClick** hodisasi uchun kod yozamiz:

ListBox1.Items.Add(Edit1.Text);

i:=i+1;

a[i]:=StrToInt(Edit1.Text);

Edit1.SetFocus

8. ‘**yechish**’ tugmasini ikki bosib, quyidagini joylashtiramiz:

max:=a[1];

imax:=1;

min:=a[1];

imin:=1;

for i:=2 to 10 do

begin

if max<a[i] then

begin

max:=a[i];

imax:=i

end;

if min>a[i] then

begin

min:=a[i];

imin:=i

end;

end;

a[imax]:=min;

a[imin]:=max;

for i:=1 to 10 do

ListBox2.Items.Add(IntToStr(a[i]));

9. Ilova bilan qulay bo’lishi uchun **Enter** klavishini bosganda ‘**Kiritish**’ tugmasiga fokusni yuborish uchun **Edit1** ga **onKeyDown** hodisagakod yozamiz:

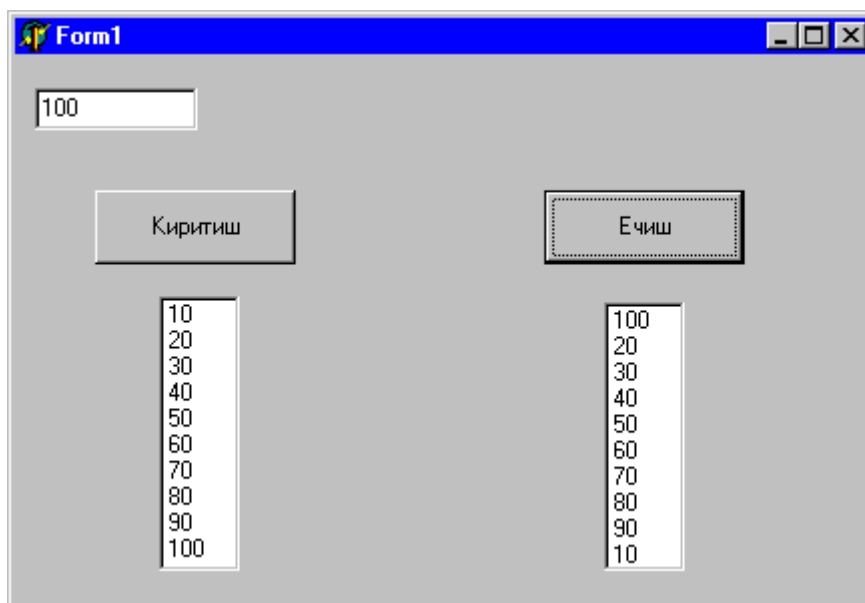
if key=13 then Button1.SetFocus

10. Loyixani bosh menyuning **File | Save All** buyrugi yordamida saqlaymiz.

11. **F9** klavishi bilan dasturni ishga tushiramiz. Butun sonlarni kiritish uchun **Edit1** satr elementidan foydalanamiz. Har bir sonni kiritishni **Enter** klavishini bosish eki ‘**Kiritish**’ tugmasida sichqonchani bosish bilan yakunlaymiz. Kiritilgan sonlar birinchi ro’yxatda tasvirlanadi. ‘**Echish**’ tugmasini bosgach ikkinchi ro’yxatda natijani olamiz.

Misol. Butun sonli A(4,4) matrictsada birinchi va oxirgi satrlar joyini almashtirish dasturini tuzamiz.

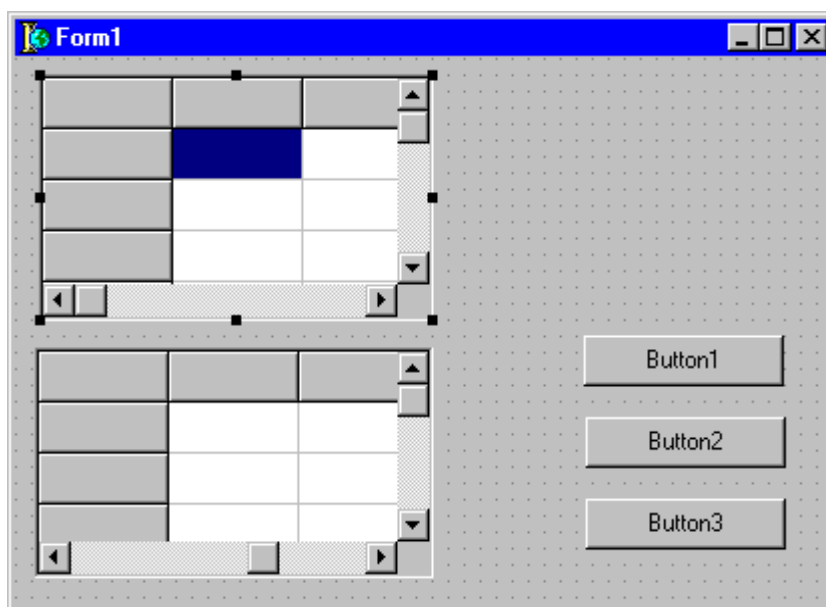
Yechish:



1. Yangi loyiha uchun papka yaratamiz, masalan, C:\MyProject\StringGrid.

2. Bosh menyuning **File|New|Application** buyrug'idan foydalanib yangi loyiha ochamiz.

3. **Form1**ga **Standard** va **Additional** sahifalaridan ikkita **StringGrid** va uchta **Button** elementlarni joylashtiramiz. (4.2-rasm)



2-rasm. **StringGrid** va **Button** elementlarni joylashtiramiz

4. Ob`yektlar Inspektori yordamida **StringGrid1** va **StringGrid2** elementlarning xossalariga quyidagi qiymatlarni beramiz:

FixedCols - 0

FixedRows - 0

ColCount - 4

RowCount - 4

StringGrid1 va **StringGrid2** elementlarning ulchamlarini shunday uzgartiramiz-ki 4 satr va 4 ustun sigadigan bo'lsin.

5. Ob`yektlar Inspektoridan ikki jadval (**StringGrid1** va **StringGrid2**) uchunham **Options** xossalarini tanlaymiz, unda sichqoncha ikki bosish qilib kiramiz. **goEditing** bayroqchani qiyamatini **True**ga teng qilamiz.

6. **Button1**, **Button2**, **Button3** elementlar uchu **Caption** xossalarining qiymati uchun mos ravishda '**Echim-1**', '**Echim-2**', '**Tozalash**' qiymatlarni beramiz.

7. **Echim-1** tugmasini faollashtirib **onClick** hodisasi uchun kod yozamiz:

```
procedure TForm1.Button1Click(Sender:TObject);
    var i:integer;
    begin
        for i:=1 to 3 do
            begin
                StringGrid2.Cells[i,0]:= StringGrid1.Cells[i,3];
                StringGrid2.Cells[i,3]:= StringGrid1.Cells[i,0];
                StringGrid2.Cells[i,1]:= StringGrid1.Cells[i,1];
                StringGrid2.Cells[i,2]:= StringGrid1.Cells[i,2];
            end
        end;
```

Ma`lumki, satr va ustunni nomerlash 0 dan boshlanadi, *Cells[j,i]* xossasidagi birinchi indeks ustun nomerini bildiradi, ikkinchi indeks satr nomerini.

8. Berilgan masalani **Rows** xossa yordamida ham echish mumkin. **Echim-2** tugmasini faollashtiramiz va **onClick** hodisasi uchun kodni echamiz:

```
procedure TForm1.Button2Click(Sender:TObject);
    begin
        StringGrid2.Rows[0]:= StringGrid1.Rows[3];
        StringGrid2.Rows[1]:= StringGrid1.Rows[1];
        StringGrid2.Rows[2]:= StringGrid1.Rows[2];
        StringGrid2.Rows[3]:= StringGrid1.Rows[0];
    end;
```

9. Ikkinchi jadvalni tozalash uchun **Tozalash** tugmasi ishlatiladi. Uni faollashtiramiz va kod yozamiz:

```
procedure TForm1.Button3Click(Sender:TObject);
    var i,j:integer;
    begin
        for i:=1 to 3 do
            for j:=1 to 3 do
                StringGrid2.Cells[j,i]:= '';
            end;
```

1. Bosh menyuning File | Save All buyrug'i yordamida loyixani saqlaymiz va uni ishga tushiramiz.

Masala: $\vec{Y} = A * \vec{B}$ vektorni aniqlash dasturini tuzing, bunda A- NxN o'lchovli kvadrat matritsa, Y, B- N tartibli bir o'lchovli massivlar. Y vektorning elementlari $Y_i = \sum_{j=1}^N A_{ij} \cdot B_j$ formula bilan topiladi.

The screenshot shows a window titled 'Form1' with two 4x4 grids. The top grid contains the following values:

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4


The bottom grid contains the following values:

4	4	4	4
2	2	2	2
3	3	3	3
1	1	1	1

To the right of the grids are three buttons: 'Ечим-1', 'Ечим-2', and 'Тозалаш'.

N ning qiymatini TEdit komponentiga, A va B – TStringGrid komponentiga kiriting. Tbutton tipidagi knopkani bosgandan so`ng natijani TStringGrid komponentiga chiqaring.

TStringGrid komponentni sozlash

TStringGrid komponentini formaga o`rnatish uchun Additional sahifada sichqoncha bilan  piktogrammani bosish kerak. Shundan so`ng sichqoncha bilan formaning kerakli joyida bosiladi. Komponent chekkalarini belgilab uning o`lchovini to`g`rilanadi. Ob`yektlar inspektorida ColCount va RowCount xossalar qiymatlariga 2 ni (ikkita satr va ikkita ustun) o`rnatamiz, FixedCols va FixedRows larda 1 ni (tayinlangan sohalarda bitta satr va bitta ustun). StringGrid2 va StringGrid3 komponentlar faqat bir ustunga ega bo`lgani uchun ularda ColCount=1, RowCount=2, FixedCols=0 va FixedRows=1. Jimlik qoidasiga ko`ra TStringGrid komponentga axborotni klaviaturadan kiritish man qilingan, shuning uchun StringGrid1 va StringGrid2 komponentlar uchun Options goEditing xossaga True holatni o`rnatish zarur.

Dialog paneli 4-rasmda keltirilgan.

The screenshot shows a Windows form titled 'Form1' with a dialog panel. At the top, there are three labels: 'SrtingGrid1', 'StringGrid2', and 'StringGrid3', each with a line pointing to a corresponding grid in the form. The form contains the following elements:

- A label 'Массив улчови:' followed by a text box containing the value '3'.
- A button labeled 'Массив улчовини узгартириш'.
- A large grid labeled 'A - массив:' with columns 'i= 1', 'i= 2', and 'i= 3'. The data is as follows:

A - массив:	i= 1	i= 2	i= 3
i= 1	4	4	4
i= 2	3	3	3
i= 3	2	2	2
- A vertical list box labeled 'B - массив:' containing the values 3, 3, and 3.
- A vertical list box labeled 'Y - массив:.' containing the values 36,00, 27,00, and 18,00.
- A button labeled 'Хисоблаш' at the bottom center.

4-rasm. Dastur formasi

Dastur matni quyida keltirilgan.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dia-
logs,
  StdCtrls, Grids;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    Button2: TButton;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    StringGrid3: TStringGrid;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  end;

```

```

public
  { Public declarations }
end;
const
  Nmax=10;
Type
  Mas2 = array[1..Nmax,1..Nmax] of extended;
  Mas1 = array[1..Nmax] of extended;
var
  Form1: TForm1;
  A : Mas2;
  B,Y : Mas1;
  N,i,j : integer;
implementation
  {$R *.DFM}
  procedure TForm1.FormCreate(Sender: TObject);
  begin
    N:=3;
    Edit1.Text:=FloatToStr(N);
    StringGrid1.ColCount:=N+1;
    StringGrid1.RowCount:=N+1;
    StringGrid2.RowCount:=N+1;
    StringGrid3.RowCount:=N+1;
    StringGrid1.Cells[0,0]:='A - massiv: ';
    StringGrid2.Cells[0,0]:='B - massiv: ';
    StringGrid3.Cells[0,0]:='Y - massiv: ';
    for i:=1 to N do begin
      StringGrid1.Cells[0,i]:=' i= '+IntToStr(i);
      StringGrid1.Cells[i,0]:=' j= '+IntToStr(i);
    end;
  end;
  procedure TForm1.Button1Click(Sender: TObject);
  begin
    N:=StrToInt(Edit1.Text);
    StringGrid1.ColCount:=N+1;
    StringGrid1.RowCount:=N+1;
    StringGrid2.RowCount:=N+1;
    StringGrid3.RowCount:=N+1;
    for i:=1 to N do begin
      StringGrid1.Cells[0,i]:=' i= '+IntToStr(i);
      StringGrid1.Cells[i,0]:=' j= '+IntToStr(i);
    end;
  end;
  procedure TForm1.Button2Click(Sender: TObject);
  var s: extended;

```


```

begin
  for i:=1 to N do
    for j:=1 to N do
      A[i,j]:=StrToFloat(StringGrid1.Cells[j,i]);
    for i:=1 to N do
      B[i]:=StrToFloat(StringGrid2.Cells[0,i]);
    for i:=1 to N do begin
      s:=0;
      for j:=1 to N do s:=s+A[i,j]*B[j];
      Y[i]:=s;
      StringGrid3.Cells[0,i]:=FloatToStrf(y[i],ffixed,6,2);
    end;
  end;
end.

```

Mavzu: Fayllar bilan ishlash. Fayllar bilan ishlashning ob'yektli yondashuvi. C++ Buliderda muloqot oynalari. OpenFileDialog, SaveDialog, ColorDialog, FontDialog komponentalari bilan ishlash.

C++ Bulider tizimida muloqot oynalarini qo'llash uchun maxsus **Dialogs** nomli komponentalar palitrasi mavjud bo'lib, u o'z ichiga bir necha vizual bo'lmagan komponentalarni oladi. Ulardan **OpenDialog**, **SaveDialog** va **FontDialog** komponentalarini ko'rib chiqamiz.

OpenDialog komponentasi kompyuter fayl tizimini ko'rish va undan kerakli fayl nomini tanlash imkonini beradi. Bu komponenta piktogrammasi  ko'rinishga ega. U vizual bo'lmagan komponenta bo'lib, uni formaga sichqonchada bir marta bosib qo'yiladi va keyin uning xossalari o'rnatiladi.

Uning asosiy xossalarini ko'rib chiqamiz:

DefaultExt -faylning kengaytma nomini saqlaydi.

FileName -tanlangan fayl nomini saqlaydi.

Filter -fayl nomlarini muloqot darchasiga ko'rsatilgan kengaytma nom bo'yicha filtrlab chiqaradi. Masalan, agar .pas ko'rsatilgan bo'lsa muloqot oynasida faqat .pas kengaytmali fayllar chiqadi.

Filter xossasiga o'tilib uch nuqtali tugmacha bosilsa Filter Editor muloqot oynasi chiqadi. U ikki qismdan iborat bo'lib, birinchi qismda filtr matni ikkinchi qismida esa filtrning o'zi beriladi.

Masalan:

filtr matni nomlari:

Fayly moduley C++ Bulider (*.pas)

Текстовые документы (*.txt,*.doc)

Vse fayly (*.*)

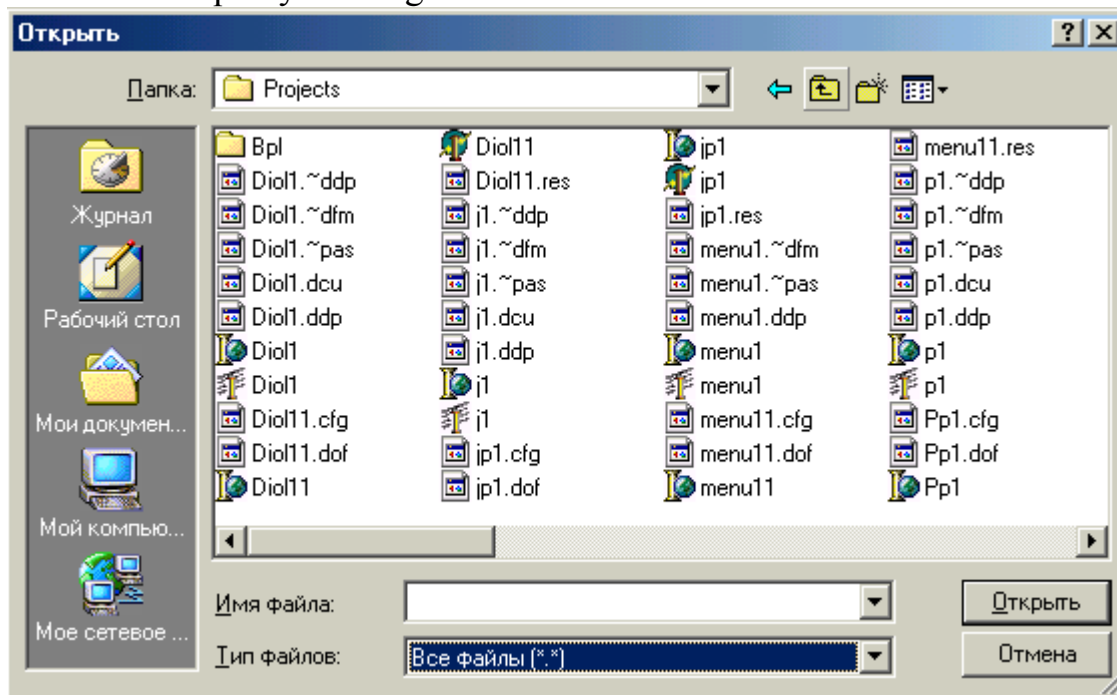
va boshqa


mos filtrlar:

*.pas

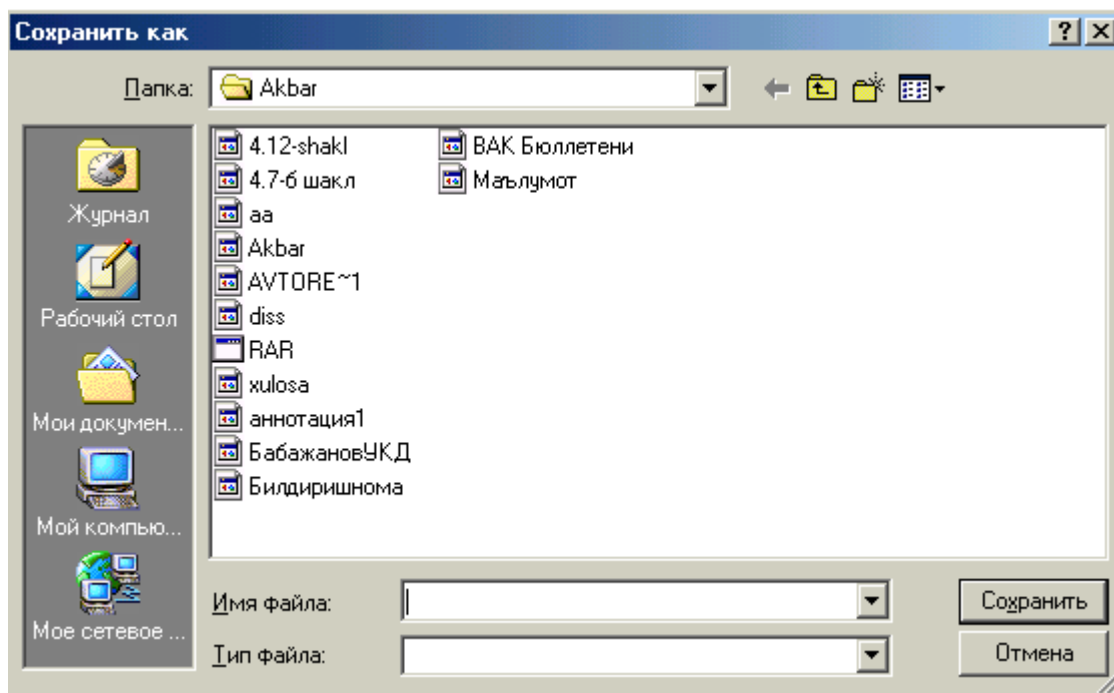
*.txt; *.doc


**
Fayl ochish muloqot oynasining ko‘rinishi.



SaveDialog komponentasi kompyuter xotirasiga fayllarni saqlash imkonini beradi. Bu komponenta piktogrammasi  ko‘rinishga ega. U vizual bo‘lmagan komponenta bo‘lib, uni formaga sichqonchada bir marta bosib qo‘yiladi va keyin uning xossalari o‘rnatiladi. Agar uning DefaultExt xossasi qiymati .txt qilib tenglashtirilsa, faylni saqlashda avtomatik ravishda uning kengaytmasi .txt qilib saqlanadi.

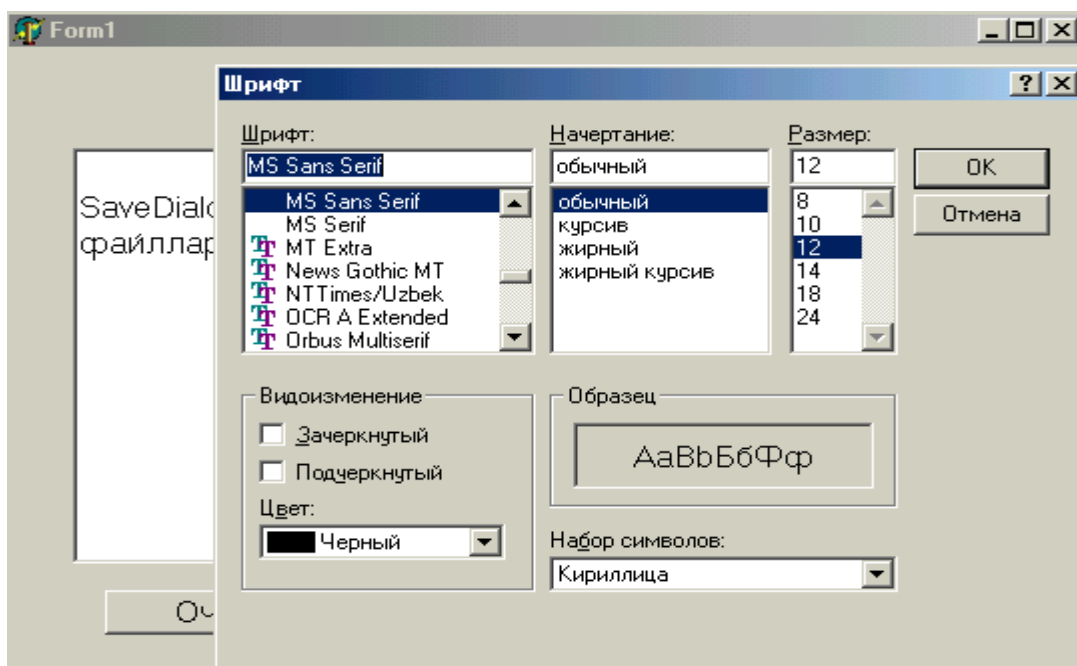
Faylni saqlash muloqot oynasining ko‘rinishi.



FontDialog komponentasi foydalanuvchiga shriftlarni tanlaydi va uning xarakteristikasini belgilaydi. Bu komponenta piktogrammasi  ko‘rinishga ega.

U vizual bo‘lmagan komponenta bo‘lib, uni formaga sichqonchada bir marta bosib qo‘yiladi va keyin uning xossalari o‘rnatiladi. Uning Font xossasi shrift xarakteristikasini beradi.

SHriftni tanlash muloqot oynasining ko‘rinishi.

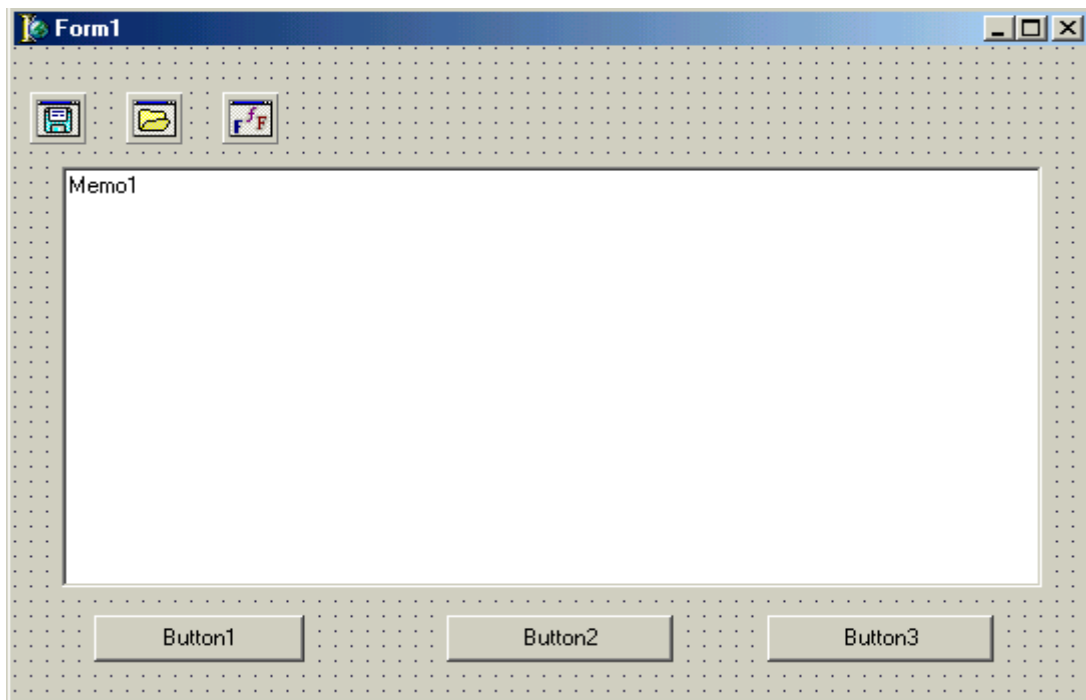


Misol

OpenDialog, **SaveDialog** va **FontDialog** komponentalarini ishlatgan holda oddiy matn muharriri yaratilsin.

Yechish

1. Yangi ilova yaratamiz.
2. Formaga matnlarni chiqarish uchun Standart komponentalar palitrasidan Memo komponentasini Memo1 nom bilan o‘rnatamiz.
3. Forma yuqorisiga OpenFileDialog, SaveDialog va FontDialog komponentalarini o‘rnatamiz. Bu komponentalarni vizual bo‘lmaganligi sabab, istalgan joyga o‘rnatsa bo‘ladi. Chunki dastur ishlashi vaqtida bu komponentalar ko‘rinmaydi.
4. Formaning pastki qismiga Standart komponentalar palitrasidan Botton komponentasini uch marta Botton1, Botton2 va Botton3 nomlar bilan o‘rnatamiz.



5. Memo1 komponentasining Lines xossasigakelib, uch nuqtali tugmachanibosamiz va muloqot oynasidan Memo1 soʻzinioʻchiramiz va Ok tugmasinibosamiz. (Bu degani ilovani ishga tushirganda muharrir oynasi boʻsh chiqadi.)

6. OpenFileDialog komponentasi xossalarini oʻrnatamiz. Buning uchun Filter xossasiga kirib, uning uch nuqtali tugmasini bosamiz. Hosil boʻlgan Filter Editor muloqot darchasiga quyidagilarni kiritamiz va Ok tugmasini bosamiz.

Filter Name qismiga

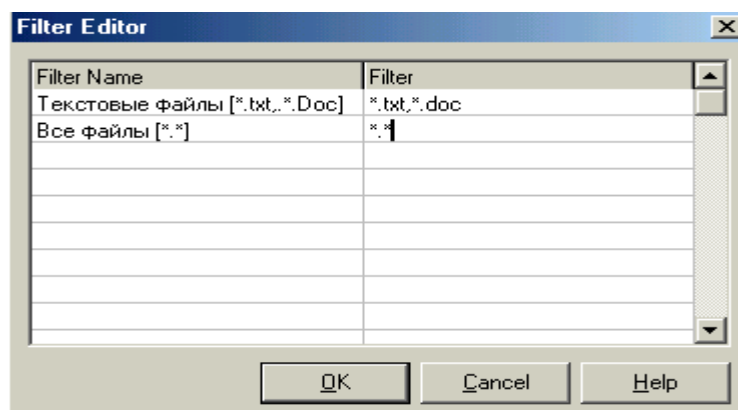
Текстовые документы (*.txt;*.doc)

Все файлы (*.*)

Filter qismiga

*.txt; *.doc

.



7. SaveDialog komponentasi xossalarini oʻrnatamiz. DefaultExt xossasi qiymatini .txt qilib tenglashtiramiz.

8. Botton1, Botton2 va Botton3 tugmachalar nomlarini ularning Caption xossasiga kirib, mos ravishda “Ochish”, “Saqlash” va “SHrift” nomlariga oʻzgartiramiz.

9. Botton1 tugmasini ikki marta tez-tez bosib, dastur kodlarini kiritish darchasiga o‘tib quyidagi operatorlarni kiritamiz.

With Opendialog1 do

Begin

If not Execute then Exit;

Memo1.Lines.LoadFromFile(Filename)

End;

10. Botton2 tugmasini ikki marta tez-tez bosib, dastur kodlarini kiritish darchasiga o‘tib quyidagi operatorlarni kiritamiz.

With Savedialog1 do

Begin

If not Execute then Exit;

Memo1.Lines.SaveToFile(Filename);

End;

11. Botton3 tugmasini ikki marta tez-tez bosib, dastur kodlarini kiritish darchasiga o‘tib quyidagi operatorlarni kiritamiz.

With Fontdialog1 do

Begin

If not Execute then Exit;

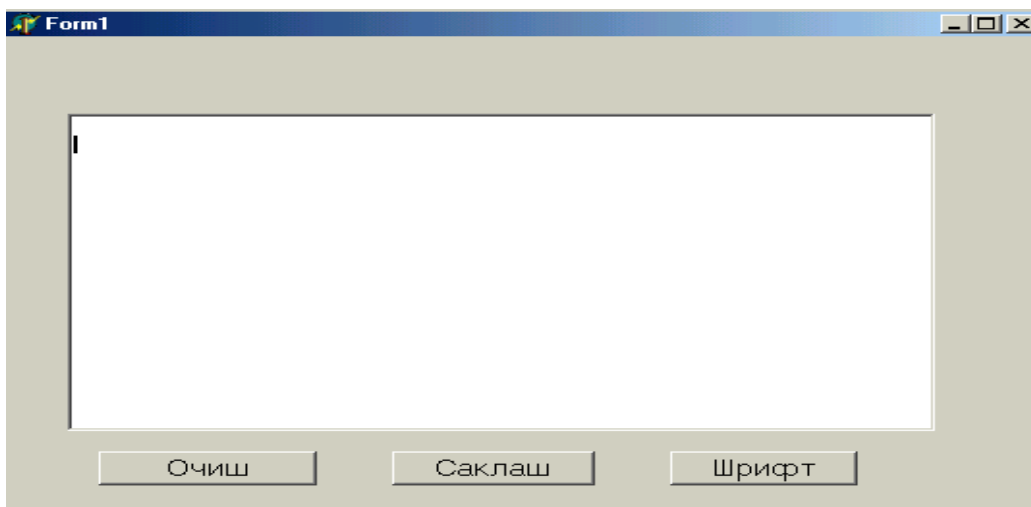
Memo1.Font:=Font;

End;

12. Tuzilgan loyiha (proekt) ya’ni Project1 va Unit1 standart modul nomlarini mos nomlar bilan almashtirib saqlaymiz.

13. Yangi nom bilan saqlangan proekt, ya’ni ilova F9 tugmachasini bosish bilan ishga tushiriladi.

Ilova ishga tushirilganda uning quyidagi ko‘rinishi ekranda namayon bo‘ladi.



Tashkil qilingan modulning to‘liq ko‘rinishini keltiramiz.

unit Diol1;
interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Menus;**

type

```
TForm1 = class(TForm)
  Button1: TButton;
  Button2: TButton;
  Button3: TButton;
  Memo1: TMemo;
  OpenFileDialog: TOpenDialog;
  SaveDialog1: TSaveDialog;
  FontDialog1: TFontDialog;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
```

private

```
{ Private declarations }
```

public

```
{ Public declarations }
```

end;

var

```
Form1: TForm1;
```

implementation

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
  With OpenFileDialog do
```

```
    Begin
```

```
      If not Execute then Exit;
```

```
      Memo1.Lines.LoadFromFile(Filename)
```

```
    End;
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
  With SaveDialog1 do
```

```
    Begin
```

```
      If not Execute then Exit;
```

```
      Memo1.Lines.SaveToFile(Filename);
```

```
    End;
```

```
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
```

```
  With FontDialog1 do
```

```
    Begin
```

```
      If not Execute then Exit;
```

```

Memo1.Font:=Font;
End;
end;
end.

```

Mavzu: C++ Bulider dasturlash tilining grafik imkoniyatlari, C++ Buliderdagi maxsus TCanvas, TFont, TPen, Tbrush klasslari.

C++ Bulider dasturchiga grafik dasturlar sxema, chertej, illyustratsiyalar yaratishga imkon beradi. Dastur grafikani ob'ekt (forma eki Image komponentasi) yuzasiga chiqaradi. Ob'ekt yuzasiga canvas xossasi moc keladi. Ob'ekt yuzasiga grafik element (to'g'ri chiziq, aylana, turtburchak va hokazo), chiqarish uchun bu ob'ektning canvas xossasiga mos usul qo'llash lozim. Misol uchun Form1.anvas. Rectangle (10,10,100,100) instruksiyasi dastur oynasida turtburchak chizadi.

Chizish sohasi

YUkorida ko'rilgan canvas xossasi -TCanvas tipidagi ob'ektdir. Grafik primitivlarni chiqarish usullari Canvas xossasini abstrakt chizish sohasi deb qaraydi. CHizish sohasi alohida nuqtalar - piksellardan iborat. Pikel holati uning gorizontal (X) va vertikal (Y) koordinatalari bilan aniqlanadi. CHap yuqori piksel koordinatalari (0,0). Koordinatalar yuqoridan pastga va chapdan o'ngga qarab o'sib boradi.

Soha o'lchovlarini image komponentasining Height i width xossalari va formaning ClientHeight va Clientwidth xossalari orqali aniqlash mumkin.

Qalam

Qalam geometrik figuralarni chizish uchun ishlatiladi. CHiziq ko'rinishi Tren ob'ektining quyidagi jadvalda ko'rsatilgan xossalari orqali aniqlanadi.

Tren (qalam) xossalari.

<i>Xossa</i>	<i>Ta'rifi</i>
<i>Color</i>	<i>CHiziq rangi</i>
<i>Width</i>	<i>CHiziq kalinligi</i>
<i>Style</i>	<i>CHiziq ko'rinishi</i>
<i>Mode</i>	<i>Akslantirish rejimi</i>

Quyidagi jadvalda color xossasi qiymati sifatida beriluvchi nomlangan konstantalar sanab o'tilgan.

Color xossasi qiymatlari.

<i>Konstanta</i>	<i>Rang</i>	<i>Konstanta</i>	<i>Rang</i>
<i>clBlack</i>	<i>Qora</i>	<i>clSilver</i>	<i>Serebristy</i>
<i>clMaroon</i>	<i>Kashtanovy</i>	<i>clRed</i>	<i>Qizil</i>
<i>clGreen</i>	<i>YAshil</i>	<i>clLime</i>	<i>Salatny</i>
<i>clOlive</i>	<i>Olivkovy</i>	<i>clBlue</i>	<i>Ko'k (zangori)</i>

<i>clNavy</i>	<i>Tim-ko'k</i>	<i>clFuchsia</i>	<i>YArko-rozoviy</i>
<i>clPurple</i>	<i>Rozoviy</i>	<i>clAqua</i>	<i>Biryuzoviy</i>
<i>clTeal</i>	<i>Zeleno-goluboy</i>	<i>clWhite</i>	<i>Oq</i>
<i>clGray</i>	<i>Kul rang</i>		

CHiziq qalinligi width xossasi orqali piksellarda beriladi.

CHiziq turini style xossasi belgilaydi. Quyidagi jadvalda chiziq turini belgilovchi nomlangan konstantalar sanab o'tilgan.

Style xossasi qiymatlari.

<i>Konstanta</i>	<i>CHiziq ko'rinishi</i>
<i>psSolid</i>	<i>Uzluksiz chiziq</i>
<i>psDash</i>	<i>Punktir chiziq, uzun shtrixlar</i>
<i>psDot</i>	<i>Punktir chiziq, qisqa shtrixlar</i>
<i>psDashDot</i>	<i>Punktir chiziq, uzun va qisqa shtrixlar ketma ketligi</i>
<i>psDashDotDot</i>	<i>Punktir chiziq, bitta uzun va ikkita qisqa shtrixlar ketma ketligi</i>
<i>psClear</i>	<i>CHiziq aks etmaydi</i>

Mode xossasi chiziq rangining fon rangiga munosabatini ko'rsatadi. Odatda chiziq rangi Pen.Color xossasi qiymati bilan belgilanadi.

Dasturchi chiziq uchun fon rangiga nisbatan invers rang berishi mumkin. Bu holda hatto chiziq va fon rangi bir xil berilgan bo'lsa ham chiziq ajralib turadi.

Quyidagi jadvalda Mode xossasi qiymati sifatida ishlatish mumkin bo'lgan konstantalar berilgan.

Mode xossasi qiymatlari

<i>Konstanta</i>	<i>CHiziq rangi</i>
<i>pmBlack</i>	<i>Qora, Pen. Color xossasi qiymatiga bogliq emas</i>
<i>pmWhite</i>	<i>Ok, Pen. Color xossasi qiymatiga bog'liq emas</i>
<i>pmCopy</i>	<i>CHiziq rangi Pen. Color xossasi qiymatiga bog'liq</i>
<i>pmNotCopy</i>	<i>CHiziq rangi Pen. Color xossasi qiymatiga invers</i>
<i>pmNot</i>	<i>CHiziq rangi sohaning mos nuqtasi rangiga invers</i>

Muyqalam

Muyqalam (Canvas.Brush) yopiq sohalarni chizish va soha ichini bo'yash uchun mo'ljallangan usullardan foydalaniladi. Muyqalam ob'ekt jadvalda ko'rsatilgan ikki xossaga ega.

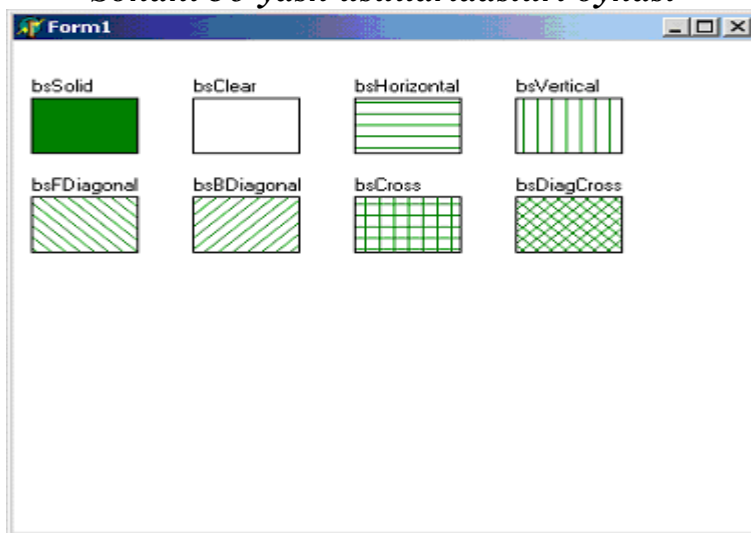
TBrush (muyqalam) xossalari.

<i>Xossa</i>	<i>Ta'rifi</i>
<i>Color</i>	<i>Epiq sohani bo'yash rangi</i>
<i>Style</i>	<i>Sohani to'ldirish uslubi</i>

**Kontur ichidagi soha bo'yalishi yoki shtrixlanishi mumkin.
Sohani to'ldirish usulini belgilovchi konstantalar quyidagi jadvalda berilgan.
Brush.style xossasi qiymatlari.**

<i>Konstanta</i>	<i>Soha bo'yash uslubi</i>
<i>bsSolid</i>	<i>Uzluksiz bo'yash</i>
<i>bsClear</i>	<i>Soha bo'yalmaydi</i>
<i>bsHorizontal</i>	<i>Gorizontal shtrixlash</i>
<i>bsVertical</i>	<i>Vertikal shtrixlash</i>
<i>bsFDiagonal</i>	<i>Diagonal shtrixlash, oldinga og'ish</i>
<i>bsBDiagonal</i>	<i>Diagonal shtrixlash, orqaga og'ish</i>
<i>bsCross</i>	<i>Katakli gorizontal-vertikal shtrixlash</i>
<i>bsDiagCross</i>	<i>Katakli diagonal shtrixlash</i>

**Misol tariqasida sohalarni bo'yash usullari dasturini keltiramiz.
Sohani bo'yash usullaridasturi oynasi**



Sohani bo'yash usullaridasturi matni

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
```

```
type
  TForm1 = class(TForm)
```

```

    Button1: TButton;
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;
implementation

{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
const
    bsName: array[1..8] of string =
('bsSolid','bsClear','bsHorizontal',
'bsVertical','bsFDiagonal','bsBDiagonal',
'bsCross','bsDiagCross');
var
    x,y: integer;
    w,h: integer;
    bs: TBrushStyle;
    k: integer;
    i,j: integer;

begin
    button1.visible:=false;
    w:=60; h:=40;
    y:=80;
    for i:=1 to 2 do
    begin
        x:=10;
        for j:=1 to 4 do
        begin
            k:=j+(i-1)*4;
            case k of
            1: bs:= bsSolid;
            2: bs:= bsClear;
            3: bs:= bsHorizontal;
            4: bs:= bsVertical;
            5: bs:= bsFDiagonal;
            6: bs:= bsBDiagonal;
            7: bs:= bsCross;
            8: bs:= bsDiagCross; end;

```

```

Canvas.Brush.Color := clGreen;
Canvas.Brush.Style := bs;
Canvas . Rectangle (x, y, x+w, y-h) ;
Canvas.Brush.Style := bsClear;
Canvas.TextOut(x, y-55, bsName[k]);
x := x+w+30;
end;
y:= y+h+30;
end;
end;end.

```

Matnni chiqarish

Grafik ob'ekt yuzasiga matn chiqarish uchun TextOut usuli qo'llaniladi. Bu usulni chaqirish instruksiyasi quyidagi ko'rinishga ega:

Ob'ekt.Canvas.TextOut(x, u, Tekst)

Matn shrifti Font xossasi qiymati bilan aniqlanadi. Font xossasi TFont tipidagi ob'ekdir. Quyidagi jadvalda TFont ob'ekti xossalari keltirilgan.

TFont ob'ekti xossalari

Xossa	Ta'rifi
Name	<i>SHrift nomi, masalan Arial</i>
Size	<i>SHrift punktlarda kattaligi</i>
Style	<i>Simvollar chiqarish uslubi. Quyidagi konstantalar orqali beriladi: fsBold (polujirnyy), fsItalic (kursiv), fsUnderline (podcherknuty), fsStrikeOut (perecherknuty).</i>

Bu xossa bir necha uslublarni kombinatsiyasini olishga imkon beradi. Masalan: Ob'ekt. Canvas . Font : = [fsBold, fs Italic]

Color *Simvollar rangi.*

Matn chiqarish sohasi muyqalam joriy rangiga buyaladi. SHuning uchun matn chiqarishdan oldin Brush. Color xossasiga bsClear qiymatini yoki soha rangiga mos qiymatni berish lozim.

Misol:

```

with Form1.Canvas do begin
Font.Name := 'Tahoma';
Font.Size := 20;
Font.Style := [fsItalic, fsBold] ;
Brush.Style := bsClear;
TextOut(0, 10, 'Borland C++ Bulider 6');
end;

```


Textout uslubi orqali matn ekranga chiqarilgandan so'ng qalam matn chiqarish sohasining yuqori o'ng burchagiga keltiriladi.

Agar matn uzunligi ma'lum bo'lmasa, chiqarilgan matn o'ng chegarasi koordinatalarini PenPos xossasiga murojaat qilib aniqlash mumkin.

Misol:

with Form1.Canvas do begin

TextOut(0, 10, 'Borland ');

TextOut(PenPos.X, PenPos.Y, 'C++ Bulider 6');

end;

Grafik primitivlarni chizish usullari

Chiziq

To'g'ri chiziq LineTo usuli orqali amalga oshiriladi.

Komponent.Canvas.LineTo(x,u)

LineTo usuli qalam joriy pozitsiyasidan berilgan koordinatali nuqttagacha to'g'ri chiziq chizadi. Boshlangich nuqtani kerakli nuqtaga ko'chirish uchun MoveTo usulidan foydalanish mumkin.

Tutashgan chiziq

O'zaro tutashgan kesmalardan iborat shaklni chizish uchun polyline usulidan foydalaniladi. Bu usul parametri TPoint tipli massivdan iborat.

Polyline usuliga misol tariqasida ma'lum qiymat o'zgarishi grafigini chizuvchi protseduracini keltiramiz:

procedure TForm1.Button1Click(Sender: TObject);

var

gr: array[1..50] of TPoint;

x0,y0: integer;

dx,dy: integer;

i: integer; **begin**

x0 := 10; u0 := 200; dx :=5; dy := 5;

for i:=1 to 50 **do begin**

gr[i].x := x0 + (i-1)*dx;

gr[i].y := y0 - Data[i]*dy;

end;

with form1.Canvas **do begin**

MoveTo(x0,y0); LineTo(x0,10);

MoveTo(x0,y0); LineTo(200,y0);

Polyline(gr);

end;

end;

Polyline usuli yordamida yopiq ko'pburchak chizish uchun massivning birinchi va oxirgi elementi bir nuqtaning koordinatalaridan iborat bo'lishi kerak.

Aylana va ellips

Aylana yoki ellips chizish uchun Ellipse usuli chaqiriladi. Usulni chaqirish instruksiyasi umumiy ko'rinishi:

Ob'ekt.Canvas.Ellipse(x1,y1, x2,u2).

Bu erda x_1, y_1, x_2, u_2 – ellipsni o‘z ichiga olgan minimal turtburchak koordinatalari. Agar turtburchak kvadrat bo‘lsa aylana chiziladi.

Yoy

Yoyni chizish uchun Arc usuli qo‘llaniladi va u quyidagi umumiy ko‘rinishga ega:

Ob’ekt.Canvas.Arc($x_1, y_1, x_2, u_2, x_3, u_3, x_4, u_4$)

Bu erda:

- *x_1, y_1, x_2, u_2 - yoyga tegishli bo‘lgan ellips yoki aylana parametrlari;*
- *x_3, u_3 - yoy boshlang‘ich nuqtasi parametrlari;*
- *x_4, u_4 - so‘ngi nuqtasi parametrlari.*

Yoy soat miliga teskari tartibda chiziladi.

To‘rtburchak

To‘rtburchak Rectangle usuli bilan chizilib, bu usulni chaqirish instruksiyasi umumiy ko‘rinishi quyidagicha:

Ob’ekt.Canvas.Rectangle(x_1, y_1, x_2, y_2)

Bu erda x_1, y_1 va x_2, u_2 — chapgi yuqori va o‘nggi pastgi burchaklar koordinatalari.

RoundRec usuli burchaklari yumaloq to‘rtburchak chizishga imkon beradi.

RoundRec usulini chakirish instruksiyasi quyidagi kurinishga ega:

Ob’ekt.Canvas.RoundRec($x_1, y_1, x_2, u_2, x_3, u_3$)

Bu yerda:

- *x_1, y_1, x_2, u_2 – turtburchak parametrlari;*
- *x_3 i u_3 — chorak kismi yumalok burchak chizish uchun ishlatiladigan ellips kattaligi.*

YA’na ikki usul muyqalamdan foydalanib to‘rtburchak chizishga imkon beradi.

FillRect usuli ichi bo‘yalgan to‘rtburchak chizadi, FrameRect - faqat kontur.

Bu usullarda faqat bitta parametrga ega -TRect tipidagi struktura. Quyidagi misolda FillRect va FrameRect usullari orqali forma yuzasiga qizil turtburchak soha va yashil konturli to‘rtburchak chizuvchi protsedura keltirilgan.

procedure TForm1.Button1Click(Sender: TObject);

var

r1, r2: TRect;

begin

r1 := Rect(20,20,60,40);

r2 := Rect(10,10,40,50);

with form1.Canvas do begin

Brush.Color := clRed;

FillRect(r1);

Brush.Color := clGreen;

FrameRect(r2};

end;

end;

Ko'pburchak

Polygon usuli ko'pburchak chizishga mo'ljallangan bo'lib, parametri TPoint tipidagi massivdir. Massivning har bir elementi (x,u) maydonlari ko'pburchak uchi koordinatalaridan iborat bo'lgan yozuvdir.

Quyida polygon usuli yordamida uchburchak chizish protsedurasi keltirilgan:

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
pol: array[1..3] of TPoint;  
begin  
pol[1].x := 10;  
pol[1].y := 50;  
pol[2].x := 40;  
pol[2].y := 10;  
pol[3].x := 70;  
pol[3].u := 50;  
Form1.Canvas.Polygon(pol);  
end;
```

Sektor

Ellips yoki aylana sektori pie usuli bilan chizilib, chaqirish instruksiyasi quyidagi umumiy ko'rinishga ega:

Ob'ekt. Canvas.Pie(x1,y1,x2,y2,x3,u3,x4,u4)

Bu erda:

- *x1, y1, x2, u2 - ellips yoki aylana parametrlari;*
- *x3, u3, x4, u4 - sektor chegarasini tashkil qiluvchi to'g'ri chiziqlar oxirgi nuktalari koordinatalari.*

Nuqta

Canvas ob'ektining pixels xossasi tipidagi ikki o'lchovli massiv bo'lib har bir soha nuqtasining rangi haqidagi ma'lumotni o'z ichiga oladi. Pixels xossasidan foydalanib ixtiyoriy nuqta rangini o'zgartirish, ya'ni nuqta chizish mumkin. Misol uchun

```
Form1.Canvas.Pixels[10,10]:=clRed
```

Instruksiyasi soha nuqtasini qizil ranga bo'yaydi.

