

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

GULISTON DAVLAT UNIVERSITETI
FIZIKA-MATEMATIKA FAKULTETI Axborot texnologiyalari
kafedrası

“5110700-Informatika o'qitish metodikasi” ta'lim yo'nalishi
6-16 guruh talabasi Namozov G'ayratninig

BITIRUV MALAKAVIY ISHI

Mavzu: Dasturlash tillarini o'qitishning interaktiv metodlari.

Ilmiy rahbar _____

Guliston 2020.

Mavzu: Dasturlash tillarini o‘qitishning interaktiv metodlari.

MUNDARIJA:

KIRISH.....	3
I-BOB. DASTURLASH TEXNOLOGIYALARI	7
1.1 Dasturlash texnologiyalari: tarixi, muammolari va yechimlari	7
1.2 Dasturlash texnologiyasi	12
1.3. Obektga yo‘naltirilgan dasturlash texnologiyalari	14
II-BOB. DASTURLASH TILLARINING IMKONIYATLARI	19
2.1. Paskal dasturlash tili va imkoniyatlari	19
2.2. Delphi dasturlash tili va imkoniyatlari	21
2.3. C++ dasturlash tili va imkoniyatlari.....	23
2.4. Dasturlash tillarining qiyosiy tahlili.....	24
III-BOB. DASTURLASH TEXNOLOGIYALARINI O‘QITISH.....	30
3.1. Umumta’lim maktablarida Pascal dasturlash tilini o‘rgatishning interaktiv metodlari.....	30
3.2. Akademik litsey va umumta'lim maktablarida Delphi va C++ dasturlash tilini o‘rgatishning interaktiv metodlari	33
3.3. Dasturlash tillarini o‘qitishda multimediya texnologiyalaridan foydalanish....	47
Umumiy xulosa va tavsiyalar	55
FOYDALANILGAN ADABIYOTLAR RO‘YXATI	56

Kirish

O'zbekistonning har bir sohada amalga oshirayotgan islohotlari, buyuk kelajakni ko'zlab qilinayotgan ishlari va loyihalari bilan bir qatorda iqtisodiyot, ta'lim, ilm-fan kabi ko'plab asosiy sohalarda jahon tajribalaridan kelib chiqib yangiliklarni kiritish, yurtimiz taraqqiyotida juda muhim omil hisoblanadi. Jumladan Prezidentimizning zamonaviy axborot texnologiyalarini har bir sohada keng qo'llashga doir ko'plab qaror va farmonlari aynan shu masalani qo'llab quvvatlaydi. O'zbekiston Respublikasi Prezidentining 2018-yil 13 dekabrda "O'zbekiston Respublikasi davlat boshqaruviga raqamli iqtisodiyot, elektron hukumat hamda axborot tizimlarini joriy etish bo'yicha qo'shimcha chora-tadbirlar to'g'risida"gi PF-5598-son Farmoni, O'zbekiston Respublikasi Prezidentining 2018-yil 21-noyabrda "raqamli iqtisodiyotni rivojlantirish maqsadida raqamli infratuzilmani yanada modernizatsiya qilish chora-tadbirlari to'g'risida"gi PQ-4022-son Qarori, "Axborot sohasi va ommaviy kommunikatsiyalarni yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida" O'zbekiston Respublikasi Prezidentining, 02.02.2019 yildagi PF-5653-son Farmoni, shuningdek "1million dasturchi loyihasi" ham Axborot Texnologiyalari sohasiga katta e'tibor qaratilayotganidan dalolatdir.

Ta'limda informatika fanining ahamiyati va zaruriyati yildan yilga o'sib bormoqda. Chunki axborot olamida yangiliklar kun sayin emas soat sayin ko'payib borayotgan, zamonaviy axborot texnologiyalari rivojlanib borayotgan bir paytda bu sohaga e'tiborsizlik yoki yangiliklar kiritmaslik yurt taraqqiyotida katta oqsashga olib keladi. Yosh avlod yangidan-yangi axborotlarni o'z vaqtida olishi, ilm-fan, texnika yutuqlaridan bahramand bo'lishlari muhim hisoblanadi.

Informatikaning bir bo'lagi sifatida dasturlash tillarini ko'rsatish mumkin. Dasturlash sohasining yildan yilga rivojlanib ommalashib borayotgani o'quvchilarga dasturlash tillarini o'qitishning ahamiyatini oshirmoqda. Yurtimizga ham aynan ushbu sohada mutaxassislar zarurligini inobatga olsak dasturlash tillarini o'qitishda eng samarali usul va metodlardan, shuningdek zamonaviy axborot texnologiyalaridan foydalanish zarur.

Maktablar informatika kurslarida algoritmlash asoslari, Pascal va Delphi dasturlash tillari o'qitiladi. Algoritmnı o'rgatish bu dasturlash tillariga kirishning birinchi bosqichi bo'lib, o'quvchılarda fundamental bilim va aqliy ko'nikmani shakllantiradi. Pascal dasturlash tili sodda ko'rinishdagi yuqori darajali dasturlash tili hisoblanadi. Aynan ushbu dastur o'quvchılarning yosh xususiyatlariga to'la mos keladi. Xuddi shuningdek Delphi dasturlash tili ham dasturlashni o'rganishda o'quvchılar uchun juda qulay hisoblanadi. Aynan Pascal dasturlash tilining mantiqiy davomi sifatida va sodda interfeysga egaligi uchun ham Borland Delphi dasturi informatika kursiga mos tushgan. Ushbu bitiruv malakaviy ishining mavzusi va mohiyatidan kelib chiqqan holda ushbu tavsıflarnı keltirib o'tish mumkin:

Masalaning qo'yilishi: Bitiruv malkaviy ishining maqsadi "Dasturlash tillarini o'qitishning interaktiv metodlari"ni yoritish, dasturlash tillarini o'qitishga yangicha yondoshish haqida batafsil tushuntirishlar berish. Ushbu manbaadan o'qituvchi pedagoglar foydalanishi va ish jarayoniga tadbıq qilishlari nazarda tutilgan.

Mavzuning dolzarbligi: Hozirgi paytda dunyo miqyosida axborot texnologiyalari va aynan dasturlash sohasiga katta ahamiyat berilayotgan bir paytda, yurtimiz taraqqiyoti uchun ushbu sohada yoshlarimiz bilim, ko'nikmalarini oshirish muhim ahamiyatga ega ekanligini aytib o'tish mumkin.

Dasturlashni o'rganish o'quvchılar ongida nafaqat ushbu sohadagi bilimlarini oshirishda balki, yaxshi fikrlay olishlari, masalaga teran munosabatda bo'la olishlari uchun ham o'rgatilishi maqsadga muvofiq.

Muammoning o'rganilganlik darajasi: Ushbu mavzuni yoritishda kerakli manbaalar to'plandi, aynan ushbu sohadagi kamchiliklar, muammolar o'rganildi va shu asosda zarur fikrlar va rejalar bayon etildi. Yurtimizda dasturlash tillarining o'qitilishidagi, zamonaviy axborot texnologiyalaridan samarali foydalanishda bir qancha muammolar borligi ochiqlandi va ularni bartaraf etishga qaratilgan mavzular yoritildi.

Tadqiqot maqsadi: Dasturlash tillarini o'qitishda zamonaviy dars texnologiyalari, yangicha va samarali interaktiv metodlarnı va multimedıya vosıtarini qo'llagan holda darslarnı olib borib, o'quvchılarning bilimi, bilim saviyasini va amaliy ko'nikmalarini

oshirish. Jahon talablari asosida dars berish va yosh avlodni dasturlash tillarini saviyali o'rganishi uchun yangicha yondoshishni ochib berish.

Tadqiqot vazifalari: Ta'lim tizimida yangicha noan'anaviy texnologiyalarni qo'llashni o'rgatish, dasturlash tillarini o'rgatishda samarali interaktiv metodlardan foydalanish o'quvchilar bilim va qiziqishini yanada oshirishini ko'rsatish, ko'plab metodlarning dars jarayonida qo'llanilishi va samarsini ochib berish.

Tadqiqot obyekti va predmeti: Bitiruv malakaviy ishining obyekti sifatida bevosita dasturlash tillarini aytish mumkin. Uning predmeti esa zamonaviy interaktiv metodlar hisoblanadi. Aynan dasturlash tillarini o'qitishda interaktiv metodlardan foydalanish mavzusi o'zaro ma'zmunli uyg'unlashgan bo'lishi zarur. Mavzu to'liq yoritilishi va saviyali bo'lishi tadqiqot obyekti va predmetining mutanosibligiga bog'liq.

Kutiladigan natija: Darsni olib borishda zamonaviy interaktiv metodlardan foydalanish va dasturlash tillarini o'qitishda, har bir mavzuni o'tishda eng samarali dars texnologiyasi va metodalarini bilgan holda dars o'tish va o'quvchilar bilim saviyasi, qiziqishi va amaliy ko'nikmasini yuksaltirish.

Tadqiqot metodlari: mavzuga doir umumiy ma'lumatlarni keltirgan holda mavzuning dolzarbligini ochib berish, har bir dasturlash tili haqida zarur ma'lumotlarni bergan holda, o'quvchilar uchun qanchalik zaruriy bilimlar ekanligini ko'rsatish, hozirgi davrda ta'lim jarayoniga yangicha yondoshish zarurligi, yangicha interaktiv metodlardan foydalanishning bir qancha usullarini yoritgan holda, yuqori natijalarga erishishni ko'rsatish.

Tadqiqotning ilmiy yangiligi va amaliy ahamiyati:

Hozirgi vaqtda ta'lim jarayonida o'qitishning zamonaviy metodlari keng qo'llanilmoqda. O'qitishning zamonaviy metodlarini qo'llash o'qitish jarayonida yuqori samaradorlikka erishishga olib keladi. Bu metodlarni har bir darsning didaktik vazifasidan kelib chiqib tanlash maqsadga muvofiq. An'anaviy dars shaklini saqlab qolgan holda uni ta'lim oluvchilar faoliyatini faollashtiradigan turli-tuman metodlar bilan boyitish ta'lim oluvchilarning o'zlashtirish darajasi o'sishiga olib keladi.

Bugungi kunda bir qator rivojlangan mamlakatlarda ta'lim-tarbiya jarayonining samaradorligini kafolatlovchi zamonaviy pedagogik texnologiyalarni qo'llash borasida katta tajriba asoslarini tashkil etuvchi metodlar interfaol metodlar nomi bilan yuritilmoqda. Interfaol ta'lim metodlari hozirda eng ko'p tarqalgan va barcha turdagi ta'lim muassasalarida keng qo'llanayotgan metodlardan hisoblanadi. Shu bilan birga, interfaol ta'lim metodlarining turlari ko'p bo'lib, ta'lim-tarbiya jarayonining deyarlik hamma vazifalarini amalga oshirish maqsadlari uchun moslari hozirda mavjud. Amaliyotda ulardan muayyan maqsadlar uchun moslarini ajratib tegishlicha qo'llash mumkin. Bu holat hozirda interfaol ta'lim metodlarini ma'lum maqsadlarni amalga oshirish uchun to'g'ri tanlash muammosini keltirib chiqargan.

Foydalanilgan manbaa va adabiyotlarning tanqidiy taxlili: Bitiruv malakaviy ishini tayyorlashda dasturlashga oid turli kitoblar va o'quv qo'llanmalardan foydalanildi. O'zbek olimlarining kitoblaridan tashqari jahonga mashhur axborot texnologiyalari sohasi bo'yicha kitoblardan ham foydalanildi. Shuningdek internet manbaalaridan ham kerakli ma'lumotlar to'plangan holda mavzular umumlashtirildi. Zamonaviy interaktiv metodlarni aynan dasturlash tillarini o'rgatishda qo'llash mavzusini ochib berishda internet manbaalaridan keng foydalanildi.