Quyida keltirilgan dastur pixels xossasidan foydalanib, $u = 2 \sin(jc) e^{x/5}$ funksiyasi grafigini chiqaradi.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Button1: TButton;
```

```
procedure Button1Click(Sender: TObject);
```

```

private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
  Function f(x:real):real;
begin
f:=2*Sin(x)*exp(x/5) ;
end;
procedure GrOfFunc;
var
x1,x2:real;
y1,y2:real;
x:real;
y:real;
dx:real;
l,b:integer;
w,h:integer;
mx,my:real;
x0,y0:integer;
begin
l:=10;
b:=Form1.ClientHeight-20;
h:=Form1.ClientHeight-40;
w:=Form1.Width-40;
x1:=0;
x2:=25;
dx:=0.01;
y1:=f(x1);
y2:=f(x1);
x:=x1;
repeat
y := f (x);
if y < y1 then y1:=y;
if y > y2 then y2:=y;
x:=x+dx; until (x >= x2);
my:=h/abs(y2-y1);
mx:=w/abs(x2-x1);
x0:=1;
y0:=b-Abs(Round(y1*my)) ;
with form1.Canvas do

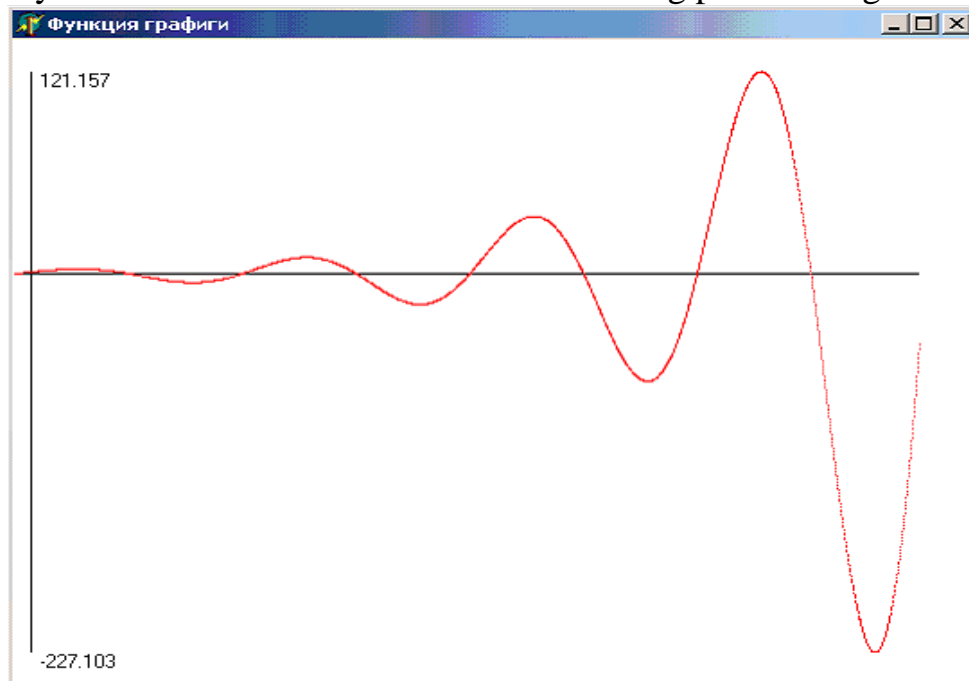
```

```

begin
// osi
MoveTo(l,b);LineTo(l,b-h);
MoveTo(x0,y0);LineTo(x0+w,y0);
TextOut(l+5,b-h,FloatToStrF(y2,ffGeneral,6,3));
TextOut(l+5,b,FloatToStrF(y1,ffGeneral,6,3));
x:=x1; repeat
y:=f(x);
Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
x:=x+dx;
until (x >= x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
Button1.Visible:=false;
GrOfFunc;
end;end.

```

Asosiy vazifani GrOfFunc protsedurasi bajaradi. Avval $[x_1, x_2]$ oraliqda funktsiyaning maksimal (y_2) va minimal (y_1) qiymatlari hisoblanadi. Soʻngra koordinatalar u ki boʻyicha masshtab hisoblanadi. SHundan soʻng protsedura grafikni quradi.



GrOfFunc protsedurasi tomonidan qurilgan grafik.

Keltirilgan dastur universal xarakterga ega. Oʻzga funktsiya grafigini chizish uchun $f(x)$, tanasini oʻzgartirish etarli.

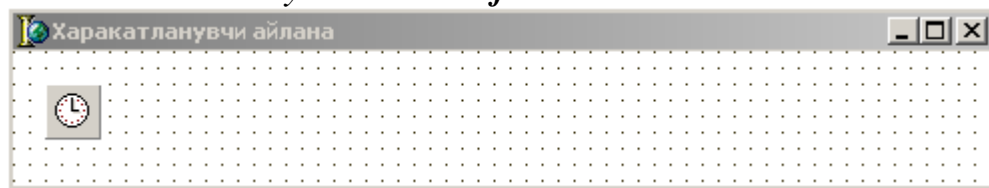
Dastur toʻgʻri ishlaydi, agar funktsiya ham musbat, ham manfiy qiymatlarni qabul qilsa.

Multiplikatsiya

Multiplikatsiya deyilganda harakatlanuvchi rasm tushuniladi. Rasmni harakatlantirish uchun avval u ekranga chiziladi, ma'lum vaqtdan so'ng rasmni uchirib yangi joyga chiziladi.

Quyidagi dastur, aylananing chapdan o'ngga harakatini ko'rsatadi.

Harakatlanuvchi aylana dasturi formasi



Harakatlanuvchi aylana dasturi matni

unit Unit1;

interface

uses

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls;*

type

TForm1 = class(TForm)

Timer1: TTimer;

procedure FormActivate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

procedure Ris;

var

Form1: TForm1;

x,y: byte;

implementation

*{\$R *.dfm}*

procedure Ris;

begin

form1.Canvas.Pen.Color:=form1.Color;

form1.Canvas.Ellipse(x,y,x+10,y+10);

x:=x+5;

form1.Canvas.Pen.Color:=clBlack;

form1.Canvas.Ellipse(x,y, x+10, y+10) ;

end;

procedure TForm1.FormActivate(Sender: TObject);

begin

x:=0;


```

y:=10;
timer1.Interval:=50;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
Ris;
end;
end.

```

Grafik komponentalar


Image komponentasi formaga rasmlarni joylashtirish uchun ishlatiladi. Joylashtirilishi lozim bo'lgan rasmlar bitli fayllar (kengaytmalari .bmp), piktogramma-li (kengaytmalari .ico), meta fayllar (kengaytmalari .wmf) bo'lishi kerak.

Image komponentasi Additional palitrasida joylashgan bo'lib, u  ko'rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan rasm uchun joy ajratiladi va keyin esa xossalar bo'limidan Picture xossasi tanlanib, u erdan uch nuqtali tugmacha bosiladi. Natijada ekranda rasmni aniqlash va joylash uchun mu-loqat darchasi ochiladi. Muloqat darchasi quyidagi tugmachalarga ega:


- Load -fayldan rasmni chaqirish;
- Save -rasmni faylga saqlash;
- Clear -tanlangan rasmni olib tashlash;
- Ok -tanlangan rasmni ajratilgan joyga yozish;
- Cancel -qilingan o'zgartirishlarni bekor qilish.

Shape komponentasi formaga aylana, to'rtburchak, ellips va boshqa shakllarni joylashtirish uchun ishlatiladi. Uning quyidagi xossalari mavjud:


- Brush -shaklni bo'yash uchun cho'tkacha;
- Pen -shakl chetini chizish uchun qalam;
- Shape -ekranga chiqadigan shaklni aniqlaydi:
 - StRectangle -to'rtburchak;
 - StSquare -kvadrat;
 - StRoundRect -chetlari aylana to'rtburchak;
 - StRoundSquare -chetlari aylana kvadrat;
 - StEllipse -ellips;
 - StCircle -aylana.

Shape komponentasi ham Additional palitrasida joylashgan bo'lib, u  ko'rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan shakl uchun joy ajratiladi va keyin esa xossalar bo'limidan Shape xossasiga kirilib kerakli shakl tanlanadi.

PaintBox komponentasi formaga chegaralangan maydonda shakllarni chizish imkonini beradi.

PaintBox komponentasi System palitrasida joylashgan bo'lib, u  ko'rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan shakl uchun joy ajratiladi va keyin esa xossalar bo'limidan Shape xossasiga kirilib kerakli shakl tanlanadi.

Timer vizual bo‘lmagan komponenta bo‘lib, formada bajariladigan ma’lum bir operatsiyalarni vaqt bo‘yicha boshqaradi.

Timer komponentasi System palitrasida joylashgan bo‘lib, u  ko‘rinishdagi piktogrammaga ega. Bu tugmachani bosib formaga olib kelib qo‘yiladi.

U quyidagi xossalarga ega:

Enabled -true qiymati o‘rnatilgan bo‘lsa u bo‘ladigan jarayonga ta’sir qiladi;

Interval -millisekundlarda vaqt intervalini aniqlaydi va jarayonning ekranga chiqishiga ta’sir ko‘rsatadi. Tegmagan holda 1000 (1 sekund) ko‘rsatadi.

M i s o l.

Ilova uchun “zastavka” yaratish.

Ye ch i sh

Zastavka grafik tasvirlar ko‘rinishida bo‘lib, programmalar ishga tushirilganda bir necha sekunddan so‘ng ekranda paydo bo‘ladi. Unda programma nomi va uning avtorlari haqida ma’lumot bo‘lishi mumkin.

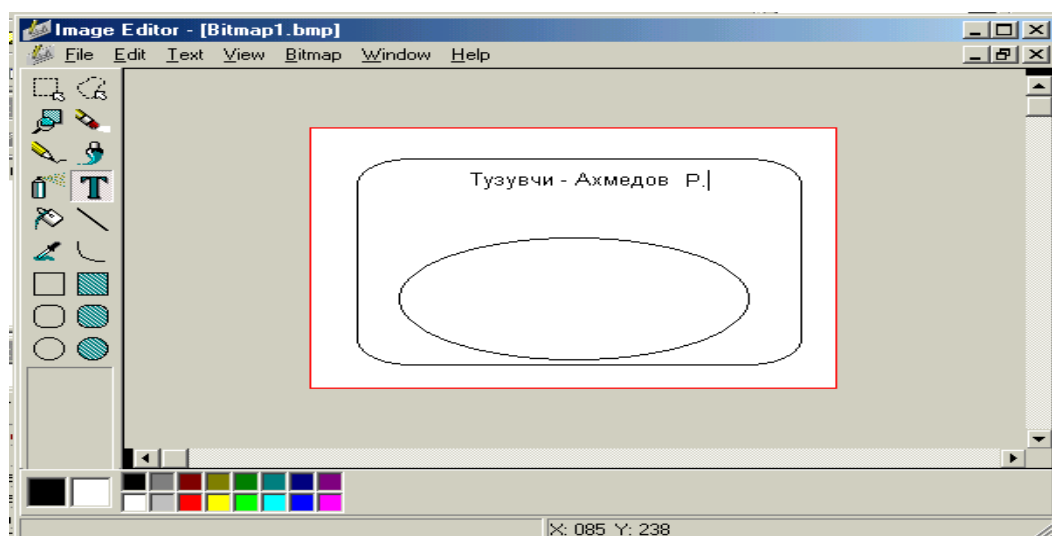
Grafik tasvirni, ya’ni .bmp kengaytmaga ega bo‘lgan faylni grafik muharriri yordamida tayyorlaymiz. C++ Bulider sistemasini ishga tushirishdan avval tuziladigan ilovani saqlash uchun o‘zimizga papka tashkil qilamiz.

1.Oldin tuzilgan biror bir ilovani ochamiz yoki yangi ilova tashkil etamiz.

2.Bosh menyudan grafik muharrirni ishga tushiramiz: Tols=>Image Editor. (Bu C++ Bulider grafik muxariri oddiy Paint grafik muharriridan uncha katta farq qilmaydi)

3.C++ Bulider grafik muxariri Image Editor menyusidan File=>New=>Bitmap File(.bmp) buyrug‘i beriladi. Natijada ekranda rasm parametrlarini berish uchun muloqat darchasi paydo bo‘ladi. Muloqat darchasidan kerakli parametrlar tanlanib Ok tugmasi bosiladi. Tayyor mavjud rasm foyllaridan ham foydalanish mumkin.

4.Grafik muharriri oynasidan ajratilgan joyga ixtiyoriy rasm chizilib, u saqlanadi. Masalan, aylana va unga tashqi chizilgan rasm chizib, ichiga “Tuzuchi – Axmedov P.” so‘zi yozib qo‘yilsin. Matnni yozish uchun uskunalar panelining “T” (Text) tugmachasidan foydalaniladi.



5.Grafik fayli saqlanadi va undan chiqiladi.

6.System palitrasidan Timer komponentasining tugmachani bosib formaga olib kelib qo'yiladi va u Timer1 nom oladi. Interval xossasini 3000 ga tenglashirib olamiz.

7.Additional palitrasidan Image komponentasi tugmachasini bosib formadan rasm uchun joy ajratiladi va keyin esa xossalar bo'limidan Picture xossasi tanlanib, u erdan uch nuqtali tugmacha bosiladi. Natijada ekranda rasmni aniqlash va joylash uchun muloqat darchasi ochiladi. Muloqat darchasidan Load buyrug'i berilib, saqlangan rasm faylimiz tanlanadi va Ok tugmasi bosiladi. Rasm to'liq formaga joylashishi uchun Autosize xossasiga True qiymatini o'rnatamiz.

8.Timer1 komponentini aktivlashtiramiz, ya'ni uni ikki marta tez-tez bosamiz va kodlarni yozish oynasiga quyidagi qora yozilgan kodlarni kiritamiz.

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    Image1.Free;
    Timer1.Free;
end;
```

Bu shuni bildiradiki programma ishga tushgandan so'ng 3000 milisekunddan o'tishi bilan Image1 va Timer1 komponentalari kompyuter xotirasidan va mos ravishda ekrandan o'chiriladi.

9.Tuzilgan loyiha (proekt) ya'ni Project1 va Unit1 standart modul nomlarini mos nomlar bilan almashtirib saqlanadi.

10.YAngi nom bilan saqlangan proekt, ya'ni ilova F9 tugmachasini bosish bilan ishga tushiriladi.

Ilova ishga tushirilganda ekranda yuqoridagi 4 punktdagi rasm "zastavka" ko'rinishida namoyon bo'ladi.

Tashkil qilingan modulning to'liq ko'rinishini keltiramiz.

```
unit px1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
type
TForm1 = class(TForm)
    procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Timer1Timer(Sender: TObject);
begin
```

```
Image1.Free;  
Timer1.Free;  
End;  
end.
```

Yaratilgan “zastavka”miz programma ishlash davomida o‘chib-yonib turishi uchun quyidagi OnTimer xodisasini qayta ishlash kodini yozishimiz kerak bo‘ladi.

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    If Image1.visible=true then Image1.hide  
        Else Image1.Show;  
End;
```

M i s o l 3.

Oyning Er atrofida aylanishini namoyish etuvchi ilova yaratish.

Ye ch i sh

1. Yangi ilova yaratamiz.

2. Formaga Timer komponentasini Timer1 nom bilan joylashtiramiz. Uning Interval xossasini 55 qilib o‘rnatamiz. Jarayon ya’ni xodisa 55 millisekundda paydo bo‘ladi (uyg‘onadi).

3. Additional palitrasidan Shape komponentasini Shape1 nom bilan formaga joylashtiramiz va uning quyidagi xossalarini o‘rnatamiz.

```
Shape -StCircle;  
Height -121;  
Width -121;  
Left -240;  
Top -104.
```

Brush xossasini tanlab ikki marta sichqonchada bosamiz, natijada ikkita yana qo‘shimcha xossalar paydo bo‘ladi: Color va Style. Color xossasini tanlab unga slBlue qiymatni o‘rnatamiz.

4. Formaga ikkinchi Shape komponentasini Shape2 nom bilan joylashtiramiz va uning quyidagi xossalarini o‘rnatamiz.

```
Shape -StCircle;  
Height -41;  
Width -41;  
Left -400;  
Top -152.
```

Brush xossasiga clYellow rangini o‘rnatamiz.

5. Formaning yuqori qismiga Label komponentasini Label1 nom bilan joylashtiramiz va uning Caption xossasini “Oyning Er atrofida aylanishi” qiymatiga o‘zgartiramiz. Font xossasiga kirib kerakli shriftni va uning o‘lchamini aniqlaymiz (agar kerak bo‘lsa). Masalan,

```
SHrift -Courier New,  
Nachertanie -polujirnyy,  
Razmer -16,  
Nabor simvolov -krilitsa.
```

Transparent xossasi qiymatini True qilib oʻrnatamiz.

6.Timer1 komponentini aktivlashtiramiz, yaʼni uni ikki marta tez-tez bosamiz va kodlarni yozish oynasiga quyidagi qora yozilgan kodlarni kiritamiz.

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    x:=x+0.1;
    Shape2.Left:=265+Trunc(50*Cos(x));
    Shape2.top:=150-Trunc(50*Sin(x));
end;
```

7.Tuzilgan loyiha (proekt) yaʼni Project1 va Unit1 standart modul nomlarini mos nomlar bilan almashtirib saqlanadi.

8.Yangi nom bilan saqlangan proekt, yaʼni ilova F9 tugmachasini bosish bilan ishga tushiriladi.

Ilova ishga tushirilganda ekranda Oyning Er atrofida aylanishi namoyish qilinadi.

Tashkil qilingan modulning toʻliq koʻrinishini keltiramiz.

```
unit px2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dia-
logs, StdCtrls, ExtCtrls;
type
TForm1 = class(TForm)
    Timer1: TTimer;
    Shape1: TShape;
    Shape2: TShape;
    Label1:Tlabel;
    procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
    X: Real;
implementation
{$R *.dfm}
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    x:=x+0.1;
    Shape2.Left:=265+Trunc(50*Cos(x));
    Shape2.top:=150-Trunc(50*Sin(x));
End;
Initialization
```

$x:=0$
end.

Programmada o'zgaruvchi x global holda e'lon qilingan. SHu tufayli uning boshlang'ich qiymati Initialization seksiyasida berilgan.

C++ dasturlash tili

Mavzu: Programmash asoslari haqida umumiy ma'lumotlar. C++ dasturlash tili va muhiti. C++ dasturlash tili to'g'risida umumiy ma'lumotlar. C++ tilining alfaviti va asosiy elementlari.

Reja:

1. Programmash asoslari va ular to'g'risida umumiy ma'lumotlar.
2. C++ tili va uning leksik asoslari, dastur tuzilishi va uning kompilyatsiyasi.
3. C++ tili alfaviti, leksimalar, identifikatorlar va kalit so'zlar.

Ma'ruza bayoni

Biz axborot, kod, kompyuter, blok- sxema, algoritm, ijrochi va boshqa tushunchalarni o'rgandik. Bo'lar masalani kompyuterda hal qilishda kerak bo'ladigan tushunchalarni o'rgandik. Bo'lar masalani kompyuterda hal qilishda kerak bo'ladigan tushunchalardir. Ammo kompyuter bilan muloqot qilish uchun, bo'lardan tashqari, kompyuter "tilini" ham bilishimiz talab qilinadi. Biror masalani kompyuterda hal qilish uchun, avval, uning algoritmini tuzishimiz, so'ng bu algoritmi kompyuter tushunadigan ko'rsatmalar va qonun qoidalar asosida yozishimiz lozim bo'ladi. Bu yozuv dastur deb ataladi.

Shunday qilib, dastur biror masalani yechishda kompyuter bajarishi mumkin bo'lgan ko'rsatmalarning izchil tartibidan iborat ekan. Kompyuter uchun dastur tuzish jarayoni dasturlash va dastur tuzadigan kishi dasturchi deb ataladi.

Kompyuter tushunadigan "til" dasturlash tili deb ataladi. (Eslang: algoritm ijrochi-sining ko'rsatmalar tizimi. Bu yerda kompyuter ijrochi sifatida qatnashmoqda.)

Hozircha yaratilgan Programmash asoslarini, ulardagi ko'rsatmalar, qonun qoidalarga ko'ra uch guruhga ajratish mumkin:

1. *Quyida darajadagi.*
2. *O'rta darajadagi.*
3. *Yuqori darajadagi.*

Quyida darajadagi Programmash asoslari da ko'rsatmalar raqamlar yordamida beriladi. Barcha amallar ma'lum raqamlar bilan ifodalangan bo'lib, ular mashina qurilmalari adreslar bilan bevosita ishlashga mo'ljallangan. Bu tilda dastur tuzuvchi amallarning sonli ifodasi va adreslarni puxta bilishi kerak.

O'rtadarajadagidasturlashtillaridako'rsatmalarsifatidainsontiligayaqinbo'lgan, ammoqisqartirilganholdaishlatiladiganso'zlartanlanadi. Bunda ham dasturchi adreslar bilan ishlashi lozim. Qisqartirilgan ko'rsatmalar mnemokodlar, bu turdagi Programmash asoslari esa assemblerlar deb ataladi.

Yuqoridarajadagidasturlashtillaridagi-ko'rsatmalarinsontiligayaqinbo'lganso'zlardaniborat. Ularda amallarni bajarish avvalgilardan oson bo'lib, dasturchi, biror maxsus amal bo'lmasa, adreslar, qurilmalar bilan bevosita bog'liq ko'rsatmalarni bilmasligi ham mumkin.

Kompyuteramallarnibajarishuchunularniraqam-liko'rinishgakeltirishkerakbo'ladi. Bu vazifani translyatorlar deb nomlanadigan maxsus dasturlar bajaradi. Translyatorlar dasturlashtilida yozilgan dasturlarni mashina tiliga tarjima qiladi. Ular 2 xil bo'ladi: interpretator va kompilyator.

Keyingiyillardaju-dako'pyuqoridarajadagidasturlashtillariyaratilganligisizlargama'lum. UlarqatorigaPaskal, Simula, dBasse, KARAT, Sivaboshqatillarnikiritishmumkin. Dasturlashtillariningko'payibborayotganligigaqaramay, harbirdasturlashtiliningo'zmaqsadi, vazifasivaqo'llanishsohalaribor. Masalan, hisoblashgabog'liqmasalalarniyechishuchunPaskal, Fortran, belgili-ma'lumotvaro'yxatlarniqaytaishlashuchunLISP, matnlima'lumotlarbilanishlashuchunSI, muloqotuchunBeysikPaskal, iqtisodiy-masalalarniyechishuchundBasse, Kabolva-ro'yxatlarniqaytaishlashuchundasturlashtillaridanfoydalaniladi.

C++ dasturlash tili C tiliga asoslangan. C esa o'z navbatida B va BCPL tillaridan kelib chiqqan. BCPL 1967 yilda Martin Richards tomonidan tuzilgan va operatsion sistemalarni yozish uchun mo'ljallangan edi. Ken Thompson o'zining B tilida BCPL ning ko'p hossalarini kiritgan va B da UNIX operatsion sistemasining birinchi versiyalarini yozgan. BCPL ham, B ham tipsiz til bo'lgan. Yani o'zgaruvchilarning ma'lum bir tipi bo'lmagan - har bir o'zgaruvchi kompyuter hotirasida faqat bir bayt yer egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun sonmi, kasrli sonmi yoki harfdekmi, dasturchi vazifasi bo'lgan.

C tilini Dennis Ritchie B dan keltirib chiqardi va uni 1972 yili ilk bor Bell Laboratoriyasida, DEC PDP-11 kompyuterida qo'lladi. C o'zidan oldingi B va BCPL tillarining juda ko'p muhim tomonlarini o'z ichiga olish bilan bir qatorda o'zgaruvchilarni tiplashtirdi va bir qator boshqa yangiliklarni kiritdi. Boshlanishda C asosan UNIX sistemalarida keng tarqaldi. Hozirda operatsion sistemalarning asosiy qismi C/C++ da yozilmoqda. C mashina arhitekturasiga bog'langan tildir. Lekin yahshi rejalashtirish orqali dasturlarni turli kompyuter platformalarida ishlaydigan qilsa bo'ladi.

1983 yilda, C tili keng tarqalganligi sababli, uni standartlash harakati boshlandi. Buning uchun Amerika Milliy Standartlar Komiteti (ANSI) qoshida X3J11 texnik komitet tuzildi. Va 1989 yilda ushbu standart qabul qilindi. Standartni dunyo bo'yicha keng tarqatish maqsadida 1990 yilda ANSI va Dunyo Standartlar Tashkiloti (ISO) hamkorlikda C ning ANSI/ISO 9899:1990 standartini qabul qilishdi. Shu sababli C da yozilgan dasturlar kam miqdordagi o'zgarishlar yoki umuman o'zgarishlarsiz juda ko'p kompyuter platformalarida ishlaydi.

C++ 1980 yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan tarzda tuzildi. C++ juda ko'p qo'shimchalarni o'z ichiga olgan, lekin eng asosiysi u ob'ektlar bilan dasturlashga imkon beradi.

Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoda. Buni ta'minlash uchun ob'ektlilik dasturlash g'oyasi ilgari surildi. Huddi 70-chi yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi ob'ektlilik orqali tuzish dasturlash sohasida inqilob qildi.

C++ dan tashqari boshqa ko'p ob'ektli dasturlashga yo'naltirilgan tillar paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xerox ning Palo Altoda joylashgan ilmiy-qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalk da hamma narsa ob'ektlarga asoslangan. C++ esa gibril tildir. Unda C ga o'hshab strukturali dasturlash yoki yangicha, ob'ektlar bilan dasturlash mumkin. Yangicha deyishimiz ham nisbiydir. Ob'ektli dasturlash falsafasi paydo bo'lganiga ham yigirma yildan oshayapti.

C++ funksiya va ob'ektlarning juda boy kutubhonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubhonasidagi tayyor ob'ekt/funksiyalarni qo'llashni o'rganishdir.

Alfavit, identifikator, xizmatchi so'zlar.

Alfavit. C++ alfavitiga quyidagi belgilar kiradi.

- Katta va kichik lotin alfaviti xarflari (A,B,..,Z,a,b,...,z)
- Raqamlar: 0,1,2,3,4,5,6,7,8,9
- Maxsus belgilar: “ , { } | [] () + - / % \ ; ‘ . : ? < = > _ ! & * # ~ ^
- Ko'rinmaydigan belgilar (“umumlashgan bushliq belgilari”). Leksemalarni uzaro ajratish uchun ishlatiladigan belgilar (misol uchun bushlik, tabulyatsiya, yangi qatorga o'tish belgilari).

Izohlarda, satrlarda va belgili konstantalarda boshqa literalalar, masalan rus xarflarini ishlatilishi mumkin.

C++ tilida olti xil turdagi leksemalar ishlatiladi: ehtkin tanlanadigan va ishlatiladigan identifikatorlar, xizmatchi so'zlar, konstantalar(konstanta satrlar), amallar(amallar belgilari), azhratuvchi belgilar.

Identifikator. Identifikatorlar lotin xarflari, ostki chiziq belgisi va sonlar ketma ketligidan iborat bo'ladi. Identifikator lotin xarfidan yoki ostki chiziq belgisidan boshlanishi lozim. **Misol uchun:** *A1, _MAX, adress_01, RIM, rim*

Katta va kichik xarflar farklanadi, shuning uchun ohirgi ikki identifikator bir biridan farq qiladi.

Borland kompilyatorlaridan foydalanilganda nomning birinchi 32 xarfi, ba'zi kompilyatorlarda 8 ta xarfi inobatga olinadi. Bu holda NUMBER_OF_TEST va NUMBER_OF_ROOM identifikatorlari bir biridan farq qilmaydi.

Xizmatchi so'zlar.

Tilda ishlatiluvchi-ya'ni dasturchi tomonidan uzgaruvchilarning nomlarini sifatida ishlatish mumkin bulmagan identifikatorlar xizmatchi so'zlar deyiladi. C++ tilida quyidagi xizmatchi so'zlar mavjud:

*int extern else char register for float type def do double static while struct
goto switch union return case long sizeof default short
break entry unsigned continue auto if*

Sinov savollari:

1. C++ tizimining umumiy tuzilishi qanday?
2. Menuning tarkibiy qismlari qaysi?
3. C++ tilining tarkibiy qismlari qanday?

4. Identifikatorni tushuntiring?

Uyga vazifa : C++ tizimida ishlash qoidalarini mukammal o'rganish va ularni amaloyotga tatbiq etish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu: C++ tilida ma'lumotlar va ularning turlari. O'zgaruvchi va o'zgarmaslarning turlari.

Reja:

1. C++ da berilganlar va ularning turlari .
2. O'zgaruvchilar va o'zgarmaslar.
3. Ma'lumotlarning turlari.

Ma'ruza bayoni

O'zgaruvchilar. (VARIABLES)

O'zgaruvchilar ob'ekt sifatida. C++ tilining asosiy tushunchalaridan biri nomlangan hotira qismi – ob'ekt tushunchasidir. Ob'ektning xususiy holi bu o'zgaruvchidir. O'zgaruvchiga qiymat berilganda unga ajratilgan hotira qismiga shu qiymat kodi yoziladi. O'zgaruvchi qiymatiga nomi orqali murojaat qilish mumkin, hotira qismiga esa faqat adresi orqali murojaat qilinadi. O'zgaruvchi nomi bu erkin kiritiladigan identifikatordor. O'zgaruvchi nomi sifatida xizmatchi so'zlarni ishlatish mumkin emas.

O'zgaruvchilar tiplari. O'zgaruvchilarning qo'yidagi tiplari mavjuddir:

char – bitta belgi;

long char – uzun belgi;

int – butun son;

short yoki **short int** – qisqa butun son;

long yoki **long int** – uzun butun son;

float - haqiqiy son;

long float yoki **double** – ikkilangan haqiqiy son;

long double – uzun ikkilangan haqiqiy son;

Butun sonlar ta'riflanganda ko'rilgan tiplar oldiga unsigned (ishorasiz) ta'rifi kushilishi mumkin. Bu ta'rif qushilgan butun sonlar ustida amallar mod 2n arifmetikasiga asoslangandir . Bu erda n soni int tipi hotirada egallovchi razryadlar sonidir. Agar ishorasiz k soni uzunligi int soni razryadlar sonidan uzun bulsa, bu son qiymati k mod 2n ga teng bo'ladi. Ishorasiz k son uchun ga -k amali 2n - k formula asosida hisoblanadi. Ishorali ya'ni signed tipidagi sonlarning eng katta razryadi son ishorasini ko'rsatish uchun ishlatilsa unsigned (ishorasiz) tipdagi sonlarda bu razryad sonni tasvirlash uchun ishlatiladi.

O'zgaruvchilarni dasturning ixtiyoriy qismida ta'riflash yoki qayta ta'riflash mumkin. Misol uchun:

Int a, b1, ac; eki

Int a;

int b1;

int ac;

O'zgaruvchilar ta'riflanganda ularning qiymatlari aniqlanmagan bo'ladi. Lekin

o'zgaruvchilarni ta'riflashda initsializatsiya ya'ni boshlang'ich qiymatlarini ko'rsatish mumkin. Misol uchun:

Int I=0;

Char c='k';

Typedef ta'riflovchisi yangi tiplarni kiritishga imkon beradi.

Misol uchun yangi COD tipini kiritish:

Typedef unsigned char COD;

COD simbol;

Konstantalar. (constants)

Konstanta bu o'zgartirish mumkin bulmagan qiymatdir. C++ tilida besh turdagi konstantalar ishlatilishi mumkin: butun sonlar, haqiqiy sonlar, belgilar, sanovchi konstantalar va nul ko'rsatkich.

1. Ma'lumotlarning butun son turi.

Butunsonlar o'nlik, sakkizlik yoki unoltilik sanoq sistemalarida berilishi mumkin. O'nlik sanoq sistemasida butun sonlar 0-9 raqamlari ketma ketligidan iborat bo'lib, birinchi raqami 0 bulishi kerak emas. Sakkizlik sanoq sistemasida butun sonlar 0 bilan boshlanuvchi 0-7 raqamlaridan iborat ketma ketlikdir. O'n oltilik sanoq sistemasida butun son 0x eki 0X bilan boshlanuvchi 0-9 raqamlari va a-f yoki A-F xarflaridan iborat ketma ketlikdir.

Masalan 15 va 22 o'nlik sonlari sakkizlikda 017 va 026, un oltilikda 0xF va 0x16 shaklda tasvirlanadi.

Ma'lumolarning uzun butun son turi.

Oxiriga l eki L harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik butun son.

Ma'lumotlarning ishorasiz (unsigned) butun son turi:

Ohiriga u yoki U harflari quyilgan o'nlik, sakkizlik yoki o'n oltilik oddiy yoki uzun butun son.

2. Ma'lumotlarning haqiqiy son turi:

Olti qismdan iborat bulishi mumkin: butun qism, nuqta, kasr qism, yoki E belgisi, o'nlik daraja, F eki f suffikslari.

Masalan: **66. .0 .12 3.14F 1.12e-12**

Ma'lumolarning uzun haqiqiy son turi:

Ohiriga L eki l suffikslari quyilgan haqiqiy son.

Masalan: **2E+6L;**

3. Belgili konstanta.

Bittalik qavslarga olingan bitta yoki ikkita belgi. Misol uchun 'x', '*', '\012', '\0', '\n' - bitta belgili konstanta; 'dd', '\n\t', '\x07\x07' ikki belgili konstantalar.

'\`' belgiidan boshlangan belgilar eskeyp belgilar deyiladi. Belgili konstanta qiymati belgining kompyuterda qabul qilingan sonli kodiga tengdir.

ESC (eskeyp) belgilar jadvali:

Yozilishi	Ichki kodi	Belgii (nomi)	Ma'nosi
\a	0x07	bel (audible bell)	Tovush signali
\b	0x08	Bs (backspace)	Bir qadam qaytish
\f	0x0C	Ff (form feed)	Sahifani qaytarish
\n	0x0A	lf (line feed)	Qatorni o'tkazish

\r	0x0D	Cr (carriage return)	Karetkani qaytarish
\t	0x09	Ht (horizontal tab)	Gorizantal tabulyatsi
\v	0x0B	Vt (vertical tab)	Vertikal tabulyatsi
\\	0x5C	\ (backslash)	Teskari chiziq
\'	0x27	' (single quote)	Apostrif (oddiy qavs)
\"	0x22	" (double quote)	Ikkilik qavs
\?	0x3F	? (question mark)	Savol Belgisi
\000	000	Любой (octal number)	Belgi sakkizlik kodi
\xhh	0xhh	Любой (hex number)	Belgi o'n oltilik kodi

Satrlı konstanta.

Satrlı konstantalar C++ tili konstantalariga kirmaydi, balki leksemalari alohida tipi hisoblanadi. Shuning uchun adabiyotda satrlı konstantalar satrlı leksemalar deb ham ataladi..

Satrlı konstanta bu ikkilik qavslarga olingan ihtiyoriy belgilar ketma ketligidir. Misol uchun "Men satrlı konstantaman".

Satrlar orasiga eskeyp belgilar ham kirishi mumkin. Bu belgilar oldiga \ belgisi quyiladi. Misol uchun : "***n Bu satr \n uch katorga \n zhoylashadi***".

Satr belgilari hotirada ketma-ket joylashtiriladi va har bir satrlı konstanta ohiriga avtomatik ravishda kompilyator tomonidan '\0' belgii qo'shiladi. Shunday satrning hotiradagi hazhmi belgilar soni+1 baytga tengdir.

Ketma-ket kelgan va bushlik, tabulyatsiya yoki satr ohiri belgisi bilan ajratilgan satrlar kompilyatsiya davrida bitta satrga aylantiriladi. Misol uchun: "**Salom**" "**Toshkent**" satrlari bitta satr deb qaraladi.

"Salom Toshkent"

Bu qoidaga bir necha qatorga yozilgan satrlar ham buysinadi. Misol uchun :

"O'zbekistonga "

"bahor "

"keldi"

qatorlari bitta qatorga mos: "**O'zbekistonga bahor keldi"**

Agar satrda '\ ' belgisi uchrasa va bu belgidan so'ng to '\n' satr ohiri belgisigacha bushlik belgisi kelsa bu bushlik belgilari '\ ' va '\n' belgisi bilan birga satrdan uchiriladi. Satrning uzi keyingi satrda kelgan satr bilan qo'shiladi.

**"Ozbekistonga **

" bahor

" keldi"

qatorlari bitta qatorga mos: "**Uzbekistonga bahor keldi"**

Sanovchi konstanta.

Sanovchi konstantalar enum hizmatchi so'zi yordamida kiritilib, int tipidagi sonlarga qulay so'zlarni mos quyish uchun ishlatiladi.

Misol uchun: **enum{one=1,two=2,three=3};**

Agar son qiymatlari ko'rsatilmagan bulsa eng chapki so'zga 0 qiymati berilib qolganlariga tartib buyicha usuvchi sonlar mos quyiladi:

Enum{zero,one,two};

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, two=2;

Konstantalar aralash ko'rinishda kiritilishi ham mumkin:

Enum(zero,one,for=4,five,seeks}.

Bu misolda avtomatik ravishda konstantalar quyidagi qiymatlarni qabul qiladi:

Zero=0, one=1, for=4;five=5,seeks=6;

Yana bir misol:

Enum BOOLEAN {NO, YES};

Konstantalar qiymatlari:

NO=0, YES=1;

Nomlangan konstantalar.

Ci++ tilida o'zgaruvchilardan tashqari nomlangan konstantalar kiritilishi mumkin. Bu konstantalar qiymatlarini dasturda o'zgartirish mumkin emas. Konstantalar nomlari dasturchi tomonidan kiritilgan va hizmatchi so'zlardan farqli bo'lgan identifikatorlar bulishi mumkin. Odatda nom sifatida katta lotin harflari va ostiga chizish belgilari kombinaciyasidan iborat identifikatorlar ishlatiladi. Nomlangan konstantalar quyidagi shaklda kiritiladi:

Const tip *konstanta_nomi* =*konstanta_kiyjmati*.

Misol uchun:

Const double EULER=2.718282;

Const long M=99999999;

Const R=765;

Ohirgi misolda konstanta tipi ko'rsatilmagan, bu konstanta **int** tipiga tegishli deb hisoblanadi.

Nul ko'rsatkich.

NULL- ko'rsatkich yagona arifmetik bulmagan konstantadir. Konkret realizatsiyalarda null ko'rsatkich 0 eki 0L eki nomlangan konstanta NULL orqali tasvirlanishi mumkin. Shuni aytish lozimki bu konstanta qiymati 0 bo'lishi eki '0' belgii kodiga mos kelishi shart emas.

Quyidagi jadvalda konstantalar chegaralari va mos tiplari ko'rsatilgan:

Ma'lumotlar turi	Hajm, bit	Qiymatlar chegarasi	Tip vazifasi	
Unsignedchar	8	0...255	Kichik butun sonlar va belgilar kodlari	
Char	8	-128...127	KichikbutunsonlarvaASIIkodlar	
Enum	16	-32768...32767	Butun sonlar tartiblangan katori	
Unsigned int	16	0...65535	Katta butun sonlar	
Short int	16	-32768...32767	Kichik butun sonlar, tsikllarni boshqarish	
Int	16	-32768...32767	Kichik butun sonlar, tsikllarni boshqarish	

Unsignedlong	32	0...4294967295	Astronomik masofalar
Long	32	-147483648... ...2147483647	Katta sonlar
Float	32	3.4E-32...3.4E+38	Ilmiy hisoblar (7 raqam)
Double	64	1.7E-308...1.7E+308	Ilmiy hisoblar(15 raqam)
Long double	80	3.4E-4932... 1.1E+4932	Moliyaviy hisoblar (19 raqam)

Sinovuchun savollar:

1. C++ tili qanday ma'lumotlardan iborat?
2. Ma'lumotlarning qanday turlari?
3. Ma'lumotlar qanday tavsiflanadi?
4. C++ tilida o'zgarmas nima?
5. Qanday o'zgarmlarni bilasiz?
6. Ma'lumotlarning nechta standart turi ishlatiladi?

Uyga vazifa : C++ tizimida ma'lumotlar va ularning turlari bilan ishlash qoidalarini mukammal o'rganish va ularni amaliyotga tatbiq etish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu:C++ tilida amallar. Arifmetik amallar, qiymat berish operatori. Inkrement va dekriment amallar. Mantiqiy amallar, taqqoslash amallar..

Reja:

1. Arifmetik amallar, qiymat berish operatori.
2. Ifoda.
3. Inkrement va dekriment amallar

Ma'ruza bayoni

C++ da arifmetik amallar

Ko'p programmalar ijro davomida arifmetik amallarni bajaradi. C++ dagi amallar quyidagi jadvalda berilgan. Ular ikkita operand bilan ishlatildi.

C++ dagi amal **Arifmetik operator** **Algebraik ifoda** **C++** **dagi ifodasi:**

<i>Qo'shish</i>	+	$h+19$	$h+19$	
<i>Ayirish</i>	-		$f-u$	$f-u$
<i>Ko'paytirish</i>	*		sl	$s*l$
<i>Bo'lish</i>	/		$v/d,$	v/d
<i>Modul olish</i>	%		$k \text{ mod } 4$	$k\%4$

Bo'larning ba'zi birlarinig hususiyatlarini ko'rib chiqaylik. Butun sonli bo'lishda, yani bo'luvchi ham, bo'linuvchi ham butun son bo'lganda, javob butun son bo'ladi. Javob yahlitlanmaydi, kasr qismi tashlanib yuborilib, butun qismining o'zi qoladi.

Modul operatori (%) butun songa bo'lishdan kelib chiqadigan qoldiqni beradi. $x\%y$ ifodasi x ni y ga bo'lgandan keyin chiqadigan qoldiqni beradi. Demak, $7\%4$

bizga 3 javobini beradi. % operatori faqat butun sonlar bilan ishlaydi. Vergulli (real) sonlar bilan ishlash uchun "math.h" kutubhonasidagi fmod funksiyasini qo'llash kerak.

C++ da qavslarning ma'nosi huddi algebradagidekdir. Undan tashqari boshqa boshqa algebraik ifodalarning ketma-ketligi ham odatdagidek. Oldin ko'paytirish, bo'lish va modul olish operatorlari ijro ko'radi. Agar bir necha operator ketma-ket kelsa, ular chapdan o'nga qarab ishlanadi. Bu operatorlardan keyin esa qo'shish va ayirish ijro etiladi.

Misol keltiraylik. $k = m * 5 + 7 \% n / (9 + x);$

Birinchi bo'lib $m * 5$ hisoblanadi. Keyin $7 \% n$ topiladi va qoldiq $(9 + x)$ ga bo'linadi. Chiqqan javob esa $m * 5$ ning javobiga qo'shiladi. Qisqasini aytsak, amallar matematikadagi kabi. Lekin biz o'qishni osonlashtirish uchun va hato qilish ehtimolini kamaytirish maqsadida qavslarni kengroq ishlatishimiz mumkin. Yuqoridagi misolimiz quyidagi ko'rinishga ega bo'ladi.

$$k = (m * 5) + ((7 \% n) / (9 + x));$$

Amallar jadvali

Arifmetik amallar	Razryadli amallar	Nisbat amallari	Mantiqiy amallar
+ qo'shish	&va	== teng	&&va
- bo'lish	yoki	!= teng emas	yoki
* ko'paytirish	^ inkor	>katta	! inkor
/ bo'lish	<<chapga surish	>= katta yoki teng	
% modul olish	>>o'ngga surish	<kichik	
- unar minus	~ inkor	<= kichik yoki teng	
+ unar plyus			
++ oshirish			
-- ka-maytirish			

Amallar jadvali (davomi)

Imlo amallar	Qiymat berish va shartli amallar	Tipli amallar	Adresli amallar
() – doirali qavs	= - oddiy qiymat berish	(tip) – tipni o'zgartirish	& - adresni aniqlash
[] – kavadrat qavs	op= - murakkab qiymat berish	sizeof- hajmni hisoblash	* - adres bo'yicha qiymat aniqlash yoki joylash
, - vergul	? – shartli amal		

Arifmetikamallar.

Amallar odatda unariya'ribitta operandga qo'llaniladigan amallarga binariya'niik kioperandga qo'llaniladigan amallarga ajratiladi.

Binaramallaradditivya'ni + qo'shuvva – ayirishamallariga , hamdamultiplikativya'ni * kupaytirish, / bulishva % modulolishamallarigaajratiladi. Additivamallariningustivorligimultiplikativamallariningustivorligidanpastroqdir. Butunsonnibutunsongabo'lgandanatijabutunsongachayahlitlanadi. Misoluchun $20/3=6$; $(-20)/3=-6$; $20/(-3)=-6$.

Modulamalibutunsonnibutunsongabulishdanhosilbo'ladiganqoldikkatengdir. Agaromodulamalimusbatoperandlargaqo'llanilsa, natijahammusbatbo'ladi, aksholdanati jaishorasikompilyatorgabog'likdir.

Binararifmetikamallar bajarilgandanatiplarnikeltirishquyidagiqoidalarasosidaamalgaoshiriladi:

shortvachartiplariinttipigakeltiriladi;

Agaroperandlarbirilongtipigategishlibo'lsaikkinchioperandhamlongtipigakeltiriladivanatijahamlongtipigategishlibo'ladi;

Agaroperandlarbirifloattipigategishlibo'lsaikkinchioperandkhamfloattipigakeltiriladivanatijahamfloattipigategishlibo'ladi;

Agaroperandlarbiridoubletipigategishlibo'lsaikkinchioperandhamdoubletipigakeltiriladivanatijahamdoubletipigategishlibo'ladi;

Agaroperandlarbirilongdoubletipigategishlibo'lsaikkinchioperandhamlongdoubletipigakeltiriladivanatijahamlongdoubletipigategishlibo'ladi;

Unaramallargaishorano'zgartiruvchiunarminus – vaunar + amallarikiradi.

Bundantashqari ++ va -- amallarihamunaramallargakiradi.

++ unaramaliqiyatni 1 gaoshirishniko'rsatadi. Amalniprefiksyani postfiksyaniiko'rinishdaishlatisholdino'zgaruvchiqiyatinioshiribso'ngrafoydalanishlozimligini, ko'rinishdaishlatisholdino'zgaruvchiqiyatidanfoydalanibso'ngraoshirishkerakliginiko'rsatadi. Misoluchuniqiyati 2 gatengbo'lsin, uholda $3+(++i)$ ifodaqiyati 6 ga, $3+i++$ ifodaqiyati 5 gatengbo'ladi. Ikkalaholdahamiqiyati 3 gatengbo'ladi.

-- unaramaliqiyatni 1 gakamaytirishniko'rsatadi. Buamalhamprefiksvapostfiksko'rinishdaishlatilishimumkin. Buikkiamalni-faqato'zgaruvchilargaqo'llashmumkindir.

Unaramallarningustivorligibinaramallardanyuqoridir.

Razryadliamal-

lar.Razryadliamallarnatijasibutunsonlarniikkilikko'rinishlariningharbirrazryadigamosmantikiyamallarniqo'llashdanhosilbo'ladi. Masalan 5 kodi 101 ga teng va 6 kodi 110 ga teng.

$6&5$ qiyjmati 4 ga ya'ni 100 ga teng.

$6|5$ qiyjmati 7 ga ya'ni 111 ga teng.

6^5 qiyjmati 3 ga ya'ni 011 ga teng.

~ 6 kiyjmati 4 ga yajhni 010 ga teng.

Bu misollarda amallar ustivorligi oshib borishi tartibida berilgandir.

Bu amallardan tashqari $M \ll N$ chapga razryadli siljitish va $M \gg N$ unga razryadli siljitish amallari qo'llaniladi. Siljitish M butun sonning razryadli ko'rinishiga

qo'llaniladi. N nechta pozitsiyaga siljitish kerakligini ko'rsatadi.

Chapga N pozitsiyaga surish bu operand qiymatini ikkining N chi daraasiga kupaytirishga mos keladi. Misol uchun $5 \ll 2 = 20$. Bu amalning bitli ko'rinishi:

$$101 \ll 2 = 10100.$$

Agar operand musbat balsa N pozitsiyaga unga surish chap operandni ikkining N chi darajasiga bo'lib kasr qismini tashlab yuborishga mosdir. Misol uchun $5 \gg 2 = 1$. Bu amalning bitli ko'rinishi $101 \gg 2 = 001 = 1$. Agarda operand qiymati manfiy balsa ikki variant mavjuddir: arifmetik siljitishda bushatilayotgan razryadlar ishora razryadi qiymati bilan to'ldiriladi, mantiqiy siljitishda bushatilayotgan razryadlar nullar bilan tuldiriladi.

Razryadli surish amallarining ustivorligi o'zaroteng, razryadli inkor amalidan past, qolgan razryadli amallardan yuqoridir. Razryadli inkor amali unar qolgan amallar binar amallarga kiradi.

Nisbat amallari. Nisbat amallari qiymatlari 1 ga teng agar nisbat bajarilsa va aksincha 0 ga tengdir. Nisbat amallari arifmetik tipdagi operandlarga yoki ko'rsatkichlarga qo'llaniladi.

Misollar:

$1 != 0$ qiymati 1 ga teng;

$1 == 0$ qiymati 0 ga teng;

$3 >= 3$ qiymati 1 ga teng;

$3 > 3$ qiymati 0 ga teng;

$2 <= 2$ qiymati 1 ga teng;

$2 < 2$ qiymati 0 ga teng;

Katta $>$, kichik $<$, katta ekiteng $>=$, kichik ekiteng $<=$ amallarining ustivorligi bir hildir.

Teng $=$ vateng emas $!=$ amallarining ustivorligi uzaroteng va qolgan amallardan pastdir.

Funksiya	Aniqlanishi	Misol
ceil(x)	x ni x dan katta yoki unga teng b-n eng kichik butun songacha yahlitlaydi	ceil(12.6) = 13.0 ceil(-2.4) = -2.0
cos(x)	x ning trigonometrik kosinusi (x radianda)	cos(0.0) = 1.0
exp(x)	e ning x chi darajasi (eskponetsial f-ya)	exp(1.0) = 2.71828
fabs(x)	x ning absolut qiymati	$x > 0 \Rightarrow \text{abs}(x) = x$ $x = 0 \Rightarrow \text{abs}(x) = 0.0$ $x < 0 \Rightarrow \text{abs}(x) = -x$
floor(x)	x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi	floor(4.8) = 4.0 floor(-15.9) = -16.0
fmod(x,y)	x/y ning qoldig'ini kasr son tipida beradi	fmod(7.3,1.7) = 0.5
log(x)	x ning natural lagorifmi (e asosiga ko'ra)	log(2.718282) = 1.0
log10(x)	x ning 10 asosiga ko'ra lagorifmi	log10(1000.0) = 3.0
pow(x,y)	x ning y chi darajasini beradi	pow(3,4) = 81.0

$\sin(x)$ x ning trigonometrik sinusi (x radianda) $\sin(0.0) = 0.0$
 $\text{pow}(16,0.25) = 2$
 $\text{sqr}t(x)$ x ning kvadrat ildizi $\text{sqr}t(625.0) = 25.0$
 $\text{tan}(x)$ x ning trigonometrik tangensi (x radianda) $\text{tan}(0.0) = 0$

Qiymat berish operatorlari

Bu qismda keyingi bo'limlarda kerak bo'ladigan tushuncha-larni berib o'tamiz. C++ da hisoblashni va undan keyin javobni o'zgaruvchiga beruvchi bir necha operator mavjuddir. Misol uchun:

```
k = k * 4; ni
k *= 4;
```

deb yozsak bo'ladi.

Bunda operatorining chap argumenti o'ng argumentga qo'shiladi va javob chap argumentdasaqlanadi. Biz har bir operatorni ushbu qisqartirilgan ko'rinishda yozsa olamiz (+=, -=, /=, *= %=). Ikkala qism birga yoziladi. Qisqartirilgan operatorlar tezroq yoziladi, tezroq kompilyatsiya qilinadi va ba'zi bir hollarda tezroq ishlaydigan mashina kodi tuziladi.

1 ga oshirish va kamaytirish operatorlari (increment and decrement) C++ da bir argument oluvchi **inkrement** (++) va **dekrement** (--) operatorlari mavjuddir. Bo'lar ikki ko'rinishda ishlatiladi, biri o'zgaruvchidan oldin (**++f** - **preinkrement**, **--d** - **predekrement**), boshqasi o'zgaruvchidan keyin (**s++** - **postinkrement**, **s--** - **postdekrement**) ishlatilgan holi. Bo'larning bir-biridan farqini aytin o'taylik. Postinkrementda o'zgaruvchining qiymati ushbu o'zgaruvchi qatnashgan ifodada shlatiladi va undan keyin qiymati birga oshiriladi. Preinkrementda esa o'zgaruvchining qiymati birga oshiriladi, va bu yangi qiymat ifodada qo'llaniladi. Predekrement va postdekrement ham aynan shunday ishlaydi lekin qiymat birga kamaytiriladi. Bu operatorlar faqatgina o'zgaruvchining qiymatini birga oshirish/kamaytirish uchun ham ishlatilinishi mumkin, yani boshqa ifoda ichida qo'llanilmasdan. Bu holda preva post formalarining farqi yo'q.

Masalan:

```
++r;
r++;
```

Yuqoridagilarning funksional jihattan hech qanday farqi yo'q, chunki bu ikki operator faqat r ning qiymatini oshirish uchun qo'llanilmoqda. Bu operatorlarni oddiy holda yozsak:

```
r = r + 1;
d = d - 1;
```

Lekin bizning inkrement/dekrement operatorlarimiz oddiygina qilib o'zgaruvchiga bir qo'shish/ayirishdan ko'ra tezroq ishlaydi. Yuqoridagi operatorlarni qo'llagan holda bir dastur yozaylik.

```
//Postinkremet, preinkrement va qisqartirilgan teglashtirish operatrlari
# include <iostream.h>
int main()
{
int k = 5, l = 3, m = 8;
```

```

cout << k++ << endl; //ekranga 5 yozildi, k = 6 bo'ldi.
l += 4; // l = 7 bo'ldi.
cout << --m << endl; // m = 7 bo'ldi va ekranga 7 chiqdi.
m = k + (++l); // m = 6 + 8 = 14;
return (0);
}

```

Dasturdagi o'zgaruvchilar e'lon qilindi va boshqangich qiymatlarni olishdi. cout << k++ << endl; ifodasida ekranga oldin k ning boshlangich qiymati chiqarildi, keyin esa uning qiymati 1 da oshirildi. l += 4; da l ning qiymatiga 4 soni qo'shildi va yangi qiymat l da saqlandi. cout << --m << endl; ifodasida m ning qiymati oldin predekrement qilindi, va undan so'ng ekranga chiqarildi. m = k + (++l); da oldin l ning qiymati birga ishirildi va l ning yangi qiymati k ga qo'shildi. m esa bu yangi qiymatni oldi. Oshirish va kamaytirish operatorlari va ularning argumentlari orasida bo'shliq qoldirilmasligi kerak. Bu operatorlar sodda ko'rinishdagi o'zgaruvchi-larga nisbatan qo'llanilishi mumkin halos. Masalan:

```
++(f * 5);
```

ko'rinish noto'g'ridir.

Mantiqiy amallar. C ++ tilidamantiqiytipyukdir. Shuninguchunmantiqiyamallarni-butunsonlargaqo'llanadi. Buamallarningnatijalariqo'yidagichaaniqlanadi:

*x//yamali 1 gatengagarx>0 ekiy>0 bo'lsa, aksincha 0 gateng
x&&y amali 1 ga teng agar x>0 va y>0 bo'lsa, aksincha 0 ga teng
!x amali 1 ga teng agar x>0 bulsa, aksincha 0 ga teng*

Bu misollarda amallar ustivorligi oshib borish tartibida berilgandir.

Inkor ! amali unar kolganlari binar amallardir.

Bu amallardan tashqari quyidagi amallar ham mavjuddir:

Qiymat berish amali. Qiymat berish amali = binar amal bo'lib chap operandi odatda o'zgaruvchi ung operandi odatda ifodaga teng bo'ladi. Misol uchun **Z=4.7+3.34**

Bu qiymati 8.04 ga teng ifodadir. Bu qiymat Z o'zgaruvchiga ham beriladi.

Bu ifoda ohiriga nuqta vergul ; belgisi quyilganda operatorga aylanadi.

```
Z=4.7+3.34
```

Bitta ifodada bir necha qiymat berish amallari qo'llanilishi mumkin. Misol uchun:

```
C=y=f=4.2+2.8;
```

Bundan tashqari C ++ tili da murakkab qiymat berish amali mavjud bo'lib, umumiy ko'rinishi quyidagichadir:

O'zgaruvchi_nomi amal= ifoda;

Bu erda amal quyidagi amallardan biri *,/,%,+,-, & , ^, |, <<, >>.

Misol uchun:

X+=4 ifoda x=x+4 ifodaga ekvivalentdir;

X*=a ifoda x=x*a ifodaga ekvivalentdir;

X/=a+b ifoda x=x/(a+b) ifodaga ekvivalentdir;

X>>=4 ifoda x=x>>4 ifodaga ekvivalentdir;

Sinov uchun savollar:

1. Qanday arifmetik amallarni bilasiz?
2. Qiymat berish amali qanday?

3. Ifoda qanday tuzilgan?
4. Inkrement va dekriment amallarini izoxlab bering?

Uyga vazifa : C++ tizimida arifmetik amallarni mukammal o'rganish va ularni amaloyotga tatbiq etish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu: C++ tilida standart funksiyalar va ifodalar. C++ tilida dasturning umumiy tuzilishi.

Reja:

1. Sizeof amali.
2. Mantiqiy va taqqoslash amallar.

Ma'ruza bayoni

C++ tilida standart funksiyalar.

C++ tilida **math.h** matematik funksiyalar kutubxonasiga tegishli quyidagi funksiyalar mavjud:

<i>C++ tilida matematik funksiyalar</i>			
Funksiya	Belgilash	Funksiya	Belgilash
absolyut qiymat	<i>fabs(x)</i>	darajaga ko'tarish x^n	<i>pow(x,n)</i>
arctangens	<i>atan(x)</i>	qoldiqni hisoblash	<i>fmod(x,y)</i>
kosinus	<i>cos(x)</i>	x dan kichik bo'lmagan butun son	<i>ceil(x)</i>
sinus	<i>sin(x)</i>	x dan katta bo'lmagan butun son	<i>floor(x)</i>
natural logarifm	<i>ln(x)</i>	arcsinus	<i>asin(x)</i>
kvadrat ildiz	<i>sqrt(x)</i>	arccos	<i>acos(x)</i>
eksponenta	<i>exp(x)</i>	o'nli logarifm	<i>log10(x)</i>

Masalan : $ceil(4.8) = 5$, $floor(4.8) = 4$, $pow(2,5) = 32$, $fmod(5,3) = 2$.

Trigonometrik funksiyalarning qiymatlari radian o'lcho birligida hisoblanadi. **Masalan:** $\sin(1.2) = 0.932039$, $\cos(1.2) = 0.362358$, $\tan(1.2) = 2.57215$.

Agar trigonometrik funksiyalarni qiymatlarin graduslarda hisoblash kerak bo'lsa, ular radianga aylantiriladi yoki π orqali ifodalani. **Masalan:** $\sin 60^\circ = \sin(\pi/3) = \sin(\pi * 60/180) = 0.86576$.

C++ tilida ifodalar.

C++ tilida ifodalar o'zgarmlardan, o'zgaruvchilardan, funksiyalardan, amallar ishorasidan va qavslardan tuzilgan bo'lib, ostki va uctki indeksiz bir satrga yoziladi.

C++ tilida dasturlash tilida arifmetik, mantiqiy va satrli ifodalar ishlatiladi.

Arifmetik ifoda. C++ tilida arifmetik ifodalar o'zgaraslardan, o'zgaruvchilardan, funksiyalardan, amallar ishorasidan va qavslardan tuzilgan bo'ladi. Ifodalarda qatnashayotgan o'zgaraslar, o'zgaruvchilar va funksiyalar dasturda e'lon qilingan bo'lishi yoki standart nomiga ega bo'lishi kerak. Sonli qiymatlarni hisoblash uchun arifmetik ifoda qo'llaniladi.

Arifmetik ifodada bo'lish va ko'paytirish amallari qo'shish va ayirish amallaridan oldin bajariladi. **Masalan:** $3 + 2 * 5 - 3$ ifodaning bajarilishidan 10 hosil bo'ladi.

Bir xil o'rinli amallar ishtirok etgan ifodalarda ular chapdan o'ngga qarab bajariladi. **Masalan.** $4 * 6 / 3$ ifoda natijasi 8 ga teng.

Qavs ichiga olingan ifodalar birinchi navbatda bajariladi.

Masalan. $(7-2) * (5+4)$ ifoda natijasi 45 ga teng

Arifmetik ifodada haqiqiy va butun turdagi ma'lumotlar qatnashishi mumkin, natija ham mos ravishda haqiqiy yoki butun bo'ladi.

Oddiy yozuvda	C++ tilida
$\frac{a + 12,686}{c_2 - 1,87 \cdot 10^3} + \gamma$	$(a + 12.686) / (C2 - 1.87E3) + \text{gamma}$
$\frac{ 2 \sin 4x + x }{3x}$	$\text{fabs}(2 * \sin(4 * x) + x) / (3 * x)$
$\frac{2^2 + \ln 1,13}{\text{tg} 2x}$	$(2 * 2 + \ln(1.13)) / \tan(2 * x)$
$\frac{a^2 + 2^{1,17}}{\sqrt{x + y}}$	$(\text{pow}(a, 2) + \text{pow}(2, 1.17)) / \text{sqrt}(x + y)$
$\frac{ x - y }{1 + xy }$	$(\text{fabs}(x) - \text{fabs}(y)) / (\text{float})(1 + \text{fabs}(x * y))$
$\ln \left(\left(y - \sqrt{ x } \right) \left(x - \frac{y}{z + x^2 / 4} \right) \right)$	$\log(\text{fabs}((y - \text{sqrt}(\text{fabs}(x))) * (x - y / (z + \text{pow}(x, 2) / 4.0))))$

Mantiqiy ifoda. Mantiqiy ifodalar mantiqiy qiymatlar, mantiqiy va munosabat amallari, arifmetik hamda belgili qiymatlar qatnashib, natijasi **true** va **false** qiymat bo'lishi mumkin. Unda amallar quyidagi tartibda bajariladi:

1. not (yo'q);

2. * (ko'paytirish), / (bo'lish), div, mod, and (va)

3. + (qo'shish), - (ayirish), or (yoki);

4. < (kichik), <= (kichik yoki teng), = (teng), <> (teng emas), >= (katta yoki teng), > (katta);

Bitta satrda berilgan amallar bir xil bajarilish o'rniga ega. Mantiqiy ifodalarda faqat oddiy qavslar yoziladi.

Masalan. 1) $(A > 3)$ and $(B = A + 6)$ or not $(c = 4)$

2) $(A \leq x)$ and $(x \leq B)$ or $(c \leq x)$ and $(x \leq d)$

Dastur tuzilishi.

Sodda dastur tuzilishi. Dastur preprocessor komandalari va bir necha funktsiyalardan iborat bo'lishi mumkin. Bu funktsiyalar orasida **main** nomli asosiy funktsiya bo'lishi shart. Agar asosiy funktsiyadan boshqa funktsiyalar ishlatilmasa dastur quyidagi ko'rinishda tuziladi:

Preprocessor komandalari

```
Void main()
```

```
{
```

```
Dastur tanasi.
```

```
}
```

Preprocessor direktivalari kompilyatsiya jarayonidan oldin preprocessor tomonidan bajariladi. Natijada dastur matni preprocessor direktivalari asosida o'zgartiriladi.

Preprocessor komandalaridan ikkitasini ko'rib chiqamiz.

```
#include <fayl_nomi>
```

Bu direktiva standart bibliotekalardagi funktsiyalarni dasturga joylash uchun foydalaniladi.

```
#define <almashtiruvchi ifoda><almashinuvchi ifoda>
```

Bu direktiva bajarilganda dastur matnidagi almashtiruvchi ifodalar almashinuvchi ifodalarga almashtiriladi.

Misol tariqasida C++ tilida tuzilgan birinchi dasturni keltiramiz:

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
    Cout << "\n Salom, Dunyo! \n";
```

```
}
```

Bu dastur ekranga Salom, Dunyo! Jumlasini chiqaradi.

Define direktivasi yordamida bu dasturni quyidagicha yozish mumkin:

```
#include <iostream.h>
```

```
#define pr    Cout << "\n Salom, Dunyo! \n"
```

```
#define begin {
```

```
#define end }
```

```
void main()
```

```
    begin
```

```
        pr;
```

```
    end
```

Define direktivasidan nomlangan konstantalar kiritish uchun foydalanish mumkindir. Misol uchun: **#define EULER 2.718282**

Agar dasturda quyidagi matn mavjud bo'lsin:

Double mix=EULER

D=alfa*EULER

Preprocessor bu matnda har bir EULER konstantani uning qiymati bilan almashtiradi, va natijada quyidagi matn hosil bo'ladi.

Double mix=2.718282

D=alfa*2.718282

Dastur matni va preprocessor. C ++ tilida matnli fayl shaklida tayyorlangan dastur uchta qayta ishlash bosqichlaridan o'tadi.

Matnni preprocessor direktivalari asosida o'zgartilishi. Bu jarayon natijasi Yana matnli fayl bo'lib preprocessor tomonidan bajariladi.

Kompilyatsiya. Bu jarayon natijasi mashina kodiga o'tkazilgan obektli fayl bo'lib, kompilyator tomonidan bajariladi.

Bog'lash. Bu jarayon natijasi to'la mashina kodiga o'tkazilgan bajariluvchi fayl bo'lib, boglagich(komponent) tomonidan bajariladi.

Preprocessor vazifasi dastur matnini preprocessor direktivalari asosida o'zgartirishdir. Define direktivasi dasturda bir jumlaning ikkinchi jumla bilan almashtirish uchun ishlatiladi. Bu direktivadan foydalanishning sodda misollarini biz yuqorida ko'rib chiqdik. Include direktivasi ikki ko'rinishda ishlatilishi mumkin.

#include fayl nomi direktivasi dasturning shu direktiva urniga qaysi matnli fayllarni qo'shish kerakligini ko'rsatadi.

#include <fayl nomi> direktivasi dasturga kompilyator standart bibliotekalariga mos keluvchi sarlavhali fayllar matnlarini qushish uchun muljhallangandir. Bu fayllarda funktsiya prototipi, tiplar, o'zgaruvchilar, konstantalar ta'riflari yozilgan bo'ladi. Funktsiya prototipi funktsiya qaytaruvchi tip, funktsiya nomi va funktsiyaga uzatiluvchi tiplardan iborat bo'ladi. Misol uchun **cos** funktsiyasi prototipi quyidagicha yozilishi mumkin: **double cos(double)**. Agar funktsiya nomidan oldin void tipi ko'rsatilgan bo'lsa bu funktsiya hech qanday qiymat qaytarmasligini ko'rsatadi. Shuni ta'kidlash lozimki bu direktiva dasturga standart biblioteka qo'shilishiga olib kelmaydi. Standart funktsiyalarning kodlari bog'lash ya'ni aloqalarni tahrirlash bosqichida, kompilyatsiya bosqichidan so'ng amalga oshiriladi.

Kompilyatsiya bosqichida sintaksis hatolar tekshiriladi va dasturda bunday hatolar mavjud bo'lmasa, standart funktsiyalar kodlarisiz mashina kodiga utkaziladi.

Sarlavhali fayllarni dasturning ixtiyoriy joyida ulash mumkin bo'lsa ham, bu fayllar odatda dastur boshida qo'shish lozimdir. Shuning uchun bu fayllarga sarlavhali **fayl (header file)** nomi berilganidir.

Dasturda kiritish va chiqarish funktsiyalaridan masalan **Cout<<** funktsiyasidan foydalanish uchun **#include <iostream.h>** direktivasidan foydalanish lozimdir Bu direktivada **iostream.h** sarlavhali fayl nomi quyidagilarni bildiradi: **st-** standart(standartni), **i-** **input(vvod),o-** **output(vihvod),h** – **head(sarlavha).**

Sinov savollari

1. Sizeof amali nima?
2. Qanday mantiqiy amallarni bilasiz?
3. Taqqoslash amallari nim a uchun ishlatiladi?
4. C++ tili dastur qanday tuzilgan?
5. Dasturning har bir qismini izohlab bering.

Uyga vazifa: *mantiqiy va taqqoslash amallariga doir 5 tadan misolni daftaringizga yozing*

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu: C++ tilining asosiy operatorlari. Operator tushunchasi.

Ma'lumotlarni kiritish va chiqarish operatorlari. Sizeof amali.

Reja:

1. Operator tushunchasi.
2. Ma'lumotlarni kiritish va chiqarish operatorlari.
3. Sizeof amali.

Ma'ruza bayoni

C++ tilining asosiy operatorlari.

Operator - bu qandaydir amalni bajarish to'g'risida kompilyatorga uzatiladigan literal(buyruq)dir. Operatorlar operandlarga ta'sir qiladi. C++ da operandlar deb alohida literallar va butun ifodalar tushuniladi.

C++ tilida dastur tuzishda yuqorida berilgan qiymat berish (=) operatoridan tashqari bir nechta operatorlar ma'lumotlarni kiritish va chiqarish, tarmoqlanish va takrorlanish operatorlari ham ishlatiladi.

Ma'lumotlarni kiritish operatori.

Ma'lumki komputerdan har qanday o'zgaruvchining qiymati xotiraga kiritiladi hamda barcha amallar xotiradagi o'zgaruvchilar ustida bajariladi va natija ham xotirada hosil bo'ladi. Ma'lumotlarni kiritish operatori dasturda o'zgaruvchilarga klaviatura orqali boshlang'ich qiymatlarni berish uchun ishlatiladi.

Ma'lumotlar cin operatori yordamida kiritiladi. Qiymati kiritiladigan o'zgaruvchilar bu operatoridan keyin bir-birlaridan ">>" belgi bilan ajratilgan holda yoziladi. Uning umumiy ko'rinishi quyidagicha:

```
cin>>a1>>a2>>...>>an;
```

Bu yerda a1,a2,...,an qiymatlari kiritilishi talab etilgan o'zgaruvchilarning nomi.

cin operatori bajarilganda dastur bajarilishi to'xtaydi va kerakli ma'lumotlarni kiritilishini kutadi.

Masalan. cin>>a>>b>>c; bunda operator bajarilganda keyin a,b,c o'zgaruvchilarga tavsifi bo'yicha mos qiymatlarni berish talab etiladi. Kiritilayotgan qiymatlar o'zgaruvchilar qanday tartibda berilgan bo'lsa, ularga va ularning turiga mos qiymatlar bir-biridan bo'shliq(probel) belgisi bilan ajratilgan holda beriladi: 5 6.8 9. Buning natijasida a=5, b=6.8, c=9 qiymatga ega bo'ladi.

Bunda quyidagilarga alohida e'tibor berish kerak:

1. Kiritilayotgan qiymatlarni o'zgaruvchilar kiritish bo'yicha qabul qiladi, ya'ni cin operatoridagi 1-o'zgaruvchi kiritilgan 1-qiymatni va hokaza n-o'zgaruvchi kiritilgan n-qiymatni qabul qiladi..
2. Kiritilayotgan qiymatlar o'zgaruvchi dasturda tafsivlangan turga mos kelishi kerak, aks holda xatolik sodir bo'lishi mumkin..
3. **cin** operatori dasturning ixtiyoriy joyida va o'zgaruvchilar foydalanishidan oldin kelishi mumkin cin operatori orqali o'zgaruvchilarga alohida-alohida qiymat berish mumkin:

cin>>a; cin>>b; cin>>c; yozuvcin>>a>>b>>c; bilan teng kuchlidir.

Agar cin operatori orqali kiritilayotgan o'zgaruvchilar haqiqiy turga mansub bo'lib, ularga kiritilayotgan qiymatlar butun bo'lsa, ularni haqiqiy son ko'rinishida qabul qiladi. Masalan:

```
int main()
{
    float a,b,c,p,s;
```

cin >> a>>b>>c;

Kiritilayotgan son mos ravishda **5 6 8** bo'lsa, natijada a=5.0 , b=6.0, c=8.0 ega bo'ladi.

Ma'lumotlarni chiqarish operatori.

Dasturda hosil bo'lgan natijalarni chiqarish uchun chiqarish operatori ishlatiladi. Dastur natijalari **cout** operatori yordamida ekranga chiqariladi.

O'zgaruvchilar **cout** operatoridan keyin bir-birlaridan "<<" belgi bilan ajratilgan holda ko'rsatiladi. Uning umumiy ko'rinishi quyidagicha:

cout<<b1<<b2<<.....<<bn; bu yerda **b1,b2,...,bn** chiqarilotgan natijalar.

cout operatorida b1,b2,...,bn sifatida quyidalar qatnashishi mumkin:

1. **Qiymati chiqarilayotgan o'zgaruvchilar:** **cout<<x1<<x2;**

2. **Satrlı o'zgarmaslar.** Ular qo'shtirnoq ichiga olingan belgilar va harflar ketma ketligidan iborat bo'lib, dastur ixtiyoriy joyida matn yoki izohlarni chop etish uchun ishlatiladi: **cout<<"a,b,c o'zgaruvchilarga qiymat bering";**

Matnda boshqaruv belgilaridan foydalanish mumkin, masalan '/n'; Bu belgi undan keyin yozilgan ma'lumotni yangi satrdan chop etishni amalga oshiradi:

cout<<"kvadrat tenglamaning ildizlari /n"<<"x1="<<x1<<"x2="<<x2;

Bu operatorning bajarilishida quyidagi hosil bo'ladi:

kvadrat tenglamaning ildizlari

x1=

x2=

3. **Sonli o'zgarmlar.** Kompyuter yordamida tilning ba'zi bir elementlarini o'rganishda qo'llaniladi. Real dasturlashda undan juda kamdan-kam holda foydalaniladi Masalan: **cout<<6;**

4. **Ifoda.** Qavs ichida yozilgan ifodalar qatnashishi mumkin. Masalan : **cout<<(a+b);**

5. **endl** kalit so'zi. Bu chiqarish manipulyatori deb yurutiladi. Bu so'zdan keyin ma'lumotlar yangi satrdan chiqailadi. Masalan:

int a=2, b=5;

cout<<"\n yig'indi " <<(a+b)<<" \n Ko'paytma "<<(a*b); yoki

cout<<endl<<" Cymma " <<(a+b)<<endl<<" Ko'paytma "<<(a*b);

Ikkala operatorning bajarilishida ham bir xil natija hosil bo'ladi.

Sizeof amali

Har xil turli o'zgaruvchilar, o'zgarmlar, turning o'zi kompyuter xotirasida har xil joylar egallaydi. Hatto bir turli o'zgaruvchilarning o'zi har xil kompyuterlarda va bir xil operatsion sistemalarda har xil joy egallashi mumkin.

O'zgaruvchilar, o'zgarmlar va turlarning xotiradan qancha joy egallashini aniqlash maqsadida **sizeof** amalidan foydalaniladi. Buni quyidagi dastur misolida ko'ramiz:

```
int main ( )
{ int a, b, c; float x, y, z;
cout << "\n butun =" << sizeof (int);
cout << "\n haqiqiy =" << sizeof (float);
cout << "\n bul turi =" << sizeof (bool);
cout << "\n a =" << sizeof (a);
cout << "\n x =" << sizeof (x);
return 0;
}
```

sizeof amali operand sifatida ko'rsatilgan ob'ektning baytlarda hotiradagi hajmini hisoblash uchun ishlatiladi. Bu amalning ikki ko'rinishi mavjud: sizeof ifoda sizeof (tip) Misol uchun:

Sizeof 3.14=8

Sizeof 3.14f=4

Sizeof 3.14L=10

Sizeof(char)=1

Sizeof(double)=8.

Amallar ustivorligi

Rang	Amallar	Yo'nalish
1	() [] -> :: .	Chapdan o'ngga
2	! ~ + - ++ -- & * (tip)	O'ngdan chapga

	sizeof new delete tip()	
3	. * ->*	Chapdan o'ngga
4	* / % (multiplikativ binar amallar)	Chapdan o'ngga
5	+ - (additiv binar amallar)	Chapdan o'ngga
6	<<>>	Chapdan o'ngga
7	<<= >= >	Chapdan o'ngga
8	= !=	Chapdan o'ngga
9	&	Chapdan o'ngga
10	^	Chapdan o'ngga
11		Chapdan o'ngga
12	&&	Chapdan o'ngga
13		Chapdan o'ngga
14	?:(shartli amal)	Chapdan o'ngga
15	= *= /= %= += -= &= ^= = <<= >>=	Chapdan o'ngga
16	, (vergul amali)	Chapdan o'ngga

Mavzu: C++ tilida dasturda izohlar berish. C++ tilida chiziqli dastur.

Reja:

1. C++ tilida dasturda izohlar berish.
2. C++ tilida chiziqli dastur.
3. Misollar

Ma'ruza bayoni

C++ tilidasturdaizohlarberish

C++ tilidasturdaoperatorlarningbajarilishihamdaundaqatnasahayotgano'zgaruvchivao'zgarvasgaizohlarberishimkoniyatimavjud.

Izohlarmatnko'rinishidaberiladi. Izohlarkompilyatordaqaaytaishlanmaydivaoperatoringbajarilishigata'sirqilmaydi.

C++ tilidaizohlarniikkixilko'rinishdajoylashtirishmumkin:

- ikkitaketma-ketkeluvchiqiyshiqchiziq (//) orqali. Unidasturniixtiyoriysatrigajoylashtirishmumkin. Bundanqisqaizohlarberishuchunfoydalaniladi.
- Izohningboshiga /* belginivaoxiriga */ belgisiniqo'yishorqali. /* va */ belgilariorasidagimatnizohnibildiradi. Bu usulda uzun izohlarni berish uchun foydalaniladi.

Masalan

```
{
int i, n, k; // i, n, k lar butun turli o'zgaruvchilar
float x,y,z; / * x,y,z o'zgaruvchilar haqiqiy tuga tegishli ekanligi ko'rsatilgan va
ular qabul qiladigan qiymat haqiqiy ko'rinishda bo'ladi */
.....
}
```

C++ tilida chiziqli dastur

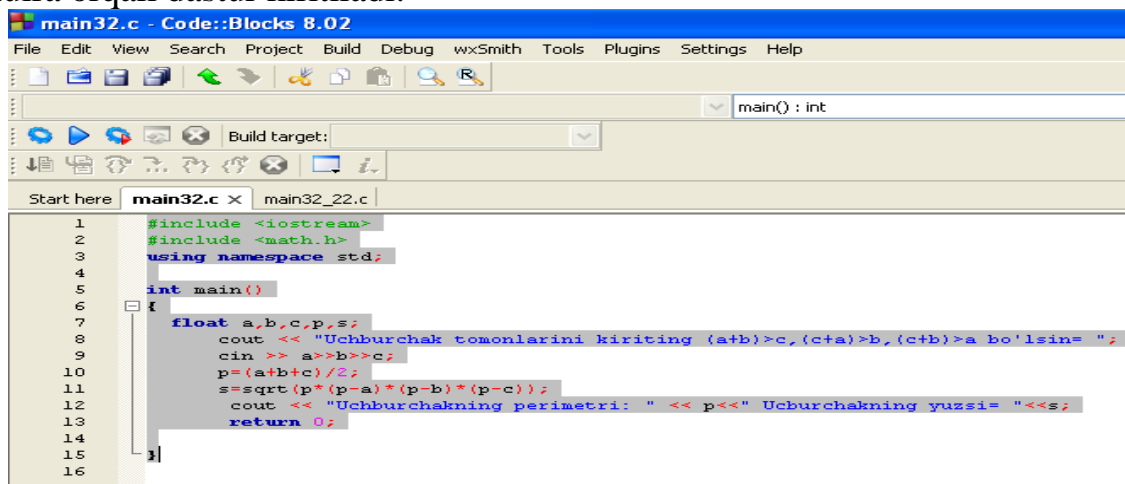
Yuqorida bayon qilingan qiymat berish, ma'lumotlarni kiritish va chiqarish operatorlari yordamida chiziqli dastur tuzish mumkin.

1-misol. Uchburchak tomonlarining uzunliklari bilan berilgan bo'lsa uning perimetri va yuzasini hisoblash dasturini tuzing.

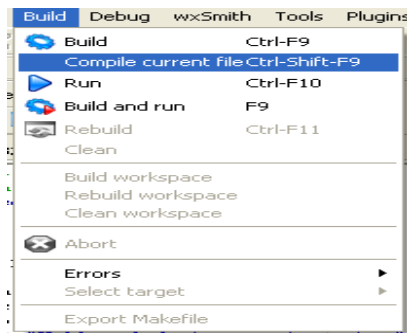
Dastur. C++ tilida dastur quyidagicha ko'rinishda bo'ladi:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a,b,c,p,s;
    cout << "Uchburchak tomonlarini kiriting (a+b)>c,(c+a)>b,(c+b)>a
bo'lsin= ";
    cin >> a>>b>>c;
    p=(a+b+c)/2;
    s=sqrt(p*(p-a)*(p-b)*(p-c));
    cout << "Uchburchakning perimetri: " << p<<" Uchburchakning yu-
zasi= "<<s;
    return 0;
}
```

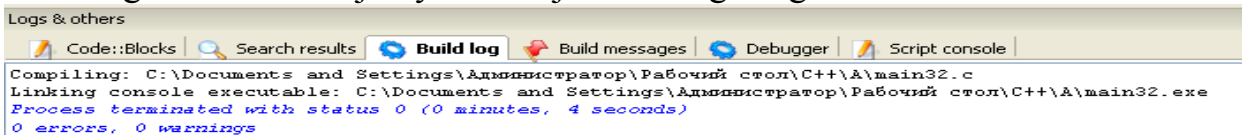
Bu tuzilgan dasturni ishlatish uchun eng avval **Code::Bloks** muhitini ishga tushirish (*Пуск→Все программы→CodeBloks→CodeBloks*) kerak. Shundan so'ng yangi fayl (*File→New→Project*) yaratiladi. Songra uning ishchi sohasi klaviatura orqali dastur kiritiladi.

The image shows a screenshot of the Code::Blocks IDE. The window title is "main32.c - Code::Blocks 8.02". The menu bar includes File, Edit, View, Search, Project, Build, Debug, wxSmith, Tools, Plugins, Settings, and Help. The toolbar contains various icons for file operations and execution. The main editor area shows the C++ code from the previous block, with line numbers 1 through 16 on the left. The code is: 1 #include <iostream>, 2 #include <math.h>, 3 using namespace std;, 4 int main(), 5 {, 6 float a,b,c,p,s;, 7 cout << "Uchburchak tomonlarini kiriting (a+b)>c,(c+a)>b,(c+b)>a bo'lsin= ";, 8 cin >> a>>b>>c;, 9 p=(a+b+c)/2;, 10 s=sqrt(p*(p-a)*(p-b)*(p-c));, 11 cout << "Uchburchakning perimetri: " << p<<" Uchburchakning yuzasi= "<<s;, 12 return 0;, 13 }, 14, 15, 16. The status bar at the bottom indicates "main() : int".

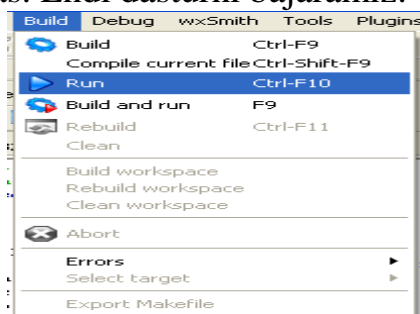
Dastur kiritlgandan keyin u kompilyatsiya qilinadi, ya'ni xatoliklar tekshiriladi.



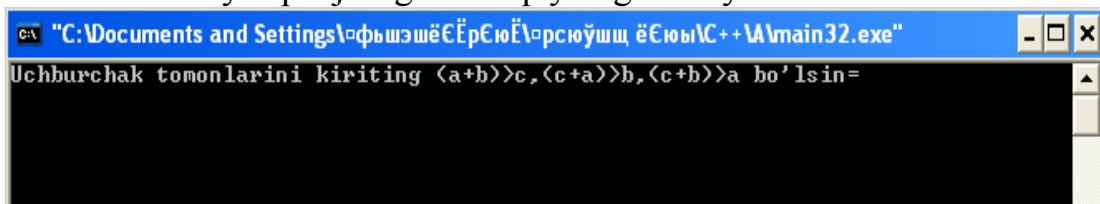
Agar xatolik mavjud yoki mavjud emasligi to'g'risida xabar hosil bo'ladi:



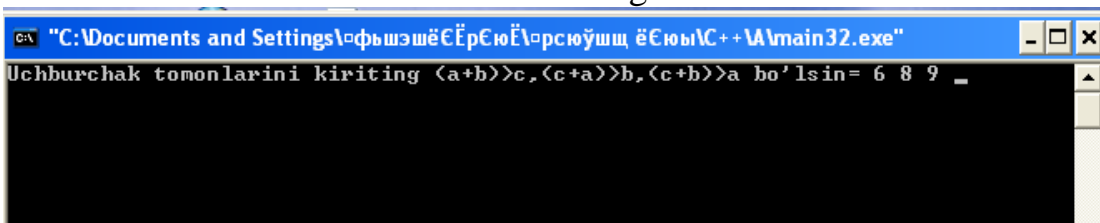
Agar xatolik mavjud bo'lsa u tuzatilib, yana kompilyatsiya qilinadi. Hozirgi holatda xatolik mavjud emas. Endi dasturni bajaramiz:



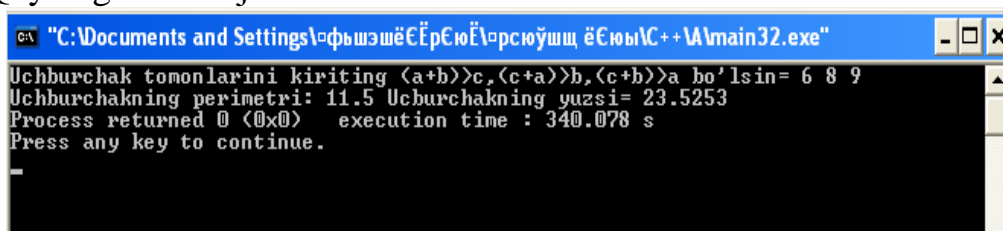
Bu buyruq bajarilgandan quyidagicha oyna hosil bo'ladi:



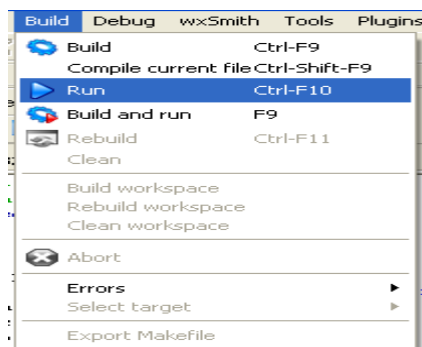
Bu a,b,c o'zgaruvchilarga qiymat berish kerakligini bildiradi. Biz 6 8 9 qiymatlarni klaviaturadan kiritamiz va enter tugmachasini bosamiz:



Quyidagicha natija hosil bo'ladi:



Dasturni a,b,c o'zgaruvchilarning boshqa qiymatlarida ishlatish uchun



Buyrug'ini bajarish yetarli.