Ishning taqribiy tuzilishi: BMI mavzusi "Dasturlash tillarini o'qitishda interaktiv metodlar" ekanligini inobatga olsak kirish qismini aynan informatika kursidagi dasturlash tillari haqida va ahamiyati, dolzarbligi yoritilgan. I-bobda dasturlash texnologiyalari haqida umumiy ma'lumotlar keltirilgan. II-bobda hozirda ta'lim tizimimizda bo'lgan dasturlash tillarining har biri uchun alohdida to'xtalib o'tilgan. III-bobda esa dasturlash tillarini o'qitishda interaktiv metodlardan foydalanish haqida ma'lumotlar va misollar keltirilgan. BMI so'ngida mavzu yuzasidan xulosalar va takliflar berilgan.

I-BOB. DASTURLASH TEXNOLOGIYALARI

1.1. Dasturlash texnologiyalari: tarixi, muammolari va yechimlari

Texnik vositalari sohasida ma'lumotlarga ishlov berish va ularni qayta ishlash so'nggi yillarda har tomonlama jiddiy muvaffaqiyat qozondi. Har ikki-uch yilda yangilanadigan kompyuterlarning yanada samaraliroq avlodlari buning natijasidir.

Texnik vositalarining o'zgarishi, shuningdek yangi kompilyatorlar va operatsion tizimlar paydo bo'lishi sababli dasturiy ta'minot xizmat muddati davomida o'zgartirishlarning katta miqdori zarur bo'lib qoladi. Ma'lum darajada ular "yuqoridan pastga" qoidasi ishlamaydi, chunki yangi muhit eski dasturlarga xizmat ko'rsata olmaydi va eski turdagi kompilyatorlar komandasini uzoqroq tutib turmaydi. Bundan tashqari mazkur yangi vositalar eski dasturlashni osonlashtiradigan yangi vositalarning katta miqdoriga ega.

Keyingi sabab shuki, kompilyatorlarning dolzarb versiyasi apparat vositalari yangi platformasidagi ortiq ishlashga qodir emas (ya'ni funktsional bo'lmay qoladi) va ularni tegishlisi bilan almashtirishga tayyorgarlik ko'rish lozim. Bu asosda o'zgarishlarning natijasi shuki, ko'p hollarda iloji boricha ko'proq dasturlarni tez o'zgartira olish uchun ularni notartib deb atalgan o'zgartirishlar natijasidir. Etarli darajada hujjatlashtirishni o'zgartirishni amalga oshirmaslik dasturlarni tushunib bo'lmaydigan qiladi.

Shu bilan birga tan olishmaydiki yangi standart dasturiy ta'minotni rejalashtirish va kiritishning o'zi ko'pincha juda uzoq davom etadi va o'z kuchlari bilan ishlab chiqilgan va yana kutish kerak bo'ladigan bir qator to'ldirishlar bilan tugaydi. Mana shu sabablarga ko'ra sanoati industrallashgan mamlakatlarda eski odatlarni yengib o'tish yoki texnik xizmat ko'rsatish tarmog'i bo'yicha dasturiy ta'minot sohasida mashg'ul bo'lgan barcha xodimlarning 50 dan 80 foizigina mehnat qiladi. Qolgan 20-50 foiz xodim tizimli va amaliy dasturiy ta'minotni ishlab chiqish va uni kelgusida rivojlantirish bilan shug'ullanadi.

O'zbekistonda zamonaviy dasturiy ta'minot bilan baravar qo'llaniladigan va albatta, dasturiy ta'minot bilan soddalashtirilmagan bir qator o'ziga xos dasturiy ta'minot mavjudki, mamlakatda umuman olganda bu o'ziga xos xususiyat bilan qiyoslasa

bo'ladigan vaziyat hukmronligini ifoda etadi. Bu yerda dasturchilar uchun dasturiy ta'minot xizmatini ko'rsatish, uni bundan keyin rivojlantirish (takomillashtirish) bo'yicha, zarurat tug'ilganda esa mamlakatning iqtisodiy va ma'muriy jarayonlari uchun zarur bo'lgan yangi dasturiy ta'minotni ishlab chiqish bo'yicha faoliyatning keng maydoni mavjud.

Eng avvalo, dasturiy ta'minotning turlari va ularning vazifalari haqidagi tasavvurlar dasturlar yoki butun bir amaliy tizimlarni loyihalash, amalga oshirish va xizmat ko'rsatishdagi harakat usullari va chizmalari bilan birga beriladi.

XX asrning 80 - yillaridan oldin ishlab chiqilgan kompyuterlar uchun katta dasturiy sistemalarni ishlab chiqish juda ham mushkul vazifa edi. Buning eng katta sababi shu davrga xos bo'lgan kompyuterlarning imoniyatlarining chegaralanganidadir. Dasturiy komplekslarni ishlab chiqishda asosiy cheklanishlar kompyuter tezkor xotirasining sig'imi, ma'lumotlarni ikkilamchi xotira qurilmalaridan (magnit lentalar, barabanlar va h.k.) o'qish tezligi, protsessorning ishlash tezligi (ularning takt chastotalari bir necha yuz mikrosekund bo'lgan) bilan bog'liq. Bu davrdagi kompyuterlar xalq ho'jaligining hisob-kitob bilan bog'liq bo'lgan masalalarini yechish uchun mo'ljallangan edi. Kompyuterlarda boshqa xarakterdagi masalalarning yechishning iloji yo'q edi. Dasturchilar uchun shu kompyuterlarda yechilayotgan masala uchun berilgan ma'lumotlarni hisobga olgan holda masalaning yechish algoritmini ishlab chiqish birinchi o'rindagi vazifa hisoblangan. Bu borada N. Virtning mashhur ***ma'lumotlar+algoritm=dastur*** formulasini yodga olish yetarli.[12-14]

1980 yillardan keyin ishlab chiqilgan kompyuterlarning i-mkoniyatlari kengayib, ishlab chiqarish narxi keskin pasaygani tufayli ulardan foydalanish samarasi ortib bordi. Natijada, dastlab matn va grafiklar uchun muharrirlari ishlab chiqildi, keyinchalik multimedia qurilmasi yordamida tovushli ma'lumotlarni qayta ishlash imkoniyatlari ham paydo bo'ldi. Kompyuter tasviriyl imkoniyatlarining kengayishi foydalanuvchilar uchun qulay bo'lgan grafik muloqot interfeysini yaratilishiga sabab bo'ldi. Natijada yuqori murakkablikdagi katta amaliy dasturlarni yaratish foydali hamda zarur bo'lib qoldi. Dasturiy vositalarni ishlab chiqish uchun qurol sifatida yuqori bosqichli algoritmik

tillardan foydalanildi. Bu dasturiy vositalar dasturchi va dasturchilar guruhi imkoniyatlarini kengaytirib, dasturiy mahsulotlarning murakkablik darajasining ortishiga ham sabab bo'ldi.

XX asrning 70-80 yillarda dasturiy mahsulotlarning murakkablik darajasini hal qilish uchun ko'plab metodlar ishlab chiqildi. Ularning ichida yuqoridan pastga usulidagi strukturali dasturlash keng tarqaldi. Bu usul yuqori bosqichli FORTRAN va COBOL tillarining topologiyasiga asoslangan edi. Bu tillarda asosiy bazaviy birlik bo'lib qism dastur xizmat qiladi. Dastur shakli jihatidan daraxtni eslatib, bitta qism dastur boshqasiga murojaat qilar edi. Strukturali dasturlashda aynan shunday yondoshuvdan foydalaniladi: algoritmik dekompozitsiya katta masalalarni kichik masalachalarga bo'lish uchun qo'llanadi.

XX asrning 60-70 yillarida xalq xo'jaligining ko'plab masalalarini kompyuter yordamida yechish uchun BASIC, PASCAL, FORTRAN kabi dasturlash tillari, DBASE. CLIPPER, FOXBASE, KARAT kabi ma'lumotlar bazasini boshqaruv tizimlari yetarli bo'lgan. Bu til va tizimlar asosan IBM*86 tipidagi kompyuterlarga mo'ljallangan bo'lib, ularning asosiy kamchiliklari shu davrga hos bo'lgan kompyuterlarning imkoniyatlari bilan bog'liq edi:

- foydalanuvchi uchun qulay bo'lgan grafik interfeys yaratish imkoniyatining yo'qligi yoki juda ham katta mehnat talab qilishi;
- bitta dasturda foydalanish mumkin bo'lgan o'zgaruvchilar sonining chegaralanganligi;
- dasturda foydalanish mumkin bo'lgan sonli ma'lumotlar dipazonining chegaralanganligi;
- kiritilayotgan va chiqarilayotgan ma'lumotlarni nazorat qilish;
- buyruqlar tizimining «kambag'alligi»;
- grafiklar bilan ishlash imkoniyatining chegaralanganligi;
- tovushli ma'lumotlarni qayta ishlashdagi kamchiliklari;
- ma'lumotlar bazasi va fayllar bilan ishlashdagi imkoniyatlarning chegaralanganligi;

- yangi tipdagi ma'lumotlar faqat shartli ravishda yaratish mumkin mumkinligi.

Keyingi yillarda strukturali dasturlashning o'ndan ortiq usullari ishlab chiqildi. Bu usullarning ko'pchiligi bitta mavzuning turli ko'rinishlaridan iborat bo'ldi. Ammo, Sommervil ularni uchta guruhga bo'lishni taklif qildi:

- yuqoridan quyiga qarab strukturali loyihalash usuli;
- ma'lumotlar oqimini tashkil qilish usuli; - ob'ektga yo'naltirilgan loyihalash.

Strukturali loyihalash usuliga namunalar Yordan va Konstantin keltirilgan. Asosiy g'oyalar esa Virtning ko'plab ilmiy ishlarida ochib berilgan. Bu asarlarda strukturali dasturlashga nisbatan bo'lgan har bir yondoshuvda algoritmik dekompozitsiya ishtirok etadi. Yozilgan dasturlarning kattagina qismining asosini strukturali dasturlash prinsipi tashkil qiladi. Strukturali dasturlash abstraksiyalarni ajratishga imkon bermaydi, ma'lumotlarni samarali himoya qilishni ta'minlamaydi, parallelizmni tashkil qilish uchun yetarlicha vositalarni taklif qila olmaydi. Strukturali dasturlash murakkab sistemalarni ishlab chiqishga imkon bermaydi, ob'ektga yo'naltirilgan dasturlashda samara bermaydi. [12;17]

Ma'lumotlar oqimini tashkil qilish usuli Djekson hamda Orrlarning ishlarida keng yoritilgan. Bu usulda dasturiy sistemaning strukturasi xuddi kiruvchi ma'lumotlarni chiquvchi ma'lumotlar qimiga almashtirish tashkil qilish kabi quriladi. Ma'lumotlar oqimini tashkil qilish usuli ham strukturali dasturlash usuli kabi bir qator murakkab masalalarda, hususan axborot ta'minot sistemalarida qo'llanilgan. Bu masalalarda kiritilyotgan va chiqarilayotgan ma'lumotlar o'rtasida bevosita bog'liqlik mavjud hamda dasturlarning bajarish tezligiga katta e'tibor berilmaydi.