2-misol. Arifmetik progressiyaning birinchi hadi, ayirmasi berilgan bo'lsa uning n-hadi va n ta hadlar yig'indisini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a1,d,xn,sn;
    int n;
    cout<<"arifmetik progressiyaning 1-hadini kiritng= ";
    cin>>a1;
    cout<<"arifmetik progressiyaning ortirmasini kiritng= ";
    cin>>d;
    cout<<"arifmetik progressiyaning hadlar sonini kiritng= ";
    cin>>n;
    xn= a1+(n-1)*d;
    sn=(a1+xn)/2*n;
    cout<<"arifmetik progressiyaning "<<n<<" hadi= "<<xn;
    cout<<"\narifmetik progressiyaning "<<n<<" ta hadi yig'indisi= "<<sn;
    return 0;
}
```

3-misol. $w = \frac{(x+y)^2}{2}$, $r = \sqrt{2\pi} + w$, $s = \pi r^2 + vw$ ifodani qiymatini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float x,y,w,v,s,r,Pi=3.1415;
    cout << "Ixtiyoriy uchta son kiritng= ";
    cin >> x>>y>>v;
    w=((x+y)*(x+y))/2;
    r=sqrt(2*Pi)+w;
```

```

    s=Pi*r*r+v*w;
cout << "Ifoda= " << s;
    return 0;
}

```

Nazorat uchun savollar

1. Operator nima?
2. C++ tilida dastur qanday tuzilgan?
3. Dasturning har bir qismini izohlab bering.
4. Ma'lumotlarni kiritish uchun qanday operatorlar ishlatilsadi?
5. Kiritish operatoridan foydalanish usullarini tushintiring.
6. Ma'lumotlarni chiqarish uchun qanday operatorlar ishlatilsadi?
7. Chiqarish operatoridan foydalanishni tushintiring.
8. O'zgaruvchilar qiymati va o'zgarmaslar qanday chiqariladi?
9. C++ tilida dasturi izohlar qanday beriladi?
10. Chiziqli dastur nima?

Uyga vazifa:

1. Ifodani qiymatini hisoblash dasturini tuzing:

$$2. \quad a = \sqrt[3]{x + \sqrt{y}}; \quad b = \sqrt{y}l^{-(y+4/2)},$$

3. Tomonlari bilan berilgan uchburchakning perimetri va yuzasini hisoblash dasturini tuzing.
4. Uchlarining koordinatalari bilan berilgan uchburchakning perimetri va yuzasini topish dasturini tuzing.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu:Tarmoqlanuvchi jarayonlarni dasturlash. Shartsiz va shartli o'tish operatorlari.

Reja:

1. Tarmoqlanuvchi jarayonlarni dasturlash.
2. Tarmoqlanuvchi jarayonlarni dasturlash.
3. ? shart operatori. Goto operatori va nishonlar.
4. ? shart operatori. Goto operatori va nishonlar.
5. Misollar

Ma'ruza bayoni

Tarmoqlanuvchi jarayonlarni dasturlash

Agar dasturning bajarilishi davomida buyruqlar ketma-ketligi biror shartga asosan o'zgarsa, bunday hollarda tarmoqlanish jarayonini tashkil etadigan operatorlardan foydalaniladi. C++ tilida tarmoqlanuvchi jarayonlarga dastur tuzishda tarmoqlanish operatorlaridan foydalaniladi. Tarmoqlanish operatorlari shartli o'tish, shartsoz o'tish va tanlash operatorlarinidan iborat bo'ladi.

Shartsiz o'tish operatori.

Bu operator hech imkon bo'lmay qolgan holda dasturning tabiiy bajarilishini buzib, boshqaruvni dasturning ixtiyoriy joyiga uzatish uchun xizmat qiladi.

O'tish operatori qo'yidagicha beriladi:

goto nishon(belgi);

bu yerda **goto(..ga o'tish)** - operator nomi, **nishon** – bu belgi bo'lib, u harf, harflar yoki harf va sonlar bo'lishi mumkin. Bu operatoridan foydalanish uchun boshqaruv shartsiz ravishda uzatiladigan operator belgilanadi. Operatorni belgilash uchun uning oldiga **nishon** va **:** qo'yiladi: **nishon : operator;**

Masalan: **goto a;**

.....

a: y=x*x+c;

Bu yerda **goto a;** operatori bajarilgandan keyin boshqaruv to'g'ridan-tog'ri **a:y=x*x+c;**operatorga uzatiladi.

Shartli o'tish operatori.

C++ tilida shartli o'tish operatorining qisqa va to'liq ko'rinish-laridan foydalaniladi.

Qisqa shartli o'tish operatori. **Shartli operatorning qisqartirilgan ko'rinishi quyidagicha:**

if (ifoda) {s1;s2;...;sn};

Bu yerda **if** kalit so'zi bo'lib, **agar** degan ma'noni bildiradi hamda har doim lotin alifbosining kichik haraflari bilan yoziladi. Qavs ichida yozilgan **ifoda**arifmetik, mantiqiy, taqqoslash va boshqa amallardan tashkil topgan mantiqiy ifoda bo'lib, u **true** yoki **false** qiymatlarini qabul qiladi. **s1,s2,..sn** operatorlar ketma-ketligi.

Bu operator quyidagicha bajariladi: agar mantiqiy ifodaning qiymati rost (**true**) bo'lsa **{s1;s2;...;sn};**operatorlar ketma – ketligi bajariladi, aks holda keyingi qatorga o'tiladi.

1-misol.

$$y = \begin{cases} \cos x + 2x, & \text{agar } x < 3 \text{ bo'lsa} \\ \sqrt[4]{x^3}, & \text{agar } x \leq 3 \text{ bo'lsa} \end{cases}$$
 ifodani qiymatini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float x,y;
    cout << "x o'zgaruvchining qiymatini kiriting = ";
    cin >> x;
    if (x<3)    { y=cos(x)+2*x; goto a1;}
    y=pow(x,4/3);
    a1: cout << "Ifodaning qiymati: " << y;
    return 0;
```

Dasturda:

1) standart funksiyalardan foydalanilgani uchun qo'shimcha **#include <math.h>**chaqirildi;

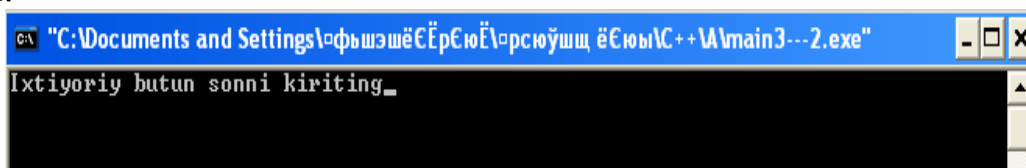
- 2) x va y o'zgaruvchilar haqiqiy turga mansub bo'lganligi uchun x o'zgaruvchiga klaviaturadan ixtiyoriy qiymat beriladi;
- 3) agar $x < 3$ dan kichik bo'lsa $y = \cos x + 2x$ funksiyasi hisoblanadi va jarayon **goto a1**; orqali natajani chiqarish operatoriga uzatiladi. Bu yerda $y = \cos x + 2x$; **goto a1**;lar birga kelgani uchun alohida { }belgilar ichiga olingan.
- 4) Agar $x < 3$ shart bajarilmasa u holda $y = \text{pow}(x, 4/3)$; operator bajariladi va natija chiqariladi.

2-misol. Berilgan sonning juft yoki toqligini aniqlash dasturini tuzing:

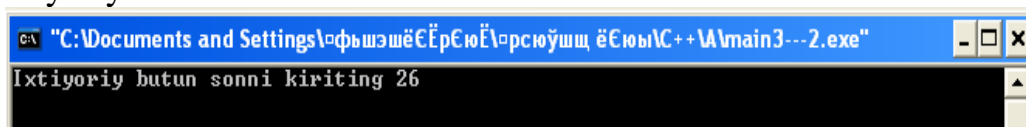
Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main ()
{ int n;
cout << "Ixtiyoriy butun sonni kiriting ";
cin>>n;
if(n%2 == 0) cout << "\n bu "<<n<<" soni juft "<<endl;
if (n%2!= 0) cout << "\n bu "<<n<<" soni toq "<<endl;
return 0;
}
```

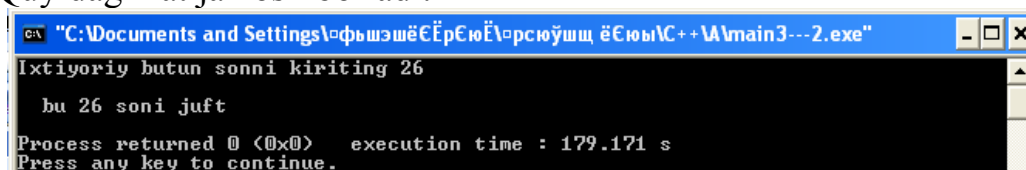
Dasturga izoh. Dastur bajarilganda ekranda quyidagicha yozuv hosil bo'ladi:



Ixtiyoriy butun son beramiz:



Quyidagi natija hosil bo'ladi:



C++ tilida dastur tuzish jarayonida goto operatoridan foydalanish uncha maqsadga muvofiq bo'lmaydi. Shuning uchun tarmoqlanish operatorining boshqa ko'rinishlaridan foydalaniladi.

To'liq shartli o'tish operatori. Shartli operatorning to'liq ko'rinishi quyidagicha:

if (ifoda) {s1;s2;...;sn;} else {p1;p2; ...;pm;}

Bu yerda **if** va **else** kalit so'zlar bo'lib, mos ravishda **agar** va **aks holdadegan** ma'noni bildiradi hamda ular hardoim lotin alifbosining kichik harflari bilan yoziladi. Qavs ichida yozilgan **ifoda** arifmetik, mantiqiy, taqqoslash va boshqa amal-

lardan tashkil topgan mantiqiy ifoda bo'lib, u **true** yoki **false** qiymatlarini qabul qiladi. **s1;s2;...;sn;** va **p1;p2; ...;pm** lar operatorlar ketma-ketligi.

Bu operator quyidagicha bajariladi: agar mantiqiy ifodaning qiymati rost (**true**) bo'lsa **{s1;s2;...;sn;}** operatorlar ketma – ketligi bajariladi, aks holda **{p1;p2; ...;pm;}** operatorlar ketma – ketligi bajariladi. Bu yerda bir nechta operatorlar birgalikda kelsa ular **{ }** belgilar ichiga olinadi.

Masalan.a) if (x<0) i=i+1 else k=k+1;

b) if (a>=0) y=sqrt(a+1) else y=pow(a,2);

1-misol.

$$y = \begin{cases} \cos x + 2x, & \text{agar } x < 3 \text{ bo'lsa} \\ \sqrt[4]{x^3}, & \text{agar } x \leq 3 \text{ bo'lsa} \end{cases} \quad \text{ifodani qiymatini hisoblash dasturini tuzing.}$$

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float x,y;
    cout << "x o'zgaruvchining qiymatini kiriting = ";
    cin >> x;
    if (x<3) y=cos(x)+2*x;
    else y=pow(x,4/3);
    cout << "Ifodaning qiymati: " << y;
    return 0;
}
```

2-misol. Berilgan uchta a,b,c haqiqiy sonlaridan eng kattasini topish dasturini tuzing.

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a,b,c,z,max;
    cout << "Ixtiyoriu 3 ta haqiqiy qiymatni kiriting: ";
    cin >> a>>b>>c;
    if (a>b)
        z=a; else z=b;
    if (z>c)
        max=z; else max=c;
    cout << "Sonning kattasi= " << max;
    return 0;
}
```

C++ tilida shartli o'tish operatoridan foydalanishda mantiqiy **or (&&)**, **and (| |)**, **not(!)** amallarini ham qo'llash mumkin:

a) if ((x<y) and (x<c)) z=2*x+y; else z=pow(x,3)+pow(y,4) ;

b)if ((a==0) or (b==0) or (c==0)) goto d5;

3-misol. $ax^2 + bx + c = 0$ ($a \neq 0, b \neq 0, c \neq 0$) kvadrat tenglamaning ildizlarini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
int main()
```

```
{
    float a,b,c,d,x1,x2;
    b:cout << "Kvadrat tenglama koefitsientlarini kiritng a<>0,b<>0,c<>0
bo'lsin= ";
    cin >> a>>b>>c;
    if ((a==0) or (b==0) or (c==0)) goto b;
    d=b*b-4*a*c;
    if (d<0)
    {
        cout<<"Haqiqiy yechim mavjud emas";
        return 0;
    }
    if (d>0)
    {x1=(-b+sqrt(d))/(2*a);
    x2=(-b-sqrt(d))/(2*a);
    cout << "Ikkita yechim mavjud: x1= " << x1<<" x2= " <<x2;
    } else cout<< "Karrali yechim mavjud: x1=x2=" <<x1;
    return 0;
}
```

Shartli o'tish operatorlari **ichma- ich joylashgan** ko'rinishda ham bo'lishi mumkin: *if (B1) S1; else if (B2) S2; else S3;* Bu yerda **B1,B2** - mantiqiy ifodalalar, **S1,S2,S3**- operatorlar. Bu operatorlarda *if (B2) S2; else S3;* ichki operator **B1** mantiqiy ifodaning qiymati *yolg'on(false)* bo'lsa bajariladi.

if (1-ifoda)

1-operatorlar ketma-ketligi;

else if (2-ifoda)

2-operatorlar ketma-ketligi; ...

else if (n-ifoda)

n-operatorlar ketma-ketligi;

else

(n+1)-operatorlar ketma-ketligi;

Ichma- ich joylashgan shartli operatorning boshqacha ko'rinishi quyidagicha: *if (B1) if (B2) S1; else S2; else S3;*

Ichki joylashgan *if (B2) S1; else S2;* operator **B1** ifoda *chin(true)* qiymatga ega bo'lsa bajariladi va buni C++ tilida quyidagicha, ya'ni har bir ichki va tashqi shartli

operatorlarni alohida { } belgilari ichiga yozish maqsadga muvofiq:

```
    if (B1)
{ if (B2) S1; else S2;}
    else S3;
```

Umumiy holda ichma- ich joylashgan shartli operatorni quyidagicha yozish mumkin:

if (1-ifoda)

1-operatorlar ketma-ketligi;

else if (2-ifoda)

2-operatorlar ketma-ketligi; ...

else if (n-ifoda)

n-operatorlar ketma-ketligi;

else

(n+1)-operatorlar ketma-ketligi;

Masalan: if (d or b) { if (x<0) x:=2+x;}

else { if (x<y) x:=x+0.5; else y:=y+0.5; }

Misol. Talabalarni test yakunlari bo'yicha baholash dasturini tuzing. (eng yuqori bal x=100).

Dastur.

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float x;
```

```
    cout << "Talabani balini kiriting = ";
```

```
    cin >> x;
```

```
    if (x >= 85) cout << "\ta'lo"; else
```

```
        { if ((x < 85) and (x >= 75)) cout << "\tyaxshi";
```

```
            else { if ((x < 75) and (x >= 55)) cout << "\tqoniqarli";
```

```
                else cout << "\tqoniqarsiz";
```

```
        }
```

```
        return 0;
```

```
}
```

Nazorat uchun savollar

1. Tarmoqlanuvchi jarayonni tushuntiring.
2. Tarmoqlanuvchi jarayonlarni dasturlash uchun qanday operatorlar ishlatiladi?
3. Shartli o'tish operatori qanday tuzilmalarga ega?
4. To'liq shartli o'tish operatorini bajarilishini tushuntiring.
5. Murakkab shartli o'tish operatorlari qanday bo'ladi?
6. Shartsiz va shartli o'tish operatorlarini vazifasini tushuntiring.
7. Tanlash operatorining vazifasi nimadan iborat va uning umumiy ko'rinishi?
8. Tanlash operatorining shartli operatoridan farqi qanday?
9. ? shart operatorini bajarilishini tushuntiring.

10. Nima uchun goto operatori ko'p ishlatilmaydi?

Mustaqil bajarish uchun topshiriqlar

1. a,b,c sonlarini o'sish tartibida joylashtirish dasturini tuzing.
2. a,b,c kesmalar uzunliklari bilan berilgan. Ular orqali uchburchak yasash mumkin yoki mumkin emasligini aniqlash dasturini tuzing.
3. Yuzasi S1 ga teng bo'lgan doira, yuzasi S2 ga teng bo'lgan kvadtaga joylashishini aniqlash dasturini tuzing .
4. Koordinatalari bilan berilgan a(x,y) nuqtani koordinatalar tekisligida joylashish o'rnini aniqlash dasturini tuzing.
5. Berilgan ikkita a va b sonlarining kichigini ularning yig'indisi, kattasini esa ularning ko'paytmasi bilan almashtirish dasturini tuzing.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu: ? shart operatori. Goto operatori va nishonlar. Tanlash operatori.

Reja:

1. ? shart operatori. Goto operatori va nishonlar.
2. Goto operatori va nishonlar.
3. Tanlash operatori
4. Misollar

Ma'ruza bayoni

? shart operatori.

? shart operatorining umumiy ko'rinishi quyidagicha:

X=B1 ? S1: S2;

bu yerda X-operatorni bajarilish natijasini qabul qiluvchi o'zgaruvchi; B1-mantiqiy ifoda bo'lib **chin(true)** yoki **yolg'on(false)** qiymatni qabul qiladi; S1-mantiqiy ifodaning qiymati true bo'lgan bajariladigan operator; S2-mantiqiy ifodaning qiymati **false** bo'lgan bajariladigan operator.

Amalning bajarilish tartibi shartli o'tish operatorining bajarilish tartibiga o'xshaydi. Ya'ni mantiqiy ifodaning qiymati chin yoki 0 dan farqli bo'lsa, birinchi operator, yolg'on (false) yoki 0 bo'lsa, ikkinchi operator bajariladi. Shartli o'tish operatoridan farqi shundaki, amal natijasi biror bir o'zgaruvchiga qiymat sifatida beradi.

Misol. Ixtiyoriy berilgan to'rtta haqiqiy sondan eng kattasini topish dasturini tuzing.

Dastur.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a,b,c,x,x1,x2,d;
    cout<<" ixtiyoriy 4 ta son kiriting ";
    cin>>a>>b>>c>>d;
    x=a>b?a:b;
```

```

    x1=c>d?c:d;
    x2=x>x1?x:x1;
cout<<"\t katta son ="<<x2;
    return 0;
}

```

Tanlash operatori.

Algoritmning xarakterli xususiyati shundan iboratki, ular uchun berilgan hisoblash jarayonlari ko'p tarmoqli bo'ladi. Bunday hisoblash jarayonlarini Paskal tilida yozishning usullaridan biri bizga ma'lum bo'lgan operatoridir. Agar hisoblash jarayonida qo'yilgan shartlar soni ikkitadan ortiq bo'lsa, u holda shartli operatoridan foydalanish ancha qiyinchilik tug'diradi. Ana shunday hollarda C++ tilida tanlash operatori qo'llaniladi.

Tanlash operatori Paskal tilida hosil qilinadigan operatorlardan hisoblanib, shartli operatorning umumlashmasidan iboratdir.

Tanlash operatori quyidagicha umumiy ko'rinishga ega :

```

switch (ifoda)
{
  case 1-o'zgarmas ifoda: 1-operatorlar ketmaketligi; break;
  case 2-o'zgarmas ifoda: 2-operatorlar ketmaketligi; break;
  .....
  case n-o'zgarmas ifoda: n-operatorlar ketmaketligi; break;
  default;
    operatorlar ketmaketligi;
}

```

Tanlash operatorida ifodaning qiymatiga qarab, dasturda berilgan operatorlar ketma-ketligidan birini tanlaydi.

break operatori dastur bajarilishini **switch** operatoriga taaluqli blokdan chiqarib yuboradi.

Agar ifodaning qiymati ro'yxatda berilgan ifodalarning qiymati bilan bir xil bo'lmasa **default** operatoridan keying operatorlar bajariladi va tanlash operatorini bajarilishi tugaydi.

Switch (x<=0)

```

{ case true: y:=sqrt(sqr(x)-x+2);cout<<y;break;
  case false: y:=ln(x+3);cout<<y;break;
  default:
    }

```

Masalan, 5 ballik tizimda o'quvchilar bahosini tahlil qilish dasturi.

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{
  int a,b,c,x,x1,x2,d;
  cout<<" bahoni kiriting ";
cin>>a;
  switch(a)

```

```

{
    case 1:cout<<"juda yomon"; break;
    case 2: cout<<"yomon"; break;
    case 3:cout<<"qoniqarli";break;
    case 4: cout<<"yaxshi"; break;
    case 5:cout<<"a'lo";break;
    default:
        cout<<"bunday baho yo'q";
}
return 0;
}

```

Tanlash operatorida ketma-ket bir nechta ifodaning qiymatlariga bitta bir xil operatorlar ketma-ketligini ko'rsatish ham mumkin.

Masalan, 10 ballik tizimda o'quvchilar bahosini tahlil qilish dasturi.

Dastur.

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int z;
    cout<<"Balni kiriting"
    cin>>z;
    switch (z)
    { case 1: case 2: case 3: cout <<"qoniqarsiz" ; break;
    case 4: case 5:cout <<"qoniqarli"; break;
    case 6: case 7: case 8: cout <<"yaxshi"; break;
    case 9: case 10: cout <<"a'lo" ; break;
    default: cout<<"bunday ball yo'q";
    }
return 0;

```

Misol. Quyidagi ifodani qiymatini hisoblash dasturini tuzing

$$y = \begin{cases} \sin x, \text{ agar } x = 10 \\ \cos x, \text{ agar } x = 2 \\ \text{tg } x, \text{ agar } x = 13 \\ \sqrt{|x|}, x \text{ ning boshqa qiymatlarida} \end{cases}$$

Dastur.

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{ int x; float y;
    cout << "x o'zgaruvchining qiymatini kiriting = ";
    cin >> x;

```

```

switch(x)
{
    case 10 : y=sin(x);break;
    case 2: y=cos(x); break;
    case 13 : y=tan(x); break;
    default: y=sqrt(fabs(x));
}
cout << "Ifodaning qiymati: " << y;
return 0;}

```

Nazorat uchun savollar

1. Tarmoqlanuvchi jarayonni tushuntiring.
2. Tarmoqlanuvchi jarayonlarni dasturlash uchun qanday operatorlar ishlatiladi?
3. Shartli o'tish operatori qanday tuzilmalarga ega?
4. To'liq shartli o'tish operatorini bajarilishini tushuntiring.
5. Murakkab shartli o'tish operatorlari qanday bo'ladi?
6. Shartsiz va shartli o'tish operatorlarini vazifasini tushuntiring.
7. Tanlash operatorining vazifasi nimadan iborat va uning umumiy ko'rinishi?

Mustaqil bajarish uchun topshiriqlar

1. a,b,c sonlarini o'sish tartibida joylashtirish dasturini tuzing.
2. a,b,c kesmalar uzunliklari bilan berilgan. Ular orqali uchburchak yasash mumkin yoki mumkin emasligini aniqlash dasturini tuzing.
3. Yuzasi S1 ga teng bo'lgan doira, yuzasi S2 ga teng bo'lgan kvadtaga joylashishini aniqlash dasturini tuzing .
4. Koordinatalari bilan berilgan a(x,y) nuqtani koordinatalar tekisligida joylashish o'rnini aniqlash dasturini tuzing.
5. Berilgan ikkita a va b sonlarining kichigini ularning yig'indisi, kattasini esa ularning ko'paytmasi bilan almashtirish dasturini tuzing.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu:Takrorlanuvchi jarayonlarni dasturlash. Sikl operatorlari. Parametrlilik sikl operatori.

Reja:

1. Sikllik jarayonlarni shartli operatorlar orqali dasturlash.
2. Sikl operatorlari. Parametrlilik sikl operatori.
3. Misollar.

Ma'ruza bayoni

Shunday hisoblash jarayonlari mavjud bo'ladiki, bunda bu jarayonning ayrim bo'laklarini bir necha marta takroran hisoblashga to'g'ri keladi. Bunday jarayonlar *takrorlanuvchi yoki siklli* jarayonlar deyiladi. Siklli jarayonlarni dasturlashda sikl operatorlaridan foydalaniladi, ya'ni dasturda bitta yoki bir nechta parametrlarning bir vaqtda o'zgarishida bir gurux operatorlarni hisoblash ko'p marotaba takrorlanadi. Masalaning qo'yilishiga qarab sikllar *takrorlanish sonima'lum bo'lgan* va *takrorlanish soni noma'lum* bo'lgan sikllarga bo'linadi.

Sikllik jarayonlarni shartli operatorlar orqali dasturlash.

Bunday holda sikl parametrining boshlang'ich qiymati berilgan bo'lishi kerak. Hisoblash shu parametr bilan bajariladi va qadam qiymati bilan o'zgartiriladi, hamda siklning tamom bo'lish sharti tekshiriladi. Agar sikl tugagan bo'lmasa, hisoblash davom ettiriladi.

Misol. $S = \sum_{i=1}^{20} \frac{i}{2} + 1$ ni hisoblash dasturini tuzing.

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int i; float S;
    S=0; i=0;
    a1: i=i+1;
    S=S+(i/2+1);
    if (i<=20) goto a1;
    cout << "Ifodaning qiymati: " << S;
    return 0;
}
```

Sikl operatorlari. Parametrli sikl operatori.

C++ tilida sikllarni tashkil qilish bir nechta sikl operatorlaridan foydalaniladi. Bu sikl operatorlarining 3 turi mavjud: *parametrli, sharti oldin berilgan va sharti keyin berilgan sikl operatorlar*. Ularni tartib bilan ko'rib chiqamiz.

Parametrli sikl operatori. Agar sikllik jarayonlarda takrorlanish soni oldindan ma'lum bo'lsa, ko'p hollarda parametrli sikl operatoridan foydalaniladi.

Parametrli sikl operatorining umumiy ko'rinishi quyidagicha:

```
for (x=x0; x<=xn; x=x+xqad)
    { 1-operator; 2-operator;...;k-operator;}
```

Bu yerda *for (uchun)* - xizmatchi so'z; *x- sikl parametri* deb ataluvchi o'zgaruvchi; x_0, x_n - sikl parametrining mos ravishda *boshlangich va oxirgi qiymatlari*; x_{qad} - *sikl parametrining o'zgarish qadami*; { 1-operator; 2-operator;...;k-operator; } - *sikl parametrning qabul qiladigan har bir qiymatida takroran bajariladigan operatorlar*. Agarbu operatorlar bittadan ko'p bo'lsa { } belgi ichiga olinadi, aks holda shart emas.

Sikl parametri o'zining boshlang'ich qiymatini oladi va oxirgi qiymati bilan solishtiriladi. Agar $x \leq x_n$ mantiqiy ifodaning qiymati true yoki 0 dan farqli bo'lsa operatorlar bir marta bajarilib, sikl parametri x_{qad} qadamga ortadi va yana $x \leq x_n$ shart tekshiriladi va hakazo bu jarayon $x > x_n$ bo'lguncha davom etadi. Boshqacha aytganda, parametrli sikl operatorida sikl parametri *x boshlang'ich qiymat x₀ dan oxirgi qiymat x_n gacha* bo'lgan qiymatlarni x_{qad} bilan o'zgarishda qabul qiladi. Uning har bir qiymatida *operatorlar* ketma-ketligi bajariladi

Masalan: *for (n=0; n<=5; n=n+1;)*

$y=pow(2,n);$

Bu yerda sikl parametri n besh marta takrorlanadi va o'zgaruvchi y ketma - ket $2^0, 2^1, 2^2, 2^3, 2^4, 2^5$ qiymatlarni qabul qiladi, $n=5$ bo'lganda hisoblash to'xtatiladi.

Misol. 1 dan n gacha bo'lgan natural sonlar yig'indisi va ko'paytmasini hisoblash dasturini tuzing.

```
#include <iostream.h>
using namespace std;
int main ( )
{
    int n,i; float s,p;
    s=0; p=1;
    cout<<"Ixtiyoriy butun son kiritng";
    cin>>n;
    for (i=1; i<= n; i=i+1)
    { s=s+i; p=p*i;}
    cout << "yig'indi =" <<s<< "\n ko'paytma="<<p;
    return 0;
}
```

Parametrlilikl operatorlarida **inkrement** (+=, -=, *=, ++, **, --) amallaridan foydalanish mumki. Yuqoridagi dasturni quyidagicha ham yozish mumkin:

```
#include <iostream.h>
using namespace std;
int main ( )
{
    int n,i; float s,p;
    s=0; p=1;
    cout<<"Ixtiyoriy butun son kiritng";
    cin>>n;
    for (i=1; i<= n; i++)
    { s+=i; p*=i;}
    cout << "yig'indi =" <<s<< "\n ko'paytma="<<p;
    return 0;
}
```

1-misol. $S = \sum_{i=1}^n (2i + 1)$ yig'indini hisoblash dasturini tuzing.

```
#include <iostream.h>
using namespace std;
int main ( )
{
    int n,i; float s;
    s=0;
    cout<<"Ixtiyoriy butun son kiritng";
    cin>>n;
```

```

for (i=1; i<= n; i=i+1)
s=s+(2*i+1);
cout << "yig'indi =" <<s;
return 0;
}

```

2-misol. Kiritilgan sonlarning eng kattasini topish dasturini tuzing.

Dastur.

```

#include <iostream.h>
#include <math.h>
int main ()
{
    int i, n, a, max;
    cout<<("kiritilayotgan sonlar soni: ");
    cin>>n;
    cout<< n<<'\n';
    cin>>max;
    for(i = 1; i < n ; i++)
    {
        cin>>a;
        if(a > max)
            max = a;
    }
    cout<< max;
    return 0;
}

```

Sharti oldin qo'yilgan sikl (while) operatori.

Sharti oldin qo'yilgan sikl (**while**) operatorining umumiy ko'rinishi quyidagicha: **while B {s1;s2;...;sn}**- bu yerda **while** (toki) xizmatchi so'z, **B**- mantiqiy ifoda, **{s1;s2;...;sn}**-operatorlar.

Bu operatorida **{s1;s2;...;sn}** operatorlar ketma-ketligi bajarilmasligi yoki bir martadan ortiq bajarilishi mumkin, chunki **{s1;s2;...;sn}** operatorlarning har bir bajarilishidan oldin **B** mantiqiy ifodaning qiymati tekshiriladi. Agar ifoda chin (true) qiymat qabul qilsa **{s1;s2;...;sn}** operatorlar bajariladi, aks holda, ya'ni **B** yolg'on (false) qiymat qabul qilsa, sikl operatorining bajarilishi tugaydi.

Agar **B** mantiqiy ifoda birinchi hisoblanishidayoq *yolg'on qiymatni* qabul qilsa, **{s1;s2;...;sn}** operatorlar bir marta ham bajarilmaydi. Ko'rinib turibdiki, bu yerda siklik jarayonning tugallanish sharti **{s1;s2;...;sn}** operatorlar bajarilishidan oldin tekshiriladi. Shuning uchun bunday ko'rinishdagi sikl operatori - *sharti oldin qo'yilgan sikl operatori* deb yuritiladi. Bu yerda ham takrorlanadigan operatorlar bittadan ortiq bo'lsa ular { } belgi ichiga olinadi, aks holda shart emas.

Masalan: 1) $y:=1; k:=0; m=5; \text{while } y \leq m \{ y:=y+3; k:=k+; \}$

2) $x:=3; y:=1; v:=1; \text{eps}:=0,0001;$

$\text{while } \text{abs}(v) \geq \text{eps} \{ V:=(x/y-y); y:=y+v; \}$

1-misol. 1 dan n gacha bo'lgan natural sonlar yig'indisi va ko'paytmasini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream.h>
using namespace std;
int main ( )
{
    int n,i; float s,p;
    s=0; p=1;i=1;
    cout<<"Ixtiyoriy butun son kiritng";
    cin>>n;
    while (i<= n)
    { s=s+i; p=p*i;i=i+1;}
    cout << "yig'indi =" <<s<< "\n ko'paytma="<<p;
    return 0;
}
```

2-misoll. $S = \sum_{i=1}^n (2i + 1)$ yig'indini hisoblash dasturini tuzing.

Dastur.

```
#include <iostream.h>
using namespace std;
int main ( )
{
    int n,i; float s;
    s=0;
    cout<<"Ixtiyoriy butun son kiritng";
    cin>>n;
    while (i<= n)
    { s=s+(2*i+1);i=i+1;}
    cout << "yig'indi =" <<s;
    return 0;
}
```

Bu misolda qaysi operatoridan foydalanish - **for** yoki **while** danmi farq qilmaydi. Agar takrorlanish soni ma'lum bo'lsa, **for** operatori ishlatiladi, takrorlanish soni ma'lum bo'lmasa **while** operatori qo'llaniladi.

Sharti keyin qo'yilgan (do..while) sikl operatori.

Bu operator takrorlanuvchi hisoblash jarayonlarida takrorlanishlar soni ma'lum, noma'lum bo'lgan holatlarda ishlatiladi. Operatorning umumiy ko'rinishi quyidagicha Sharti oldin qo'yilgan sikl (**while**) operatorining umumiy ko'rinishi quyidagicha: **do {s1;s2;...;sn ;} while B** bu yerda **while** (toki), **do**(bajar)- xizmatchi so'zlar, **B**- mantiqiy ifoda, {s1;s2;...;sn ;} -operatorlar ketma-ketligi..

Bu operatorda {s1;s2;...;sn ;} operatorlar ketma-ketligi hech bo'lmaganda bir marta va undan ortiq bajarilishi mumkin, chunki {s1;s2;...;sn ;} operatorlarr bajarilgandan keyin **B** mantiqiy ifodaning qiymati tekshiriladi. Agar ifoda chin (true) qiymat qabul qilsa {s1;s2;...;sn;} operatorlar bajariladi, aks holda, ya'ni **B** yolg'on (false) qiymat qabul qilsa, sikl operatorining bajarilishi tugaydi.

Bu operatorlarda siklik jarayonning tugallanish sharti $\{s_1; s_2; \dots; s_n\}$ operatorlar bajarilgandan keyin tekshiriladi. Shuning uchun bunday ko'rinishdagi sikl operatori - *sharti keyin qo'yilgan sikl operatori ham* deb yuritiladi. Bu yerda ham takrorlanadigan operatorlar bittadan ortiq bo'lsa ular $\{ \}$ belgi ichiga olinadi

Operatorning bajarilishida *do...while* operatorlari orasida joylashgan operatorlar ketma- ketligi bir marta va undan ortiq bajarilishi mumkin. Bu jarayon qo'yilgan mantiqiy ifoda *yolg'on qiymat* qabul qilgunga qadar davom etadi.

do..while operatorining *while* operatoridan farqi shundaki, bunda eng avval operatorning qiymati hisoblanib, so'ngra qo'yilgan shart tekshiriladi. Bu esa bunda berilgan operatorning hech bo'lmaganda bir marta bajarilishiga imkon yarata-di.

Masalan.

$y:=1;$

do

$v:=(2*x/y-3*y)/2; y:=y+2*v$

while(abs(v)<eps);

1-misol. 1 dan n gacha bo'lgan natural sonlar yig'indisi va ko'paytmasini hisoblash dasturini tuzing.

Dastur.

```
# include <iostream.h>
```

```
using namespace std;
```

```
int main ( )
```

```
{
```

```
int n,i; float s,p;
```

```
s=0; p=1;i=1;
```

```
cout<<"Ixtiyoriy butun son kiritng";
```

```
cin>>n;
```

```
do
```

```
{s=s+i; p=p*i; i=i+1;}
```

```
while (i<= n);
```

```
cout << "yig'indi =" <<s<< "\n ko'paytma=" <<p;
```

```
return 0;
```

```
}
```

2-misoll. $S = \sum_{i=1}^n (2i+1)$ yig'indini hisoblash dasturini tuzing.

Dastur.

```
# include <iostream.h>
```

```
using namespace std;
```

```
int main ( )
```

```
{
```

```
int n,i; float s;
```

```
s=0;
```

```
cout<<"Ixtiyoriy butun son kiritng";
```

```
cin>>n;
```

```
do
```

```

{s=s+(2*i+1);i=i+1;}
while (i<= n);
cout << "yig'indi =" <<s;
return 0;
}

```

Takrorlanuvchi jarayonlarda takrorlanishlar soni ma'lum bo'lmagan holda ham **do...while** operatoridan foydalanish mumkin.

3- misol. $S = \sum_{i=1}^{\infty} \frac{1}{2^i}$ yig'indini ε aniqlikda hisoblash dasturini tuzing.

```

#include <iostream.h>
using namespace std;
int main ()
{
int n,i; float s,eps;
s=0; i=1;
cout<<"Ixtiyoriy juda kichik son kiriting";
cin>>eps;
do
{s=s+1./(2*i);i=i+1;}
while (1./(2*i)>eps);
cout << "yig'indi =" <<s;
return 0;
}

```

Sinov savollari

1. Takrorlanuvchi jarayonni tushuntiring.
2. Qanday takrorlash operatorini bilasiz?
3. while, do-while operatorlari qanday ishlaydi?
4. For operatoridan foydalanish tartibini tushuntiring.
5. Boshqaruv operatorlari qayerda ishlatiladi?

Uyga vazifa : C++ tizimida for, while, do-while, break, continue operatorlari yordamida takrorlanuvchi dastur tuzish qoidalarini mukammal o'rganish va ularga doir misollar keltirish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu: C++ da massivlar. Massivlarni ta'riflash. Bir o'lchovli massivlar.

Reja:

1. Massivlarni ta'riflash.
2. Bir o'lchovli massivlar.
3. Misollar.

Ma'ruza bayoni

Massivlarni ta'riflash. Bir o'lchovli massivlar.

Xotirada ketma-ket joylashgan bir xil turdagi qiymatlarga massiv deyiladi. Massiv –bu fiksirlangan miqdordagi ayrim elementlarning tartiblangan

majmuasidir. *Massiv* deganda bir xil turga mansub bo'lgan obyektlarning biror xususiyatiga ko'ra mujassamlangan majmuasi yoki to'plami tushuniladi

Massiv –bu ma'lumotlarning murakkab turi bo'lib, quyidagilar bilan xarakterlanadi:

- massiv elementlari boshqa tuzilmalardan farqli ravishda bir xil turga tegishli bo'ladi, shuning uchu uning har bir elementi bir xil xotira hajmini egallaydi;
- massiv fayllar kabi tashqi qurilmalarda emas, balki tezkor xotirada joylashadi;
- massiv elementlari ketma-ket keluvchi yacheykalarni egallaydi.

Barcha elementlar bir xil turda bo'lishi kerak va bu tur element turi yoki massiv uchun tayanch deb nomlanadi.

Massiv elementlariga murojoat qilish ikki xil usulda amalga oshiriladi.

1. Massiv elementining tartib raqami orqali va bu indeks deb yuritiladi. Bu juda sodda va boshqa Programmalash asoslarini biladiganlar uchun oddiy hol. Indeks sifatida butun sonlardan, butun turdagi o'zgaruvchilardan foydalanish mumkin. Indeks sifatida haqiqiy turdan foydalanib bo'lmayda.

2. C++ tilida massiv elementlarini qayta ishlashda ko'rsatkichlardan, ya'ni adreslardan foydalanish mumkin, chunki massiv va ko'rsatkichlar orasida aloqa mavjud. Shunga qaramay, birinchi usulda adreslar bilan ishlash uchun maxsus tur qatnashmaydi, ko'rsatkichlar baribir ishlatiladi..

Massivlarni tavsiflash uchun regulyar (muntazam) turlardan foydalaniladi, ya'ni ular bir xil turdagi ma'lumotlarning chekli to'plamini tavsiflash uchun ishlatiladi. Massivlarni tashkil etgan obyektlar uning elementlari bo'lib, ular tartib bilan nomerlanadi ya'ni o'z indeksiga ega bo'lib, ular ana shu indekslarning nomeri bo'yicha tartiblanadi hamda murojoat qilinadi.

Massiv bir, ikki va ko'p o'lchovli bo'lishi mumkin. Bir o'lchovli massivlar elementlari satr yoki ustundan iborat bo'lgan vektorlarni tashkil etadi. Ikki o'lchovli massivlar matrisalardan iborat bo'ladi. Uch va undan katta o'lchovli massivlar amaliyotda kamda-kam hollarda ishlatiladi, chunki ulardan foydalanish uchun katta hajmdagi tezkor xotira kerak bo'ladi.

Dasturlash jarayonida ishlatiladigan har bir massiv o'zining alohida nomiga ega bo'lishi kerak. Massivdan foydalanish uchun dastur boshida unga tavsif beriladi. Massivlarni tavsiflashda uning turi, nomi va indekslar chegarasi ko'rsatiladi.

C++ tilida massiv elementlarining indeksleri 0 dan boshlanadi va uning eng katta qiymati massiv e'lonidagi uzunlikdan bitta kam bo'ladi

Bir o'lchovli massivlar quyidagicha tavsiflanadi:

tur nom [<uzunlik>];

bu yerda **tur**-massiv elementlarining turi bo'lib, u **int**, **float**, **char** kabi ma'lumotlar turidan iborat bo'lishi mumkin; **nom** – massiv nomi bo'lib foydalanuvchi tomonida ixtiyoriy aniqlanadi va identifikarlar qoidasi bo'yicha yoziladi; **uzunlik** – butun sonidan iborat bo'lib, massivni tashkil etgan elementlar soni, yoki elementlar joyshadigan yacheykalar sonini aniqlaydi.

Masalan: 1) **float A[10];**

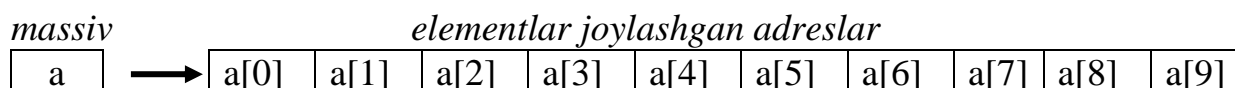
2) **n=10; float A[n];**

3) **int a[10] = {9,8,-8,-6,-5,5,4,3,12,14}**

Massiv indeksli o'zgaruvchi tushunchasiga mos keladi. Misol uchun **long int a[5]; char w[200];double f[4][5][7]; char[7][200]**. Bir o'lchovli massivlarga matematikada vektor tushunchasi mos keladi. Massivning **int z[3]** shakldagi ta'rifi, **int** turiiga tegishli z[0],z[1],z[2] elementlardan iborat massivni aniqlaydi.

Massivlar statik va dinamik ko'rinishlarda berilishi mumkin. Statik massivning uzunligi oldindan ma'lum bo'lib, u xotirada ketma-ket joylashadi. Dinamik massivni uzunligi dastur bajarilishi jarayonida aniqlanib, u dinamik xotiradagi ayni paytda bo'sh bo'lgan adresga joylashadi.

Masalan, **int a[10]** ko'rinishda e'lon qilingan statik massiv elementlari xotirada quyidagicha joylashadi:



Massiv elementiga murojoat qilish **massiv nomi[indeks]** korinishida amalga oshiriladi. Masalan, a[i] yoki a[2]; a[5].

int a[10] = {9,8,-8,-6,-5,5,4,3,12,14}; ko'rinishda e'lon qilingan massivda murojoat amalga oshirilsa, u holda **a[0]=9; a[3]=-6; a[8]=12;**

Massivlar tavsiflanganda to'liq, to'liqmas ko'rinishlarda inisializatsiya qilinishi, ya'ni boshlangich qiymatlarlari ko'rsatilishi mumkin:

1) o'lchami ko'rsatilgan massivni to'liq inisialasatsiya qilish:

int a[10] = {9,8,-8,-6,-5,5,4,3,12,14};

Bu yerda 10 ta elementdan iborat a massiv e'lon qilingan va uning barcha elementlariga boshlang'ich qiymat berilgan.

2). Agar massiv inisializatsiya qilinganda elementlar chegarasi ko'rsatilgan bo'lsa, ro'yxatdagi elementlar soni bu chegaradan kam bo'lishi mumkin, lekin ortiq bo'lishi mumkin emas o'lchami ko'rsatilgan massivni to'liqmas inisialazatsiya qilish:

int a[10] = {9,8,-8,- 3,12,14};

Bu yerda massiv boshidagi 6 ta elementga qiymat berilgan. Shuni ta'kidlash lozimki, massivning boshidagi va o'rtasidagi elementlarga qiymat bermasdan uning oxirgi elementlariga qiymat berib bo'lmaydi.

Misol uchun **int A[5]={2,-2}**. Bu holda a[0] va a[1] qiymatlari aniqlangan bo'lib, mos holda 2 va -2 ga teng.

3) O'lchami ko'rsatilmagan massiv elementlarini to'liq inisializatsiya qilish:

floatC[]={1,-1,2,10,-12.5};

Bu yerda massiv chegarasi avtomatik aniqlanadi hamda massivni barcha elementlariga qiymatlar berilgan hisoblanadi.

Massiv elentlarini kiritish

Massiv elementlarini quyidagi ko'rinishlarda berish mumkin:

1) Massiv elementlarini tavsiflash vaqtida:

a) **int a[10] = {9,8,-8,-6,-5,5,4,3,12,14}**

b) **intn=20; a[n] = {9,8,-8,-6,-5,5,4,3,12,14}**

c) **char T[11]={‘м’,‘а’,‘т’,‘е’,‘м’,‘а’,‘т’,‘и’,‘к’,‘а’};**

d) **char T[11]=“математика”;**

Bunda agar massiv elementlari soni ko'rsatilgan chegaradan kam bo'lsa, uning qolgan elementlari 0 qiymatni qabul qiladi. Agar massiv elementlari soni ko'rsatilgan chegaradan ko'p bo'lsa, u kompilyatrga uchun tushunarsiz bo'ladi.

2) Massiv elementlarini klaviaturadan kiritish. Buning uchun sikl oeratorlaridan foydalaniladi. Masalan,

```
for (i=0; i<=9; i++)
    cin>>a[i];
```

Bu yerda **a[i]** massivning **a[0]** dan **a[9]** gacha elementlariga qiymat beriladi.

```
for (i=1; i<=10; i++)
    cin>>a[i];
```

Bu holda **a[i]** massivning **a[1]** dan **a[10]** gacha elementlariga qiymat beriladi. **a[0]** element o qiymatga ega bo'ladi

3) Ayrim hollarda massiv elementlarini tasodifiy sonlar datchgi orqali ham berish mumkin:

```
const n=5; int a[n]; randomize();
for (int i=0; i<n; i++)
    a[i]=random(100);
```

randomize(); funksiyasi massiv elementlari tasodifiy bo'lishini amalga oshiradi. **random(100);** funksiyaparametri [0..99] oraliqdagi butun sonlarni tasodiy ko'rinishda hosil qiladi. Agar massiv nafaqat musbat , balki manfiy butun sonlardan iborat bo'lishi kerak bo'lsa, **a[i]= random(100)-20;**Bu yerdasonlar [-20..79] oralig'ida aniqlanadi.

Agar massiv elementlari tasodifiy haqiqiy sonlardan iborat bo'lishi kerak bo'lsa, **a[i] = random(5)/10.-0.25;** ko'rinishda yoziladi.

4) Massiv elementlari ba'zi bir qoidalar bo'yicha ham hosil qilish mumkin:

```
for ( int i=0; i<n; i++)
if (j%2) A[i]= i*10;
    else A[i]=i/100.;
```

Massiv elementlarini chiqarish

Uncha katta bo'lmagan massivlar elementlarini quyidagicha chop etish mumkin:

```
cout<< "\n Массив: \n ";
for (int i=0; i<n; i++)
    cout<<A[i]<<" ";
    cout<<endl;
```

Katta hajmdagi massivlar elementlarini quyidagicha chop etish mumkin:

```
int K=10;
for (int i=0; i< n; i++)
{ cout<<A[i]<<" ";
    if ((i+1) %K == 0) cout<<endl;
}
```

Misol. 10 ta elementdan iborat a[i] massiv berilgan:

1) Shu massiv elementlari yig'indisi va ko'paytmasini hisoblang:

```

#include <iostream.h>
#include <conio.h>
int main()
{int a[10]={8,-9,6,-5,-3,12,-13,16,17,-18}, i;
  float s,p;;
  s=0;p=1;
  for (i=0;i<=9;i++)
  { s=s+a[i]; p=p*a[i];
  }
  cout << "s="<<s<<"\t"<<"p="<<p;
  getch();
return 0;
}

```

2. Shu massiv elementlarining eng kattasi va kichigini toping:

```

#include <iostream.h>
#include <conio.h>
int main()
{int a[10]={8,-9,6,-5,-3,12,-13,16,17,-18}, i;
  int max,min;
max=a[0]; min=a[0];
  for (i=0;i<=9;i++)
  { if (a[i]>max) max=a[i];
if (a[i]<min) min=a[i];
  }
  cout << "max="<<max<<"\t"<<"min="<<min;
  getch();
return 0;
}

```

Sinov savollari:

1. Massiv deb nimaga aytiladi?
2. C/C++ tilida massivlarni qanday ifoda etiladi?
3. o'lchovli massiv elementlarini kiritish va chiqarish usullari.
4. Massiv elementlariga son qiymat berishda tasodifiy qiymatlardan foydalanish

Uyga vazifa : C++ tizimida massiv elementlari bilan ishlash qoidalarini mukammal o'rganish va ularni amaloyotga tatbiq etish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu: Ikki o'lchovli massivlar. Jadvallar bilan ishlash. Belgili va satrli massivlar.

Reja:

1. Ikki o'lchovli massivlar.
2. Jadvallar bilan ishlash.
3. Belgili va so'zlar massivlar.
4. Ko'rsatkichlar massivlari.

Ma'ruza bayoni

Ikki o'lchovli massivlar. Jadvallar bilan ishlash

Ikki o'lchovli massivlar matematikada matrisa yoki jadval tushunchasiga mos keladi. Jadvallarning insializasiya qilish qoidasi, ikki o'lchovli massivning elementlari massivlardan iborat bo'lgan bir o'lchovli massiv ta'rifiga asoslangandir. Aytaylik $a(n,m)$ o'lchovli matrisa berilgan:

$$A = \begin{vmatrix} a_{00} & a_{01} & \dots & a_{0n} \\ a_{10} & a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{m0} & a_{m1} & \dots & a_{mn} \end{vmatrix}$$

Matrisa elementlari belgilash uchun ikki indeksli o'zgaruvchi ishlatiladi, $a[n][m]$, bu yerda n -satrlar soni, m -ustunlar soni.

Matrisa quyidagicha e'lon qilingan bo'lishi mumkin:

const n=5, m=5; int A[n][m],

bu yerda n - satrlar soni (birinchi, chap o'lchov), m – ustunlar soni yoki satrdagi elementlar soni (ikkinchi o'ng o'lchov). Matrisa tezkor xotirada satr bo'yicha joylashadi va uzluksiz joyni egallaydi, hamda ining hajmi $n*m*sizeof(int)$ ga teng bo'ladi, bu yerda int – matrisa elementlarining turi.

Matrisani tavsiflashning u yoki bu usulu konkret masalaning turiga qarab belgilanishi ham mumkin. Matrisa satr va ustunlarni nomerlash 0 dan boshlanadi.

Matrisani inisialaziyalash uni tavsiflash vaqtida amalga oshirilishi mumkin, masalan:

$$a = \begin{vmatrix} 1 & -2 & 3 \\ 10 & 20 & 33 \\ -11 & 22 & 36 \end{vmatrix}$$

`int A[3][3]= {{1, -2, 3}, { 10, 20, 33}, {-11, 22, 36}};`

Agar satrda talab qilinganidan kam sondagi element berilgan bo'lsa, u holda qolganlari 0 bilan inisialaziya qilinadi:

`int A[5][3]= {{1, -2, 3, -4, 5}, { 10, 20, 33, -40}, {-11, 22, 300, 400, 500}};`

Bu misolda 2-satrning oxirxi elementi o'gat eng bo'ladi.

Misol uchun ikki qator va uch ustundan iborat bo'lgan xaqiqiy turga tegishli d massiv boshlang'ich qiymatlari quyidagicha ko'rsatilishi mumkin:

```
float a[2][3]={(1,-2.5,10),(-5.3,2,14)};
```

Bu yozuv quyidagi qiymat berish operatorlariga mosdir:

```
a[0][0]=1;a[0][1]=-2.5;a[0][2]=10;a[1][0]=-5.3;a[1][1]=2;a[1][2]=14;
```

Bu qiymatlarni bitta ro'yxat bilan hosil qilish mumkin:

```
float a[2][3]={1,-2.5,10,-5.3,2,14};
```

Inisializasiya yordamida boshlang'ich qiymatlar aniqlanganda massivning hamma elementlariga qiymat berish shart emas. Misol uchun: `int x[3][3]={(1,-2,3),(1,2),(-4)}`. Bu yozuv quyidagi qiymat berish operatorlariga mosdir: `x[0][0]=1;x[0][1]=-2;x[0][2]=3;x[1][0]=-1;x[1][1]=2;x[2][0]=-4;`

Inisializasiya yordamida boshlang'ich qiymatlar aniqlanganda massivning birinchi indeksi chegarasi ko'rsatilishi shart emas, lekin qolgan indekslar chegaralari ko'rsatilishi shart. Misol uchun:

```
Double x[] [2]={(1.1,1.5),(-1.6,2.5),(3,-4)}
```

Bu misolda avtomatik ravishda qatorlar soni uchga teng deb olinadi.

Ikki o'lchovli massiv elementlarini kiritish

Oddiy ko'rinishda ikki o'lchovli massiv elementlarini kiritish quyidagicha bajariladi:

```
for ( i=0; i<n; i++)
```

```
    for ( j=0; j<m; j++)
```

```
        cin>>A[i][j];
```

Bunda har bir satrda bitta sonni teramiz, lekin qaysi nomerdagi elementni kiritayotganimiz ko'rinmaydi. Shuning kiritlayotgan elementni ekranda ko'rib turish uchun quyidagicha yozishimiz mumkin:

```
# include <iostream.h>
```

```
using namespace std;
```

```
int main ( )
```

```
{int a[3][3],j,i, m=3, n=3; float s,eps;
```

```
for ( i=0; i<n; i++)
```

```
    { for ( j=0; j<m; j++)
```

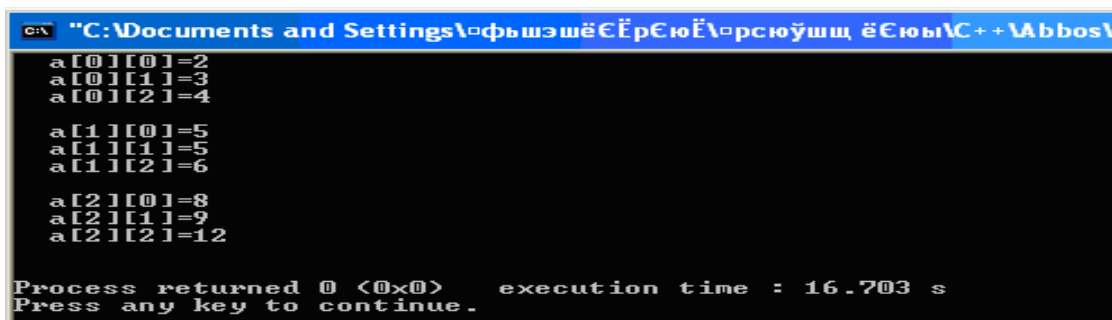
```
        { cout<<"a["<<i<<"]"<<["<<j<<"]"<<"]=";
```

```
          cin>>a[i][j];} cout<<"\n";
```

```
    }
```

```
return 0;
```

```
}
```



```
g++ "C:\Documents and Settings\ффышэщёёёрсюё\орсюўшщ ёёюы\С++\Аббос\
a[0][0]=2
a[0][1]=3
a[0][2]=4

a[1][0]=5
a[1][1]=5
a[1][2]=6

a[2][0]=8
a[2][1]=9
a[2][2]=12

Process returned 0 (0x0)   execution time : 16.703 s
Press any key to continue.
```

1-misol.Quyidagi ko'radigan misolimizda jadval kiritilib xar bir qatorning maksimal elementi aniqlanadi va bu elementlar orasida eng kichigi aniqlanadi:

```
#include <iostream.h> void main()
{ double a[4,3]; double s,max=0.0,min=0.0; int i,j;
for(i=0;i<4;i++) { for(j=0;j<3;j++)
{ Cout<<"a ["<i<<" ]["<j<<" ] ="; Cin>>s; a[i,j ] =s;
if (max<s) max=s; };
Cout<<"\n";
if (max<min) min=max; } Cout<<"\n min=",min; }
```

Matrisa elementlarini tasodifiy sonlar orqali aniqlash

Matrisaelementlarinikiritishuchunketadiganvaqtnitejashmaqsadida-ba'zimasalalarniyechishuchunmassivelementlarinitasodifiysonlargeneratoriorqalia niqlashmumkin:

```
for ( i=0; i<n; i++)
for ( j=0; j<m; j++)
A[i][j]=rand();
```

Agar masalaga qo'yilgan shart bo'yicha, matrisa elementlari [a,b) oraliqqa tegishli bo'lishi kerak bo'lsa, bu yerda $a > b$, u holda sikl operatorida $A[i][j] = \text{random}(b-a)+a$ yozish kerak. Bu usuldan matrisa elementlari uchun cheklov qo'yilmagan paytda foydalanish mumkin

Massiv elementlariga son qiymat berishda kompyuter xotirasidagi tasodifiy butun sonlardan foydalanish xam mumkin. Buning uchun standart kutubxonaning **rand ()** funksiyasini ishga tushirish kerak. **rand ()** funksiyasi yerdamida $0 \wedge 32767$ oralikdagi ixtiyoriy sonlarni olish mumkin. Bu qiymatlar umuman tasodifiydir. (**psevdo** - tasodifiy degani). Agar dastur qayta-qayta ishlatilsa, ayti tasodifiy qiymatlar takrorlanaveradi. Ularni yangi tasodifiy qiymatlar qilish uchun **srand ()** funksiyasini dasturda bir marta e'lon qilish kerak. Dastur ishlashi jarayonida ex-tiyojga karab **rand ()** funksiyasi chakirilaveradi. Tasodifiy qiymatlar bilan ishlash uchun **<stdlib.h>** faylini e'lon qilish zarur.

Masalan, [1,100) oraliqdaa(3,3) massivelementlarinihosilqilish-vachopetishquyidagichabajariladi:

```
# include <iostream.h>
# include <conio.h>
# include <stdlib.h>
#include <time.h>
int main()
{ int n=3,m=3, a[n][m],i,j;
for ( i=0; i<n; i++)
for ( j=0; j<m; j++)
a[i][j]=1+rand()%100;
for ( i=0; i<n; i++)
{
for ( j=0; j<m; j++)
cout<<" "<<a[i][j]<<" ";
```

```

    cout<<"\n";
}
return 0;
}

```

```

C:\Documents and Settings\АфшэшәЕрЄюЁ\орсюшщ ёЄюыА+
42 68 35
1 70 25
79 59 63
Process returned 0 (0x0) execution time : 0.109 s
Press any key to continue.

```

srand() funksiyasidagi qiymatni avtomatik ravishda uzgaradigan xolatga keltirish uchun **srand(time (NULL))** yozish ma'kul, shunda kompyuter ichidagi soatning qiymati **time()** funksiyasi yordamida urnatiladi va **srand**ga parametr sifatida beriladi. **NULL** yoki **0** deb yozilsa, qiymat sekundlar ko'rinishida beriladi. Vaqt bilan ishlash uchun **<time.h>**ni e'lon qilish kerak.

```

#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h> void main ( )
{ srand ( time (0)); int a[5], b[5], i;
for (i = 0; i < 5; i++) a[i] = rand ( ); for (i = 0; i < 5; i++) { b[i] =
a[i] + 64; cout <<"b="<<b[i]<<endl; } getch ( ); }

```

Izox: tasodifiy sonlar ichida manfiy sonlarning ham qatnashishini ixtiyor etsak,

a[i] = 1050 - rand (); yoki **a[i] = rand ()-1000;** deb yozish ham mumkin

Matrisa elementlarini ba'zi bir qonuniyatlar orqali ham aniqlash mumkin, masalan:

```

for ( i=0 ; i<n; i++)
for ( j=0; j<m; j++)
    A[i][j]=( i+1)*(j +1);

```

Bunda ko'rinishda aniqlangan matrisalarni ba'zi masalarni yechishda tahlil qilish osonroq.

Matrisa bunday usullarda aniqlanilishi dastur bajarilish jarayonida matrisala elementlarini qayta-qayta kiritish uchun vaqt sarflashni oldini oladi,

Ikki o'lchovli massiv elementlarini chiqarish

Oddiy ko'rinishda ikki o'lchovli massiv elementlarini kiritish quyidagicha bajariladi:

```

for ( i=0; i<n; i++)
    for ( j=0; j<m; j++)
        cout<<a[i][j];

```

Massiv elementlarini chiqarish bunday ko'rinishda berilsa, uning hamma elementlari bir satrga joylashadi:

```

# include <iostream.h>
using namespace std;
int main ( )
{int a[3][3]={1, 2 ,3 ,4 ,5 ,6, 7 ,8, 9}, j,i, m=3, n=3; float s,eps;

```

```

for ( i=0; i<n; i++) for ( j=0; j<m; j++)
cout<<a[i][j]<<" ";
return 0;}

```

Dastur natijasi:

```

C:\ "C:\Documents and Settings\...
1 2 3 4 5 6 7 8 9
Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.

```

Ikki o'Ichovli massiv elementlarini satr va ustunlar bo'yicha joylashgan matrisa ko'rinishida chop etish uchun har bir satr elementlari chop etilganda keyin "\n" belgidan ya'ni keyingi satrga o'tishdan foydalanish lozim:

```

#include <iostream.h>
using namespace std;
int main ( )
{int a[3][3]={1, 2 ,3 ,4 ,5 ,6, 7 ,8, 9}, j,i, m=3, n=3; float s,eps;
for ( i=0; i<n; i++) { for ( j=0; j<m; j++)
cout<<" "<<a[i][j]<<" "; cout<<"\n"; }
return 0;
}

```

Dastur natijasi:

```

C:\ "C:\Documents and Settings\...
1 2 3
4 5 6
7 8 9
Process returned 0 (0x0) execution time : 0.046 s
Press any key to continue.

```

2-misol. 2 ta matrisa berilgan. Ularni o'zaro ko'paytmasining yig'indisidan yangi matrisa hosil qiling. Bu yerda 1-matrisaning ustunlar soni 2-matrisaning satrlar soniga teng bo'lishi kerak.

```

#include <conio.h>
int main()
{ int a[3][3], b[3][3],c[3][3], i, j, k;
for (i=1; i<4; i++)
for (j=1; j<4; j++)
a[i][j] = i*j;
for (i=1; i<4; i++)
for (j=1; j<4; j++)
b[i][j] = 2*i+j;
for (i=1; i<4; i++)
{ for (j=1; j<4; j++)
{ c[i][j] = 0;
for (k=1; k<4; k++)
c[i][j] = c[i][j] + a[i][k]*b[k][j]; } }
for (i=1; i<4; i++)
{ for (j=1; j<4; j++)
cout <<c[i][j]<<" "; cout<<"\n";}
return 0;
}

```

```

C:\Documents and Settings\АфъшэшëЁёРёЮё\лорсюўшщ ёёЮы\С
34 40 46
68 80 92
102 120 138
Process returned 0 (0x0) execution time : 0.015 s
Press any key to continue.

```

Endi ikki o'lovchi massivlarga doir masalalarni ko'rib chiqamiz

Misol. a[3][3] massiv berilgan: int a[3][3]={8,-9,6,-5,-3,12,-13,16,17,-18}.

1) Shu massiv elementlari yig'indisi va ko'paytmasini hisoblang:

```

#include <iostream.h>
#include <conio.h>
using namespace std;
int main()
{ int a[3][3]={8,-9,6,-5,-3,12,-13,16,17},i,j;
float s,p;;
s=0; p=1;
for (i=0;i<3;i++)
for (j=0;j<3;j++)
{ s=s+a[i][j]; p=p*a[i][j]; }
cout << "s="<<s<<"\t"<<"p="<<p;
getch();
return 0;
}

```

```
C:\ "C:\Documents and Settings\офьшэшëËрëюË\орсюўшщ ëëюы\C++
s=29      p=2.74959e+08 _
```

2. Shu massiv elementlarining eng kattasi va kichigini toping:

```
#include <iostream.h>
#include <conio.h>
int main()
{ int a[3][3]={8,-9,6,-5,-3,12,-13,16,17},i,j;
  int max,min;
max=a[0][0]; min=a[0][0];
  for (i=0;i<3;i++)
  for (j=0;j<3;j++)
  { if (a[i][j]>max) max=a[i][j];
if (a[i][j]<min) min=a[i][j];
}
  cout << "max=" << max << "\t" << "min=" << min;
getch();
return 0;
}
```

```
C:\ "C:\Documents and Settings\офьшэшëËрëюË\орсюўшщ ëëюы\C++
max=17   min=-13
```

3. Ikkita(3,3) va b(3,3) massivlar berilgan: $inta[3][3]=\{8,-9,6,-5,-3,12,-13,16,-18\}$, $b[3][3]=\{11,-2,4,-8,-1,13,-14,21,-23\}$; Shu massivlarning yig'indisi va ko'paytmasidan hosil bo'lgan yangi massivni chop etish dasturini tuzing:

```
#include <iostream.h>
#include <conio.h>
using namespace std;
int main()
{ int a[3][3]={8,-9,6,-5,-3,12,-13,16,-18}, b[3][3]={11,-2,4,-8,-1,13,-14,21,-23};
int s[3][3],p[3][3],i,j;
s[0][0]=0; p[0][0]=0;
  for (i=0;i<3;i++)
  for (j=0;j<3;j++)
  { s[i][j]= a[i][j]+b[i][j]; p[i][j]= a[i][j]*b[i][j]; } cout << "\n";
cout << "\t" << "Massiv elementlari yigindisi"; cout << "\n";
for ( i=0; i<3; i++) {
  for ( j=0; j<3; j++)
  cout << "\t " << s[i][j] << " ";
```

```

cout<<"\n";}cout<<"\n";
cout<<"\t"<<"Massiv elementlari ko'paytmasi";cout<<"\n";
for ( i=0; i<3; i++)
{
for ( j=0; j<3; j++)
cout<<"\t "<<p[i][j]<<" ";
cout<<"\n";}
getch();
return 0;}

```

```

C:\Documents and Settings\... C++
Massiv elementlari yigindisi
19      -11     10
-13     -4      25
-27     37     -41

Massiv elementlari ko'paytmasi
88      18     24
40      3     156
182     336    414

```

Sinov savollari:

1. Massiv deb nimaga aytiladi?
2. C/C++ tilida massivlarni qanday ifoda etiladi?
3. 1 o'lchovli massiv elementlarini kiritish va chiqarish usullari.
4. 2 o'lchovli massiv elementlarini kiritish va chiqarish usullari.
5. Massiv elementlariga son qiymat berishda tasodifiy qiymatlardan foydalanish
6. Massiv elementlarini natija sifatida chiqarish usullari.
7. Simvollaridan iborat massivlarni xosil qilish.
8. So'zlardan tuzilgan massivlar va ular ustida amallar bajarish

Uyga vazifa : C++ tizimida massiv elementlari bilan ishlash qoidalarini mukammal o'rganish va ularni amaloyotga tatbiq etish.

Mavzu: Funksiyadan foydalanish asoslari. Foydalanuvchi funksiyalari.
Funksiyalarni ta'riflash va murojaat qilish. Funksiyaga parametrlar uzatish.
Rekursiv funksiyalar.

Reja:

1. Funksiyalarni ta'riflash va ularga murojaat qilish.
2. Funksiyaga parametrlar uzatish..

Ma'ruza bayoni Foydalanuvchi funksiyalari.

Funksiyalarni ta'riflash va ularga murojaat qilish. Funksiya ta'rifida funksiya nomi, tipi va formal parametrlar ruyxati ko'rsatiladi. Formal parametrlar nomlaridan tashkari tiplari xam ko'rsatilishi shart. Formal parametrlar ruyxati funksiya signaturasi deb xam ataladi. Funksiya ta'rifi umumiy ko'rinishi quyidagichadir: **Funksiya tipi funksiya nomi(formal_parametrlar_ta'rifi)** Formal parametrlarga ta'rif berilganda ularninga boshlangich qiymatlari xam ko'rsatilishi mumkin. Funksiya qaytaruvchi ifoda qiymati funksiya tanasidareturn<ifoda> ; operatori orkali ko'rsatiladi.

Misol: **Float min(float, float b) { if (a<b) return a; return b; }**

Funksiyaga murojaat qilish quyidagicha amalga oshiriladi:

Funksiya nomi (haqiqiy parametrlar ruyxati)

Haqiqiy parametr ifoda xam bulishi mumkin. Haqiqiy parametrlar qiymati xisoblanib mos formal parametrlar urnida ishlatiladi.

Misol uchun yukoridagi funksiya quyidagicha murojaat qilish mumkin:

Int x=5,y=6,z; z=min(x,y) yeki int z=Min(5,6) yeki int x=5; int z=min(x,6)

Funksiya ta'rifida formal parametrlar inisializasiya kilinishi, ya'ni boshlangich qiymatlar ko'rsatilishi mumkin.

Funksiyaga murojaat kilinganda biror haqiqiy parametr ko'rsatilmasa, uning urniga mos formal parametr ta'rifida ko'rsatilgan boshlangich qiymat ishlatiladi.

Misol uchun:

Float min(float a=0.0, float b) { if (a<b) return a; return b; }

Bu funksiya yukorida ko'rsatilgan murojaat usullaridan tashkari quyidagicha murojaat qilish mumkin:

Int y=6,z; z=min(,y) yeki int z=Min(,6);

Agar funksiya xech qanday qiymat qaytarmasa uning tipi **void** deb ko'rsatiladi.

Misol uchun: **Void print;**

```
{ Cout<<("\n Salom!"); }
```

Bu funksiya **Print**;shaklida murojlat qilish ekranga Salom! Yezilishiga olib keladi. Qiymat qaytarmaydigan funksiya formal parametrlarga ega bulishi mumkin:

Void Pint_Baho(Int baho);

{**Switch(baho)**

```
{case 2:Cout<<("\n yemon");geak; case 3:Cout<<("\n urta");geak; case 4:Cout<<("\n yaxshi");break; case 5:Cout<<("\n a'lo");break; default: Cout<<("\n baxo notugri kiritilgan"); }
```

Bu funksiya **Print_Baho(5)**shaklida murojaat qilish ekranga a'lo so'zi yezilishiga olib keladi. Agar programmada funksiya ta'rifi murojaatdan keyin

berilsa, yeki funksiya boshka faylda joylashgan bulsa, murojlatdan oldin shu funksiyaning prototipi joylashgan buligshi kerak. Prototip funksiya nomi va formal parametrlar tiplaridan iborat bo'ladi. Formal parametrlar nomlarini berish shart emas.

Misol uchun $y = \min(a,b) + 2 * \max(c,d)$ ifodani xisoblashni kuramiz:

```
#Include <iostream.h>
int max(int a,int b)
{if (a<b) return a;else return b};
void main()
{int a,b,c,d,y; int min(int ,int); Cin>>a>>b>>c>>d; y=min(a,b)+2* max(c,d);
Cout<<("\n %f",y); };
int min(int a,int b)
{if (a<b) return b;else return a};
```

Funksiyaga parametrlar uzatish. Funksiyaga parametrlar qiymat buyicha uzatiladi va quyidagi boskichlardan iborat bo'ladi:

Funksiya bajarishga tayerlanganda formal parametrlar uchun xotiradan joy ajratiladi, ya'ni formal parametrlar funksiyalarning ichki parametrlariga aylantiriladi. Agar parametr tipi float bulsa double tipidagi obyektlar xosil bo'ladi, char va shortint bulsa int tipidagi obyektlar yaratiladi.

Haqiqiy parametrlar sifatida ishlatilgan ifodalar qiymatlari xisoblanadi.

Haqiqiy parametrlar ifodalar qiymatlari formal parametrlar uchun ajratilgan xotira kismlariga yeziladi. Bu jarayenda float tipi double tipiga, char va shortint tiplari int tipiga keltiriladi.

Funksiya tanasi ichki obyektlar - parametrlar yerdamida bajariladi va qiymat chakirilgan joyga qaytariladi.

Haqiqiy parametrlar qiymatlariga funksiya xech qanday ta'sir utkazmaydi.

Funksiyadan chikishda formal parametrlar uchun ajratilgan xotira kismlari bushatiladi.

C tilida chakirilgan funksiya chakiruvchi funksiyadagi uzgaruvchi qiymatini uznartira olmaydi. U fakat uzining vaktinchalik nusxasini uzgartirishi mumkin xolos.

Qiymat buyicha chakirish kulaylik tugdiradi. Chunki funksiyalarda kamrok uzgaruvchilarni ishlatishga imkon beradi. Misol uchun shu xususiyatni aks ettiruvchi POWER funksiyasi variantini keltiramiz:

```
power(x,n) int x,n; int p;
for (p = 1; n > 0; --n)
p = p * x; return (p);
```

Argument N vaktinchalik uzgaruvchi sifatida ishlatiladi. Undan to qiymati 0 bulmaguncha bir ayriladi. N funksiya ichida uzgarishi funksiyaga murojlat kilingan boshlangich qiymatiga ta'sir kilmaydi.

Belgili axborot va satrlar

C++ da belgili ma'lumotlar uchun char turi qabul qilingan. Belgili axborotni taqdim etishda belgilar, belgili o'zgaruvchilar va satrli o'zgarmaslar qabul qilingan. Belgili o'zgaruvchilar odatda bir bayt joyni egallaydi va bu 256 xil belgini saqlash uchun yetarlidir.

Char turi qiymatlarini 0..255 sonlar to'plamiga yoki **ASCII** belgilar to'plamiga interpretasiya qilish mumkin.

C++ Programmash asoslarida o'zgaruvchilarning belgili va satr ko'rinishlarini ta'riflash uchun **char** turidan foydalanamiz. Satr o'zgaruvchilariga qiymat qabul qilishda belgilar qo'shtirnoq (" ") ichiga olib yoziladi. Ularni solishtirishda, aniqroq qilib aytganda o'rinlanganda apostrof (' ') ichiga yoziladi.

C++ da belgili ma'lumotlar uchun **char** turi qabul qilingan. Belgili axborotni taqdim etishda belgilar, simvulli o'zgaruvchi-lar va matniy konstantalar qabul qilingan.

Misollar:

```
const char c='c'; // belgi — bir baytni egallaydi, uning qiymati o'zgarmaydi.
```

```
char a,b; // belgili o'zgaruvchilar, bir baytdan joy egallaydi, qiymatlari o'zgaradi.
```

```
const char *s="\n satrining misoli";// matniy konstanta
```

C++ dagi satr - bu nol - belgi - '\0' (nol-terminator) bilan tugallanuvchi belgilar massivi. Nol-terminatorning holatiga qarab satrning amaldagi uzunligi aniqlanadi. Bunday massivdagi elementlar soni, satr tasviriga qaraganda, bittaga ko'p .

```
switch(operator)
```

```
{
case '+': cout<< "plyus"; break;
case '-': cout<< "minus"; break;
case '*': cout<< "ko'paytirish"; break;
case '/': cout<< "bo'lish"; break;
default : cout<< "noma'lum amal";
}
```

...

dastur bo'lagi qismi orqali yuqoridan fikrlarimizni isbotlash mumkin.

...

```
char S;
```

...

S='@'; Bizga ma'lumki ASCII kodidagi simvollarni kod tartib nomeriga mos belgini olish uchun char(N) funksiyasi orkali olishimiz mumkin. Bu yerda N simvolining ASCII dagi tartib nomeri. Masalan quyidagi dastur orqali kompyuterdagi mavjud hamma belgilarni ekranga chiqaramiz. Bizga ma'lumni kompyuterda 256 ta belgidan foydalanamiz

```
# include < string.h >
```

```
main()
```

```
{
for (int i=0; i<=256; i++)
cout << i << "-" << char(i) << ",_";
}
```

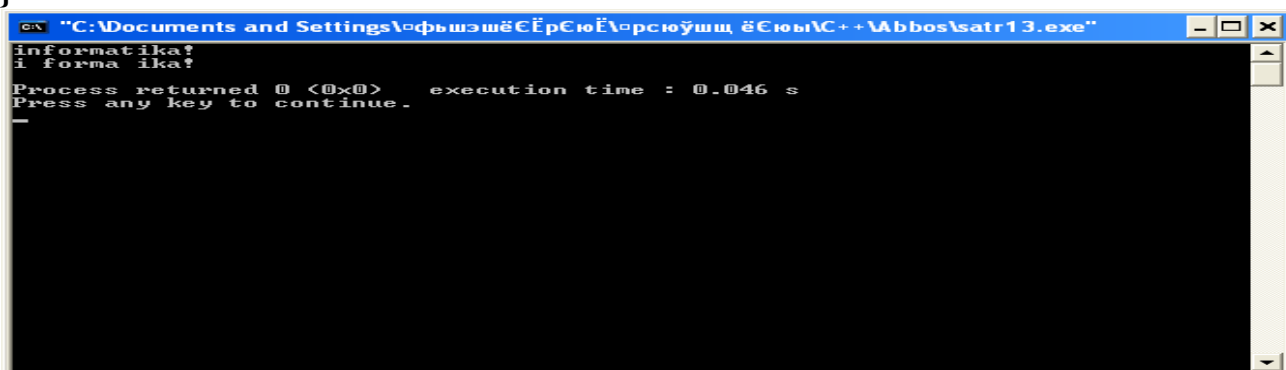
C++ tilida satrlar uchun alohida tur mavjud emas. O'zgaruvchilarning satr ko'rinishini ta'riflashda belgili o'zgaruvchilarni ta'riflashda foydalaniladigan **char** turidan foydalanamiz. **Masalan**, osonlik uchun quyidagi matn ko'rinishidagi o'zgaruvchilarni ta'riflashini C++ tilida yozib ko'ramiz:

```
char ch; char s10[10]; char s256[];
```

O'zgaruvchilarning bu yozilishiga keng ta'rif berib o'tirishga hojat yo'q. **Masalan, ch** o'zgaruvchisi faqat bitta belgi, **s10** o'zgaruvchisi **10 ta belgi** va **s256** o'zgaruvchisi **256 ta belgi** qabul qila oladi.

Endi belgili va satrli ko'rinishdagi o'zgaruvchilarni qatnashtirib dastur tuzishga harakat qilib ko'ramiz

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
int main()
{char ch,s1=' ', s2=' ';
char st[]="informatika";
ch = '!';
cout<<st<<ch<< endl;
st[1]=s1;
st[7]=s2;
cout<<st<<ch<< endl;
return 0;
}
```



Bundan tashkari satrli va belgili o'zgaruvchilar ustida qo'shish va solishtirish amallarini o'rinish mumkin. Masalan

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
main()
{ char s[10],s6='o',s7='t';
char s1[]="PC";
char s2[]="Ensiklopediya";
s=s6+s7;
cout <<s<<endl;
return 0;
}
```

Qiymat berish operatori yordamida satrga qiymat berish mumkin emas. Satrni massivga yoki kiritish paytida, yoki nomlantirish yordamida joylashtirish mumkin.

Misol:

```
#include <iostream.h>
```

```

#include <conio.h>
#include <string.h>
int main()
{ char s1[10]="string1";
int k=sizeof (s1); cout<<s1<<'\t'<<k<<endl;
char s2[]="string2"; k=sizeof(s2);
cout<<s2<< '\t'<<k<<endl;
char s3[]={ 's','t','r','i','n','g','3'};
k=sizeof(s3); cout<<s3<<'\t'<<k<<endl;
char *s4="string4";//s4 ko'rsatkichi, uni o'zgartirib bo'lmaydi. k=sizeof(s4);
cout<<s4<< '\t'<<k<<endl;
return 0;
}

```

Natijalar:

string1 10 - 10 bayt ajratilgan, shu jumladan **\0** ga **string2** 8 — 8 bayt ajratilgan (7+1 bayt **\0** ga) **string3** 8 - 8 bayt ajratilgan (7+1 bayt **\0** ga) **string4** 4 - ko'rsatkichning o'lchamlari

```

"C:\Documents and Settings\...Abbos\str1 2.exe"
string1 10
string2 8
string3 7
string4 7
Process returned 0 (0x0)   execution time : 0.046 s
Press any key to continue.

```

Sinov savollari

1. Paskaltilidafunksiyaqandayaniqlanadi?
2. Funksiyagamurojaatqandayamalgaoshiriladi?
3. Funksiyani tavsiflashda qanday parametrlar ishlatiladi?
4. Funksiyaning o'ziga xos xususiyati nimadan iborat?

Uyga vazifa : C++ tizimida function bilan mustaqil ishlash qoidalarini mukammal o'rganish va ularni amaliyotga tatbiq etish.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu: C++ da grafika. C++ tilining grafik tartibida ishlash asoslari.

Chizmalarni hosil qilish uchun fumksiya va proseduralar.

Reja:

1. C++ tilining grafik tartibida ishlash asoslari.
2. Chizmalarni hosil qilish uchun fumksiya va proseduralar.

Ma'ruza bayoni

C/C++ tilining grafik rejimida ishlash asoslari

Grafik paketga quyidagilar kiradi:

- *graphics.h* sarlavha fayli;
- *graphics.lib* kutubxona fayli;
- grafik qurilmalar drayveri (**.bgi*);
- (**.chr*) shriftlar.

Grafik tartibotida ekranni boshqarish funksiyalar majmuasi o'z qaliamalga oshiriladi. Ularning protiplari sarlavhalar fayli *graphics.h* da joylashgan bo'ladi.. Bu direktiva o'z garmaslar, o'zgaruvchilar va turli qism dasturlardan tashkil topgan bo'lib, ular yordamida turli grafik adapterlar bilan har xil tasvirlar chizish mumkin. Adapter kompyuterda **graphics.h** fayli bilan ishlash imkoniyatini yaratadigan maxsus qurilmadir. Grafik rejimiga o'tilganda ekran aloxida-aloxida nuqtalarga bo'linadi. Har bir nuqta o'z koordinatasiga egadir.

Eng ko'p ishlatiladigan adapterlar:

1. *CGA - color graphics Adapter*
2. *MCGA - multi color graphics array*
3. *EGA - enhanced graphics Adapter*
4. *VGA - video graphics array.*

Drayverlarni ko'rsatish uchun quyidagi uzmarmaslar ishlatiladi:

Detect= 0 CGA= 1; MCGA= 2; EGA=3; VGA=9.

Grafik tartibotida ishlash uchun *graphics.h* fayli **#include** direktivasi yordamida C++ tilining preprocessoriga barcha modullarda grafik qism dasturlardan foydalanish uchun qo'shilishi lozim: **#include <graphics.h>**.

Grafik funksiyalar bilan ishlash uchun ekranning matn rejimidan grafik rejimiga o'tish uchun maxsus proseduradan foydalaniladi: **initgraph (&gd, &gm, "path")**; bu yerda: **gd**- drayver nomi; **gm**- rejim nomi; **Path**- kerakli drayver faylining yo'li. Ko'pincha **gd=0** deb olinadi.

Drayverlar *.bgi* fayllarida saqlanadi. Agar drayver ishchi katalogning o'zida joylashgan bo'lsa, u holda **Path = " "** (bo'sh belgisi) bo'ladi. Masalan,

```
int D=DETECT,M;  
initgraph (&D,&M,"");
```

Grafik rejimidan yana matn rejimiga o'tish kerak bulsa, **closegraph()** funksiyasi ishlatiladi.

Umuman grafik tartibotida ishlash quyidagi ketma-ketlikda amalga oshiriladi:

- grafik tartibotiga o'tish;
- grafik bilan ishlash;
- grafik tartibotidan chiqish

Masalan:

```
#include <stdio.h>  
#include <graphics.h>  
int main()  
{
```

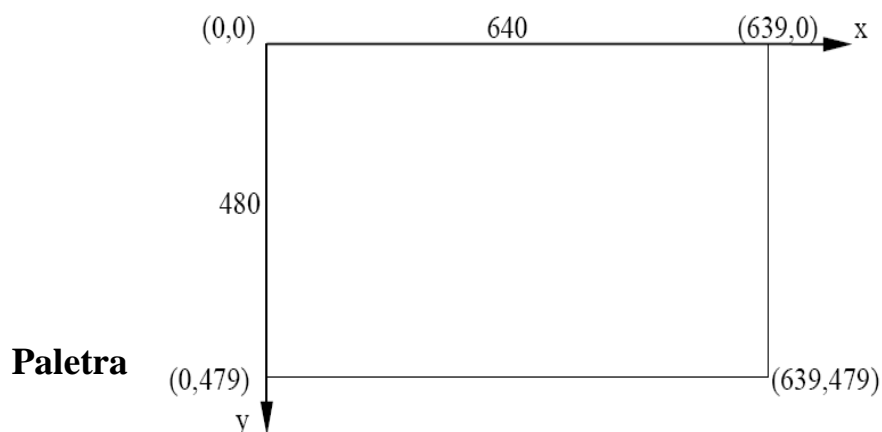
```

int DRIVER=DETECT, MODE;
initgraph(&DRIVER,&MODE,"");
outtextxy(200,200,"Chiqish");
getchar();
closegraph();
}

```

Grafik funksiyalar.

Ekraning grafik holatida koordinalar sistemasi quyidagi ko'rinishda bo'ladi:



Ekraning grafik tartibida $640 \times 480 \times 16$ o'lchovidanko'proqfoydalaniladi. Buyerda $640 \times$ o'qibo'yichanuqtalarsoni, 480 yo'qibo'yichanuqtalarsoni, 16 grafiktartibidaishlatiladiganranglarsoni.

graphics.h faylidastandardpaletraningranglarigamoso'zgarmaslarishlatiladi:

O'zgarmas nomi	rang	O'zgarmasning qiymati
<i>BLACK</i>	qora	0
<i>BLUE</i>	Ko'k	1
<i>GREEN</i>	Yashil	2
<i>CYAN</i>	Lojivard	3
<i>RED</i>	qizil	4
<i>MAGENTA</i>	Binafsha	5
<i>BROWN</i>	Jigarrang	6
<i>LIGHTGRAY</i>	Och kulrang	7
<i>DARKGRAY</i>	To'q kulrang	8
<i>LIGHTBLUE</i>	Och ko'k	9
<i>LIGHTGREEN</i>	Och yashil	10
<i>LIGHTCYAN</i>	Och lojivard	11

<i>LIGHTRED</i>	Och qizil	12
<i>LIGHTMAGENTA</i>	Och binafsha	13
<i>YELLOW</i>	Sariq	14
<i>WHITE</i>	Oq	15

Ekran nuqtasi

Grafik ekran bir qancha nuqta (piksel) lar majmuasidan iborat bo'lib, har nuqta o'z koordinasiga va rangiga ega bo'ladi. (x,y) koorditanali color rangli nuqta ekranda quyidagi funksiya orqali beriladi: putpixel (x, y, color); - bu yerda x,y – nuqtaning koordinasi, color-rang.

(x,y) koordinali nuqtaning rangini aniqlash uchun *getpixel (int x, int y)*; funksiya ishlatiladi.

Kontur figuralar

Tekislikdachiziqlarguruhikontur (yopiq) figuralarnihosilqiladi. Konturfiguralarsifatidato'g'richiziqbo'lagi, yoy, aylana, ellips, to'g'rito'rtburchakvahokasolarniqarashmumkin. Bufiguralarbir-biridankorinishidantashqarichiziqningrangi, turivaqalinligibilanfarqqiladi. Graf-iktartibidaavtomatikholdachiziqningrangi- *WHITE* (oq), qalinligi – birpiksel, turi – yaxlitchiziqqilibo'rnatiladi.

Chiziq rangini o'zgartirish*setcolor(color)*; funksiyasi orqali bajariladi. Figuraning boshqa parametrlari quyidagi funksiya orqali beriladi:

setlinestyle (linestyle, upattern, thickness);

bu yerdalinestyle – chiziq turi,thickness – uning qalinligi.

Chiziqning turi 5 xil bo'lishi nunkin:

qiymat	nomi	tavsifi
0	<i>SOLID_LINE</i>	yaxlit
1	<i>DOTTED_LINE</i>	punktirli
2	<i>CENTER_LINE</i>	shtrix-punktirli
3	<i>DASHED_LINE</i>	shtrixli
4	<i>USERBIT_LINE</i>	Foydalanuvchi tomonidan aniqlanadigan tur (<i>upattern</i>)

Kontur chiziqlarini parametrlari o'rnatilgandan keyin ekranda figuralarni hosil qilishga kirishamiz. Buning uchu standart grafik kutubxonasida mavjud bo'lgan funksiyalardan foydalanamiz. Bu funksiyalar quyidagilardan iborat.