1980-yillarga kelib, imkoniyatlari yanada katta bo'lgan kompyuterlar ishlab chiqildi. Strukturali dasturlashning ahamiyati esa o'zgarmadi. Ammo Stayn[36] shuni ta'kidlaydiki «dasturning hajmi 100 000 satrdan o'tganidan keyin strukturali dasturlash mexanizmi ishlamay qolar ekan». Demak, katta dasturiy sistemalarni ishlab chiqishda strukturali dasturlash mexanizmi ham o'z ahamiyatini yo'qotadi.

Shu munosabat bilan dasturiy ta'minotda ob'ektga yo'naltirilgan dasturlash texnologiyalariga o'tish taklif qilindi.

Ob'ektga yo'naltirilgan dasturlash atamasi dastlab 1960-yillarning o'rtalarida, Simula-67 dasturlash tilida paydo bo'lgan. Ammo, bu til FORTRAN, ALGOL, PL/I kabi tillarga nisbatan o'zining imkoniyatlari past bo'lganligi hamda shu davrda yechish talab qilingan masalalar uchun yetarli vositalarni taklif qila olmagan uchun dasturchilar orasida keng ommalasha olmadi. Keyinchalik ob'ektga yo'naltirilgan dasturlash Smalltalk, Object Pascal, C++, Ada, hozirda esa C# kabi yuqori darajali dasturlash tillarining topologiyasi asosida rivojlandi va ommalashmoqda. [7;15]

Dasturlash tillari haqida umumiy ma'lumot

Til	Yaratilgan yili	Mualliflar	Tashkilot, firma
Ada	1979-80	Jean Ichbian	Cii-Honeywell (Fransiya)
ARL	1961-1962	Kenneth Iverson, Adin Falkoff	IBM
DELPHI	1995	Chak (Chuck) va Denni (Danny)	Borland
BASIC	1964-1965	John Kemeny, Thomas Kurtz	Dartmouth Colleje
C	1972-1973	Dennis Ritchie	Bell Laboratories
C++	1980	Bjarne Stroustrup	Bell Laboratories
FORTRAN	1950-1958	John Backus	IBM
HTML	1989	Tim Berners-Li	CERN, Jeneva
LOGO	1968-70	Seymour Papert	Massachusetts Institute of Techn.
Pascal	1967-1971	Niklaus Wirth	Federal Institute of Technology (Shveytsariya)
SIMULA	1967	Ole-Yoxan Dal, Kristen Nigaard	Norvegiya XM
Java	1995	Djeyms Gosling	Sun Microsystems

[21-24]

1.2. Dasturlash texnologiyasi

Har bir o'quvchi uchun dastur, dasturlash texnologiyasi uchunchalariga oydinlik kiritib o'tishmizga to'g'ri keladi:

Dastur - bu berilgan algoritmgaga asoslangan biror bir algoritmik tilda yozilgan ko'rsatmalar, ya'ni buyruqlar yoki operatorlar to'plamidir. Algoritmnlarni to'g'ridan-to'g'ri kompyuterga kiritib yechib bo'lmagani sabab ularni biror bir algoritmik tilga o'tkaziladi.

Dasturlash – bu dastur yaratish bo'lib, uni yaratishda quyidagi bosqichlar bosib o'tiladi:

- Qo'yilgan masalani dasturlash mumkinligini tekshirish;
- Qo'yilgan masalaning algoritmini tanlash yoki qayta ishlash;
- Buyruqlarni yozish;
- Dastur xatoliklarini tekshirish;
- Testdan o'tkazish.

Dasturlash - esa bu dastur tuzish jarayoni bo'lib, u quyidagi bosqichlardan iboratdir:

- dasturga bo'lgan talablar;
- qo'yilgan masala algoritmini tanlash yoki ishlab chiqish;
- dastur kodlarini (matnlari, buyruqlarni) yozish;
- dasturni to'g'rilash va test o'tkazish.

Dastur so'zi ham komandalarning alohida blokini (berilgan kodini) aniqlovchi so'z, ham yaxlit holdagi bajariluvchi dasturiy mahsulotni belgilovchi so'z sifatida ishlatiladi.

Bu ikki xillilik o'quvchini chalg'itishi mumkin. Shuning uchun unga aniqlik kiritamiz. Demak dasturni yo dasturchi tomonidan yoziladigan komandalar to'plami, yoki amallar bajaradigan kompyuter mahsuloti sifatida tushinamiz.

Dasturlash texnologiyalari – dasturiy ta'minotning inqiroziga javob sifatida yuzaga kelgan dasturlash texnologiyalari hisoblanadi. Bu inqirozning sababi shunda ediki, strukturali dasturlash metodlari murakkablik darajasi borgan sari ortib borayotgan masalalar uchun dasturiy ta'minot yaratish imkonini bera olmay qoldi. Buning natijasida turli loyihalarni bajarish rejalari buzildi, qilinayotgan harajatlar belgilangan byudjetdan ortib ketdi, dasturiy ta'minotning funksionalligi buzildi, hatoliklari ortdi.

Dasturiy ta'minotning eng muhim tomonlaridan biri – uning murakkablik darajasidir. Biror dasturchi sistemaning barcha xususiyatlarini to'liq hisobga ola olmaydi. Shuning uchun uni ishlab chiqishda dasturchi va boshqa mutaxassislarining yirik jamoasi qatnashadi. Demak, qo'yilgan masalaga to'g'ridan-to'g'ri bog'liq bo'lgan murakkabliklarga ana shu jamoaning ishini bir maqsadga qaratilgan boshqarish ham qo'shiladi. An'naviy dasturlash tillarida bunday murakkabliklarni hal qilishda "ajrat va boshqar" prinsipidan foydalanilgan. Ya'ni, masala kichik-kichik masalalarga ajratib, keyin har bir masala uchun alohida dastur ishlab chiqilgan va birlashtirilgan.

Dasturchilar tomonidan ma'lum bir soha o'rganilib, uning alohida ob'ektlari ajratib olingan. Bu ob'ektlar uchun masalalarni yechishda qo'llash mumkin bo'lgan xususiyatlar aniqlangan. Ehtiyojga qarab har bir xususiyat ustida bajarish mumkin bo'lgan amallar aniqlangan. So'ngra o'rganilayotgan sohaning har bir real ob'ektiga mos dasturiy ob'ekt ishlab chiqilgan.

Ma'lumki, kompyuter yordamida hal qilinadigan har bir masala uchun maxsus dastur ishlab chiqish yozish talab qilinadi. Bunday masalalar sinfining kengayib borishi albatta yangi-yangi dasturlar yaratishga olib keladi. Yangi dasturlarni yaratish uchun «eski» dasturlash tillarining imkoniyati yetmay qolganda yoki dastur yaratish jarayonini mukammallashtirish uchun yangi dasturlash tiliga ehtiyoj paydo bo'ladi. Bu masalani albatta malakali va tajribali dasturchilar yoki dasturchi guruhlar tomonidan hal qilinadi. Ana shu dasturchilarni yetishtirish bugungi kunda informatika fani o'qituvchilarining eng asosiy masalasi hisoblandi. [6;15;24]

Professional dasturchi bo'lish uchun nimalarga e'tibor qaratish kerak?

Vaqt o'tishi bilan dasturchilar oldiga quyilgan masalalar o'zgarib boryapti. Bundan yigirma yil oldin dasturlar katta hajmdagi ma'lumotlarni qayta ishlash uchun tuzilar edi. Bunda dasturni yozuvchi ham, uning foydalanuvchisi ham kompyuter sohasidagi bilimlar bo'yicha professional bo'lishi talab etilardi. Hozirda esa ko'pgina o'zgarishlar ro'y berdi. Kompyuter bilan ko'proq uning apparat va dasturiy ta'minoti haqida tushunchalarga ega bo'lmagan kishilar ishlashyapti. Kompyuter odamlar tomonidan uni chuqur o'rganish

vositasi emas, ko'proq o'zlarining oldilariga qo'yilgan, o'zlarining ishlariga tegishli bo'lgan muammolarini yechish instrumenti bo'lib qoldi.

Foydalanuvchilarning ushbu yangi avlodini dasturlar bilan ishlashlarini osonlashtirilishi bilan bu dasturlarning o'zini murakkabligi darajasi oshadi. Zamonaviy dasturlar - foydalanuvchi bilan do'stona munosabatni yuqori darajada tashkil qiladigan ko'p sonli oynalar, menyu, muloqot oynalari va vizual grafikaviy muhitlardan tarkib topgan interfeysga ega bo'lishi lozim.

Dasturlashga talabni o'zgarishi nafaqat tillarning o'zgarishiga balki uni yozish texnologiyasini ham o'zgarishiga olib keldi.

Shu o'rinda quyidagi savol tug'iladi: yaxshi dasturchi bo'lish uchun nimalarga etibor qaratish kerak?

Yaxshi dasturchi bo'lishi uchun asosan quyidagilarga etibor qaratish kerak:

1. Matematika fanini puxta o'zlashtirgan bo'lishi kerak;
2. Jamoa bo'lib ishlay olishi kerak;
3. Ingliz tilini yaxshi o'zlashtirgan bo'lishi kerak;
4. Bir joyda o'tirib uzoq muddat ishlay olishi kerak.

Dasturlashni o'rganishda qanaqangi manbalardan foydalanish kerak?

Dasturlashni o'rganishda quyidagi manbalardan foydalanishimiz kerak:

1. Dasturlashni o'rgatuvchi kitob yoki ma'ruzalar matni;
 2. Dasturlashdan masalalar to'plami;
 3. Dasturlash asoslarini o'rgatuvchi video kurs;
 4. Sizda paydo bo'lgan savollarga javob beruvchi shaxs yoki manba;
 5. Tuzilgan dasturning to'g'ri yoki noto'g'riligini tekshiruvchi amaliy dasturlar.
- Shundagina dasturlash tillarini amaliyotda qo'llash ham oson va ham samarali bo'ladi.[18;24]

1.3. Obektga yo'naltirilgan dasturlash texnologiyalari

Obektga yo'naltirilgan dasturlash – bu dasturlashning shunday yangi yo'nalishi, dasturiy sistema o'zaro aloqada bo'lgan ob'ektlar majmuasi sifatida qaraladi va har bir ob'ektni ma'lum bir klassga mansub hamda har bir klass qandaydir shajarani hosil qiladi

deb hisoblanadi. Alohida olingan klass ma'lumotlar to'plami va ular ustida bajariladigan amallarning to'plami sifatida qaraladi. Bu klassning elementlariga faqat shu klassda aniqlangan amallar orqali murojaat qilish mumkin. Dasturdagi ma'lumotlar va ular ustida bajariladigan amallar o'rtasidagi o'zaro bog'liqlik an'anaviy dasturlash tillariga nisbatan dasturiy sistemalarning ishonchliligini ta'minlaydi. Ob'ektga yo'naltirilgan dasturlashning eng asosiy tushunchasi ob'ekt va klass hisoblanadi.

Obekt. Oldimizda turgan o'rik mevasini dasturlash terminologiyasi orqali ko'raylik. An'anaviy dasturlashda biz uni qismlarga bo'lib o'rganganmiz: S-o'rik po'stining yuzi, J – o'rik mevasidagi sharbat hajmi, F – po'st orasidagi meva og'irligi, D-danak og'irligi va h.k. Endi ana shu o'rikga rassom ko'zi bilan qaraylik. O'rikning rasmi bu meva emas, balki o'rik mevasining yassi tekisliklikdagi tasviridir. Uni har biri alohida va bir-biriga bog'liq bo'lmagan biror ma'lumotlar segmentida turgan bir nechta ma'lumotlardan iborat bo'lgan ko'rinishda abstraksiyalash mumkin emas. O'rikning komponentalari doimo birgalikda va bu komponentalar orasidagi o'zaro aloqalarni saqlagan holda qaraladi.