Funksiya nomi	Tavsiflanishi
<i>Line(x1, y1, x2,t y2)</i> ;	Ekranda(<i>x1, y1</i>)koordinatali nuqtadan (<i>x2, y2</i>) koordinali nuqttagacha to'g'ri chiziqni hosil qiladi.
<i>rectangle(x1, y1, x2,y2)</i> ;	Ekranda gorizontal (vertikal) joylashgan to'ri to'rtburchakni hosil qiladi, bu yerda yukori chap nuqtaning

	koordinatalari (x_1, y_1), o'ng pastki nuqtaning koordinatalari (x_2, y_2) hisoblanadi.
<i>circle</i> (x, y, R ;	markazi (x, y) nuqtada va radiusi R ga teng bo'lgan aylana chizadi (o'lchov birligi piksel)
<i>arc</i> (x, y, a, b, r)	markazi (x, y) nuqtada va radiusi r ga teng bo'lgan yoy chizadi, bu yerda x va y - markazning koordinatalari, a - bosh burchak, b - oxirigi burchak, r - yoy radiusi. Burchaklar gradusda qabul qilinadi va u soat strelkasiga teskari tartibda hisobga olinadi. Agar $a=0$, $b=360$ bo'lsa to'liq aylana hosil bo'ladi.
<i>ellipse</i> (x, y, a, b, xr, yr)	xuddi shu tartibda ellips yoyini chizadi.
<i>drawpoly</i> (n, p)	ko'pburchak hosil qiladi.. Bu yerda n – ko'pburchakning uchlari soni; p - ko'pburchak uchlarning koordinatalari.

Tekis figuralar

Tekis figuralar ekran tekisligining yopiq kontur bilan chegaralangan qimidan iborat bo'ladi.

Tekis figuralarni kontur hosil qiluvchi ichki sohasini yoki yopiq chiziqlar tashqarisini bo'yash orqali yasash mumkin. Bunday chiziqlar uzluksiz bo'lish shart.

Bu amal quyidagi funksiya yordamida bajariladi:

floodfill($x, y, border$);

Bu yerda x, y – kontur ichida yoki tashqarisida joylashgan nuqta koordinatasi, *border* – kontur hosil qiluvchi chiziq rangi. Bunda to'liq konturning rangi bir xil bo'lishi lozim. Bo'yash *color* rang bilan va *pattern* shabloni bo'yicha quyidagicha bajariladi: *setfillstyle* (*pattern, color*);

Tekis figuralarni hosil qilish eng ko'p ishlatiladigan figuralar quyidagilar hisoblanadi:

Funksiya nomi	tavsiflanish
<i>bar</i> (x_1, y_1, x_2, y_2)	joriy rang va chiziqlar yordamida ichi bo'yalgan to'g'ri to'rtburchak chizadi, bu yerda yuqori chap nuqtaning koordinatalari (x_1, y_1), o'ng pastki nuqtaning koordinatalari (x_2, y_2) hisoblanadi.
<i>fillellipse</i> (x, y, xr, yr)	joriy rang va chiziqlar yordamida markazi (x, y) nuqtada mos ravishda gorizontaal va vertikal o'qlari xr va yr (xr - kenglik, yr - balandlik) ga teng bo'lgan ichi bo'yalgan rangli ellips chizadi.
<i>fillpoly</i> (n, XY);	yopiq konturdan iborat ko'pburchakni chizadi va uni bo'yaydi, bu yerda n – burchaklar soni, XY – ko'pburchak burchaklarning koordinatalari.

<i>pieslice(x, y, a, b, r)</i>	markazi (x, y) nuqtada va radiusi R ga teng bo'lgan sektor chizadi va uni bo'yaydi, bu yerda x va y - markazning koordinatalari, a - bosh burchak, b - oxirigi burchak, r - yoy radiusi. Burchaklar gradusda qabul qilinadi va u soat strelkasiga teskari tartibda hisobga olinadi
--------------------------------	--

Chizmalarni xosil qilish uchun ishlatiladigan prosedura va funksiyalar

1. **putpixel (x, y, color)**- x va y koordinatadagi nuktani color rangda chizish;
2. **getpixel (x, y)**- x va y koordinatadagi nuqtaning rangini aniklaydi;
3. **line (x1, y1, x2, y2)**- $x1$ va $y1$ koordinatadagi nuqtadan $x2$ va $y2$ koordinatadagi nuktagacha kesma chizish;
4. **circle (x, y, r)**- markazi x va y koordinatada va radiusi R bulgan aylana chizish;
5. **rectangle (x1, y1, x2, y2)** - yukori chap nuktasi $x1$ va $y1$ koordinatada, ung pastki nuktasi $x2$ va $y2$ koordinatada bulgan tugriturtburchakni chizish;
6. **setbkcolor (color)**- orka fonga rang berish;
7. **setcolor (color)**- chizish rangini urnatish (rangli kalam); Bu yerda color - rang nomeri yoki nomi. Agar rang nomi yoziladigan bulsa, uni katta xarflarda yoziladi.
8. **bar (x1, y1, x2, y2)**- joriy rang va chiziklar yordamida ichi buyalgan tugriturtburchak chizish;
9. **fillellipse (x, y, xr, yr)**- markazi x va y da, xr kenglikda va yr balandlikda ichi buyalgan rangli ellips chizadi;
10. **getmaxx** - joriy rejim va drayverlar uchun nuqtalar sonini aniklash; **getmaxy** - joriy rejim va drayverlar uchun vertikal nuqtalar soni. Bu prosedura yordamida kompyuterning uzi ekrandagi maksimal nuqtalar sonini aniklaydi.
11. **linerel (x, y)**- x va y koordinatali nuqtadan joriy nuktagacha kesma chizish;
12. **lineto (x, y)**- joriy nuqtadan x va y koordinatali nuktagacha kesma chizish;
13. **bar3D (x1, y1, x2, y2, h, top)**- parallelopiped chizadi. Bu yerda h - parallelopipedning uzunligi; top - yukori kismini chizish uchun kerak. Agar topon - bulsa tomi bor, agar topoff - bulsa tomi yuk.
14. **arc (x, y, a, b, r)**- yoy chizish uchun. Bu yerda x va y - markazning koordinatalari, a - bosh burchak, b - oxirigi burchak, r - yoy radiusi. Burchaklar gradusda kabul kilinadi.
15. **ellipse (x, y, a, b, xr, yr)**- xuddi shu tartibda ellips yoyini chizadi.
16. **drawpoly (n, p)**- kupburchak chizish uchun. Bu yerda n - kupburchakning uchlari soni; p -. Ko'pburchak uchlarining koordinatalari.

Sinov savollari.

1. C/C++ tilidagrafikrejimidaishlashuchunkandaydirektivalarkerakbo'ladi?
2. Ko'pishlatiladiganadapterlarxakidama'lumotbering.
3. Drayver nima va uning vazifasi
4. Oddiy chizmalarni xosil kilish uchun kaynday funksiya va proseduralar mavjud?
5. Nuqta, to'g'ri chiziq, aylana, to'g'ri to'rtburchak, ichi bo'yalgan to'rtburchak, ellips, yoy, parallelopiped va shu kabilarni chizish strukturalari

Uyga vazifa : C++ tizimidagrafik ma'lumotlar va ularning funktsiya-proceduralari bilan ishlash qoidalarini mukammal o'rganing, ularni amaloyotga tatbiq etishga doir misollar keltiring.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu: Bo'yash, chizish usullari va stillari. Grafik tartibda shriftlar. Dinamik xotira. Chizmalarni harakatlantirish.

Reja:

1. Bo'yash, chizish usullari va stillari.
2. Dinamik xotira.
 2. Chizmalarni harakatlantirish.

Ma'ruza bayoni

Buyash va chizish usullari hamda stillari haqida

1. **setfillstyle (style, color)**- buyash usul va rangni urnatish. Bu yerda **style**- uzgarmas kattalik bulib, u quyidagicha bulishi mumkin: 0 - soxani fon rangi bilan tuldirish; 1- soxani rang bilan uzluksiz tuldirish;

2 - kalik gorizontal chiziklar

3 - ingichka ogma chiziklar

4 - yugon ogma chiziklar

5 - yugon ogma chiziklar (boshka stil)

6 - ogma yullar

7 - turtburchakli chiziklar

8 - ogma turtburchaklar

9 - zich ogma shrtixlar

10 - siyrak nuktalar (u yer - bu yerda)

11 - zich nuktalar bilan

2. **floodfill (x, y, color)**- joriy rang va usuldan foydalangan xolda chegaralangan soxani buyash. Bu yerda x va y - shu soxaga tegishli bulgan biror nuqta koordinatasi. Avval rang, keyin stili ko'rsatiladi. Masalan:

setcolor (4); {kizil rangli kalam, chegara rangi} **setfillstyle (1, 2);** {1-stil

bilan yashil rang bilan buyash} **circle (50, 50, 35);** {radiusi 35 bulgan aylana

chizish} **floodfill (50, 50, 4);** {aylana ichiga rang tukish, buyaladigan chegara rangi rangli kalam bilan bir xil bulishi kerak}

3. **setlinestyle (s, a, b)**- turli stildagi chiziklarni chizish uchun; Bu yerda s- **style** nomeri; a -foydalanuvchi stilini yaratishi mumkin bulgan parametr, odatda a=1 deb olinadi; b- chizikning kalinligini ko'rsatadigan parametr

0 - oddiy chizik;

1 - mayda punktir chizik;

2 - kalin va uzunchok punktir chizik

3 - yupka va uzunchok punktir chizik;

4 - siyrak nuktali chizik.

1- Misol:

```
# include <graphics.h>
```

```
# include <conio.h> void main ( )
```

```

{ int i, j, gd, gm ;
gd= 0;
initgraph (&gd, &gm, " ");
setcolor (14); // sarik kalam
for ( i=0; i<=20; i++) for ( j=0; j<=20; j++)
circle (i*50, j*30, 55); // sarik rangli aylanalar
rectangle (0, 0, getmaxx, getmaxy); //ekran buylab tugri turtburchak setcolor
(11); // tuk feruza rangli kalam
bar3d(200, 300, 100, 150, 30, topon); // parallelopiped, ichi ok setcolor
(CYAN); // och feruza rangli kalam
fillellipse (350, 360, 135, 90); //ellips, ichi ok rangda
getch( ); closegraph( ); }

```

Grafik rejimida shriftlar

Grafik rejimi xolatida turli shriftlardan foydalanib matnlarni xam yozsa bo'ladi. Shriftlar **.chr**kengaytmali fayllarda saklanadi. Ular **.bgi** fayllari bilan bitta katalogda saklanishi shart.

1. **outtextxy (x, y, 'matn');**- matnni yozish; bu yerda x va y matn boshlanadigan nuqta koordinatalari; masalan: **outtextxy (10, 10, 'Mirzaev K. 212-07 Aty');**
2. **settextstyle (sh, n, r);**mant shriftini urnatish; bu yerda sh - shrift nomeri (0 - vektorli shrift, 1 - standart shrift); n - shrift yunalishi (0 - chapdan unggga, 1 - quyidan yukoriga yozish); r - shrift razmeri (oddiy shriftda 1, vektorli shriftda 4 deb olinadi);
3. **settextjustify (h, v)-** yozilgan katorni tekislaydi. U outtextxy prosedurasidan keyin yoziladi. Bu yerda h - gorizont tekislash; v - vertikal tekislash; Gorizont tekislash uchun: 0 - chapga; 1- markazga; 2 - unggga. Vertikal tekislash uchun: 0 - pastga; 1 - markazga; 2 - yukoriga.
4. **setusercharsize** - vektor shriftlari uchun bir xil belgilarning eni va buyini urnatadi. Masalan: **setUserCharSize(x1, y1, x2, y2);**

2-misol. Funksiyalarning grafiklarini chizish.

```

# include <graphics.h>
# include <conio.h>
# include <math.h> void main ( )
{ int i, j, gd, gm ; float x, y;
gd=0; initgraph (&gd, &gm, " ");
setcolor (14);
line (320, 0, 320, 480);
line (0, 240, 640, 240);
line (480, 0, 480, 235);
line (325, 120, 635, 120);
line (160, 245, 160, 475);
line (0, 360, 315, 360);
line (480, 245, 480, 475);
line (325, 360, 635, 360);
x =-10; outtextxy(10, 20, ' y=sin(x) grafigi'); do

```

```

{ y = sin(x);
putpixel (160 + 10*x, 120 - y, 5); x = x+0.001; } while (x<=10);
x1 = -10; outtextxy(10, 20, ' y=cos(x) grafigi'); do
{ y = cos(x1);
putpixel (480 + 20*x1, 120 - 20*y1, 6); x1 = x1+0.001; } while (x1 <=10);
x2 = -10; outtextxy(10, 20, ' y=exp(x) grafigi'); do
{ y2 = exp(x2);
putpixel (160 + 10*x2, 360 - 20*y2, 7); x2 = x2+0.001; } while (x2 <=10);
x3 = -10; outtextxy(10, 20, ' y=ln(x) grafigi'); do
{ y3 = ln(x3);
putpixel (480 + 10*x3, 360 - y3, 8); x3 = x3+0.001; }
while (x3 <=10); getch( ); closegraph( ); }

```

Dinamik xotira

include <graph.h>direktivasi ning yana shunday proseduralari mavjudki, ular yordamida chizmalarni ekran buylab xarakterga keltirish mumkin. Figuralarni xarakterga keltirishning bir necha usullari bor. Ulardan biri xarakterni takrorlanish buyrug'i orkali tashkil qilishdir. Ikkinchi usul ekranda chizilgan chizma joylashgan soxani massiv ko'rinishida eslab kolib, uni maxsus prosedura yordamida ekranning kerakli nuqtasiga kuchirishdir. Bunda dinamik xotiradan foydalaniladi.

Katta miqdordagi ma'lumotlar ishlatiladigan masalalarni yechishda, kompyuterning grafik imkoniyatlaridan foydalanganimizda xotira xajmi yetishmasligi mumkin. Bunday xollarda dinamik xotira juda kul keladi. Dinamik xotira bu kompyuterning dasturga ma'lumotlar segmentidan tashkari yuklatilgan tezkor xotiradir. Bu xotira taxminan 200-300 Kbni tashkil kiladi. Dinamik xotiradan foydalanish uchun ko'rsatkichlar ishlatiladi. Bu uzgaruvchilarni (ko'rsatkichlarni) xotirada joylashtirishni kompilyator amalga oshiradi. Ko'rsatkich shunday uzgaruvchiki, uning qiymati uzgaruvchi qiymatiga emas, balki shu uzgaruvchi joylashgan xotira adresiga tengdir.

Dinamik xotira soxasidan joy ajratish uchun new operatori ishlatiladi. Bu so'zdan keyin xotiraga joylashtiriladigan obyekt tipi aniklanadi. Masalan: new int; deb yozsak, dinamik xotiradan 2 bayt joy ajratgan bo'lamiz.

Masalan: int *p; p = new int; yoki int *p = new int ;

Ajratilgan xotira soxasiga biror qiymatni joylashtirish mumkin: *r = 750 ;

Bu yozuvni quyidagicha ukiladi: « r ko'rsatkichida adresi saklanayotgan xotiraga 750 sonini yozing ».

Dinamik xotira soxasi chegaralangan, u tulib kolganda new operatori orkali joy ajratish xatolikka olib keladi. Bu xolni biz xotiraning tulib ketishi yoki okib ketishi deymiz (utechka pamyati). Shuning uchun xotira boshka kerak bulmasa uni bushatish zarurdir. Bu delete operatori yordamida bajariladi. Masalan: delete p;

Ekranda chizmalarni xarakterlantirish

Ekranda chizmalarni xarakterlantirish uchun kerak bo'ladigan proseduralar:

1. **imagesize (x1, y1, x2, y2)**- ekranning chap yukori nuqtasi va ung pastki nuqtasi koordinatalaridan tugriturburchakli soxani saklash uchun kerak bo'ladigan xotiraning ulchami (baytlarda olinadi);

2. **getimage (x1, y1, x2, y2, p)**- dinamik xotiraning berilgan r maydonida turgurturburchakli tasvirni saklash. Bu yerda r - tasvir saklanadigan joyning adresini saklaydigan uzgaruvchi, ya'ni ko'rsatkich.
3. **putimage (x, y, p, m)**- ekranning berilgan joyiga tasvirni chikarish; bu yerda x va y - xotiraning p maydonidagi tasvirdan nusxa kuchiriladigan ekran maydonining chap yukori nuktasi; m - tasvirni ekranga chikarish rejimi. Agar:
m= 0 (NormalPut)- tasvirni kuchirish. Bunda eskichi uchib, yangisi paydo bo'ladi (xuddi yurib ketayotgandek) **m= 1 (XorPut) m= 2 (Orput) - m= 3 (AndPut)**

Masalan: quyidagi dasturda kvadrat ichidagi buyalgan aylana ekran buylab xarakterlanadi.

```
# include <graphics.h >
# include <conio.h >
# include <dos.h >void main ( )
{ int gd = 0, gm, I, j, s; int *a; initgraph(&gd,&gm,""); setcolor ( 4 );
circle ( 30, 30, 20 ); putpixel ( 30, 30, 2); rectangle ( 10, 10, 50, 50); s = im-
agesize ( 9, 9, 51, 51); *a = new int; *a = s; getimage (9, 9, 51, 51, a); for ( i = 0;
i <= 585; i ++ )
{ putimage ( i, 10, a, 0); sound (20); delay (10); nosound ( ); } for ( j = 10; j <=
420; j ++ )
{ putimage(585, j, a, 0 ); sound ( 30 ); delay (10); nosound( );}
for (i = 585; i >= 10; i - -)
{ putimage(i, 420, a, 0); delay ( 10 ); }
for (j = 420; j >10; j - -)
{ putimage(10, j, a, 0); delay( 10 ); }
delete a;
getch(); //closegraph( ); }
```

Sinov uchun savollar

1. Bo'yash, chizish usullari va stillari qanday?
2. Grafik tartibda qanday shriftlardan foydalaniladi?
3. Dinamik xotira nima va undan qanday foydalaniladi?
4. Ob'ektlar qanday harakatlantiriladi?

Uygavazifa : C++ tizimidagrafikholatidamatnlarbilanishlashqoidalarini-mukammalo'rganing, tasvirlarnivagrafiklarni-hosilqiling, aularniamaloyotgatatbiqeting.

.Adabiyotlar

[6],[7],[8],[9],[10],[11],[13], [14],[15],[16]

Mavzu: C++ dafayllar. Fayltushunchasi, matnvabinarfayllar. O'qish-yozishoqimlar. Standartoqimlar. Belgilarni, satrlarni, formatlio'qish-yozishfunksiyalari.

Reja:

1. Fayltushunchasi, matnvabinarfayllar.
2. O'qish-yozishoqimlar.
3. Belgilarni, satrlarni, formatlio'qish-yozishfumksiyalari.

Ma'ruza bayoni

C/C++ da fayllar bilan ishlash

Oldindan berilgan kattaliklarni - obyektarni kiritish chikarish S++ tilida kiritish-chikarish okimlarining sinflari mavjud bulib, ular kiritish-chikarish standart kutubxonasi obyektg muljallangan ekvivalentidir. Ular quyidagilar: istream - kiritish okimi ostream - chikarish okimi iostream - kiritish/chikarish oqimi. Satrli okimlar xotirada joylashtirilgan satrli buferlardan ma'lumotlarni kiritish-chikarish uchun xizmat qiladi.

1. **istream** - satrli kiritish
2. **ostream** - satrli chikarish
3. **fstream** - satrli kiritish/chikarish

Quyidagifaylliokimlarfayllarbilanishlashuchunxizmatkiladi. **ifstream**- fayllikiritish**ofstream**- fayllchikarish**fstream**- fayllkiritish/chikarish Odatda bu okimlar include< > sifatida yoziladi.

ifstream, **ofstream**va **fstream**okimlari dasturda fayllar xosil qilish, ulardagi ma'lumotlardan foydalanish uchun ishlatiladi. Ularning kullanilishi quyidagicha: **ofstreamname(" path\ file_name");**- ma'lumotli fayl xosil qilish, ya'ni ma'lumotlar bazasi uchun ochish;

Masalan: **ofstreamfarruh("c:\ tcpp\bin\d11.dat");**
ofstream alibek ("nnn.txt");

Bu yerda**name** - ixtiyoriy nom (lotincha);ya'ni okim nomi. Keyinchalik fayldagi ma'lumotlarni yozish yoki ukish uchun shu nomdan foydalanamiz. **d11.dat**va **nnn.txt**biz xosil kilgan fayl nomlari bulib, ular okim nomlari bilan boglangandir. Endi xosil bulgan ma'lumotlardan foydalanish uchun uni ochishni kuramiz:

ifstreamname ("path");

Masalan: **ifstream farruh ("c:\ tcpp\bin\d11.dat"); ifstream alibek("nnn.txt");**Ochilgan fayllarni albatta yopish kerak! Bu jarayonni quyidagicha amalga oshiriladi: **name.close();** Masalan, **farruh.close();** yoki **alibek.close();**

Demak, **farruh** bilan **d11.dat**, **alibek**bilan **nnn.txt**nomlari uzaro ma'lumot almashinuvini ta'minlaydi. Masalan:

2ta butun sonni va ularning yigindisini uzida saklovchi fayl xosil kiling va undan keyingi dasturda foydalaning.

Sonlarni a, b, yigindini s, faylni ttt.dat, okim nomini jasur deb ataymiz.

```
include <iostream.h>
```

```
include <fstream.h>
```

```
include <conio.h>
```

```
void main ( ) {
```

```
int a=12, b=13, s; ofstream jasur ("ttt.dat");
```

```
s = a + b;
```

```
s out <<"s=" <<s <<endl; jasur <<s <<endl; jasur.close ( );
```

```
getch ( ); }
```

```
Endi undan foydalanamiz: // # include <iostream.h>
```

```
include <fstream.h>
```

```
include <conio.h>
```

```
include <math.h>
```

```

void main () {
int s; float s1; ifstream jasur ("ttt.dat"); jasur >>s; s1 = sin (s);
cout <<"s1="<<s 1<<endl; jasur.close (); getch( ); }

```

2-misol. Matrisa va vektorlar berilgan. Son qiymatlari ixtiyoriy. Ushbu qiymatlardan foydalanib, matrisani vektorga kupaytirish, matrisaning izini xisoblash va vektorning yigindisini xisoblash dasturini tuzing.

Avval matrisa va vektorlarning son qiymatlarini uzida saklovchi fayl xosil qilamiz. Sungra bu ma'lumotlardan foydalanamiz.

```

include <iostream.h>
include <fstream.h>
include <stdlib.h>
include <time.h> void main ( )
{ srand (time (0)); int a [3][3], b[3], i, j; ofstream said ("ak-
bar.txt"); for ( i=0; i<3; i++) { for (j=0; j<3; j++) { a[i][j] = rand(
); said <<a[i][j]; } } for (i=0; i<3; i++) { b[i] = rand( ); said <<b[i];
}
said.close ( );
}

```

```

include <iostream.h>
include <fstream.h>
void main ( ) {
int a[3][3], b[3], i, j, c[3], s1=0, s2=0;
ifstream said ("akbar.txt");
for ( i=0; i<3; i++)
for (j=0; j<3; j++)
said >>a[i][j];
for ( i=0; i<3; i++)
said >>b[i];
for ( i=0; i<3; i++)
{ c[i] = 0;
for (j=0; j<3; j++)
c[i] = c[i] + a[i][j] * b[i];
cout <<"c="<<c[i]<<endl; }
for ( i=0; i<3; i++)
s1 = s1 + a[i][i];
for ( i=0; i<3; i++)
s2=s2 + b[i];
cout <<"s1="<<s1<<" s2="<<s2<<endl;
}

```

Foydalanuvchining direktivalarini (include) yaratish

C/C++ tilida foydalanuvchi uzining include ini yaratishi va undan kerakli vaktlarda keng foydalanishi imkoniyati berilgan. Buning uchun kerakli operatsiyalar algoritmi funksiya va prosedura sifatidagi dasturi bloknotga yoziladi. Sungra bu dasturga ixtiyoriy nom (lotincha) beriladi, fayl nomidan sung **.h** kengaytmasi beriladi. **h - header-** sarlavxa degan ma'noni bildiradi. Bu faylni INCLUDE papkasiga

saklash shart! Foydalanuvchi bu include da avvaldan mavjud bulgan include lar-dan, istalgan uzgaruvchi, uzgarmasdar, funksiya va proseduralardan foydalanishi mumkin. Dasturchi uzining shaxsiy include ini ishlatishi uchun asosiy dasturda (C/C++ muxitida) ularni e'lon qilishi va undagi funksiya va proseduralarning nom-lari xamda haqiqiy parametrlarining urnini bilishi kerak. Shaxsiy include larni cha-kirishda <> yoki " " belgilari ishlatilishi mumkin. Masalan: **arccos x = arctg ()** formulasi yordamida tashkil etuvchi include yaratamiz va undan foydalanamiz. Bloknotdagi dasturi:

```
include <math.h >float acos (float x)
```

```
{ float y;
```

```
y = atan( sqrt ( 1-x*x) / x );
```

```
return y; }
```

Bu faylga ixtiyoriy nom beramiz, masalan: farruh . h ungra undan foy-dalanib dastur tuzamiz: (C/C++ muxitida)

```
include <iostream.h >
```

```
include <conio.h >
```

```
include <farruh.h >void main ( )
```

```
{ float x, y; cin >> x; y = acos (x);
```

```
cout <<"y="<< y <<endl;
```

```
getch ( ); }
```

Odatda yaratilayotgan include larga barcha kerakli funksiyalar guruxlab joylashtiriladi va bir yula chakirib ishlatiladi. Masalan:

```
include <math. h >
```

```
float acos ( float x) {      }
```

```
float asin ( float x) {      }
```

```
float sh ( float x )
```

```
{      }
```

va x.k.

2-misol. Uchburchak va beshburchak chizadigan prosedura uchun include yarating va undan foydalaning.

Bloknotda: (**nomi chiz . h bulsin**) # **include <graphics. h >int gd=0, gm;**

```
initgraph (&gd, &gm, " "); void uchb ( int x1, int y1, int x2, int y2, int x3, int y3) { line (x1, y1, x2, y2); line (x1, y1, x3, y3); line (x2, y2, x3, y3 ); }
```

```
void besh ( int a[ ] ) { drawpoly ( 6, a );}
```

Endi bu include dan foydalanamiz:

```
include<chiz. h>
```

```
include<conio. h>{ voidmain ( ) setcolor (4);
```

```
uchb( 10, 20, 100, 80, 250, 150);
```

```
inta[ ] = { 10, 200, 30, 50, 70, 50, 90, 200, 50, 350, 10, 200}; besh (a);
```

```
getch ( ); }
```

Sinovuchunsavollar

1. C/C++ tilida fayllar nima uchun ishlatiladi?
2. C/C++ tilida fayllar qanday xosil kilinadi?
3. Ma'lumotlar bazasini ochish yullarini ko'rsating.
4. C/C++ tilida fayllardan qanday foydalaniladi?

5. Ochiq fayl nima uchun yopilishi zarur?
6. C/C++ tilida ma'lumotlar qanday almashiniladi?
7. Ma'lumotlar bazasidagi massivlardan qanday foydalanish mumkin?
8. Foydalanuvchining direktivasi nima uchun kerak?
9. Foydalanuvchining direktivasi qanday yaratiladi?
10. Foydalanuvchining direktivasidan qanday foydalanish mumkin?

Uygavazifa : C++ tizimida fayllarga doir dasturtuzing va ularni amal yo'tga tatbiqeting.

Adabiyotlar

[6],[7],[8],[9],[10],[11],[13],[14],[15],[16]

Mavzu: Struktura va birlashmalar. Strukturalar. Struktura funktsiya argumenti sifatida. Strukturalar massivi. Strukturlarga ko'rsatkich. Dinamik strukturalar. Birlashmalar va ular ustida amallar.

Reja:

1. Strukturalar.
2. Strukturalar massivi va ko'rsatkichi.
3. Dinamik strukturalar.
4. Birlashmalar va ular ustida amallar.

Ma'ruza bayoni

Strukturali tiplar va strukturalar. Strukturali tip.

Struktura bu turli tipdagi ma'lumotlarning birlashtirilgan tipdir. Struktura xar xil tipdagi elementlar-komponentalardan iborat bo'ladi.. Strukturalar quyidagicha ta'riflanishi mumkin: **struct****strukturali_tip_nomi {Elementlar_ta'riflari}**

Misol uchun ombordagi mollarni tasvirlovchi strukturani kuramiz. Bu struktura quyidagi komponentalarga ega bulishi mumkin:

- Mol nomi (char*)
- Sotib olish narxi (long)
- Ustiga quyilgan narx, foizda (float)
- Mol soni (int)
- Mol kelib tushgan sana (char[9])

Bu struktura dasturda quyidagicha ta'riflanadi:

struct goods

```
{
char* name; long price; float percent; int vol;
char date[9];
}
```

Konkret strukturalar va strukturaga ko'rsatkichlar bu tip yordamida quyidagicha ta'riflanishi mumkin: **goodsfood, percon; goods *point_to;**

Misol uchun:

```
struct complex
{
```

```
double real;  
double imag;  
}
```

Bu misolda kompleks sonni tasvirlovchi strukturali tip **complex** kiritilgan bo'lib, kompleks son haqiqiy kismini tasvirlovchi **real** va **maxjum** kismini tasvirlovchi **imag** komponentalaridan iboratdir. Konkret strukturalar bu xolda quyidagicha tasvirlanadi: **Complex** sigma, alfa;

Quyidagi misolda kasr sonni tasvirlovchi numerator - **sur'at va denominator-maxraj** komponentalaridan iborat struktura ta'rifi keltirilgan.

```
struct fraction; {  
    int numerator; int denominator;  
}
```

Bu xolda konkret strukturalar quyidagicha tasvirlanishi mumkin:
fraction beta;

Konkret strukturalarni tasvirlash.

Strukturalar ta'riflanganda konkret strukturalar ruyxatini kiritish mumkin:
Structstruturali_tip_nomi {Elementlar_ta'riflari} Konkret_strukturalar_ruyxati.

Misol:

```
Struct student  
{  
    char name[15]; char surname[20]; int year;  
} student_1, student_2, student_3;
```

Bu xolda student strukturali tip bilan birga uchta konkret struktura kiritiladi. Bu strukturalar **student ismi (name[15])**, **familiyasi (surname[20])**, **tugilgan yilidan (year)** iborat.

Strukturali tip ta'riflanganda tip nomi ko'rsatilmay, konkret strukturalar ruyxati ko'rsatilishi mumkin:

```
Struct  
    {Elementlar_ta'riflari}  
Konkret_strukturalar_ruyxati.
```

Quyidagi ta'rif yordamida uchta konkret struktura kiritiladi, lekin strukturali tip kiritilmaydi.

```
struct { char processor [10]; int frequency;  
    int memory;  
    int disk;  
} IBM_486, IBM_386, Compaq;
```

Strukturalar uchun xotiradan joy ajratish.

Strukturali tip kiritilishi bu tip uchun xotiradan joy ajratilishiga olib kelmaydi. Xar bir konkret struktura (obyekt) ta'riflanganda, shu obyekt uchun elementlar tiplariga karab xotiradan joy ajratiladi. Xotiradan joy zich ajratilganda struktura uchun ajratilgan joy xajmi xar bir element uchun zarur bulgan xotira xajmlari yigindisiga teng bo'ladi. Shu bilan birga xotiradan joy zich ajratilmasligi xam mumkin ya'ni elementlar orasida bush joylar xam kolishi mumkin. Bu bush

joy keyingi elementni xotira kismlarining kabul kilingan chegaralari buyicha tekislash uchun koldiriladi. Misol uchun butun tipdagi elementlar juft adreslardan boshlansa, bu elementlarga murojaat tezrok amalga oshiriladi. Konkret strukturalarni joylashuviga ba'zi kompilyatorlarda #pragma preprocessor direktivasi yerdamida ta'sir utkazish mumkin. Bu direktivadan quyidagi shaklda:

Pragma pack(n)

Bu yerda n qiymati 1,2 yeki 4 ga teng bulishi mumkin.

Pack(1)- elementlarni bayt chegaralari buyicha tekislash;

Pack(2)- elementlarni so'zlar chegaralariga karab tekislash;

Pack(4)- elementlarni ikkilangan muzlar chegaralariga karab tekislash.

Struktura uchun ajratilgan joy xajmini quyidagi amallar yerdamida aniklash mumkin:

Sizeof(strukturali_tip_nomi);

Sizeof(struktura_nomi);

Sizeofstruktura_nomi.

Oxirgi xolda struktura nomi ifoda deb karaladi. Ifodaning tipi aniklanib, xajmi xisoblanadi.

Misol uchun: **Sizeof** (struct goods) **Sizeof** (tea) **Sizeof** coat

Strukturalarga murojaat.

Konkret strukturalar ta'riflanganda massivlar kabi inisializasiya kilinishi mumkin. Masalan **complex** sigma = {1.3;12.6}; **goods**

coats={ "pidjak",40000,7.5,220,"12.01.97"};

Bir xil tipdagi strukturalarga qiymat berish amalini kullash mumkin:

Complex alfa; **alfa**=sigma;

Lekin strukturalar uchun solishtirish amallari aniklanmagan. Strukturalar elementlariga quyidagicha murojaat qilish mumkin: Struktura nomi.element_nomi.

„Nukta amali' struktura elementiga murojaat qilish amali deyiladi. Bu amal kavs amallari bilan birga eng yukori ustivorlikka egadir. Misol:

Complex alfa={1.2,-4.5},beta={5.6,-7.8},sigma; **Sigma**.real=alfa. real

+beta. real; **Sigma**.imag=alfa.imag+beta.imag;

Konkret strukturalar elementlari dasturda aloxida kiritilishi va chikarilishi zarurdir.

Quyidagi misolda ikki kompleks son qiymatlari kiritilib, yigindisi xosil kilinadi:

```
#include <iostream.h> typedef struct { double real; double imag; } complex;
```

```
void main() {
```

```
complex x,y,z; Cout<<'n'; Cin>>x.real; Cout<<'n';Cin>>x.imag;
```

```
Cout<<'n';Cin>>y.real; Cout<<'n';Cin>>y.imag; z. real=x. real+y. real;
```

```
z.imag=x.imag+y.imag; Cout<<'n'<<z.real;
```

```
Cout<<'n'<<z.imag;
```

```
}
```

Atamalar

Atamaning o'zbek tilida nomlanishi	Atamaning ingliz tilida nomlanishi	Atamaning rus tilida nomlanishi	Atamaning nomlanishi
Algoritm	Algorithm	алгоритм	<p>pedagogik texnologiya tarkibiy bo'laklari (modullari) o'zaro ketma-ket joylashuvining hamda pedagogik texnologiya jarayonlarini amalga oshirishning qat'iy ketma-ketligining belgilangan qoidasi.</p> <p>Belgilangan qonun-qoidalarni bajarish (algoritmni amalga oshirish) da hech qanday ixtiyoriylikka yo'l qo'ymaslik kerak va undan mazkur yo'nalishdagi foydalana olishi kerak.</p> <p>Algoritmning bajarilish jarayonini ma'lum bir bosqichidan (uzluksiz ta'lim tizimida) keyin boshqa bosqichiga ayrim ko'rsatmalar (maqsad va vazifalar) berish yo'li bilan o'tish mumkin.</p> <p>Algoritmni qo'llanish sohasi deb, algoritm natijaviy bo'lgan boshlang'ich ma'lumotlarning eng katta sohasiga aytiladi. Boshqacha aytganda, agar masala shartlari qo'llanilish sohasidan olingan bo'lsa, u holda algoritm bu shartlarni qayta ishlaydi va to'xtash ro'y beradi, agarda shartlar bu sohadan olinmagan bo'lsa, unda yo hech qachon to'xtash ro'y bermaydi, yoki to'xtash ro'y bersa</p>

			ham biz natijasini hisobga olmaymiz.
Algoritmlash	Algorithmization	Алгоритмизация	pedagogik texnologiya tarkibiy bo‘laklari modullarining o‘zaro joylashuvi-ni hamda pedagogik texnologiya jarayonlarini amalga oshirish ketma-ketligi tartibi (qoidasi) ni belgilash jarayoni.
Axborot	Information	информация	agarda tayyorlangan ma’lumotlardan biror narsa (jarayon, hodisa) to‘g‘risida aniqlikni oshirish maqsadida foydalanilsa, u holda foydalangan ma’lumotlarga axborot deb ataladi. Axborotlashtirish axborot jarayonlarini rivojlantirish vosita va sharoitlarining butun majmui bo‘lib, tegishli texnik, ba’zani yaratish, tashkiliy – iqtisodiy, madaniy ma’rifiy, islohatlar qilishni o‘z ichiga oluvchi jarayon bo‘lib, u ta’limni texnologiyalashtirishga keng imkoniyatlar yaratadi.
Axborot izlash	Searching for information	Поиск информации	axborot massivida oldindan belgilangan izlash sharti (so‘rovi) talabini qondiruvchi yozuvlar borligi aniqlash jarayoni va agar ular mavjud bo‘lsa, bunday yozuvlar joylashishini aniqlash jarayoni
Axborotlar banki (AB)	Information Bank	Информационный банк	kishilik faoliyati sohasining yoki ilmiy bilishning ma’lum yo‘nalishi bo‘yicha axborotlarni aniq

		<p>maqsad yo'lida yig'ish, saqlash, tizimlarga ajratish va ommaviy foydalanishga mo'ljallangan texnik, dasturiy, algoritmik tillar va ularni tashkiliy vositalari majmuasi .</p> <p>Axborotlar bankining maqsadi ma'lumotlarni qayta ishlab, undan foydalanishga o'rgatishdan iborat bo'lib, uning vazifasi esa ta'lim-tarbiya jarayonini olib borishda va ta'limni boshqarishda axborot ta'minotini amalga oshirish va ularni boshqarishdagi samaradorlikni nazorat qilishdan iboratdir.</p> <p>AB ta'limni boshqarishda va ta'lim-tarbiya jarayonini ma'lumotlar bilan ta'minlashda hamda jarayonlarni kechishini nazorat qilishga boshqaruv algoritmini tuzish imkoniyatini beradi.</p> <p>AB ta'lim-tarbiya jarayonini boshqarishning axborotli ta'minotini va shular asosidagi kasbiy kompyuterli tizimlar yaratishda foydalaniladi.</p> <p>Axborotlashtirilgan jamiyatning har bir bo'g'inida AB ga murojaat qilinadi va shu asosda zamonaviy ta'lim texnologiyalari yaratiladi va ta'lim-tarbiya jarayonlarini optimal boshqarishga imkoniyat yaratiladi.</p>
--	--	--

Axborot texnologiyalari	Information technology	Информационные технологии	Jarayonlarni boshqarishdagi asosiy manba bo'lib hisoblanadi. Ular ilmiy xodim tomonidan tayyorlangan ma'lumotlarni tizimlarga ajratish va qayta ishlash hamda kerakli maqsadlar bo'yicha qayta ishlangan ma'lumotlarni uzatish algoritmlarini o'z ichiga oladi. SHuning uchun axborot texnologiyalaridan ratsional foydalanish ilmiy xodimlar mehnati samaradorligini oshiradi.
Bayt	Byte	Байт	Komp'yuter xotirasi o'lchovi birligi.
Bit	Bit	бит	Bir yoki nol qabul qiluvchi o'zgaruvchi
Bufer	Buffer	буфер	Kiritish chikarish amallari bajarilish vaqtida vaqtinchalik berilganlarni saqlash uchun ishlatiladigan xotira qismi.
Kiritish chiqarish qurilmasi	Input-output device	Устройство ввода-вывода	Kompyuter va tashqi qurilmalar o'rtasida ma'lumot almashinuvini ta'minlab beruvchi qurilma.
Operativ xotira	RAM memory	Оперативный память	Programma bajarilishi uchun vaqtincha foydalaniladigan xotira bo'lib programma bajarilish tezligini aniqlaydi.
Protessor	Processor	процессор	Kompyuter kurilmalarini boshqaruvchi va berilganlarni qayta ishlovchi qurilma
Registr	Registr	Регистр	Xotiraning tez murojaat qilish mumkin bo'lgan qism bo'lib, xajmi operativ xotiradan kichik

			bo'ladi.
Xotira segmenti	Memory segment	Сегмент памяти	Operativ xotira qismi bo'lib hajmi(o'lchami) o'zgaruvchi bo'ladi.
EHM	Electronic computing machine	Электронная вычислительная машина	Elektron hisoblash mashinasi

Atama	O'zbek tilidagi sharhi	Ingliz tilidagi sharhi
!=	Teng emas operatori; mantiqiy inkor amali qiymati bilan birhil.	The inequality operator; compares values for inequality returning a bool.
#define	Makro deriktivalarni belgilash	a directive that defines a macro.
#include	Bir source fayl ichida boshqa bir faylag murojatni amalga oshirish mexanizmi	a mechanism for textual inclusion of one source file into another.
+=	add-and-assign operatori; masalan a+=b vazifazi jihatdan a=a+b bilan bir xil	add-and-assign operator; a+=b is roughly equivalent to a=a+b.
.c file	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
.cpp file	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
.h file	Sarlavha fayli	header file
address	Hotira manzili	a memory location
aggregate	Konstruktorsiz massiv yoki struktura	an array or a struct without a constructor
algorithm	Hisoblashning aniq ketma-ketligi	a precise definition of a computation.
and	&& mantiqiy "va" (ko'paytirish) operatori sinonimi	synonym for &&, the logical and operator.
ANSI	Amerika milliy standart agentligi.	The American national standards organization.
application	Umumiy maqsadga ega dasturlar to'plami	a collection of programs seen as serving a common purpose (usually providing a common interface to their users)
bit	0 yoki 1 qiymatga ega birlik hotira	a unit of memory that can hold 0 or 1.
bool	Mantiqiy tip. Bu tip faqat rost yoki yolg'on qiymat	the built-in Boolean type. A bool can have

	qabul qiladi	the values true and false.
Borland C++ Builder	C++ tilida visual dasturlashga ihtisoslashtirilgan IDE muhiti.	Borland's implementation of C++ together with proprietary libraries for Windows programming in an IDE
bug	Xatolik termini	colloquial term for error.
byte	Xotiradagi bir nechta belgilar yig'indisi	a unit of memory that can hold a character of the C++ representation character set.
C++	Tizimli dasturlashni protsedurali qo'llab quvvatlovchi dasturlash tili.	a general-purpose programming language with a bias towards systems programming that supports procedural programming, data abstraction, object-oriented programming, and generic programming. C++ was designed and originally implemented by Bjarne Stroustrup.
char	Belgili tip. Har bir belgi 8 bit, ya'ni baytga teng.	character type; typically an 8-bit byte.
char*	Char massiviga ko'rsatkich	pointer to a char or an array of char. Typically assumed to point to a C-style string.
cin	Standart kiritish oqimi	standard istream.
class	Foydalanuvchi belgilaydigan tur, sinf. Sinf foydalanuvchi funksiyasi, foydalanuvchi ma'lumotlari va kontentlari bo'lishi mumkin.	a user-defined type. A class can have member functions, member data, member constants, and member types.
compiler	C++ da yozilgan buyruqlarni mashina tiliga o'girib beruvchi vosita.	the part of a C++ implementation that produces object code from a translation unit.
const	Faqatgina bir marta qiymat berish mumkin bo'lgan o'zgaruvchilarni e'lon qilsh.	attribute of a declaration that makes the entity to which it refers readonly.
copy()	Nusxa olish operatori	Copy operator
UML	Унификация қилинган моделлаштириш тили	Unified Modeling Language
OMG	Объектларни бошқариш	Object Management Group

	гурухи	
4GL	Тўртинчи авлод тили	Fourth-Generation Language
ANSI	Америка миллий стандартлаш институти	American National Standards Institute
AMPS		Advanced Mobile Phone Service
ERP	Корхона ресурсларини режалаштириш	Enterprise Resource Planning
CRM	Мижозлар билан ўзаро муносабатларни бошқариш	Customer Relations Management
SQL	Тузилмалашган сўровлар тили	Structured Query Language
OLAP	Хақиқий вақтда маълумотларга аналитик ишлов бериш	On-Line Analytical Processing
OLTP	Хақиқий вақтда транзакцияларга ишлов бериш	On-Line Transaction Processing
TCO	Эгалик қилишнинг ялпи қиймати	Total Cost of Ownership
JIT	Айни вақтида	Just-In-Time
LAN	Локал ҳисоблаш тармоғи	Local Area Network
MAN	Маҳаллий ҳисоблаш тармоғи	Metropolitan Area Network
WAN	Худудий ҳисоблаш тармоғи	Wide Area Network
ISO	Ҳалқаро стандартлаштириш ташкилоти	International Organization for Standardization
API	амалий дастурлаштириш махсус интерфейси	Application Programming Interface
WWW	Умумжаҳон ўргамчак тўри	World Wide Web
ASCII	Ахборот алмашишнинг Америка стандарти	American Standard Code for Information Interchange
LIFO	«Охирида келди, биринчи кетди» принципи	Last In, First Out
FIFO	«Биринчи келди, биринчи кетди» принципи	First In, First Out
PDA	персоналрақамликотиб	Personal Digital Assistant

Foydalanilgan adabiyotlar

1. A. R.Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O`quv qo`llanma. T.:“Cho`lpon”, 2013 y.
2. Sh.I.Razzoqov, M.J.Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o`quv qo`llanma. T.: “Ilim Ziyon”, 2011y.
3. T.X.Holmatov, N.I.Toyloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta`minoti.O`quv qo`llanma. T.: “Mehnat”, 2000 y.
4. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T.:“O'qituvchi”, 2008 y.
5. M.Ashurov, M.Mirmaxmudov, SH.Sapaev.Zamonaviy Programmash asoslari fanidan laboratoriya ishlari.T.:TDPU, 2008y.
6. Страуструп Б. Язык программирования С++. Специальное издание. – М.: Бинам – ПРЕСС, 2006, 1104 с
7. Павловская Т. С++ Программирование на языке высокого уровня. – СПб. : Питер, 2005. -461 с.
8. Мадрахимов Ш., Гайназаров С. С++ тилида программалаш асослари: услубий кулланма – Тошкент: УзМУ 2009 . – 196 б
9. Madrahimov Sh. va boshq. С++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma. - Toshkent: O'ZMU, 2014.-160 b
- 10.П.Дарахвелидзе, Э.Марков. Программирование в С++ Bulider7. Учебник.Санкт-Петербург, “БХВ-Петербург” 2003 г.
- 11.В.М.Пестиков, А.Н.Маслобоев. Turbo PASCAL 7.0. Изучаем на примерах. Санкт-Петербург.: “БХВ-Петербург”, 2004 г.
- 12.Фаронов В.В. Программирование на языке высокого уровня С++ Bulider. Учебник. М.: “Питер”, 2003 г.
- 13.В.Т.Безручко. Практикум по курсу информатики. М.: «Финансы и статистика», 2004 г.
- 14.Культин Н. С ++ Builderв задачах и примерах. – СПб. : ВХБ, 2005.-336с
- 15.Ахатов А., Qarshiyev Н. С++ tilida programmalash(misollar va masalalar). 2016.SamDU nashri. 140b

Elektron ta'lim resurslari

7. Программирование в С++ Bulider. Учебник. А.Я.Архангелский ([www.ozon.ru /context /detail /id /2705337/](http://www.ozon.ru/context/detail/id/2705337/))
8. www.pedagog.uz
9. www.Ziyonet.uz
10. www.edu.uz
11. <http://www.borlpasc.narod.ru/>.
12. [http:// www.intuit.ru](http://www.intuit.ru)

Mavzular bo'yicha taqdimotlar, mustaqil ta'lim uchun materiallar

Programmash asoslari» fanidan amaliy va laboratoriya mashg'ulotlar uchun masala va mashqlar to'plami

1. Chiziqli va tarmoqlanuvchi algoritmlar va dasturlar tuzish

1-topshiriq

$$a) a = u^{(x+y)/2} - \sqrt{\frac{x-1}{y+1}} \quad b = \sin(2 \arccos t), \text{ bu yerda } x = 12,65; \quad y = -2,255;$$
$$u = 3,205, \quad t = 0,88$$

$$b) K_2 = \begin{cases} \sqrt{15a^2 - 21b^2}, & \text{agar } a > b \\ \sqrt{15b^2 - 21a^2}, & \text{agar } a \leq b \end{cases}$$

2-topshiriq

$$a) a = (2 + y^2) \frac{x + y/2}{y^2 + 1/(1 + y^2)}, \quad b = \sqrt{\sin v + \cos t} \quad \text{bu yerda } x = 0,222,$$
$$y = -6,72, \quad v = 10,05, \quad t = 0,35$$

b) Tomonlari bilan berilgan uchburchakning teng tomonli bo'lishini aniqlash algoritmi va dasturini tuzing.