Obekt - biz yashayotgan o'rikdagi biror elementga hos bo'lgan barcha ma'lumot va hulqlarni, ya'ni shu element ustida bajarish mumkin bo'lgan xarakatlarni ifodalaydi hamda ma'lumotlarning tugal abstraksiyasidan iborat bo'ladi. Bu ma'lumot va hulqlar ob'ektga yo'naltirilgan dasturlash atamashunosligida mos ravishda *hususiyat* va *metod* deb ataladi. Hususiyatni ob'ektning *maydoni* deb ham yuritiladi. Masalan, shashka ob'ekti rang, vertikal maydondagi o'rni, gorizontal maydondagi o'rni kabi maydonlarga, surish, urish, «damka» ga chiqish, shashka taxtasidan chetga olish kabi metodlarga ega bo'ladi.

Maydon va metodlar birgalikda ob'ektning a'zolari deyiladi[38]. Ob'ektlarning strukturasi ularning o'zaro aloqasini ifodalaydi.

Klass. Har bir ob'ekt qandaydir bitta klassga taaluqli bo'ladi. Klass – bu murakkab struktura bo'lib, o'z ichiga ma'lumotlarni, protsedura va funksiyalarni ifodalashdan tashqari, klasslarning vakili bo'lmish ob'ektlar ustida bajarilishi mumkin bo'lgan amallarni ham oladi. Klassdagi ma'lumotlar maydonlar, protsedura va funksiyalar esa metodlar deb ataladi. Klassning strukturasi tushunchasi ham muhim hisoblanadi. U tizim

ichidagi vositalar boyligini namoyon qiladi. Ma'lumki, biror yaproqdagi fotosintez jarayonini o'rganish uchun shu yaproqdagi bitta hujayrani ko'rish yetarli, chunki, qolgan hujayralar ham o'zini ana shu o'rganilgan hujayra kabi tutadi. Biz xam biror klass, tip yoki bo'limga mansub bo'lgan ob'ektni ko'rar ekanmiz, faraz qilishimiz mumkinki, uning hulqi ham shu tipdagi boshqa ob'ektlarniki kabi bo'ladi.

Obektga yo'naltirilgan dasturlash bizning hulqimizga o'xshab ketadigan dasturlash usulini ifodalaydi. U dasturlash tillarini ishlab chiqishdagi yangiliklarning tabiiy evolyutsiyasi hisoblanadi. U barcha avvalgi dasturlash tillariga qaraganda strukturaliroq, ma'lumotlarni abstraktlashga urinishda abstraktliroq va modulliroqdir.

Obektga yo'naltirilgan dasturlash o'zining uchta asosiy hususiyatlari bilan xarakterlanadi. [7;13]

1. Inkapsulyatsiya – yozuvlarni shu yozuvlarning maydonlari ustida amallar bajarish uchun mo'ljallangan funksiyalar va protseduralar bilan birgalikda olib boradi. Inkapsulyatsiya yashirish prinsipiga ham ega bo'ladi. Buning ma'nosi shuki, ob'ekt ustida bajarish mumkin bo'lgan xarakatlarni amalga oshirish vositalari shu ob'ektdan foydalanayotgan dasturchilar ko'zidan yashirib qo'yiladi. Dasturchi faqat shu ob'ektning ayrim metod va maydonlari bilangina ishlay oladi. Boshqacha aytganda, barcha maydon va metodlar ichki va tashqi guruhlariga bo'linadi. Ob'ektning ichki a'zolari dasturchiga «ko'rinmaydi» va ob'ektning hulqi va imkoniyatlarini aniqlaydi, tashqi a'zolari esa dasturchiga «ko'rinib turadi» hamda ob'ektni boshqarish imkonini beradi. Ob'ektning dasturchiga ko'rinib, ob'ektni boshqarishga yordam beradigan metod va maydonlari (hususiyatlari) ob'ektning interfeysi deb ataladi. Dasturchi ob'ekt bilan ishlash uchun uning interfeysini bilishi kifoya. Masalan, yengil avtomobilni boshqarishni o'rganish uchun uning matorining ishlash prinsipi, g'ildiraklarning burilishini, tormoz mexanizmini o'rganishning hojati yo'q, rulni burash, pedal yoki uzatmalar qutisi richagini bosishni bilish yetarli.

2. Vorislik – yangi ob'ektni aniqlashga ehtiyoj paydo bo'lganda oldindan ma'lum bo'lgan ob'ektlardan foydalanish. Ob'ektlar o'zlarini yaratishda ishtirok etgan ota ob'ektlarning xarakteristika va hulqlarini meros qilib olishlari mumkin. OYD

konsepsiyasi yangi klasslarni mavjud klasslarga yangi maydonlar, hususiyatlar va metodlarni qo'shish orqali yaratish imkonini ham beradi. Yangi klasslarni tashkil qilishning bunday usuli yuzaga keltirish deb ataladi. Bu holda yuzaga kelgan yangi klass o'zining bazaviy ota klassiga hos bo'lgan hususiyat va metodlarini meros oladi. Hayotdan misol qilib, hasharotlar klassini olish mumkin. U ikki guruhga bo'linadi: qanotlilar va qanotsizlar. Qanotli hasharotlarga kapalaklar, pashshalari, parvonalar va x.k. lar kiradi. SHuning uchun, pashshalarni qaytadan qanotli deb ta'riflashning hojati yo'q, u bu hususiyatni ota klassi bo'lgan qanotli hasharotlar klassidan meros qilib oladi.

3. Polimorfizm – xarakatga biror nom berish hamda undan birgalikda ob'ektlar shajarasining quyi va yuqori qismlarida foydalanish. Bunda shajaraning har bir ob'ekti bu xarakatni o'zi uchun hos bo'lgan usul bilan bajaradi. Polimorfizm — bu turli klasslarga kirgan metodlar uchun bir hil nomlardan foydalanish imkoniyatidir. Polimorfizm konsepsiyasida ob'ektga nisbatan metod qo'llanganida aynan ob'ektning klassiga mos keluvchi metoddan foydalanishni ta'minlaydi.

Object Pascal dasturlash tili ob'ektga yo'naltirilgan dasturlashga hos bo'lgan barcha vositalarni bera oladi: strukturalilik, modullik, katta abstraktilik. Bu xarakteristikalarining hammasi xizmat ko'rsatish uchun juda ham sodda, osongina boshqa vaziyatlarga moslashtirila oladigan, kuchliroq strukturaga ega bo'lgan kodlarda o'z aksini topadi.

Ob'ektga yo'naltirilgan dasturlash uzoq yillar davomida an'anaviy, ya'ni standart hisoblangan dasturlashga nisbatan hos bo'lgan tasavvurlarni bir chetga qo'yishni talab qiladi. Natijada ob'ektga yo'naltirilgan dasturlash juda ham sodda, ko'rgazmaliligi yuqori bo'lib, dasturiy ta'minot yaratishdagi ko'plab muammolarni hal qilishning juda ajoyib vositasiga aylanadi.

Yuqorida aytilgan fikrlarni hisobga olsak, ob'ektga yo'naltirilgan dasturlash texnologiyalari quyidagicha masalalarni hal qilishi mumkin:

- an'anaviy dasturlash tillarida mavjud bo'lgan kamchiliklarni bartaraf qilish;
- an'anaviy dasturlash tillari yordamida yechib bo'lmaydigan yoki juda katta qiyinchiliklar bilan yechilishi mumkin bo'lgan masalalarni hal qilish;

- qayta ishlash mumkin bo'lgan ma'lumotlar va ularning tiplari doirasi an'anaviy dasturlash tillariga nisbatan ancha keng;
- foydalanuvchilar uchun qulay bo'lgan muloqot interfeysini yaratish;
- kiritilayotgan va chiqarilayotgan turli tipdagi ma'lumotlarni nazorat qilish; □ yangi tipdagi ma'lumotlar, klasslar va modullarni osongina tashkil etish va ma'lumotlarni nazorat qilish;
- multimedia va animatsion vositalaridan foydalanib, turli darajadagi tovushli va xarakatli effektlarni hosil qilish va qayta ishlash;
- ma'lumotlar bazasi va undagi ma'lumotlar ustida amallarni bajarish, SQL so'rovnomalari yordamida ma'lumotlarni qidirib topish kabi masalalar juda osonlik bilan hal qilish;
- OLE konteyneri yordamida WINDOWS muhiti uchun mo'ljallangan ilovalardagi ob'ektlar bilan ishlash;
- foydalanuvchilar uchun yaratilgan dasturiy ta'minotdan foydalanish uchun yordamchi ma'lumotnomalar tizimini yaratish;
- dastur matnini tashkil qilishda yuzaga kelishi mumkin bo'lgan xatoliklar bilan ishlash masalasini hal qilish va x.k.

Ko'rinib turibdiki, yechilayotgan masalalarni an'anaviy dasturlash tillari yordamida yechishda yuzaga kelishi mumkin bo'lgan kattakon bo'shliqni ob'ektga yo'naltirilgan dasturlash texnologiyalari to'ldiradi hamda zamonaviy dasturlashning ko'plab talablariga javob beradi. Shunday ekan, Oliy o'quv yurtlarida an'anaviy dasturlash tillaridan voz kechib, ob'ektga yo'naltirilgan dasturlash texnologiyalarini o'qitishga o'tish tavsiya qilinadi. [15;18;23]

II-BOB. DASTURLASH TILLARINING IMKONIYATLARI

2.1. Paskal dasturlash tili va imkoniyatlari

Paskal dasturlash tilining keng tarqalishi va qollanilishiga asosiy sabab dasturning soddaligi va undan foydalanishning qulayligidir. Dastavval Paskal tili universitetlarda qoʻllanilgan boʻlib, keyinchalik uning turli rusumli kompyuterlar uchun versiyalari ishlab chiqildi.

1981- yilda Paskal tilining xalqaro standarti taklif etildi. Paskal tilining Borland firmasi tomonidan ishlab chiqilgan 7.0 hozirgi davrda keng foydalaniladi. U foydalanuvchilar uchun juda qulay tizim — dasturlashning integrallashgan muhitiga ega.

Integrallashgan muhit-dasturlashga yordamlashuvchi dastur boʻlib, u quyidagi asosiy vazifalarni bajarishi lozim:

- avvalambor, u dastur matnini kiritish imkonini berishi;
- vaqti-vaqti bilan kiritilayotgan dastur matnini tashqi xotirada saqlab turishi;
- dasturni ishga tushirish uchun translyatorga ega boʻlishi;
- sintaktik xatoliklarni aniqlash vositasiga ega boʻlishi kerak.

Bugungi kunda Paskal dasturlash tilining bir nechta versiyalari mavjud boʻlib, bulardan eng koʻp ishlatilib kelayotgan versiyalari quyidagilar:

1.Turbo Pascal.

2. PascalABC.

Turbo Pascal dasturlash tili MS DOS operatsion dasturida ishlaydi. PascalABC dasturlash tili Windows muhitlarida ishlaydi. Hozirgi paytda umumtaʼlim maktablarida Turbo Pascal 7.0 versiyasidan foydalanib dasturlar oʻrgatilib kelinmoqda. Savol tugʻiladi bu dasturlash tillarida masalalarni dastur kodini kiritganimizda, dastur kodini farqi bormi?

Biror bir masalani dasturni tuzganimizda ikkala tilda ham dastur kodi bir xil.