3-topshiriq

$$a) a = \frac{1}{2}(x^{y-x} + y^{(x+y)/2}), \quad b = \lg(\sqrt[3]{u} + \sqrt{v+2}), \text{ bu yerda}$$

$$x = 3,225, \quad y = 2,98, \quad u = 125,33, \quad v = 33,075$$

$$b) S = \max(x, y, z) + \min(a, b, c)$$

4-topshiriq

$$a) a = \sqrt[3]{x + \sqrt[4]{y}}; \quad b = \sqrt{y} l^{-(y+4/2)}, \text{ bu yerda } x = 37,15; \quad y = 12,55; \quad l = 20,12.$$

$$b) z = \begin{cases} (2x + y + 3)/3, & \text{agar } x + y > 0 \text{ va } x > 0 \\ \sqrt{x^2 + 2y^{2+3}}, & \text{agar } x + y > 0 \text{ va } y < 0 \\ (x^2 + y^2)/x, & \text{qo'lgan hollarda} \end{cases}$$

5-topshiriq

$$a) a = \sqrt{|\cos^2 x^2 + y^2|}, \quad b = \frac{e^{3xy} + \ln|3x^2 - y|}{\sqrt{x^2 + y^2}}, \text{ bu yerda } x = 1,42, \quad y = 2,035$$

$$b) Z = \begin{cases} \log_3 x^2 + \sqrt[3]{x^3}, & \text{agar } x > 2 \\ \sqrt{x^2 + 16} - e^{2x}, & \text{agar } x = 2 \\ a^x + |x^3|, & \text{agar } x < 2 \end{cases}$$

2. Takrorlanuvchi algoritmlar va dasturlar tuzish

1. $S = \sum_{n=1}^5 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblash algoritmini tuzing.

2. 1 dan n gacha toq sonlar kvadratlari yig'indisini hisoblash algoritmini tuzing.

3. $P = \prod_{i=1}^{10} \frac{i^2 + ai}{i^2 + a^2} + \prod_{i=2}^4 \frac{k^3}{k + a}$ ni hisoblash algoritmini tuzing.
4. [a,b] oraliqda m soniga karrali sonlar ko'paytmasini hisoblash algoritmini tuzing.
5. $P = \sum_{i=1}^8 \frac{i^2 + ai}{i^2 + a^2}$ ni hisoblash algoritmini tuzing.
6. 1 dan 35 gacha bo'lgan toq sonlar kvadratlarining yig'indisi va juft sonlar ko'paytmasini hisoblash algoritmini tuzing.
7. Berilgan son raqamlari yig'indisini hisoblash algoritmini tuzing.
8. $S = \prod_{k=1}^{\infty} \frac{k^2}{x^k}$ ni eps = 0,0001 aniqlik bilan hisoblash algoritmini tuzing.
9. $S = \prod_{n=1}^6 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblash algoritmini tuzing.
10. 2 dan 50 gacha 4 ga va 3 ga bo'linadigan sonlarni chop etish algoritmini tuzing.

3. Ichma-ich joylashgan algoritm va dastur tuzing

1. $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblash algoritmini tuzing.
2. $S = \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^4)$ ni hisoblash algoritmini tuzing.
3. $S = \prod_{k=1}^4 \sum_{j=1}^{10} (k! + j!)$ ni hisoblash algoritmini tuzing.
4. $S = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{(k! + j!)}{a + b}$ ni hisoblash algoritmini tuzing.
5. $S = \prod_{k=1}^3 (a \prod_{n=1}^4 kn^2 + b \sum_{i=1}^5 ki^3)$ ni hisoblash algoritmini tuzing.
6. $S = \prod_{k=1}^4 \frac{\sum_{j=1}^{10} j!}{k!}$ ni hisoblash algoritmini tuzing.
7. $P = \prod_{R=1}^{15} \prod_{i=1}^{10} \frac{R^i + 1}{R^4 + 3^i * R + e^{-R}}$ ni hisoblash algoritmini tuzing.
8. $S = \sum_{R=1}^{14} \sum_{m=1}^4 \frac{R \cdot m + |R^{-m} + 2|}{\ln R + 3m}$ ni hisoblash algoritmini tuzing.
9. $P = \prod_{R=1}^{16} \prod_{i=1}^6 \frac{R + 3}{R^3 + 3R + i^3}$ ni hisoblash algoritmini tuzing.
10. $S = \sum_{R=1}^{17} \prod_{m=1}^5 \sqrt{\frac{R + m^3 + e^{-m} + 9}{\text{Log}_m R + (mR)^3}}$ ni hisoblash algoritmini tuzing.

4. C++ Bulider va C++ tilida miqdorlar bilan ishlash

1. Quyidagi yozuvlardan qaysilarini butun o'zgarmas deb qarash mumkin.

- 1) 37 6) -18 11) +46 16) 37.

- | | | | |
|-------------------|-------------|----------------------|-------------|
| 2) 0 | 7) +0 | 12) -0 | 17) 00.00 |
| 3) -18.0 | 8) 0018 | 13) 10+2 | 18) 3010-1 |
| 4) $4 \cdot 10^2$ | 9) 38/2 | 14) 7^2 | 19) -0000 |
| 5) 00008 | 10) 17845,3 | 15) $0,6 \cdot 10^3$ | 20) 1010.00 |

2. Quyidagi yozuvlardan qaysilarini haqiqiy o'zgarmas deb qarash mumkin.

- | | | | |
|-----------|-----------------------|----------------------|-------------|
| 1) -0.00 | 6) +34,00 | 11) +46 | 16) 378. |
| 2) +17.77 | 7) 2743 | 12) +13,45 | 17) 02.00 |
| 3) .13 | 8) -0015 | 13) 10+2 | 18) 40-1 |
| 4) -.1845 | 9) $13 \cdot 10^{-2}$ | 14) 7.00 | 19) -0001 |
| 5) 3,49 | 10) .00 | 15) $0,6 \cdot 10^3$ | 20) 1211.10 |

3. Quyidagi yozuvlardan qaysilarini siljuvchi nuqtali o'nlik o'zgarmas deb qarash mumkin.

- | | | | |
|----------|-----------|------------|-------------|
| 1) 5E1 | 6) 5.E-2 | 11) 2E0.2 | 16) 37. |
| 2) 0E-2 | 7) 1.1E+1 | 12) -5E.1 | 17) 00.00 |
| 3) 3E | 8) 5E/2 | 13) 666E+0 | 18) 3010-1 |
| 4) +50.5 | 9) 3+2E+2 | 14) 320 | 19) -0000 |
| 5) -E+3 | 10) 4E/2 | 15) -.1E1 | 20) 1010.00 |

5. C++ Buliderda miqdorlar bilan ishlash

4. Quyidagi yozuvlardan qaysilarini fiksirlangan nuqtali o'nlik o'zgarmas deb qarash mumkin.

- | | | | |
|----------|-----------|-----------|-------------|
| 1) 23 | 6) `1` | 11) -3.01 | 16) 37. |
| 2) 23. | 7) ++5 | 12) 256 | 17) 00.00 |
| 3) .23 | 8) 3.14 | 13) -1.0 | 18) 3010-1 |
| 4) -23.0 | 9) 1/2 | 14) 1A2 | 19) -0000 |
| 5) 00 | 10) 36.27 | 15) .0 | 20) 1010.00 |

5. Quyidagi yozuvlardan qaysilarini identifikator deb qarash mumkin.

- | | | | |
|-----------|----------|-----------|-------------|
| 1) X025Y | 6) 25A | 11) XXZ1 | 16) S37. |
| 2) 4ab | 7) GRAF | 12) X**Y | 17) 00R00 |
| 3) V10SKL | 8) A/BS | 13) R_X_Y | 18) TT3010 |
| 4) KOL_A | 9) `ABS` | 14) Z 12 | 19) -ZZ0000 |
| 5) A*4B | 10) R32 | 15) A'R' | 20) 1010.00 |

6. Quyidagi sonlarni yozing:

- a) $432 E - \text{Ç}$; á) $\text{ÇE}2$; â) $-0.00027E + 4$; ã) $2\text{Å} - \text{Ç}$; ä) $1\text{Å}2$

7. Hisoblang:

- a) $24/(3x4) - 24/3/4 + 24/3x4$; b) $22/2*3 - 25*4/2 + 28/(2*7)$;
c) $4*(15-5)/5*6/3 - 24/6$. d) $7 \bmod 2 + 21 \text{div} 6 \bmod 2$

8. D=5; E=4 va A="rost", B="YOLG'ON" qiymatlarga ega bo'lsa, quyidagi amallar natijasida qanday qiymat hosil bo'ladi?

- 1) (E>8) AND (D>8) AND (NotB)
- 2) (E=D) AND A AND B
- 3) (E<D) AND A AND (NotB)

9. Quyidagi $(-3>5)$ Or Not($7<9$) And ($0\leq 3$) mantiqiy ifodada operatsiyalar bajarilishi tartibini aniqlang va qiymatini toping.

6. Ifodalarni C++ Bulider va C++ tilida yozing

$$1. y = -2\sqrt{x^2 + \frac{4x^2}{3}} - \frac{e^{4+x}}{x+1} \quad 2. Z = -\frac{x-1}{\sqrt{x^2 - A^2}} - \frac{2A^2}{3(\sqrt{x^2 - A^2})^3} \quad 3. z(x, y) = x \frac{e^{x^2-y^2} - 1}{\sqrt{|x^2 + y^2|}}$$

$$4. W = 1 - \frac{1}{\frac{1 + \left(\frac{RG}{K}\right)^2}{e^2}} + \text{Ln}\left(\frac{RG}{K}\right) \frac{1}{S^2 \left(1 + \frac{RG}{L}\right)^2}$$

$$5. V = (2p)^{1/2} x^{x+1} e^{-x} e^{-\sqrt{\frac{W}{2px}}} \quad 6. R = \frac{1}{\text{Cos}x} + \ln\left|\text{tg}\frac{x}{2}\right| + \frac{A+Bx}{C+Dx}$$

$$7. y = 2,5\sqrt[3]{x^4} + \log_4^2 x$$

$$8. y = \sec x + \sqrt[4]{e^3} + \frac{x}{2}$$

$$9. u = \sqrt{e^x} \sqrt{\text{ctg}^2\left(\frac{x}{4}\right)} + 2\ln x$$

$$10. u = \sqrt{\ln x} + 2|e^{x+2} - 5x|$$

$$11. u = \sqrt{\text{tg}^2\left(\frac{x}{5}\right)} + 2e^x$$

$$12. u = \sqrt{e^x} \sqrt{\text{Sin}\left(\frac{x}{2}\right)} + 2|x-1|$$

$$13. u = \sqrt{\ln x} \sqrt{\text{Sin}\left(\frac{x}{2}\right)} + 2|e^x - 1|$$

$$14. u = \sqrt{\text{tg}^2\left(\frac{x}{2}\right)} + |2e^x|$$

7. Ifodani qiymatini hisoblang

a) $x = 0.4$; $y = -1,5$; bo'lsa $SQR(x) + SQR(y) \leq 4$

b) $k = 15$ bo'lsa $k \bmod 7 = k \text{div} 5 - 1$;

c) $p = 0,182$ bo'lsa $\text{odd}(\text{trunc}(10 * p))$;

d) $t = \text{true}$, $\rho = 101010$ bo'lsa $\tan d(p \bmod 3 = 0)$;

e) $x = \text{true}$; $y = \text{false}$ bo'lsa, $x \text{and} y \text{or}(\text{not}(x)) \text{and} y$

f) $n = 0$ bo'lsa $\text{notodd}(n)$;

8. C++ Buliderda ta'minlash operatori

1. Ifodalarni ta'minlash operatoridan foydalanib yozing

$$1. u = \sqrt{\ln x} + 2|e^{x+2} - 5x|$$

$$2. u = \sqrt{e^x} \sqrt{\text{Sin}\left(\frac{x}{2}\right)} + 2|x-1|$$

$$3. y = e^{\sin x^2} + \frac{x}{2e^{x-1}}$$

$$4. y = 2,5\sqrt{x^4} + \text{Sin}\frac{4}{x}$$

$$5. u = \sqrt{\ln x} + 2|e^{x+2} - 15|$$

$$6. u = \sqrt{\ln x} + 2|e^{x+2} - 5x|$$

2. Operatorning bajarilishida x o'zgaruvchi qanday qiymat qabul qiladi?

- a) $x := 5; x := x + 2; x := \text{SQR}(x) - x;$
 b) $x := 2; y := 4; x := x + y; y := x; x := y + 3;$
 c) $x := 3; y := x; x := x + y$
 d) $x := 6; y := 2; y = x / y; x := y; x := x + y;$

9. C++ Buliderda ifoda

1. $b = 1,0$ va $c = -2,0$ berilgan bo'lsa, dasturning bajarilishdan nima hosil bo'ladi.

Prograt ifoda:

Var b,c,d: real;

```
begin read (b,c); d: = SQRT (SQR(b) - 4+a+c);
  writeln ('x1=', (-b+d/2), 'x2=', (-b-d(2)));
end
```

2. Dastur natijasi qanday bo'ladi?

Program ifodasi ;

Var x,y,z: real;

beginx:=3; y:=2; z:=x*x*y+y; x:=c2+y

z=x+y; writeln (^z=1,z); end.

3. Agar 1,2 va 3 sonlari berilgan bo'lsa, dastur natijasi qanday bo'ladi?

Program son;

Var x,y,: integer;

beginreal (x,y,z) writeln (x,y,x); end.

4. Dastur natijasi nima bo'ladi?

Program o'zgarmas;

Var a:real;

begina:=3; writeln (^a+1`);writeln `a=`, a+3)

5. Agar $b=1.0$ va $c=-2.0$ bo'lsa, quyidagi dastur natijasini aniqlang:

program ildiz (input, output);

var b,c,d:real;

begin

read (b,c);

d:=b*b-4*b*c;

writeln('x1=',(-b+sqrt(d))/(2*a));

writeln('x2=',(-b-sqrt(d))/(2*a))

end.

6. Quyidagi dastur uchun $a=5.0$ va $b=6.2$ bo'la oladimi ?

program masala;

var a: integer; b: real;

begin

readln(a,b); writeln (a*b) end.

7. Quyidagi dastur uchun $k=1$ va $m=-12,5$ bo'la oladimi ?

program masala;

var k:integer ; l,m: real;

begin

readln(k,m); l:=m/(k-1) writeln (l) end.

8. Berilgan dasturlardagi xatoliklarni aniqlang :

a) program a;

```
const d=5;
```

begin

```
d:=sqr(d); writeln('d**2=',d) end.
```

b) program b ; var a,b,c: integer;

```
begin read (a,b) ; c:=a/b writeln('c=',c) end.
```

c) program b ;

```
var x:real;
```

```
begin read (x) ; y:=sqr(x)+1 writeln('y=',y) end.
```

10. C++ Buliderda chiziqli dastur tuzing

1. Kvadratning tomonlari a berilgan. Uning perimetrini toping $P=4 \cdot a$.

2. Kvadratning tomonlari a berilgan. Uning yuzasini toping $P=a^2$.

3. To'g'ri to'rtburchak tomonlari a va b berilgan. Uning perimetrini $P=2 \cdot (a+b)$ va yuzasini toping $P=a \cdot b$.

4. Aylana diametri d berilgan. Uning uzunligini toping $L=\pi \cdot d$.

5. Kubning tomonlari a berilgan. Uning hajmini $V=a^3$ va sirti maydonini toping $P=6 \cdot a^2$.

6. To'g'ri burchakli paralelopipedning a , b , s tomonlari berilgan. Uning hajmini va sirti maydonini toping: $V=a \cdot b \cdot s$; $S=2 \cdot (a \cdot b + b \cdot s + a \cdot s)$.

7. Aylana radiusi R beilgan. Aylana uzunligi L va uning ichi maydoni S ni toping: $L=2 \cdot \pi \cdot R$, $S=\pi \cdot R^2$. Bu erda π sonini 3.14 ga teng qilib oling.

8. Ikkita a va b haqiqiy sonlari berilgan. Ularning o'rta arifmetik qiymatini toping: $(a+b)/2$.

9. Ikkita a va b haqiqiy musbat sonlari berilgan. Ularning o'rta geometrik qiymatini toping: $\sqrt{a \cdot b}$.

10. Ikkita a va b nol qiymatga teng bo'lmagan sonlar berilgan. Ularning yig'indisi, ayirmasi, ko'paytmasi va bo'linmasini toping.

11. C++ Buliderda hisoblash dasturini tuzing

1. Berilgan butun son musbat bo'lsa unga 1 qo'shilsin, aks holda o'zgartirilmasin. Natijani chiqaring.

2. Berilgan butun son musbat bo'lsa unga 1 qo'shilsin, aks holda 2 ni ayrib tashlansin. Natijani chiqaring.

3. Berilgan butun son musbat bo'lsa unga 1 qo'shilsin, manfiy bo'lsa 2 ayrib tashlansin, nol bo'lsa, uni 10 ga o'zgartirib natijani chiqaring.

4. Uchta butun son a , b , s berilgan. Ularning $a+b$, $a+s$, $b+s$ yig'indilari ichidan kattasini aniqlang.

5. Berilgan ikkita sonning eng kichigi tartib raqamini chiqaring.

6. Berilgan ikkita sondan avval kattasi, so'ng kichigini chiqaring.

7. Berilgan uchta son orasidan eng kichigini aniqlang va chiqaring.

8. Berilgan uchta son orasidan avval eng kichigi, so'ng kattasini chiqaring.

9. Berilgan 3 ta son orasidan 2 kattasining yig'indisini aniqlang.

10.A va V butun o'zgaruvchilar berilgan. Ular qiymati teng bo'lmasa har bir o'zgaruvchiga kattasining qiymati o'zlashtirilsin, aks holda o'zgaruvchilar 0 ga tenglashtirilsin. A va V o'zgaruvchilarning yangi qiymatlarini chiqaring.

12. Tarmoqlanuvchi jarayonlarga dastur tuzing

1. Tekislikda nuqta berilgan. Agar nuqta koordinata o'qi boshiga mos tushsa, 0 ni chop eting. Agar nuqta koordinata o'qi boshiga mos tushmasa 0x yoki 0y o'qida yotsa, mos ravishda 1 yoki 2 ni chiqaring. Agar nuqta koordinata o'qida yotmasa 3 ni chiqaring.
2. Ox va Oy koordinata o'qida yotmagan nuqta berilgan. Nuqta qaysi chorakda joylashganligini aniqlang.
3. Butun sonlar berilgan. Uni so'zda satr ko'rinishida "manfiy juft son", "0 ga teng son", "musbat toq son" va h.k ko'rinishida chiqaring.
4. 1-999 oraliqdagi butun son berilgan. Agar u juft bo'lsa "juft son", toq bo'lsa "toq son" so'zlarini qator ko'rinishida chiqaring.
5. Funktsiyani hisoblang

$$D = \begin{cases} x_1 - x_2, & \text{agar } \lg 16 > x_3, \\ (x_1 - x_2)^2, & \text{agar } \lg 16 \leq x_3. \end{cases} \quad x_1 = -19.54; x_2 = -1.938; x_3 \in R.$$

6. Funktsiyani hisoblang

$$F = \begin{cases} \sqrt{|x_1 + x_2|}, & \text{agar } 2^{x_1} < 1, \\ x_3 - x_4^2, & \text{agar } 2^{x_1} \geq 1. \end{cases} \quad x_1 = -19.54; x_2 = -1.938; x_3 = 18.11; x_4 = 2.804.$$

7. Quyidagi avtomobil markalari berilgan: MAZ; GAZ; UAZ; DAMAS; AZLK. Mos ishlab chiqaradigan shaharlar nomini chiqaruvchi dastur tuzing (Minsk, Nijniy Novgorod, Ulyanovsk, Andijon, Moskva).
8. Quyidagi davlatlar domen Internet adresi berilgan: Uz, Us, Ru, Kz, Fr. Mos davlatlar nomini chiqaruvchi dastur tuzing (Uzbekiston, Amerika, Rossiya, Qozog'iston, Fransiya).
9. 1 dan 20 gacha sonlar yig'indisini hisoblash algoritmi va dasturini tuzing. Dasturda sikl operatori ishlatilmasin.
- 10.1 dan 10 gacha sonlar ko'paytmasini hisoblash algoritmi va dasturini tuzing. Dasturda sikl operatori ishlatilmasin.

13. C++ Buliderda dastur tuzing

1. K va N (N>0) butun sonlari berilgan. K sonini N marta ekranga chiqaring.
2. Ikkita A va V (A<V) butun sonlari berilgan. A va V oralig'ida joylashgan sonlarni o'sish tartibida (A va V larni ham hisobga olgan holda) ekranga chiqaring, hamda ular sonini hisoblang.
3. Ikkita A va V (A<V) butun sonlari berilgan. A va V oralig'ida joylashgan sonlarni kamayish tartibida (A va V larni ham hisobga olgan holda) ekranga chiqaring, hamda ular sonini hisoblang.
4. Bir kilogramm konfet narxini ifodalovchi haqiqiy son berilgan. 1 kg, 2 kg,...10 kg konfetlar baholarini hisoblang.
A) 0.1, 0.2,...1 kg. konfetlar narxlarini hisoblab chiqaring.

B) 1.2, 1.4, ...2 kg konfetlar narxlarini hisoblab chiqaring.

5. Ikkita A va V ($A < V$) butun sonlari berilgan. A va V sonlar orasidagi barcha butun sonlar yig'indisini toping (A va V larni ham hisobga olgan holda).

6. Ikkita A va V ($A < V$) butun sonlari berilgan. A va V sonlar orasidagi barcha butun sonlar kvadrati yig'indisini toping (A va V larni ham hisobga olgan holda).

7. Butun N va K sonlari berilgan. Quyidagi yig'indini toping
$$N^2 + (N+1)^2 + (N+2)^2 + \dots + (N+K)^2$$

8. Butun $N > 0$ soni berilgan. Quyidagi N ta ko'paytuvchilar ko'paytmasini toping.

$$1.1 * 1.2 * 1.3 * \dots$$

9. Butun $N > 0$ soni berilgan. N ta ishorasi almashuvchi yig'indilardan iborat ifoda qiymatini toping (Shartli o'tish operatori ishlatilmasin).

$$1.1 - 1.2 + 1.3 - \dots$$

10. Butun $N > 0$ toq soni berilgan. Quyidagi formula orqali hisoblanuvchi sonning kvadratini toping.

$$1 + 3 + 5 + \dots + N$$

14. C++ Buliderda takrorlanuvchi jarayonlarga dastur tuzing

1. Cheksiz qatorning hadlari sonini va yig'indisini $\varepsilon = 10^{-5}$ aniqlikda hisoblang

$$\sum_{n=0}^{\infty} \frac{1}{2^n} = 1 + \frac{1}{2} + \frac{1}{4} + \dots$$

2. Cheksiz qatorning hadlari sonini va yig'indisini $\varepsilon = 10^{-4}$ aniqlikda hisoblang

$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

3. Cheksiz qatorning hadlari sonini va yig'indisini $\varepsilon = 10^{-5}$ aniqlikda hisoblang

$$\sum_{n=0}^{\infty} (-1)^i \frac{i}{2i+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

4. Raqamlari yig'indisi kubiga teng bo'lgan barcha uch xonali sonlarni toping.

5. 2,3,4,5,6,7,8,9 sonlariga ko'paytirilganda raqamlarining yig'indisi o'zgamaydigan ikki xonali sonlarni toping.

6. Raqamlari yig'indisi berilgan butun songa teng uch xonali barcha sonlarni toping.

7. Barcha shunday uch xonali sonlarni toping-ki, ularni kvadratga ko'targanda uchata raqami bir xil bo'lsin, nol bundan mustasno.

8. Do'konda 16 kgli, 17 kgli va 21 kgli pol buyoqlari yashiklarda mavjud. Yashiklarni ochib qaramasdan 185 kg pol buyog'i tanlang. Barcha variantlarni qarang.

9. $42 * 4^*$ yulduzchalar o'rniga shunday raqamlarni tanlang, hosil bo'lgan besh xonali son 72 ga bo'linsin.

10. O'zi va raqamlari yig'indisi 7ga karrali uch xonali sonlarni toping.

15. Icma-ich sikllarga hisoblash dasturini tuzing

1. $S = \sum_{n=1}^{10} \prod_{k=1}^5 (a^k + b^n)$ ni hisoblang.

2. Ko'paytirish jadvali elementlarini hosil qilish dasturini tuzing.

$$3. S = \sum_{i=1}^3 \frac{(\sum_{j=1}^{10} j + \prod_{k=1}^4 \frac{1}{k})}{i!} \text{ ni hisoblang.}$$

$$4. S = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{(k! + j!)}{a + b} \text{ ni hisoblang.}$$

$$5. S = \sum_{k=1}^3 (\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3) \text{ ni hisoblang.}$$

$$6. S = \sum_{k=1}^3 (\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3) \text{ ni hisoblang.}$$

$$7. S = \prod_{k=1}^4 \sum_{j=1}^{10} (k! + j!) \text{ ni hisoblang.}$$

$$8. S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n) \text{ ni hisoblang.}$$

$$9. S = \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^4) \text{ ni hisoblang.}$$

$$10. S = \sum_{n=1}^5 \prod_{i=1}^8 (n^2 + i^3) \text{ ni hisoblang.}$$

16. Massivlar bilan ishlash

1. $A(n)$ massiv berilgan. Uning elementlarining eng kattasi va eng kichigini topish algoritmi va dasturini tuzing

2. $A(n)$ massiv berilgan. Massiv elementlarini o'sib borish tartibida joy-lashtirish algoritmi va dasturini tuzing.

3. $A(n)$ massiv berilgan. Uning elementlarining eng kattasini 1-soni bilan, eng kichigini esa -1 soni bilan almashtirish algoritmi va dasturini tuzing.

4. $A(n)$ massiv berilgan. Uning elementlarining eng kattasi va eng kichigini hamda ularning o'rnini aniqlash algoritmi va dasturini tuzing.

5. $A(n)$ massiv berilgan. Uning manfiy, musbat va nol elementlari sonini aniqlash algoritmi va dasturini tuzing.

6. x_1, x_2, \dots, x_{55} massiv berilgan: $x_1(x_2+x_3) \cdot (x_4+x_5+x_6) \dots (x_{46}+x_{47}+\dots, x_{55})$ ni hisoblash algoritmi va dasturini tuzing.

7. x_1, x_2, \dots, x_n massiv berilgan. Uning toq elementlaridan U , juft elementlaridan Z massiv hosil qilish algoritmi va dasturini tuzing.

8. $A(n)$ massiv berilgan. Uning birinchi va oxirgi elementlari o'rnini almashtirish algoritmi va dasturini tuzing.

9. $A(n)$ massiv berilgan. Uning manfiy elementlarining eng kattasini aniqlash algoritmi va dasturini tuzing.

10. A va B ikkita massivning elementlari butun sonlarda iborat. A massivning toq elementlardan tashkil topgan, lekin B massivning elementi bo'lmagan C massivni hosil qiling.

17. Ikki o'ldovli massivlar bilan ishlash

1. $A(N,N)$ massivning barcha musbat elementlari yig'indisi va manfiy elementlari ko'paytmasini hisoblash algoritmi va dasturini tuzing.
2. $A(N,N)$ massivning barcha musbat, manfiy va nol elementlari sonini aniqlash algoritmi va dasturini tuzing.
3. $A(N,N)$ massivning bosh diagonal elementlari yig'indisi va manfiy elementlari ko'paytmasini hisoblash algoritmi va dasturini tuzing.
4. $A(N, N)$ massivning bosh diagonalining eng katta elementi va u turgan ustunni aniqlash algoritmi va dasturini tuzing.
5. $A(N,M)$ massiv har bir ustuni elementlari yig'indisini, so'ngra har bir satri elementlari ko'paytmasini topish algoritmi va dasturini tuzing.
6. $A(N,M)$ massivning har bir satrining eng katta elementini toping va ularning indeks tartib raqamini aniqlash algoritmi va dasturini tuzing.
7. $A(N,M)$ massiv har bir satr elementlari yig'indisini hisoblash va ulardan eng kattasi va eng kichigini topish algoritmi va dasturini tuzing.
8. $A(3,3)$ massiv berilgan. Uning birinchi va uchinchi ustun elementlarini 3 ga ko'paytirib yangi massiv hosil qilish algoritmi va dasturini tuzing.
9. $A(N,N)$ massivning eng kichik elementi joylashgan satr va ustun elementlarini o'chirish algoritmi va dasturini tuzing.
10. $A(3,3)$ massiv berilgan. Uning satr va ustun elementlarini kvadratga ko'tarib yangi massiv hosil qilish algoritmi va dasturini tuzing.

18. Procedure va function dan foydalanib hisoblash dasturini tuzing

1. $U = \frac{3,2P(1+y) + P(0,1y)}{P(1+2y) + P(1-2y)}$, bu yerda $P(x) = \prod_{k=1}^{\infty} \frac{x^2}{k!}$
2. $U = \frac{\min(a,b,c,d)}{\min(a,3,b+2,c+4)} + \frac{\max(a,b,c,d)}{\max(2a,b,3c,d+1)}$
3. $U = \frac{EKUB(m,n)}{EKUB(m^2,n+m)} + EKUB(n+m,n-m)$
4. n ta muntazam uchburchak tomonlari bilan berilgan, bu uchburchakka tashqi chizilgan aylana radiuslarini hisoblang va ularni kamayib borish tartibida joylashtiring.
5. $U = \frac{EKUB(m+n,n,k+m)}{EKUB(m,n+k,m+n)} + EKUB(m+n,n+k,m+k)$
6. N ta uchburchak o'z tomonlari bilan berilgan. Bu uchburchaklar yuzlarini hisoblang va ulardan kattasini aniqlang.
7. $B = \frac{P(x+2) + P(2y+1)}{P(2x+1)}$; bu yerda $P(a) = \frac{(1+a)^2}{2a}$.
8. $t = \frac{\max(2x,y,z)}{\max(x,2y,z)} + \max(x,y,2z)$.

19. Belgili va satri ma'lumotlar bilan ishlashga doir dastur tuzing.

1. Berilgan matnda 'a' va 'c' harflarining sonini aniqlash algoritmi va dasturini tuzing.

2. Berilgan matnda ‘a’ va ‘c’ harflaridan boshlanadigan so‘zlar sonini aniqlash algoritmi va dasturini tuzing.
3. Berilgan matnda eng uzun va eng qisqa so‘zni hosil qilish algoritmi va dasturini tuzing.
4. Berilgan matnda ‘n’ va ‘s’ harflari bilan tugaydigan so‘zlarni hosil qilish algoritmi va dasturini tuzing.
5. Berilgan matnda ‘a’ va ‘o’ harflari o‘rnini almashtirish algoritmi va dasturini tuzing.
6. «AABB...ZZ» satrni chop etish algoritmi va dasturini tuzing.
7. Berilgan matndagi eng qisqa so‘zni topish va uni teskari tartibda hosil qilish algoritmi va dasturini tuzing.
8. Berilgan so‘zdan juft o‘rindagi va toq o‘rindagi harflar ketma-ketligini ajratish algoritmi va dasturini tuzing.
9. «ZZYY...AA» satrni chop etish algoritmi va dasturini tuzing.
10. Berilgan matnda necha marta qo‘sh harflar ketma-ketligi uchrashini aniqlash algoritmi va dasturini tuzing.

20. Harakanuvchi ob’ektlar uchun dastur tuzing

1. To‘g‘ri chiziq bo‘yicha harakatlanuvchi shar tasvirini chizing.
2. Berilgan hamma topshirihlardagi harakat biror bir tugma bosilguncha davom etishi lozim. Aylana bo‘ylab harakatlarni tasvirlarni hosil qilishda qutb koordinatalar sistemasidan foydalanishni tavsiya qilinadi va strelkasi bo‘yicha aylana bo‘ylab o‘zgarmas burchak tezligida harakat qilayotgan nuqta tasvirini hosil qiling.
3. Ekranda soat miliga qarshi yo‘nalishda aylana bo‘ylab " harfi o‘zgarmas burchak tezligida harakat qilayotgan "O" tasvirini yasang.

Laboratoriya ishlari

Mavzu: Chiziqli algoritm va dastur tuzish

1-laboratoriya ishi topshiriqlari

a) *Chiziqli algoritm va dastur tuzish.*

b) *Tarmoqlanuvchi algoritm va dastur tuzish.*

1-topshiriqlarni bajarishda har bir talaba dastlab chiziqli, tarmoqlanish va tanlash buyruqlari haqida qisqacha ma’lumot berib, so’ngra berilgan amaliy tooshiriqlarni bajaradi.

1-variant

$$a) a = u^{(x+y)/2} - \sqrt{\frac{x-1}{y+1}} \quad b = \sin(2 \arccos t), \quad \text{bu yerda} \quad x = 12,65;$$

$$y = -2,255; \quad u = 3,205, \quad t = 0,88$$

$$b) K_2 = \begin{cases} \sqrt{15a^2 - 21b^2}, & \text{agar } a > b \\ \sqrt{15b^2 - 21a^2}, & \text{agar } a \leq b \end{cases}$$

2-variant

a) $a = (2 + y^2) \frac{x + y/2}{y^2 + 1/(1 + y^2)}$, $b = \sqrt{\sin v + \cos t}$ bu yerda $x = 0,222$,
 $y = -6,72$, $v = 10,05$, $t = 0,35$

b) Tomonlari bilan berilgan uchburchakning teng tomonli bo'lishini aniqlash algoritmi va dasturini tuzing.

3-variant

a) $a = \frac{1}{2}(x^{y-x} + y^{(x+y)/2})$, $b = \lg(\sqrt[3]{u} + \sqrt{v+2})$, bu yerda
 $x = 3,225$, $y = 2,98$, $u = 125,33$, $v = 33,075$

b) $S = \max(x, y, z) + \min(a, b, c)$

4-variant

a) $a = \sqrt[3]{x + \sqrt[4]{y}}$; $b = \sqrt{yl^{-(y+4/2)}}$, bu yerda
 $x = 37,15$; $y = 12,55$; $l = 20,12$.

b) $z = \begin{cases} (2x + y + 3)/3, & \text{agar } x + y > 0 \text{ va } x > 0 \\ \sqrt{x^2 + 2y^{2+3}}, & \text{agar } x + y > 0 \text{ va } y < 0 \\ (x^2 + y^2)/x, & \text{qolg'an hollarda} \end{cases}$

5-variant

a) $a = \sqrt{|\cos^2 x^2 + y^2|}$, $b = \frac{e^{3xy} + \ln|3x^2 - y|}{\sqrt{x^2 + y^2}}$, bu yerda $x=1,42$, $y=2,035$

b) $Z = \begin{cases} \log_3 x^2 + \sqrt[2]{x^3}, & \text{agar } x > 2 \\ \sqrt{x^2 + 16} - e^{2x}, & \text{agar } x = 2 \\ a^x + |x^3|, & \text{agar } x < 2 \end{cases}$

6-variant

a) Tomonlari bilan berilgan uchburchakning perimetri va yuzasini hisoblang.

b) a, b, c sonlari berilgan. Ularning manfiylarini 2 marta oshiring, musbatlarini 2 ga bo'ling, nolga teng bo'lganlarini o'zgarishsiz qoldiring.

7-variant

a) $a = \frac{(x + y)^3 + \sqrt[3]{|x + y|}}{3}$, $b = \frac{\ln \left| \frac{x}{y} \right| + \lg \left| \frac{y}{x} \right|}{e^{-xy}}$; bu yerda $x=1,0645$,
 $y=2,1365$.

b) $e = \begin{cases} \operatorname{tg} x + \sqrt{x+1}, & \text{agar } x > 0 \\ x^3 + 3x^2 - 4x + 7, & \text{agar } x \leq 0 \end{cases}$

8-variant

$$a) z = (1+u) \frac{x + \frac{y}{u}}{a - \frac{1}{1-x^2}}; u = \frac{\sin^2 x}{x^2 + y^2}; x = \frac{1}{2}; a=10,5, y=2,5.$$

$$b) \begin{cases} \sqrt[3]{|x|} + x^2 + 7, & \text{agar } x < 0 \\ x^3 - 3x^2 + 9\sqrt{x}, & \text{agar } x \geq 0 \end{cases}$$

9-variant

$$a) t = e^{|x|} \cdot \operatorname{tg} \frac{a+b}{a-b}, \quad b = \frac{\cos x + \sin x^2}{e^x + a}, \text{ bu yerda } a=10, v=5, x=2.$$

$$b) Z = \begin{cases} \sin \ln|x| + \sqrt{|x|^3 + 5,3}, & \text{agar } x < -3 \\ x^2 + e^{|x|} + \operatorname{tg} x, & \text{agar } x = -3 \\ 0,8 \cdot \sin x + 10^{-5,6}, & \text{agar } x > -3 \end{cases}$$

10-variant

$$a) l = \operatorname{tg} 4,5x + \frac{|x^{-10}| + b}{\sin 0,5x + a}, \quad t = \frac{e^{a+b} + \sqrt{x^2 + 3}}{a}, \text{ bu yerda } a=10, b=16,$$

$x=-2,5.$

$$b) Z = \begin{cases} \sqrt{a + \sin 2x} + ax^3 + b \log|2x|, & \text{agar } \sqrt{a-b} \leq x \\ \frac{\operatorname{arctg} 5x}{b \cdot \cos x + \ln a \cdot x}, & \text{agar } \sqrt{a-b} > x \end{cases}$$

11-variant

$$a) a = \frac{y^{x+1}}{\sqrt[3]{y-2+3}} + \frac{x+y/2}{2x+y}; \quad b = (x-1)^{-1/\sin^2 z}; \quad x = 1,625; \quad y = 15,4; \quad z = 0,255$$

b) Ixtiyoriy a, b, c sonlar uchburchakning tomonlari bo'lishi yoki bo'lmasligini aniqlang.

12-variant

$$a) a = \frac{x^{y-z} + e^{y-1}}{1+xy - \operatorname{tg} z}; \quad b = 1 + y - \frac{y-x^2}{2} + \frac{(x-y)^3}{3}; \quad x = 2,444; \quad y = 0,869; \quad z = -0,166.$$

$$b) z = \begin{cases} \lg x + 1/x + e^{2\pi}, & \text{agar } x < 3 \\ \sqrt{x^2 + 2x + 3}, & \text{agar } x < 3 \text{ bo'lsa,} \\ \lg x + x^2 + \sqrt{x}, & \text{agar } x = 3 \end{cases}$$

13-variant

$$a) a = \lg(\sqrt{e^{x-y}} + x^y + z); \quad b = x - \frac{x^3}{3!} + \frac{x^5}{5!}; \quad x = 1,542; \quad y = -3,264; \quad z = 8,05$$

b) Uchta har xil butun son berilgan. Shu sonlarning arifmetik progressiya tashkil qilish yoki qilmasligini aniqlang.

14-variant

a) $a = \frac{\sqrt[3]{8-x-y^2+1}}{x^2+y^2+1}$; $b = e^{x-y}(tg^2z+1)$; $x = 4,5$; $y = 0,007$; $z = 0,845$

b) Uch o'ldovli fazoda uchta x, y, z vektorlar o'z koordinatalari bilan berilgan. Shu vektorlarning qaysilari juftt-jufti bilan kollingar bo'lishini aniqlang.

15-variant

a) $a = \cos x + \cos y^{1-2\sin^2 y}$; $b = 1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}$; $x = 0,4$; $y = -0,875$; $z = -0,475$

b) $y = \begin{cases} \sqrt{x^2 + |\sin x|}, & \text{agar } x > 0 \\ |x^2 - \sin x|, & \text{agar } x = 0 \\ |tg^4 x + |ctg^3 x||, & \text{agar } x < 0 \end{cases}$

16-variant

a) $a = \sqrt{10(\sqrt[3]{x} + x^{y+2})}$; $b = (\arcsin z)^2 + x + y$; $x = 16,55$; $y = -2,75$; $z = 0,15$

b) Tomonlari a, b, c larga teng bo'lgan uchburchakning turini aniqlang.

17-variant

a) Uchlarining koordinatalari bilan berilgan uchburchakning perimetri va yuzasini toping.

b) $z = \begin{cases} \frac{1}{x} + \frac{1}{y} + \frac{1}{(x+y)}, & \text{agar } x \neq 0 \text{ va } y \neq 0 \\ \frac{1}{x} + \frac{1}{(x+1)}, & \text{agar } x \neq 0 \text{ va } y = 0 \\ \frac{1}{y} + \frac{1}{(y+1)}, & \text{agar } x = 0 \text{ va } y \neq 0 \end{cases}$

18-variant

a) $S = 2\pi R^3 + \sqrt{R+vt}$, $t = \frac{h+a+c}{2}$; $R = \sqrt{t+2h}$

b) $y = \begin{cases} x^2 + 1, & \text{agar } x > \frac{\pi}{2} \\ x - 2,1, & \text{agar } x = \frac{\pi}{2} \\ \sin x & \text{agar } x < \frac{\pi}{2} \end{cases}$

19 - variant

a) $a = u^{(x+y)/2} - \sqrt{\frac{x-1}{y+1}}$, $b = \sin(2\arccost)$ $x = 12,65$; $y = -2,255$; $u = 3,205$; $t = 0,88$

b) a, b, c sonlarining eng kattasini toping.

20 - variant

$$a) a = \frac{1}{2}(x^{y-x} + y^{(x+y)/2}); b = \lg(\sqrt[3]{u} + \sqrt{v+2}), x = 3,225; y = 2,98; u = 125,33; v = 33,075$$

$$b) S = \max(x, y, z) + \min(a, v, c)$$

21 - variant

$$a) y = a * e^{\lg x} - \sqrt{ax + b}; Z = t^2 * \lg(|a + b|) + |x^3 - 2x|,$$

$$\text{buyerd} a=3,34; b=-2,18; x=1,128; t=3,028.$$

$$b) Y = \begin{cases} 2x^2 - 3x + e^x, & \text{agar } x < 3 \\ \ln(e^x + 2) - x, & \text{agar } x = 3 \\ 4x^2 - \ln \frac{1}{x}, & \text{agar } x > 3 \end{cases}$$

22 - variant

$$a) C = \sqrt{\frac{x^2 + y^2}{3}} + \ln|x + y|; Z = \log_3|t| + \frac{\text{tg}|2 \cdot t|}{2};$$

$$\text{buyerd} a = 3,0011; y = -2,018; t = 0,0386$$

$$b) Y = \begin{cases} \ln(|x^2 + 1|) + \frac{1}{3}, & \text{agar } x > 1 \\ e^{x+2} + 2 \sin x, & \text{agar } x < 1 \\ \frac{100}{x^2} - \ln x, & \text{agar } x = 3 \end{cases}$$

23 - variant

$$a) b = e^{x \ln a} + \sin x^2; c = \sin^2 x^2 + \frac{x-a}{x+b}; \text{buyerd} a=10, b=28,7, x=-0,25.$$

$$b) y = \begin{cases} \ln x^2 - e^{ax+b} + \lg|a-b|, & \text{agar } 2a+b < 0 \\ \text{tg}(ax-b)^2 - b|x^{-10}|, & \text{agar } 2a+b = 0 \\ \text{arctg}(2x-0,5 + \sqrt{a+bx}), & \text{agar } 2a+b > 0 \end{cases}$$

24 - variant

$$a) y = \frac{\sqrt[3]{x+2} + \text{tg}^3 \sin|x|}{4 \lg \sin x^2}, \quad z = \frac{e^{|x+1|} + 2}{\text{ctg}^2 3x-1}, \text{ bu yerda } x=0,792.$$

$$b) y = \begin{cases} ax^3 + b, & \text{agar } x < 0,2 \\ 16,5b + \cos + e^{|x|}, & \text{agar } x = 0,2 \\ b \cdot \sqrt{x+25} + e^{x^2}, & \text{agar } x > 0,2 \end{cases}$$

25 - variant

$$a) t = a^x \cdot b + e^x \cdot b + e^x + \cos^2(4x-0,2); c = b^2 \cdot \text{tg}^4 x + x^{10};$$

bu yerda $a=3,127$; $b=-2,087$; $x=1,298$;

$$b) y = \begin{cases} e^x \cdot \sin x, & \text{agar } x \geq 0 \\ e^x + \cos x, & \text{agar } x < 0 \end{cases}$$

Mavzu: Takrorlanuvchi jarayonlarga dastur tuzish.
2-laboratoriya ishi topshiriqlari

a) Parametrlil sikl operatoridan foydalanib dastur tuzing;

b) Sharti oldin va qo'yilgan sikl operatoridan foydalanib dastur tuzing;

c) ichma-ich joylashgan sikllardan foydalanib dastur tuzing;

2-mustaqil ish topshiriqlarni bajarishda har bir talaba dastlab takrorlanish buyruqlari haqida qisqacha ma'lumot berib, so'ngra berilgan amaliy topshiriqlarni bajaradi.

1-variant

a) $S = \sum_{n=1}^5 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblang.

b) 1 dan n gacha toq sonlar kvadratlari yig'indisini hisoblang.

c) $S = \sum_{n=1}^5 \sum_{i=1}^{12} (n^2 + i^3)$ ni hisoblang

2-variant

a) $P = \prod_{i=1}^{10} \frac{i^2 + ai}{i^2 + a^2} + \prod_{i=2}^4 \frac{k^3}{k+a}$ ni hisoblang.

b) $[a,b]$ oraliqda m soniga karrali sonlar ko'paytmasini hisoblang

c) $S = \prod_{a=1}^{10} \prod_{i=2}^4 \frac{k^3}{k+a}$ ni hisoblang.

3-variant

a) $P = \sum_{i=1}^8 \frac{i^2 + ai}{i^2 + a^2}$ ni hisoblang.

b) 1 dan 35 gacha bo'lgan toq sonlar kvadratlarning yig'indisi va juft sonlar kvadratlarning ko'paytmasini toping.

c) $S = \sum_{n=1}^5 \prod_{i=1}^8 (n^2 + i^3)$ ni hisoblang

4-variant

a) Berilgan son raqamlari yig'indisini hisoblash dasturini tuzing.

b) $S = \prod_{k=1}^{\infty} \frac{k^2}{x^k}$ ni $\text{eps} = 0,0001$ aniqlik bilan hisoblang.

c) $S = \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^4)$ ni hisoblang

5-variant

a) $S = \prod_{n=1}^6 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblang.

b) 2 dan 50 gacha 4 ga va 3 ga bo'linadigan sonlarni chop eting.

c) $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblang.

6-variant

a) $[a,b]$ oraliqdagi m soniga karrali sonlar yig'indisini hisoblang.

b) $S = \sum_{k=1}^{\infty} \frac{k^2}{x^k}$ ni $\epsilon = 0,001$ aniqlik bilan hisoblang.

c) $S = \prod_{k=1}^4 \sum_{j=1}^{10} (k! + j!)$ ni hisoblang

7-variant

a) $y = \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$ ni hisoblang.

b) $y = \text{tg}(x+c)$ funksiya qiymatini $[a,b]$ oraliqda h qadam bilan hisoblang.

c) $S = \sum_{n=1}^3 \prod_{k=1}^4 \sum_{j=1}^5 (k! + j! + n!)$ ni hisoblang.

8-variant

a) $S = m! + n! + k!$ ni hisoblang.

b) $S = 1 + \frac{4}{3} + \frac{9}{5} + \dots + \frac{n^2}{2n-1}$ ni hisoblang.

c) $S = \sum_{k=1}^3 (\prod_{n=1}^4 kn^2 + \sum_{i=1}^5 ki^3)$ ni hisoblang.

9-variant

a) $[a,b]$ oraliqdagi n va m larga karrali bo'lgan sonlar yig'indisini hisoblang.

b) $S = \sum_{n=1}^{\infty} \frac{n^2 + 1}{n^n}$ ni $E = 0,001$ aniqlik bilan hisoblang.

c) $S = \prod_{k=1}^3 (a \prod_{n=1}^4 kn^2 + b \sum_{i=1}^5 ki^3)$ ni hisoblang

10-variant

a) $S = \sum_{i=1}^5 (i^2 + a) + \prod_{k=1}^n \frac{k^3 - 1}{k!}$ ni hisoblang.

b) EKUB(a,b,c) ni topish dasturini tuzing.

c) $S = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{(k! + j!)}{a + b}$ ni hisoblang.

11-variant

a) $y = n!!$ ($n > 0$) ni hisoblang.

b) EKUK(a,b,c) ni topish dasturini tuzing.

c) $S = \prod_{k=1}^4 \sum_{j=1}^{10} \frac{a(k+j)}{(k! + j!)}$ ni hisoblang

12-variant

a) $P = \sin 1 + \sin 2 + \dots + \sin m$ ($m > 0$) ni hisoblang.

b) $[a,b]$ oraliqdagi toq sonlar kvadtlarining ko'paytmasini hisoblang.

c) $S = \prod_{k=1}^4 \frac{\sum_{j=1}^{10} j!}{k!}$ ni hisoblang.

13-variant

a) $S = \sum_{n=1}^{10} 2^n + \prod_{k=1}^5 3^k$ ni hisoblang.

b) 10 dan 120 gacha bo'lgan juft sonlar yig'indisi va toq sonlar ko'paytmasini hisoblang.

$$c) S = \sum_{i=1}^3 \frac{(\sum_{j=1}^{10} j + \prod_{k=1}^4 \frac{1}{k})}{i!} \text{ ni hisoblang.}$$

14-variant

$$a) S = \sum_{n=1}^{10} \frac{x^{4n+1}}{4n-1} \text{ ni hisoblang.}$$

$$b) S = \sum_{n=1}^{\infty} \frac{a^{n-1}}{n^n} \text{ ni hisoblang.}$$

c) Ko'paytirish jadvali elementlarini hosil qilish dasturini tuzing.

15-variant

$$a) S = \sum_{n=1}^{10} b^n + \prod_{k=1}^5 a^k \text{ ni hisoblang.}$$

$$b) \varepsilon=0,0003 \text{ aniqlik bilan } S = \sum_{n=1}^{\infty} \frac{x^{n-1}}{a^n} \text{ ni hisoblang.}$$

$$c) S = \sum_{n=1}^{10} \prod_{k=1}^5 (a^k + b^n) \text{ ni hisoblang.}$$

16-variant

a) [a,b] oraliqdagi tub sonlarni toping.

$$b) \varepsilon=0,0003 \text{ aniqlik bilan } S = \prod_{n=1}^{\infty} \frac{(a+c)^{n-1}}{n^n} \text{ ni hisoblang.}$$

$$c) S = \prod_{n=1}^3 \prod_{k=1}^5 (x^k + y^n) \text{ ni hisoblang.}$$

17-variant

$$a) S = \prod_{n=1}^m \frac{(a+c)^{n-1}}{n^n}$$

$$b) S = \sum_{n=1}^{\infty} \frac{n!}{(2n)!}, \quad \varepsilon = 0.003 \text{ aniqlik bilan hisoblang.}$$

$$c) S = \sum_{n=1}^5 \prod_{k=1}^5 (x^k + y^n) \text{ ni hisoblang.}$$

18-variant

$$a) P = \prod_{i=1}^{10} \frac{i^2 + a}{i^2 + b^2} \text{ ni hisoblang.}$$

$$b) S = \sum_{n=1}^{\infty} \frac{1}{(3n-2)(3n+1)}, \quad \varepsilon = 0.0004 \text{ aniqlik bilan hisoblang.}$$

$$c) P = \prod_{a=1}^5 \prod_{k=2}^4 \frac{b^2}{k+a} \text{ ni hisoblang.}$$

19-variant

$$a) P = \prod_{i=1}^{10} \frac{i^2 + ai}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^2}{k+a} \text{ ni hisoblang.}$$

$$b) S = \sum_{k=1}^{\infty} \frac{\sin k}{k!} \text{ ni } E = 0,001 \text{ aniqlik bilan hisoblang.}$$

$$c) P = \sum_{i=1}^3 \frac{\prod_{a=1}^5 \prod_{k=2}^4 \frac{b^2}{k+a}}{i} \text{ ni hisoblang}$$

20 - variant

$$a) S = \prod_{i=1}^{20} \frac{1}{i+2} + \sum_{k=1}^{10} \frac{\sin k}{k!} \text{ ni hisoblang.}$$

$$b) S = \sum_{k=1}^{\infty} \frac{a+b}{k!}, \quad \varepsilon = 0.003 \text{ aniqlik bilan hisoblang.}$$

$$c) P = \sum_{i=1}^3 \frac{\prod_{a=1}^5 a+b}{i+c} \text{ ni hisoblang}$$

21 - variant

a) Berilgan N sonining barcha bo'luvchilarini toping.

$$b) P = \prod_{k=1}^{\infty} \frac{k}{2+k^k} \text{ ni } E = 0,001 \text{ aniqlik bilan hisoblang.}$$

$$c) S = \prod_{a=1}^5 \sum_{k=1}^8 \frac{a^3}{k+b} \text{ ni hisoblang.}$$

22 - variant

a) Berilgan sonning tub yoki murakkabligini aniqlang.

$$b) P = \prod_{k=1}^{\infty} \frac{1}{k!} \text{ ni eps} = 0,001 \text{ aniqlik bilan hisoblang.}$$

$$c) S = \prod_{b=1}^4 \sqrt{\prod_{a=1}^5 \sum_{k=1}^8 \frac{a^3}{k+b}} \text{ ni hisoblang.}$$

23 - variant

a) $S = 3 + 6 + 9 + \dots + 63$ yig'indini hisoblang.

$$b) P = \sum_{k=1}^{\infty} \frac{a+b}{2k!} \text{ ni eps} = 0,001 \text{ aniqlik bilan hisoblang.}$$

$$c) S = \prod_{b=1}^m \sqrt{\sum_{k=1}^l \frac{a^3}{k+b}} \text{ ni hisoblang.}$$

24 - variant

$$a) P = \prod_{k=1}^5 \frac{k!}{2+k} + \sum_{j=1}^6 (2j-1)^2 \text{ ni hisoblang.}$$

$$b) P = \sum_{k=1}^{\infty} \frac{x+y}{x^k} \text{ ni eps} = 0,001 \text{ aniqlik bilan hisoblang.}$$

$$c) S = \prod_{b=1}^m \sqrt{\sum_{k=1}^l \frac{a^3}{k+b}} \text{ ni hisoblang}$$

25 - variant

$$a) S = \sum_{k=1}^5 \frac{a+b}{k!} \text{ ni hisoblang.}$$

$$b) S = 2 + \frac{4}{3} + \dots + \frac{2n}{2n-1} \text{ ni hisoblang.}$$

$$c) P = \sum_{i=1}^3 \frac{\prod_{a=1}^5 a!}{i!} \text{ ni hisoblang}$$

26-variant

a) $S = \sum_{n=1}^5 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblang.

b) $S = \prod_{k=1}^{\infty} \frac{k^2}{x^k}$ ni eps = 0,0001 aniqlik bilan hisoblang.

c) $S = \sum_{n=1}^5 \prod_{i=1}^8 (n^2 + i^3)$ ni hisoblang

27-variant

a) $P = \prod_{i=1}^{10} \frac{i^2 + ai}{i^2 + a^2} + \prod_{i=2}^4 \frac{k^3}{k+a}$ ni hisoblang.

b) 1 dan n gacha toq sonlar kvadratlari yig'indisini hisoblang.

c) $S = \prod_{a=1}^{10} \prod_{i=2}^4 \frac{k^3}{k+a}$ ni hisoblang.

28-variant

a) $P = \sum_{i=1}^8 \frac{i^2 + ai}{i^2 + a^2}$ ni hisoblang.

b) [a,b] oraliqda m soniga karrali sonlar ko'paytmasini hisoblang

c) $S = \sum_{n=1}^5 \sum_{i=1}^{12} (n^2 + i^3)$ ni hisoblang

29-variant

a) Berilgan son raqamlari yig'indisini hisoblash dasturini tuzing.

b) 2 dan 50 gacha 4 ga va 3 ga bo'linadigan sonlarni chop eting.

c) $P = \sum_{i=1}^3 \frac{\prod_{a=1}^5 a!}{i!}$ ni hisoblang

30-variant

a) $S = \prod_{n=1}^6 n^2 + \sum_{i=1}^{12} i^3$ ni hisoblang.

b) 1 dan 35 gacha bo'lgan toq sonlar kvadratlarning yig'indisi va juft sonlar kvadratlarning ko'paytmasini toping

c) $S = \sum_{n=1}^3 \prod_{k=1}^5 \sum_{j=1}^{10} (k^2 + j^2 + n)$ ni hisoblang.

Mavzu: Massiv elementlari bilan ishlash.

3 – laboratoriya ishi topshiriqlari

a) Bir o'lchovli massiv elementlari bilan ishlash.

b) Ikki o'lchovli massiv elementlari bilan ishlash.

c,d) Finction dan foydalanib dastur tuzish.

Ha bir talaba ushbu nazariy savollarning mazmunini yoritib, so'ngra o'z variantidagi amaliy topshiriqlarni bajaradi.

1-variant

a) x_1, x_2, \dots, x_n sonlar berilgan.

$M = \frac{1}{n} \sum_{k=1}^n x_k$ va $D = \left(\frac{1}{n-1} \sum_{k=1}^n (x_k - M)^2 \right)^2$ ni hisoblang.

b) $A(10, 10)$ matritsaning bosh diagonalining eng katta elementini va u turgan ustunni aniqlang.

$$c) U = \frac{3,2P(1+y) + P(0,1y)}{P(1+2y) + P(1-2y)}, \text{ bu yerda } P(x) = \prod_{k=1}^{\infty} \frac{x^2}{k!} .$$

$$d) U = \frac{\min(a,b,c,d)}{\min(a,3,b+2,c+4)} + \frac{\max(a,b,c,d)}{\max(2a,b,3c,d+1)} .$$

2-variant

a) x_1, x_2, \dots, x_{55} sonlar berilgan: $x_1(x_2+x_3) \cdot (x_4+x_5+x_6) \dots (x_{46}+x_{47}+\dots, x_{55})$ ni hisoblang.

b) $S(M,N)$ kvadrat matritsa bosh diagonalining barcha elementlarini, so'ngra musbat va manfiy elementlarini yig'indisini toping.

$$c) U = \frac{\min(a,b) + \max(2a,b+c)}{\min(a+b,b+c) + \max(2a^2+c,b^2+a)}$$

$$d) U = \frac{Y\hat{E}O\hat{A}(m,n)}{Y\hat{E}O\hat{A}(m^2,n+m)} + Y\hat{E}O\hat{A}(n+m,n-m)$$

3-variant

a) n elementdan iborat A massiv berilgan. Massiv elementlarini o'sib borish tartibida joylashtiring.

b) $B(N,M)$ matritsa har bir ustuni elementlari yig'indisini, so'ngra har bir satri elementlari ko'paytmasini toping.

$$c) U = \frac{C(0,2) + 2C(x+1,5)}{3B(2x+1) + B(2,5)}, \text{ bu yerda}$$

$$C(y) = \sum_{n=1}^5 \frac{2y^2 + 4}{n!}; \quad B(y) = \frac{2y-1}{y^2} .$$

$$d) U = \frac{(2m^2+n)! + (3n^2-1)! + (nm)!}{(3m^2-n)! + (n+m+1)!}$$

4-variant

a) a_1, a_2, \dots, a_{55} massiv berilgan. Massiv elementlarining eng kattasini 1-soni bilan, eng kichigini esa -1 soni bilan almashtiring.

b) $A(N,M)$ har bir satrining eng katta elementini toping va ularning indeks tartib raqamini aniqlang.

$$c) U = f(a,b) + f(a^2, b^2) + f(a^2-1, b), \text{ bu yerda}$$

$$f(u,t) = \begin{cases} u^2 + t^2, & \text{agar } z > 0, t > 0 \\ u + t^2, & \text{agar } u \leq 0, t \leq 0 \\ u^2 + t, & \text{qolgan hollarda} \end{cases}$$

$$d) U = \frac{EKUB(m+n, n, k+m)}{EKUB(m, n+k, m+n)} + EKUB(m+n, n+k, m+k)$$

5-variant

a) x_1, x_2, \dots, x_{20} massiv elementlarining eng kichik musbat elementining tartib nomerini aniqlang va undan keyin turgan elementlar sonini toping.

b) $A(N, M)$ matritsa har bir ustunining eng kichik elementlarini toping.

c) $S = f(a, b) + f(a^2, b^2) + f(a^2 - 1, b)$, bu yerda

$$f(u, t) = \begin{cases} u^2 + t^2, & \text{agar } u > 0, \quad t > 0 \\ u + t^2, & \text{agar } u \leq 0, \quad t \leq 0 \\ u^2 + t, & \text{qo'lgan xollarda} \end{cases}$$

d) Ntauchburchako'ztomonlaribilanberilgan. Buuchburchaklaryuzlarini-hisoblangvaulardankattasinianiqlang.

6-variant

a) $Z = \sum_{i=1}^m \sum_{k=1}^n \frac{\sin^2 x_i}{k}$, bu yerda x_1, x_2, \dots, x_m massiv berilgan.

b) $A(N, M)$ matritsa har bir satr elementlari yig'indisini hisoblang va ulardan eng kattasi va eng kichigini toping.

c) $Z = f(\sqrt{x}, y) + f(a, b) + f(\sqrt{x} + 1, y) + f(x + y, x)$, bu yerda

$$f(u, t) = \begin{cases} u + 2t, & \text{agar } u > 0 \\ u + t, & \text{agar } u \leq -1 \\ u^2 - 2t + 1, & \text{qo'lgan xollarda} \end{cases}$$

d) n ta uchburchakning uchlari koordinatalari bilan berilgan. Bu uchburchaklarning yuzlarini hisoblang va ulardan kichigini toping.

7-variant

a) x_1, x_2, \dots, x_n massivning manfiy elementlarining eng kattasini toping.

b) $A(N, M)$ matritsa har bir ustuni elementlari ko'paytmasini hisoblang. Ularning eng kichigi va kattasini aniqlang.

$$c) P = \frac{EKUB(n, m, n + n) + EKUB(n + m, n + k, m + k)}{EKUB(m^2, n^2, m + n)}, \quad m, n, k \in N$$

d) n ta uchburchak o'z tomonlari bilan berilgan. Bu uchburchaklarga ichki chizilgan aylana radiuslarini hisoblang va ularning kattasini aniqlang.

8-variant

a) x_1, x_2, \dots, x_n massivning manfiy, musbat va nol elementlarining sonini aniqlang.

b) $A(3, 3)$ massiv berilgan. Uning satr elementlari yig'indisidan yangi bir o'lchovli massiv hosil qiling.

$$c) B = \frac{P(x + 2) + P(2y + 1)}{P(2x + 1)}; \text{ bu yerda } P(a) = \frac{(1 + a)^2}{2a}.$$

$$d) t = \frac{\max(2x, y, z)}{\max(x, 2y, z)} + \max(x, y, 2z).$$

9-variant

a) $S = \frac{1}{n} \sum_{i=1}^n x_i + \frac{1}{m} \prod_{k=1}^m Y_k$ ni hisoblang.

b) $A(3,3)$ massiv berilgan. Uning ustun elementlari yig'indisidan bir o'lchovli massiv hosil qiling.

c) $Z = e^{-t(0,5)} + \frac{15}{t(2x+1)} + \ln|t(x)|$; bu yerda $t(a) = \sqrt{2a^2 + 3a + 1}$

d) $t = \frac{n! + (2n + m)!}{(n^2 - 2m)!} + (2n^2 - m)!$

10-variant

a) x_1, x_2, \dots, x_n massiv berilgan. Uning toq elementlaridan U, juft elementlaridan Z massiv hosil qiling.

b) $B(3,3)$ massiv berilgan. Uning birinchi va uchinchi ustun elementlarini 3 ga ko'paytirib yangi massiv hosil qiling.

c) $y = \frac{\ln(y^2 + 6y + 1)}{1 + \cos(z^2 + 6z + 1)} + \frac{\sin(t^2 + 6t + 1)}{10} + x^2 + 6x + 1$

d) $T = \frac{EKUK(m, n, k)}{EKUK(x, y, z)} + EKUK(a, b, c).$

11-variant

a) x_1, x_2, \dots, x_{20} massiv berilgan. Toq elementlardan U, juft elementlardan Z massiv hosil qiling.

b) $B(3,3)$ massiv berilgan. Uning satr elementlarini kvadratga ko'tarib yangi massiv hosil qiling.

c) $B = \frac{1,8 + |1 + \cos^2 xy| + e^{-|1 + \cos^2(x+2y)|}}{9,94 - |1 + \cos^2(2x + y)|}$

d) $T = \frac{EKUB(m, n, k)}{EKUB(x, y, z)} + EKUB(a, b, c).$

12-variant

a) x_1, x_2, \dots, x_n massivning musbat elementlaridan y_i , manfiy elementlaridan z_i , massiv hosil qiling.

b) $A(4,4)$ va $B(4,4)$ massivlar berilgan $C = A \cdot B + A - B \cdot C$ massivini hosil qiling.

c) $Z = f(\sin x + \cos x, x + y) + f(\sin x, \cos x) + f(x - y, x) + f(\sin^2 x, \cos^2 x)$

bu yerda $f(d, t) = \begin{cases} d + t, & \text{agar } 4 > 1 \\ d - t, & \text{agar } 0 \leq 4 \leq 1 \\ t - d, & \text{agar } 4 < 0 \end{cases}$

d) n ta muntazam besh burchak tomonlari bilan berilgan. Bu besh burchakka tashqi chizilgan doira yuzlarini hisoblang va ularni kamayish tartibida joylashtiring.

13-variant

- a) z_1, z_2, \dots, z_n massiv elementlarini kamayib borish tartibida joylashtiring.
- b) $A(N, M)$ matritsa har bir satrining eng katta va eng kichik elementlarini, so'ngra ularning yig'indisi va ko'paytmasini hisoblang.
- c) $R = \sqrt{x^2 + y^2 + \sin xy} + \sqrt{y^2 + z^2 + \sin yz} + \sqrt{x^2 + z^2 + \sin xz} + \sqrt{z^2 + a^2 + \sin za}$
- d) n ta muntazam uchburchak berilgan. Bu uchburchakka ichki chizilgan doira yuzlarini hisoblang va ularni kamayib borish tartibida joylashtiring.

14-variant

- a) x_1, x_2, \dots, x_n massivning manfiy elementlarining eng kattasi va musbat elementlarining eng kichigini aniqlang.
- b) $A(N, M)$ matritsa har bir ustunning eng katta va eng kichik elementlarini, so'ngra ularning yig'indisi va ko'paytmasini toping.

c) $B = \begin{cases} T(3x+z, y+x) + T(x^2-y, z^2) \\ T(x-y^2, z) + T(2x+y, z+y) \end{cases}$ bu yerda, $T(a, b) = \begin{cases} \sqrt{a^2 + b^2}, & \text{agar } a > b \\ \sqrt[3]{2a^2 - b}, & \text{agar } a \leq b \end{cases}$

- d) n ta muntazam uchburchak tomonlari bilan berilgan, bu uchburchakka tashqi chizilgan aylana radiuslarini hisoblang va ularni kamayib borish tartibida joylashtiring.

15-variant

- a) x_1, x_2, \dots, x_n massiv berilgan.
- $Z = \sum_{n=1}^m \frac{(m-n)\sin x_n}{n} + \prod_{k=1}^r \frac{\operatorname{ctg}(r-x_k)}{k}$ ni hisoblang.
- b) $A(N, M)$ matritsa har bir ustun elementlarini eng kattasini toping. Topilgan elementlar ichida eng kichigini va uning tartib nomerini aniqlang.

c) $A = \begin{cases} \min(y_1, y_2, y_3) + \min(z_1, z_2, z_3), & \text{agar } \min(a, b, c) > 5 \\ \sqrt{\min(t_1, t_2, t_3)} + 5, & \text{agar } \min(a, b, c) \leq 5 \end{cases}$

- d) n ta uchburchakning uchlari koordinatalari bilan berilgan. Bu uchburchaklarga ichki chizilgan doira yuzlarini hisoblang va ularni kamayib borish tartibida joylashtiring.

16-variant

a) x_1, x_2, \dots, x_n sonlar berilgan. $M = \frac{1}{n} \sum_{k=1}^n x_k$ va $D = \left(\frac{1}{n-1} \sum_{k=1}^n (x_k - M)^2 \right)^2$

larni hisoblang

- b) $A(20, 20)$ matrisaning bosh diagonalining katta elementini va u turgan ustunni aniqlang.

c) $U = \frac{3,2P(1+y) + P(0,1y)}{P(1+2y) + P(1-2y)}$; bu yerda $P(x) = \prod_{k=1}^{\infty} \frac{x^2}{k!}$

d) $U = \frac{\min(a, b, c, d)}{\min(a+3, b+2, c, 4)} + \frac{\max(a, b, c, d)}{\max(2a, b, 3c, d+1)}$

17-variant

- a) x_1, x_2, \dots, x_{55} sonlar berilgan: $x_1(x_2+x_3) (x_4+x_5+x_6) \dots (x_{46}+x_{47}+\dots, x_{55})$ ni hisoblang.

b) S(M, N) kvadrat matritsa bosh diagonalining barcha elementlarini, soʻngra musbat va manfiy elementlarining yigʻindisini toping.

$$c) U = \frac{\min(a, b) + \max(2a, b + c)}{\min(a + b, b + c) + \max(2a^2 + c, b^2 + a)}.$$

$$d) U = \frac{(2m^2 + n)! + (3n^2 - 1)! + (nm)!}{(3m^2 - n)! + (n + m + 1)!}.$$

18-variant

a) a_1, a_2, \dots, a_{55} massiv berilgan. Massiv elementlarining eng kattasini 1 soni bilan, eng kichigini esa -1 soni bilan almashtiring.

b) A(N, M) har bir satrining eng katta elementini toping va ularning indeks tartib raqamini aniqlang.

c) $U = f(a, b) + f(a^2, b^2) + f(a^2 - 1, b)$, bu yerda

$$f(u, t) = \begin{cases} u^2 + t^2, & \text{agar } u > 0, t > 0 \\ u + t^2, & \text{agar } u \leq 0, t \leq 0 \\ u^2 + t, & \text{qoʻlgan xollarda} \end{cases}$$

$$d) U = \frac{EKUB(m + n, n, k + m)}{EKUB(m, n + k, m + n)} EKUB(m + n, n + k, m + k)$$

19-variant

a) $Z = \sum_{n=1}^m \sum_{k=1}^n \frac{\sin^2 x_n}{k}$ bu yerda x_1, x_2, \dots, x_m massiv berilgan.

b) A(N, M) matritsa har bir satr elementlari yigʻindisini hisoblang va ulardan eng kattasi va eng kichigini toping.

c) $Z = f(\sqrt{x}, y) + f(a, b) + f(\sqrt{x} + 1, y) + f(x + y, x)$, bu yerda

$$f(u, t) = \begin{cases} u + 2t, & \text{agar } u > 0 \\ u + t, & \text{agar } u \leq -1 \\ u^2 - 2t + 1, & \text{qoʻlgan xollarda} \end{cases}$$

d) n ta uchburchakning uchlari koordinatalari bilan berilgan. Bu uchburchakning yuzini hisoblang va ulardan kichigini toping.

20-variant

a) x_1, x_2, \dots, x_n massivning manfiy elementlarining eng kattasini toping.

b) A(N, M) matritsa har bir ustuni elementlari koʻpaymasini hisoblang va ularning eng kichigi va kattasini aniqlang.

$$c) P = \frac{EKUB(n, m, n + n) + EKUB(n + m, n + k, m + k)}{EKUB(m^2, n^2, m + n)}, m, n, k \in N$$

d) n ta muntazam oltiburchak oʻz tomonlari bilan berilgan. Bu oltiburchaklarga chizilgan aylana radiuslarini hisoblang va ularning kattasini aniqlang.

21-variant

a) A(n) massiv berilgan. Uning manfiy elementlarini kvadratga koʻtarib, musbatlarini 2 ga boʻlib, yangi massiv hosil qiling.

b) $V(3,3)$ massiv berilgan. Uning har bir satr elementlari yig'indisining o'rta arifmetigi va geometrigini toping.

$$c) Z = \frac{tha + th^2(a-b)}{th(a^2 - b^2)} + \sqrt{th(2ab)}, \text{ bu yerda } th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad d)$$

$$S = \frac{\min(a_1, a_2, a_3) + \min(b_1, b_2, b_3)}{\min(x_1, x_2, x_3) + \min(t_1, t_2, t_3)}.$$

22-variant

a) u_1, u_2, \dots, u_{10} massiv berilgan. Uning elementlarini kamayib borish tartibida joylashtiring.

b) $A(3,3)$ massiv berilgan. Uning har bir ustun elementlari yig'indisining o'rta arifmetigi va geometrigini toping.

$$c) T = \frac{1,2 \cdot P(1+x^2)}{2 \cdot P(2x+3)} + \sqrt{3 \cdot P(x^2)}; \text{ bu yerda } P(y) = \sum_{i=1}^3 \frac{y+i}{2}$$

$$d) A = \frac{\max(a_1, a_2, a_3) + \max(b_1, b_2, b_3)}{\max(x_1, x_2, x_3) + \max(t_1, t_2, t_3)}.$$

23 variant

a) z_1, z_2, \dots, z_{10} massiv berilgan. Manfiy elementlaridan alohida, musbat elementlaridan alohida massiv hosil qiling.

b) $X(3,3)$ massiv berilgan. Uning toq tartib raqamli elementini 2 ga, juft tartib raqamli elementini 3 ga ko'paytirib, yangi massiv hosil qiling.

$$c) \delta = \frac{A(0,5y) + A(x+y)}{A(2\delta - y)}; \text{ bu yerda } A(z) = \sum_{i=1}^6 \frac{z}{i!}.$$

$$d) R = \frac{\max(a, b, c, d) + \min(x, y, z, t)}{\max(m, n, k, l) + \min(e, f, g, t)}.$$

24 variant

a) t_1, t_2, \dots, t_{15} massiv berilgan. Juft elementlarini 2 ga, toq elementlarini 3 ga, ko'paytirib, yangi massiv hosil qiling.

b) $A(3,3)$ massiv berilgan. Uning bosh diagonallari elementlarini 2 ga ko'paytirib, yangi massiv hosil qiling.

$$c) B = \frac{e^{x+y} + \ln(x+y)}{2\sqrt{e^{2x+y} + \ln(2x+y)}} + \sin(e^{3x-1} + \ln(3x-1));$$

$$d) T = \frac{EKUK(a, b, c)}{EKUK(m, n, k)} + EKUK(x, y, z).$$

25 variant

a) z_1, z_2, \dots, z_{20} massiv berilgan. Uning toq raqamli elementlarining kvadratini, juft raqamli elementlarini kubini hisoblab, yangi massiv hosil qiling.

b) $V(3,3)$ massiv berilgan. Uning har bir ustunining eng kichik elementlarining o'rta arifmetigini toping.

$$c) A = \frac{B(2x^2 + y) + B^2(x+y)}{B(2x-y)}; \text{ bu yerda } B(z) = \sum_{i=1}^5 \frac{z^2}{i} + \prod_{k=1}^5 \frac{k^2}{z}.$$

$$d) S = \frac{EKUB(a, b, c)}{EKUB(m, n, k)} + EKUB(x, y, z)$$

26-variant

a) x_1, x_2, \dots, x_n massiv berilgan. Uning toq elementlaridan U, juft elementlaridan Z massiv hosil qiling.

b) $B(3,3)$ massiv berilgan. Uning satr elementlarini kvadratga ko'tarib yangi massiv hosil qiling.

$$c) P = \frac{EKUB(n, m, n+n) + EKUB(n+m, n+k, m+k)}{EKUB(m^2, n^2, m+n)}, m, n, k \in N$$

$$d) U = \frac{\min(a, b) + \max(2a, b+c)}{\min(a+b, b+c) + \max(2a^2 + c, b^2 + a)}$$

27-variant

a) x_1, x_2, \dots, x_{20} massiv berilgan. Toq elementlardan U, juft elementlardan Z massiv hosil qiling.

b) $B(3,3)$ massiv berilgan. Uning birinchi va uchinchi ustun elementlarini 3 ga ko'paytirib yangi massiv hosil qiling.

$$c) Z = f(\sin x + \cos x, x+y) + f(\sin x, \cos x) + f(x-y, x) + f(\sin^2 x, \cos^2 x)$$

$$\text{bu yerda } f(d, t) = \begin{cases} d+t, & \text{agar } 4 > 1 \\ d-t, & \text{agar } 0 \leq 4 \leq 1 \\ t-d, & \text{agar } 4 < 0 \end{cases}$$

d) n ta muntazam besh burchak tomonlari bilan berilgan. Bu besh burchakka tashqi chizilgan doira yuzlarini hisoblang va ularni kamayish tartibida joylashtiring.

28-variant

a) x_1, x_2, \dots, x_n massivning musbat elementlaridan y_i , manfiy elementlaridan z_i , massiv hosil qiling.

b) $A(4,4)$ va $B(4,4)$ massivlar berilgan $C = A \cdot B + A - B \cdot C$ massivini hosil qiling.

$$c) y = \frac{\ln(y^2 + 6y + 1)}{1 + \cos(z^2 + 6z + 1)} + \frac{\sin(t^2 + 6t + 1)}{10} + x^2 + 6x + 1$$

$$d) T = \frac{EKUB(m, n, k)}{EKUB(x, y, z)} + EKUB(a, b, c).$$

29-variant

a) x_1, x_2, \dots, x_n massivning manfiy elementlarining eng kattasini toping.

b) $A(4,4)$ va $B(4,4)$ massivlar berilgan $C = A \cdot B + A - B \cdot C$ massivini hosil qiling.

$$c) B = \frac{1,8 + |1 + \cos^2 xy| + e^{-|1 + \cos^2(x+2y)|}}{9,94 - |1 + \cos^2(2x+y)|}$$

$$d) T = \frac{EKUK(m, n, k)}{EKUK(x, y, z)} + EKUK(a, b, c).$$

30-variant

- a) x_1, x_2, \dots, x_{55} sonlar berilgan: $x_1(x_2+x_3)$ $(x_4+x_5+x_6)\dots(x_{46}+x_{47}+\dots, x_{55})$ ni hisoblang.
- b) $A(N,M)$ matritsa har bir ustuni elementlari ko'paytmasini hisoblang va ularning eng kichigi va kattasini aniqlang.
- c) $A = \begin{cases} \min(y_1, y_2, y_3) + \min(z_1, z_2, z_3), & \text{agar } \min(a, b, c) > 5 \\ \sqrt{\min(t_1, t_2, t_3)} + 5, & \text{agar } \min(a, b, c) \leq 5 \end{cases}$
- d) n ta muntazam oltiburchak o'z tomonlari bilan berilgan. Bu oltiburchaklarga chizilgan aylana radiuslarini hisoblang va ularning kattasini aniqlang.

Mavzu: C++ Buliderda modullar va uning grafik imkoniyatlari . 4 - laboratoriya ishi topshiriqlari

Modullar va grafik imkoniyatlardan foydalanib dastur tuzish:

- a) *Satrlı ma'lumotlar bilan ishlash.*
- b) *Grafik ma'lumotlar bilan ishlash.*
- c) *Grafiklar bilan ishlash.*

1-variant

- a) «Informatika» so'zidan yangi 6 ta so'z hosil qiling.
- b) Ekranda aylana va uning ichida joylashgan uchburchak tasvirini hosil qilish dasturini tuzing.
- c) $y = \sin x + \cos 2x$ funksiyaning grafigini chizing.

2-variant

- a) Berilgan matndagi eng qisqa so'zni toping va uni teskari tartibda hosil qiling.
- b) Ekranda uy rasmini hosil qilish dasturini tuzing.
- c) $y = x^2 + 6x + 2$ funksiya grafigini chizing.

3-variant

- a) Berilgan matnda uchraydigan «s» va «t» harflari sonini aniqlang.
- b) Ekranda televizor tasvirini hosil qilish dasturini tuzing.
- c) $y = x + \sin x$ funksiya grafigi chizing.

4-variant

- a) «Komputer» so'zidan yangi 6 ta so'z hosil qiling.
- b) Ekranda samolyot rasmini hosil qilish dasturini tuzing.
- c) Asosi besh burchak bo'lgan og'ma prizma va uning kesimlarini chizing.

5-variant

- a) Berilgan matnda eng uzun va eng qisqa so'zlarni toping.
- b) Ekranda tog' va orqasida ko'rinib turgan quyosh rasmini hosil qiling.
- c) Ikkita parallel va perpendikulyar tekisliklar chizing.

6-variant

- a) «Modellashtirish» so'zidan yangi 4 ta so'z hosil qiling.
- b) Svetofor rasmini hosil qilish dasturini tuzing.
- c) Piramida va uning kesmalarini chizing.

7-variant

- a) Berilgan matndagi soʻzlarni teskari tartibda hosil qiling.
- b) Ekkranda asosi toʻrtburchakdan iborat boʻlgan piramida va prizma hosil qilish dasturini tuzing.
- c) Ekkranda konus va uning kesmalarini chizing.

8-variant

- a) «Algoritmash» soʻzidan yangi 5 ta soʻz hosil qiling.
- b) Ekkranda shaxmat doskasi hosil qilish dasturini tuzing.
- c) $y = |\sin x \cdot \cos x|$ funksiyaning grafigini chizing.

9-variant

- a) «Kibirnetika» soʻzidan yangi 7 ta soʻz hosil qiling.
- b) Ekkranda trapesiya va uning ichida joylashgan uchburchakni hosil qilish dasturini tuzing.
- c) $y = |\sin^i x \cdot \cos^i x|$ funksiyaning grafigini chizing.

10-variant

- a) Berilgan matndagi soʻzlardan nechtasi «b» va «d» harflar bilan boshlanishini aniqlang.
- b) Ekkranda toʻrtburchak va uning ichida joylashgan ochiq konvert tasvirini hosil qilish dasturini tuzing.

c) $y = \frac{8}{x} + \frac{x}{2}$ funksiyaning grafigini chizing.

11-variant

- a) «Matematika» soʻzidan yangi 6 ta soʻz hosil qiling.
- b) Ichma-ich joylashagan ellips, aylana, kvadrat va uchburchakni hosil qilish dasturini tuzing.

c) $y = \frac{10 - 2x}{x - 3}$ funksiyaning grafigini chizing.

12-variant

- a) Berilgan soʻzdan juft oʻrindagi va toq oʻrindagi harflar ketma-ketligini ajrating.
- b) Ekkranda stol tasvirini hosil qilish dasturini tuzing.
- c) Shar va uning kesimlarini chizing.

13-variant

- a) «Internet» soʻzidan yangi 4 ta soʻz hosil qiling.
- b) Ekkranda konus va silindr tasvirini hosil qilish dasturini tuzing.

c) $y = \frac{\sin x}{x}$ funksiyaning grafigini chizing.

14-variant

- a) Berilgan matnda necha marta qoʻsh harflar ketma-ketligi uchrashini aniqlang.
- b) Ikkita vagondan iborat poyezd tasvirini hosil qilish dasturini tuzing.
- c) $y = 2 \lg(x - 2)$ funksiyaning grafigini chizing.

15-variant

- a) «Dasturlash tili» soʻzidan yangi 6 ta soʻz hosil qiling.
- b) Choynak va piyola tasvirini hosil qilish dasturini tuzing.

c) $y = 2tg|2x + 1| + e^{3x}$ funksiyaning grafigini chizing.

16-variant

a) To'rt xonali sonni so'zlar bilan ifodalang.

b) Yengil avtomobil tasvirini hosil qilish dasturini tuzing.

c) $y = Cos|3x - 1| + ln 3x$ funksiyaning grafigini chizing.

17-variant

a) «Deraza» so'zidan yangi 5 ta so'z hosil qiling.

b) Shkaf tasvirini hosil qilish dasturini tuzing.

c) $y = x + e^x$ funksiyaning grafigini chizing.

18-variant

a) Berilgan sonning raqamlar yig'indisini toping.

b) «Parta» tasvirini hosil qilish dasturini tuzing.

c) $y = ctgx + 3e^x$ funksiyaning grafigini chizing.

19-variant

a) N ta so'z berilgan. Bulardan nechtasi bir xil harflar bilan boshlanadi va tugaydi. Shuni aniqlash dasturini chizing.

b) Monitor tasvirini hosil qilish dasturini tuzing.

c) $y = (x - 1)^3 - 3(x - 1)$ funksiyaning grafigini chizing.

20 - variant

a) Berilgan matnda qo'sh harflarni bitta harf bilan almashtiring.

b) Silindr tasvirini hosil qilish dasturini tuzing.

c) $y = 3 \ln(x + 1) + 1$ funksiyaning grafigini chizing.

21 - variant

a) Berilgan matnda «k»harflarini «d» harfiga almashtiring.

b) Avtomobil rasmini hosil qilish dasturini tuzing.

c) $y = 3 \ln(x + 1) + 1$ funksiyaning grafigini chizing.

22 - variant

a) Berilgan matnda mavjud so'zlarni uzunliklarini o'sish tartibida joylashtiring.

b) Uch o'lchovli koordinatalar sistemasini chizing, yo'nalishlarini ko'rsating va o'qlarini x, y, z bilan belgilang.

c) $y = ax + b$ funksiyaning grafigini chizing.

23 - variant

a) Berilgan matnda mavjud so'zlarning uzunliklarini kamayib borish tartibida joylashtiring.

b) Stol va uning ustida gul vazasini chizish dasturini tuzing.

c) $y = ax^2 + bx + c$ funksiyaning grafigini chizing.

24 - variant

a) Berilgan matnda so'zlarning bosh va oxirgi harflari o'rinlarini almashtiring.

b) To'g'ri chiziq bo'yicha harakatlanuvchi shar tasvirini chizing.

c) $y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ funksiyaning grafigini chizing.

25 - variant

a) Berilgan matnda eng qisqa soʻzni toping va shu soʻzni oʻzini qoʻshib yangi soʻz hosil qiling.

b) Vertikal toʻgʻri chiziq boʻyicha harakatlanuvchi shar tasvirini chizing.

c) $y = \frac{\ln|x^2 + 1|}{x} + \sqrt[3]{x^2}$ funksiyaning grafigini chizing.

26-variant

a) Berilgan matndagi soʻzlardan nechtasi «k» va «t» harflar bilan boshlanishini aniqlang.

b) Ekranda uchburchak va uning ichida joylashgan ellips tasvirini hosil qilish dasturini tuzing.

c) $y = \frac{x+2}{x^2}$ funksiyaning grafigini chizing.

27-variant

a) «Operator» soʻzidan yangi 6 ta soʻz hosil qiling.

b) Ichma-ich joylashgan kvadrat, romb hosil qilish dasturini tuzing.

c) $y = \frac{x^3}{3}$ funksiyaning grafigini chizing.

28-variant

a) Berilgan soʻzni polindrom yoki polindrom emasligini aniqlang.

b) Ekranda choynak va piyola tasvirini hosil qilish dasturini tuzing.

c) Silindr va uning kesimlarini chizing.

29-variant

a) «global» soʻzidan yangi 5 ta soʻz hosil qiling.

b) Ekranda kub va parallelogramm tasvirini hosil qilish dasturini tuzing.

c) $y = \frac{\operatorname{tg} x}{x}$ funksiyaning grafigini chizing.

30-variant

a) Berilgan matnda necha marta qoʻsh harflar ketma-ketligi uchrashini aniqlang.

b) Yuk avtomashinasi tasvirini hosil qilish dasturini tuzing.

c) $y = \cos^2(2x - 2)^2$ funksiyaning grafigini chizing