Ammo grafik imkoniyatiga kelganda farqlanadi. Masalan:

Turbo Paskalda	PascalABC
<i>uses graph; var gd,gm:integer; begin gd:=detect;initgraph(gd,gm, " "); circle(320,240,100); readln; closegraph; end.</i>	<i>uses GraphABC; begin circle(320,240,100); readln; end.</i>

Hozirgi kunda barcha kompyuterdan Windows operatsion dasturi tarkibida ishlaydi. Shuning uchun foydalanuvchi Paskal dasturlash tilida masalalarni dasturini tuzganimizda Windows muhitidag ishlovchi PascalABC dasturlash tilidan foydalansak maqsadga muvofiq bo'lar edi. [8;24]

Paskal tilida ma'lumotlarning toifalari ikki xil bo'ladi oddiy skalyar va murakkab. Skalyar toifa o'z navbatida o'zgaruvchi va standart toifalarga bo'linadi. O'zgaruvchi toifaga qayd qilingan va chegaralangan toifalar kirsa, standart toifaga butun - INTEGER, haqiqiy - REAL, mantiqiy - BOOLEAN, xarakter yoki simvol - CHAR toifalar kiradi. Bundan tashqari Turbo Paskalda qator - STRING toifasi kiritilgan. Murakkab toifalarga esa ma'lumotlarning muntazam to'plam, aralash (yozuvlar), murojaat va faylli toifalari kiradi. Ma'lumotlarning standart toifalari. Standart turdagi ma'lumotlar dasturda maxsus xizmatchi so'zlar yordamida tavsiflanadi: Integer - Butun toifadagi ma'lumotlar faqat butun sonlarni qabul qiladi. Ular Paskal tilida quyidagicha ifodalanadi: -7, 10, 89, -35, 0 va h.k. Real – haqiqiy toifadagi ma'lumotlar haqiqiy sonlarni qabul qiladi va 2 xil ko'rinishda bo'ladi: a) qo'zg'almas nuqtali haqiqiy sonlar (3.56, 0.88,-150.45) b) qo'zg'aluvchan nuqtali haqiqiy sonlar (0.546q546E-03, 96.78*102q96.78E02 va h.k.) Char – belgi (simvol) toifadagi ma'lumotlar qiymat sifatida

ASCII kodlar jadvalidagi belgilarning bittasini qabul qiladi. Bu ma'lumotlar apostrof ichiga olib yoziladi. Masalan ('F', 'A', 'R', '5', '8' va h.k.) Boolean – mantiqiy toifadagi

ma'lumotlar. Bu nom ingliz matematigi Jorj Bul sharafiga shunday deb ataladi va shu turdagi o'zgaruvchilar faqat 2 ta qiymat qabul qiladi True yoki False. [12;21]

2.2. Delphi dasturlash tili va imkoniyatlari

Delphi – Windows muhitida ishlaydigan dastur tuzish uchun qulay bo'lgan vosita bo'lib, kompyuterda dastur yaratish ishlarini avtomatlashtiradi, xatoliklarni kamaytiradi va dastur tuzuvchi mehnatini yengillashtiradi.

Delphi muhiti zamonaviy 32 razryadli grafikali operatsion tizimlar (OT) Windows 95, Windows 98, Windows NT, Windows 2000 (ya'ni windows32 deb yuritiladi) WindowXP> Windows 7, Windows 8 boshqaruvida ishlovchi dasturlar tuzish uchun mo'ljallangan.

Oldingi MS-DOSda dasturlashga nisbatan Windows tizimida dasturlash ancha qiyinroqdir. Microsoft va Borland korporatsiyalari Windows 3.1 paydo bo'lgandan boshlaboq mos vositalar yarata boshladi. Masalan, 1991 yil Turbo Pascal for Windows, 1992 yil bu dasturlash tizimining yangilangan versiyasini Borland Pascal with Object 7.0 ni chiqardi. Bu dastlabki maxsuslashtirilgan uskunalar Windows ning anchagina bilimlarini talab qilar edi. 1993 yilga kelib Microsoft firmasi birinchi vizual dasturlash muhiti - Visual Basicni ishlab chiqdi hamda Windows uchun dasturlashtirish bir muncha osonlashdi. Bunga javoban Borland 1995 yilda Delphi dasturlash tilining birinchi versiyasi yaratildi. [4]

Bu dasturlash tili ob'ekt - orientirlashtirilgan tillar majmuiga kirib, 1995 yildan hozirgacha bir qancha avlodlari yaratilgan va ularga quyidagilarni aytib o'tish mumkin:

Delphi 1.0, Delphi 2.0, Delphi 3.0, Delphi 4.0, Delphi 5.0, Delphi 6.0, Delphi 7.0, Delphi 8.0, Delphi 9.0, Delphi 10.0.

Bu til o'zining keng qamrovli imkoniyatlariga egaligi bilan birga, boshqa dasturlash tillaridan o'zining ba'zi bir xususiyatlari bilan ajralib turadi. Borland Delphining paydo bo'lishi dasturlashni rivojlantirish tarixida yorqin ko'rinish bo'ldi. Delphi dasturlash tilining dunyoga kelishiga quyidagi tendensiyalar sabab bo'ldi:

➤ Windows uchun dasturlash va komponentlar texnologiyasi;

- masalalarni yechish uchun ob'ektga yo'naltirilgan usul;
- komponentlar texnologiyasiga asoslangan ilovalarni tez yaratishning vizual muhitlari;
- interpretatsiyadan emas, kompilyatsiyadan foydalanish. Bu shundan iboratki, interpretator bilan ishlashga qaraganda kompilyator bilan ishlash tezligi o'n marta ustunlikka ega bo'ladi;
- universal usullar yordamida ma'lumotlar bazasi bilan ishlash imkoniyatlarining mavjudligi, masalan, lokal va shu bilan bir qatorda server ma'lumotlari faylidan mijoz - server arxitekturasiga yoki ko'p bosqichli sxemasiga o'tishni ta'minlash. Borland Delphi yuqorida bayon etilgan tendensiyalarni joriy etish maqsadida yaratilgan. Ammo, uning eng asosiy elementi Object Pascal tili bo'lib hisoblanadi.

Delphi – bu bir qancha muhim texnologiyalar yig'indisidir, ya'ni:

- mashina kodiga o'tkazuvchi yuqori unumdorlikka ega bo'lgan kompilyator mavjudligi;
- komponentlar ob'ekt-orientirlashtirilgan modelga egaligi;
- dasturlarni vizual tarzda yaratish (tuzish) imkoniyatining mavjudligi;
- ma'lumotlar bazasini yaratish (tuzish) imkoniyatlarining yetarliligi.

Dastur yuqori unumdorlikka va tezlikka ega bo'lgan kompilyatorga ega. Har bir yangi versiyasi yaratilganda dasturlash tiliga qandaydir qulaylik yoki imkoniyatlilik kiritildi. Delphi dasturlash tili vizual instrumentlar yig'indisiga ega bo'lib, u esa dasturlash jarayonini yanada osonlashtiradi. Dasturchi dastur tuzish jarayonida komponentlar palitrasidagi tayyor tugmachalarni (ob'ektlarni) dasturga joylashtirishi mumkin. Mahsulotning standart turi ikkita bazaviy sinflardan tashkil topgan. Dasturlash tili qobig'i korporativ ma'lumotlar bazasiga ulanishni qo'llab quvvatlovchi, dastur tuzishni tezlashtiruvchi RAD (Rapid Application Development) vizual asboblarning interfeysiga ega. VCL – visual komponentlar kutubxonasi foydalanuvchining standart ob'ektlari, ma'lumotlarni boshqarish ob'ektlari, grafik ob'ektlar, multimedia ob'ektlari, muloqot ob'ektlari, fayllarni boshqarish ob'ektlari, DDE va OLE larni boshqarish ob'ektlaridan tarkib topgandir. Ma'lumotlar bazasi ob'ektlari SQL ma'lumotlar bazasini boshqarish

tizimi asosida Borland Database Engineni to'liq o'z ichiga oladi. Shuningdek, dasturlash tili tarkibiga Borland SQLLink ham kiradi. Bu esa Oracle, Sybase, Informix va Interbase ma'lumotlar bazasini boshqarish tizimlariga ruxsatni amalga oshiradi. Mahsulotni ikki xil varianti mavjuddir:

1. «Mijoz - Server» (Delphi Client-Server).

2. Delphi for Windows.

«Mijoz - Server» varianti quyidagi imkoniyatlarga ega:

- SQLLinks: Oracle, Sybase, Informix va InterBase larga ruxsat beruvchi (dostup) maxsus drayverlarning mavjudligi;

- InterBase lokal serveri: lokal tarmoqqa ulanmagan kompyuterlar uchun ma'lumotlar bazasini boshqarish tizimlari dasturlarini tuzish;

- Report Smith Client/Server Edition: SQL – server uchun hisobotlar generatorining mavjudligi;

- Visual Query Builder – SQL so'rovlarini tuzish modullarini mavjudligi.

Delphi for Windows varianti personal kompyuterlar uchun dasturlar tuzish hamda dBase va Paradox ma'lumotlar bazasini boshqarish tizimlarida dasturlar tuzish imkoniyatlariga ega. Shuningdek, shaxsiy DLL komponentlarini tuzish imkoniyati ham mavjuddir.

Delphi dasturlash tili bir necha muhim ahamiyatga ega bo'lgan texnologiyalar kombinatsiyasini o'zida mujassam etgan:

- yuqori darajali mashinali kodda tuzilgan kompilyator;

- ob'ektga yo'naltirilgan komponentalar modellari; ➤ dastur ilovalarini visual tuzish;

- ma'lumotlar bazasini tuzish uchun yuqori masshtabli vosita. [4-5;20]

2.3. C++ dasturlash tili va imkoniyatlari

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 hossalarni kiritgan va B da UNIX operatsion tizimining birinchi versiyalarini

yozgan. BCPL ham, B ham tipsiz til bo'lgan. Ya'ni o'garuvchilarning ma'lum bir tipi bo'lmagan - har bir o'zgaruvchi kompyuter xotirasida faqat bir bayt joy egallagan. O'zgaruvchini qanday sifatda ishlatish esa, yani butun son-mi, kasrli son-mi yoki harfdek-mi, 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 arxitekturasiga bog'langan tildir. Lekin yaxshi 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'zgarishsiz juda ko'p kompyuter platformalarida ishlaydi. [9-11]

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'ektli dasturlash g'oyasi ilgari surildi. Huddi 1970 - yillar boshida strukturali dasturlash kabi, programmalarni hayotdagi jismlarni modellashtiruvchi ob'ektlar 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 gibrid tildir. Unda C ga o'xshab 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 kutubxonasiga ega. Yani C++ da dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ ni o'zini o'rganish, ikkinchisi esa C++ ning standart kutubxonasidagi tayyor ob'ekt funksiyalarni qo'llashni o'rganishdir. [10;15;19]

2.4. Dasturlash tillarining qiyosiy tahlili

Chiziqli dasturlar:

1-masala. Berilgan ikkin sonni yig'indisini hisoblash dasturi.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
<pre>Var a,b,s:real; Begin Write('a='); read(a); Write('b='); read(b); s:=a+b; Write('s=',s);</pre>	<pre>{ float a,b,s; cout <<"a="; cin >> a; cout <<"b="; cin >> b; s:=a+b; { cout<<"s="<<s<<endl; } getch(); }</pre>

O'tish va tarmoqlanish operatori:

2-masala. Uchta son berilgan. Ular ichida manfiy sonlarning kubini hisoblovchi dastur.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
<pre>Var a,b,c:real; Begin Write('a='); read(a); Write('b='); read(b); Write('c='); read(c); If a<0 then write('a=', a*a*a); If b<0 then write('b=', b*b*b); If c<0 then write('c=', c*c*c); end.</pre>	<pre>{ float a,b,c; cout <<"a="; cin >> a; cout <<"b="; cin >> b; cout <<"c="; cin >> c; if (a<0) { cout<<"a="<<a*a*a<<endl; } if (b<0) { cout<<"b="<<b*b*b<<endl; } if (c<0)</pre>

	<pre> { cout<<"c="<<c*c*c<<endl; } getch(); } </pre>
--	--

Tarmoqlanuvchi strukturali dasturlar tuzish:

3-masala. Berilgan butun n son musbat va 5 ga karrali bo'lsa, shu sonning kvadrat ildizini, aks holda kvadratini hisoblash dasturi.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
<pre> Var n:integer; Begin Write('n='); read(n); If (n>0) and (n mod 5=0) then write('Ildizi=', sqrt(n)) else write('n*n=', n*n); end. </pre>	<pre> { int n; cout <<"n="; cin >> n; if ((n>0) and (n%5==0)) { cout<<"ildizi="<<sqrt(n)<<endl; } else { cout<<"n*n="<<n*n<<endl; } getch(); } </pre>

Parametrli takrorlash operatori:

4-masala. $S = \frac{7}{11} + \frac{17}{21} + \frac{27}{31} + \dots + \frac{2007}{2011}$ yig'indini hisoblash dasturi.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
<pre> Var a,b,c:real; Begin Write('a='); read(a); Write('b='); read(b); Write('c='); read(c); </pre>	<pre> { int i; float s; for(i=7; i<=17; i++) { if (i % 10==7) </pre>

<p><i>If $(a+b>c)$ and $(b+c>a)$ and $(c+a>b)$ then write('Bajariladi') else write('Bajarilmaydi'); end.</i></p>	<pre><i>{ s+=i/(i+4); cout<<"s="<<s<<endl; } } getch(); }</i></pre>
--	---

Shart bo'yicha takrorlash operatori:

5-masala. $y = x \sin x$ funksiyaning qiymatlarini $[-\pi; \pi]$ oraliqda 0.3 qadam bilan hisoblash dasturi.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
<pre><i>var x,y:real; Begin x:=-pi; while x<= pi do begin y:=x*sin(x); x:=x+0.3; write('y=', y); end; end.</i></pre>	<pre><i>{ float y,x=-pi; do { y=x*sin(x); x=x+0.3; { cout<<"y="<<y<<endl; } } while (x<=pi); getch(); }</i></pre>

Bir o'lchovli massivlar:

6-masala. $A[N]$ massivdagi manfiy elementlar sonini aniqlovchi dastur.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
--	----------------------------

<pre> var n,i,k:integer; a:array[1..100] of integer; begin write('n='); read(n); for i:=1 to n do begin write('a['i,']= '); readln(a[i]); end; for i:=1 to n do if a[i]<0 then begin k:=k+1; end; writeln('manfiy elementlari soni k=',k); end. </pre>	<pre> { int n,a[100],i,j; cout<<"Elementlar sonini kiriting:\nn="; cin>>n; cout<<"Massiv elementlarini kiriting:\n"; for(i=1;i<=n; i++) { cout<<"a["<<i<<"]=""; cin>>a[i];} cout<<"Massivning umimiy ko'rinishi:\n"; for(i=1;i<=n; i++) cout<<a[i]<<" "; j=0; for(i=1;i<n; i++) </pre>
	<pre> if(a[i]<0) j++; cout<<"manfiy sonlar soni:"<<j<<"ta"<<endl; getch(); } </pre>

Ikki o'lchovli massivlar:

7-masala. $A[N, N]$ massivning eng katta elementi chiqarish dasturi.

Paskal va Delphi dasturlash tillari	C++ dasturlash tili
--	----------------------------

<pre> var n, i, j:integer; max:real; a:array[1..20,1..20] of real; begin write('n='); read(n); for i:=1 to n do for j:=1 to n do begin write('a['i','j']='); read(a[i,j]); end; min:=a[i,j]; for i:=1 to n do for j:=1 to n do if a[i,j]>max then max:=a[i,j] ; write('max=',max); end. </pre>	<pre> int a[20][20], i, j, n, max, p; cout<<"Massiv o'lchamlar sonini kiriting:\nn="; cin>>n; for(i=1;i<=n; i++) for(j=1;j<=n; j++) { cout<<"a["<<i<<","<<j<<"]="; cin>>a[i][j]; } max=a[1][1]; cout<<"Qator: "; for(i=1;i<=n; i++) for(j=1;j<=n; j++) cout<<a[i][j]<<",";cout<<endl; for(i=1;i<=n; i++) for(j=1;j<=n; j++) if(max<=a[i][j]) { max=a[i][j]; } cout<<"\nmax="<<max<<endl; getch(); } </pre>
--	---

Yuqoridagi ma'lumotlar shuni ko'rsatadiki, bu tillar bir xil vazifani bajarishda boshqa-boshqa sintaktik qoidalar asosida yozilgan. Pascal va Delphi tillari deyarli bir xil ko'rinishda, C++ tili esa biroz boshqacha ko'rinishga ega. [18]

III-BOB. DASTURLASH TEXNOLOGIYASINI O‘QITISH USLUBIYOTI

3.1. Umumta’lim maktablarida Pascal dasturlash tilini o‘rgatishning interaktiv metodlari.

Umumiy o‘rta ta’limda «Informatika va hisoblash texnikasi asoslari» kursining maqsadi – o‘quvchilarga axborotlarni qayta ishlash texnologiyalari va ulardan foydalanish jarayonlari haqida puxta bilim berishni ta’minlashdan, ularga o‘quv hamda keyingi ish faoliyatlarida kompyuterlardan oqilona foydalanish ko‘nikma va malakalarini berishdan, shular asosida yangi axborot texnologiyalarining mamlakatimiz taraqqiyotiga qo‘shadigan hissasi to‘g‘risida tasavvur hosil qilishdan iborat.

Umumiy o‘rta ta’limda «Informatika va hisoblash texnikasi asoslari» ta’limining asosiy vazifalaridan biri o‘quvchilarga kompyuterda masalalar yechish texnologiyalari va uning asosiy bosqichlari haqida bilim berishni ifodalaydi. 2006 o‘quv yilidan boshlab Paskal dasturlash tili darslikka kiritildi. Paskal dasturlash tili dasturlashni o‘rgatish maqsadida ishlab chiqilgan bo‘lsada, yuqori malakali dasturchilar orasida keng tarqalgan.

Paskal dasturlash tili umumta’lim maktablarining 9-sinf darsligining ikkinchi bobida o‘qitiladi. Bunda jami 38 dasrdan iborat. Savol tug‘iladi umumta’lim maktab o‘quvchilariga qanday tartibda o‘qitilsa samarali bo‘ladi? **Avvalom bor biz darslarni zamonaviy ta’lim texnologiyalaridan, interaktiv metodlardan hamda multimediali darslar yordamida tashkil etsak samaraliroq hisoblanadi.** [12;18]

Shu bois biz paskal dasturlash tili mavzularni o‘qitish bo‘yicha quyidagi zamonaviy ta’lim texnologiyalar va darsni olib borish texnologiyasi bo‘yicha quyidagilarni tavsiya etamiz:

Mavzular	Foydalaniladigan metodlar
Paskal tili haqida ma’lumot	Bu darsimizda “Klaster” texnologiyasidan foydalanildi
Paskal tilida algoritmi tizim ko‘rinishida ifodalash	Bu darsimizda “Charxpalak” texnologiyasidan foydalanildi
O‘zgaruvchilar bo‘limi	Bu darsimizda “Qora quti” texnologiyasidan foydalanildi
O‘zlashtirish operatori	Bu darsimizda “Qora quti” texnologiyasidan foydalanildi.

Ma'lumotlarni kiritish va chiqarish	Bu darsimizda "O'yin" texnologiyasidan foydalanildi
Shartli o'tish operatori	Bu darsimizda "Charhpalak" texnologiyasidan foydalanildi
Tanlash operatori	Bu darsimizda "Katta aylana" texnologiyasidan foydalanildi
Sikl operatorlari	Bu darsimizda "Aqliy hujum" texnologiyasidan foydalanildi
Massivlar bilan ishlash	Bu darsimizda "Baliq skeleti" texnologiyasidan foydalanildi

Paskal dasturlash tilida o'tish va tarmoqlanish operatori darsini olib borish texnologiyasi

Dasturlash tillarini o'rgatish va o'quvchilarda amaliy bilim va ko'nikmalar hosil qilish uchun yangicha interaktiv metodlardan qo'llash maqsadga muvofiq. Shunday interaktiv metodlardan biri – bu **"Assesment"** metodidir. Ushbu metod o'quv mashg'ulotlarining barcha turlarida o'tilgan mavzuni o'zlashtirganlik darajasini baholash takrorlash va mustahkamlash uchun mo'ljallangan.[24]

Mavzu: O'tish va tarmoqlanish operatori

Paskal dasturlash tilida tarmoqlanish **If** operatori bilan amalga oshiriladi. **If** operatori- bu shartli operator hisoblanadi. Paskal tilida shart - bu mantiqiy turdagi ifoda bo'lib, u faqat «chin»(True) yoki «yolg'on »(False) qiymatni qabul qiladi, quyidagi mantiqiy belgilar ishlatiladi: katta ">",kichik "<"0, kichik yoki teng "<=", katta yoki teng ">=", teng emas "<>", teng "=". Quyidagilar munosabat amallari yoki taqqoslash amallari ham deyiladi. Quyidagi mantiqiy amallar ishlatiladi:

- NOT-«inkor»;
- AND-«mantiqiy va»;
- OR-«mantiqiy yoki».

Bu mantiqiy amallarning bajarilish natijalari quyidagicha:

Op1	Op2	Op1 AND Op2	Op1 OR Op2	NOT Op1
-----	-----	----------------	---------------	------------

False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Masalan: $(5 < 6)$ AND $(6 < 50)$ - mantiqiy ifoda rost (True),

$(20 > 0)$ OR $(20 < 0.5)$ - mantiqiy ifoda rost (True),

$(10 < 8)$ AND $(10 < 15)$ - mantiqiy ifoda yolg'on (False), NOT $(100 > 3)$ - mantiqiy ifoda yolg'on (False).

O'zlashtirish mantiqiy ifodalarni biror bir mantiqiy o'zgaruvchiga yuborish ham mumkin. Masalan: $F := (A < B) \text{ AND } (A < C)$;

Bu yerda, agar ikkala shart bajarilgandagina F mantiqiy o'zgaruvchi "chin"

(True) qiymatni qabul qiladi. Aks holda "yolg'on" (False) qiymatni qabul qiladi.

Delphi dasturlash tilida shartli o'tish operatorining ikki xil ko'rinishi mavjud:

1. To'liq ko'rinish operatori:

If <shart> then begin

<shart rost bo'lganda bajariladigan operatorlar> end

else begin

<shart yolg'on bo'lganda bajariladigan

operatorlar>

end;

2. Qisqa ko'rinish operatori:

If <shart> then begin

<shart rost bo'lganda bajariladigan operatorlar> end;

Bu yerda If -agar; then -u holda; else -aks holda ma'nosini bildiruvchi xizmatchi (kalit) so'zlar. Birinchi ko'rinishdagi shartli operatorlarda, agar shart bajarilsa birinchi begin va end ichidagi operatorlar ketma-ket bajariladi, aks holda ikkinchi begin va end ichidagi operatorlar ketma-ket bajariladi. Ikkinchi ko'rinishdagi shartli operator quyidagicha ishlaydi: agar berilgan shart bajarilsa begin va end ichidagi operatorlar ketma-ket

bajariladi, aks holda ular bajarilmaydi. Agar bajariluvchi operatorlar soni bitta bo'lsa, begin va end so'zlarini yozish shart emas.

Misollar: 1) **If** A>0 **then begin** C: =1; B:=C+1; **end; else begin** C: =0; B: =4; **end;**
2) **If** D=A **then** D: =A **else** A: =D;

Bir shartli o'tish operatori ichida boshqa ichki shartli operatorlar joylashishi ham mumkin, *Masalan:* **If** b1 **then** a1 **else If** b2 **then** a2 **else** a3;

Misollar. A: =0.5; B: =-1.7; **If** A<B **then** A: =B **else** B: =A;

Javob: 0.5<-1.7 yolg'on bo'lganligi sababli B:=A operator bajariladi, va bunda A=0,5 va B=0,5 ekanligi kelib chiqadi.

A: =0.1; B: =0.1; C: =0.5; D: =0;

If (A<B) **or** (A>C) **then** D: =B+C **else**

If B=A **then begin** D: =C; C: =A; **end;**

Javob: (0.1<0.1) yoki (0.1>0.5) bu mantiqiy ifoda yolg'on bo'lganligi sababli B=A shart tekshiriladi. Bu shart chin bo'lganligi sabab D=0.5 ga, S=0.1 qiymatlarga teng ekanligi kelib chiqadi. [8;12]

Ushbu mavzu bo'yicha "ASSESSMENT" metodini qo'llab topshiriq beriladi.

1.TEST.

1) Agar A=rost B=yolg'on qiymatda bo'lsa ushbu ifodaning qiymati qaysi qatorda to'g'ri ko'rsatilgan? A or B and(not B and A)

a) yolg'on b) Rost

c) aniq yechim mavjud emas

2.Muammoli vaziyat.

Ertaga tomorqangizga urug' ekmoqchisiz, lekin ba'zi internet manbalari ertaga yomg'ir yog'ishini aytmoqda, ba'zilar yo'q shu ma'lumotlarning rostlik jadvalini tuzgan holda ertaga urug' ekiladimi yoki yo'qligini ifodalang.

3.Simpton.

Shart operatoriga ta'rif bering. If operatorining ishlash prinsipini ilmiy asoslang.

4.Amaliy ko'nikma.

If operatori bo'yicha 2 ta dastur tuzing

Ushbu qo'llanilgan imteraktiv metod orqali o'quvchilar o'tilgan mavzu yuzasidan bilimlarini yanada mustahkamlaydi. Bundan tashqari savollar va topshiriqlarning turlichaligi o'quvchining ham nazariy va ham amaliy bilimlarini oshiradi. Bu metodni qo'llashda savollar va topshiriqlarni to'g'ri tanlash juda muhim. Tuzilgan savollar va topshiriqlar butun bir mavzuni qamrab olishi kerak. O'qituvchining **fantaziyasi** qanchalik yaxshi bo'lsa savollar va topshiriqlar saviyasi shuncha baland bo'ladi. [24]

3.2. Akademik litsey va umumta'lim maktablarida Delphi va C++ dasturlash tilini o'rgatishning interaktiv metodlari.

Hozirgi kunda akademik litsey va maktablarda Delphi va C++ dasturlash tillari o'qitilib kelinmoqda. Bu dasturlash tillarini qanaqangi tartibda o'qitilish kerak degan savol tug'iladi.

Albatta bu dasturlash tillarini ham interaktiv metodlardan, zamonaviy ta'lim texnologiyalaridan hamda multimediali darslar yordamida tashkil etilsa samaraliroq hisoblanadi.

Shu bois biz Delphi va C++ dasturlash tillarini mavzularni o'qitish bo'yicha quyidagi zamonaviy ta'lim texnologiyalar va darsni olib boorish texnologiyasi bo'yicha quyidagilarni tavsiya etamiz:

Delphi dasturlash tilini o'qitish bo'yicha ta'lim texnologiyalaridan foydalanish

Quyidagi mavzuni o'rgatishda interaktiv metodlardan bir nechtasini qo'llash mumkin.

Masalan: "modellashtirish", "Ijodiy ish" yoki "Keys stadi" metodlaridan birini qo'llash mumkin. Bunda "**Keys stadi**" metodi ko'proq samarali deb hisoblasak darsni olib borishni shaklini ko'ramiz:

Mavzu:Standard komponentalar palitrasi

O'quvchilarga berilgan vazifa: Standart komponentalardan foydalangan holda kichik dastur tayyorlash.

Keys topshiriq: 1) O'zingiz haqingizdagi kichik va ko'proq ma'lumotni ekranga chiqarish.

2) **Test** hosil qilgan holda unda savol va javoblarni ekranda aks ettirish.

Kerakli jihozlar: Shaxsiy kompyuter, o'quv qurollari.

Kerakli dasturiy ilova: Borland Delphi dasturi.

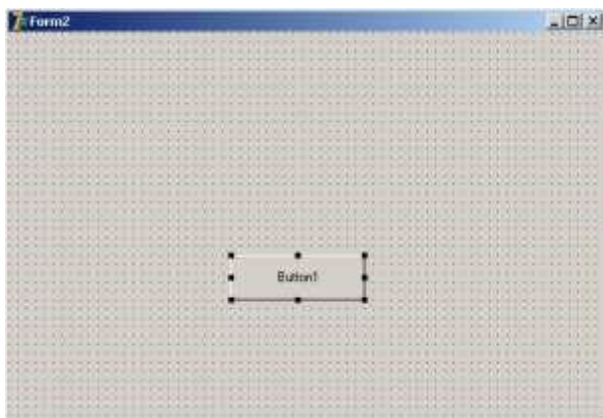
Vazifani bajarish bosqichlari:

- kerakli ma'lumotlar bilan tanishish;
- ma'lumotlar asosida dasturda amaliy shaklda tanishish;
- dasturni tuzish uchun algoritm tuzish va nazariy modellashtirish;
- berilgan vazifa bo'yicha dasturni tuzish;
- tayyor bo'lgan dastur to'g'ri ishlayotganligiga ishonch hosil qilish.

Nazariy ma'lumotlar va keys dasturiy kartasi:

1. Button tugmachasi.

Button tugmachasini bosilishi natijasida kutilishi lozim bo'lgan jarayonlar (masalan, hisoblashlar yoki bajarilishi lozim bo'lgan operatsiyalar) bajarilishga xizmat qiladi.



Button tugmachasidan foydalanish.

Button tugmachasini ekranga joylash uchun Delphi dasturlash tilining Standart komponentalar palitrasidan “Ok” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugmacha qo'yilishi lozim bo'lgan joy ajratiladi. Natijada “**Button1**” tugmachasi hosil qilinadi. Ikkinchi marta tanlansa, “**Button2**” va hokazo “**ButtonN**” ko'rinishda davom etadi.

Agar siz tugmachadagi “**Button1**” so'zini boshqa so'zga almashtirmoqchi bo'lsangiz, ob'ektlar inspektorining “**Caption**” xossasidan foydalanib o'zgartiriladi.



Button tugmachasidan foydalanish.

Tugmachadagi soʻzni kattartirish yoki kichiklashtirishga toʻgʻri kelsa, obʼektlar inspektorining Font-{Tfon} bandi tanlanadi va bu yerdan kerakli oʻlchamlarni tanlash mumkin.

Yuqorida paydo boʻlgan koʻrinishdan nafaqat **Button** tugmachasi ustida amallar bajaribgina qolmasdan, boshqa tugmalar ustida ham bajarish, shu bilan bir qatorda dasturlash maydonchasiga turli xil bezaklar berishda ham foydalanishimiz mumkin.

Demak, **Button** tugmachasi bosilishi natijasida kutilishi lozim boʻlgan jarayonlar bajarilishga xizmat qilar ekan. Biz bu tugmachaga doir bir nechta misollar koʻrib chiqamiz.

1-misol. Dasturdan chiqish buyrugʻini bajaruvchi dastur tuzing.

Standard komponentalar palitrasidan “**Button1**” tugmasini hosil qilamiz va sichqoncha tugmasini ikki marta bosgan holda “**Until1.pas**” maydoniga “**form1.close**” buyrugʻini kiritamiz.

```
{ $R *.dfm}

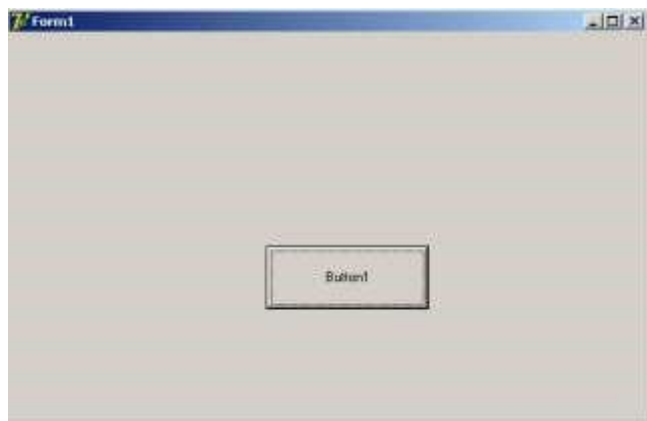
procedure TForm1.Button1Click(Sender: TObject);
begin
  form1.close;
end;

end.
```

Dasturlash oynasida “Button” tugmasinidan foydalanish.

Yuqoridagi topshiriq bajarilgandan so‘ng, bu dasturni birinchi marotaba

“**File→Save As**”, Ikkinchi marotaba “**File→Save project As**” ketma-ketlikni bajaramiz va dasturdan chiqib, “**project1**” faylni qayta yuklaymiz. Natijada quyidagi ko‘rinish paydo bo‘ladi.



Button tugmachasidan foydalanish.

Agar siz tugmacha ustiga sichqoncha chap tugmasini birmarta bossangiz hosil qilingan dasturdan chiqishga muyassar bo‘lasiz.

2. Label komponentasi.

Label komponentasi nafaqat ma’lumotlarni ekranga joylashtirish uchun xizmat qiladi, balki dastur natijalarini chiqarishda ham ishlatish mumkin. Buning uchun dasturda `Labeln.caption:='Dastur natijasi';` buyrug‘i berilishi lozim.

Label tugmachasini Forma oynasiga joylashtirish uchun Delphi dasturlash tilining Standart komponentalar palitrasidan “**A**” piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugmacha qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada “**Label1**” tugmachasi hosil qilinadi. Ikkinchi marta tanlansa “**Label2**” va hokazo “**LabelN**” ko‘rinishda davom etadi.

2-misol. Label tugmasi yordamida ma’lumotlarni ekranga joylashtirish va dastur natijasini chiqarish yo‘lini ko‘rib chiqamiz.

1. Standart komponentalar palitrasidan “**A**” piktogrammasi belgilaymiz va forma ustiga sichqoncha tugmachasini bosgan holda tugmacha qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada “**Label1**” komponentasi hosil bo‘ladi, ob’ektlar inspektorining

“Caption” bandidagi “Label1” so‘zini o‘chirib “Dastur tuzishni o‘rganamiz” jumlasini kiritamiz.

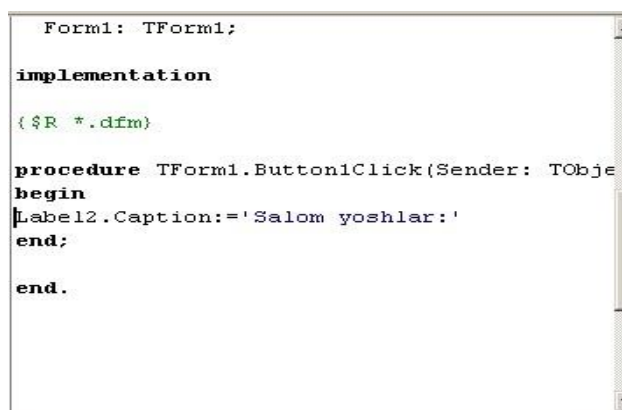
2. Ikkinchi marotaba xuddi shu yo‘lni tanlab, “Label2” komponentasini hosil qilamiz.

3. “Button1” tugmasini hosil qilib, “Caption” bandiga “Hisoblash” jumlasini kiritamiz.



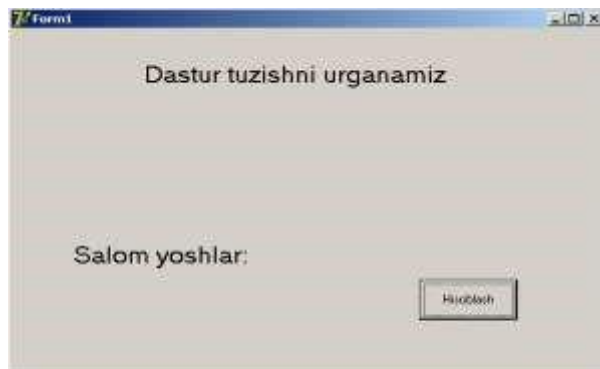
Label tugmachasidan foydalanish.

4. Button1(hisoblash) bandiga sichqoncha tugmachasi ikki marta bosilib dasturlash maydonchasiga “Label2.caption:= ‘Salom yoshlar:’” jumlasini kiritamiz.



Label tugmachasidan foydalanish

5. Yuqoridagi amallar ketma-ketligi kiritilgandan so‘ng “F9” tugmasini bosamiz. Natijada quyidagi ko‘rinish hosil bo‘ladi.



Label tugmachasidan foydalanish.

3. Edit (kiritish) qatori.

Edit kiritish qatori matnni bir qatordan kiritish va uni tahrirlash uchun ishlatilibgina qolmasdan balki ma'lumotlarni chiqarishda ham xizmat qiladi. Natijada **"Edit1"** matn kiritish maydoni hosil qilinadi. Matnni kiritish dastur ishchi holatiga o'tilganda bajariladi.

3-misol. "Edit" matn kiritish va chiqarish qatoridan foydalanishni ko'rib chiqamiz.

1. Standard komponentalar palitrasidan **"ab"** piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **"Edit1"** matn kiritilishi va ikkinchi marotaba **"Edit2"** chiqarilishi lozim bo'lgan joy ajratiladi.

2. Standard komponentalar palitrasidan **"Button1"** tugmasini hosil qilamiz.

3. Natijada hosil bo'lgan **"Button1"** tugmasi ustiga sichqoncha tugmachasini ikki marta bosib, dasturlash maydonchasini hosil qilamiz.

4. Demak, hosil bo'lgan maydonchaga quyidagi dasturni kiritamiz: **var m:integer; begin m:=strtoint(edit1.Text); edit2.text:=edit2.text+'Kiritilgan son='+inttostr(m); end; end.**

```
Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var m:integer;
begin
m:=strtoint(edit1.Text);
edit2.text:=edit2.text+'Kiritilgan son='+inttostr(m);
end;
end.
```

Edit tugmachasidan foydalanish.

5. Dastur bajarilish uchun **"F9"** tugmasini tanlaymiz.


6. Shundan so‘ng, hosil bo‘lgan katakchaga (**Edit1**) maydoniga 6 sonini kiritamiz. Natijada “**Edit2**” maydonida “**Kiritilgan son=6**” ko‘rinish paydo bo‘ladi.




Edit tugmachasidan foydalanish.

4.ScrollBar tugmachasidan foydalanish.

ScrollBar – formada siljitish vositasini yaratishda uchun xizmat qiladi.

Siljitish vositasini forma oynasiga joylashtirish uchun Delphining Standard komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash, lozim bo‘lgan joy ajratiladi. Natijada “**ScrollBar1**” siljitish tugmachasi hosil qilinadi. Siljitish dastur ishchi holatiga o‘tilganda bajariladi.

4-misol. “ScrollBar” tugmachasidan foydalanish.

1. Standard komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda “**ScrollBar1**” maydonini hosil qilamiz.

2. Standard komponentalar palitrasidan “**Label1**” tugmasini hosil qilamiz.

3. Ob’ektlar inspektorining “**Caption**” xossasidan “**Label1**” yozuvi o‘rniga “**0**” sonini kiritamiz.

4. “**RadioButton1**” tugmachasini ustiga sichqoncha tugmachasini ketma-ket ikki marta bosamiz va dasturlash maydonchasiga quyidagi dasturni kiritamiz:

Label1.Caption:=IntToStr(ScrollBar1.Position);


```

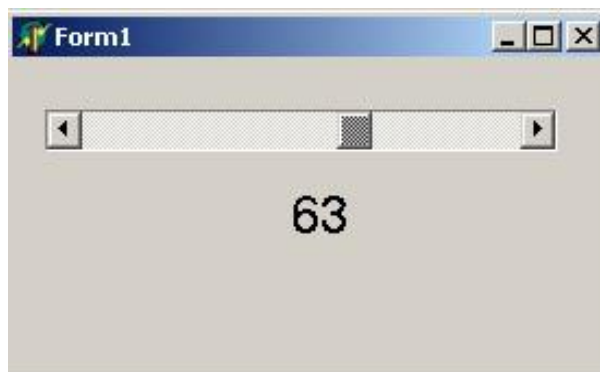
{$R *.dfm}
procedure TForm1.ScrollBar1Change(Sender:
begin
    Label1.Caption:=IntToStr(ScrollBar1.Position);
end;

end.

```

ScrollBar tugmachasidan foydalanish


5. Amallar ketma-ketligi bajarilishi uchun “F9” tugmasini tanlaymiz.




ScrollBar tugmachasidan foydalanish


7. RadioButton tugmachasidan foydalanish.

RadioButton – bir nechta variantdan birortasini tanlash imkoniyatini

yaratish uchun xizmat qiladi. Variantdan birortasini tanlash tugmachasini forma oynasiga joylashtirish uchun Delphining Standard komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash lozim bo‘lgan joy ajratiladi. Natijada “**RadioButton1**” tanlash tugmachasi hosil qilinadi. Tanlash dastur ishchi holatiga o‘tilganda bajariladi.

6-misol. “RadioButton” tugmachasidan foydalanish.

1. Standard komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda “**GroupBox1**” maydonini hosil qilamiz.

2. Standard komponentalar palitrasidan  piktogrammasi belgilanib “**GroupBox1**” forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda “**RadioButton1**” va ikkinchi marotaba “**RadioButton2**” tanlash lozim bo‘lgan joy ajratiladi.

3. Standard komponentalar palitrasidan “**Label1**” tugmasini hosil qilamiz.

4. Ob'ektlar inspektorining “**Caption**” xossasidan “**Label1**” yozuvi o‘rniga “**Dastur**” so‘zini kiritamiz.

5. “**RadioButton1**” tugmachasini ustiga sichqoncha tugmacahsini ketma-ket ikki marta bosamiz va dasturlash maydonchasiga quyidagi dasturni kiritamiz:

Label1.Enabled := False;

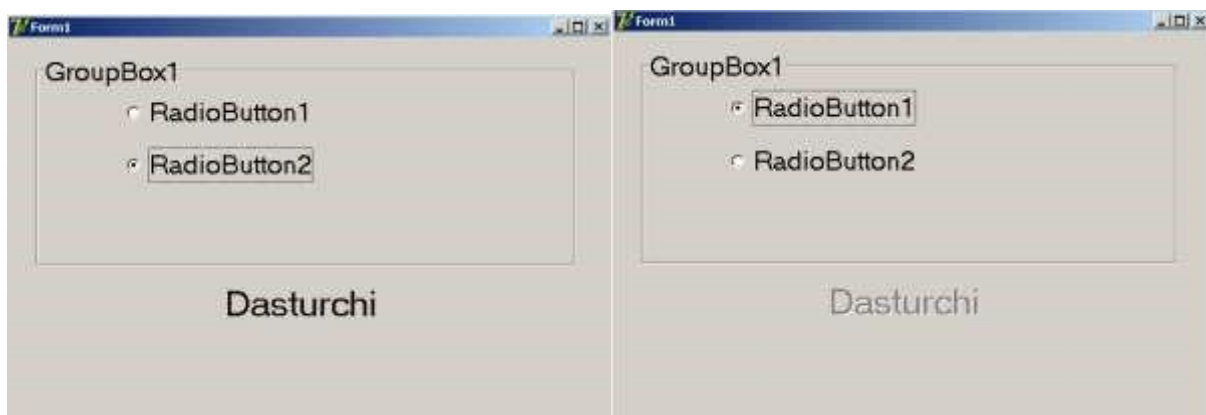
6. “**RadioButton2**” tugmachasini ustiga sichqoncha tugmacahsini ketma-ket ikki marta bosamiz va dasturlash maydonchasiga quyidagi dasturni kiritamiz:

Label1.Enabled := True;

```
{ $R *.dfm}

procedure TForm1.RadioButton2Click(Sender:
begin
Label1.Enabled := True;
end;
procedure TForm1.RadioButton1Click(Sender:
begin
Label1.Enabled := False;
end;
end.
```

3.2.15-rasm. Radio Button tugmachasidan foydalanish 7. Dastur bajarilishini ta’minlash uchun “**F9**” tugmasini bosamiz.



Radio Button tugmachasidan foydalanish

8. Agar siz “**RadioButton2**” tugmasini tanlasangiz dastur so‘zi faollashadi va aksincha “**RadioButton1**” tanlasangiz dastur xira holga keladi. [4-5;13]

“**Keys stadi**” interaktiv metodi orqali o‘tilgan ushbu mavzu o‘quvchilarda ham nazariy bilimlar egallash uchun balki amaliy bilim, ko‘nikmalar hosil bo‘lishida samarali metod desak mubolag’a bo‘lmaydi. Kerakli topshiriqlar, ularni bajarish bosqichlari aniq va lo‘nda bayon qilinganligi, o‘quvchilarning individual ishlashlariga va dars jarayonida ko‘proq izlanishlariga zamin yaratadi. Berilgan masalalarning hayotiy bo‘lishi esa o‘quvchilar olgan bilimlarini amaliyotga tadbiq qilishlaridagi bilimlarini va ularning ushbu sohaga qiziqishlarini yanada oshiradi. Masalan yuqoridagi

mavzuni o'rganayotgan o'quvchilar **Delphi** dasturlash tilining komponentalari bilan tanishibgina qolmay, unda bir nechta dasturlarni mustaqil tuzishlari o'quvchilar bilimlarining yanada oshishiga olgan bilimlarini mustahkamlashlariga zamin yaratadi. [17;24]

C++ dasturlash tilini o'qitish bo'yicha ta'lim texnologiyalaridan foydalanish

Yana bir ommabop dasturlash tillaridan biri bu C++ dasturlash tilidir. O'qituvchi qanday metod tanlashi haqida o'zi qaror qabul qiladi. Albatta bunda darsning turi, mavzuning murakkabligi, o'quvchilarning yosh xususiyatlari inobatga olinishi shart.

Quyida C++ dasturini o'rganishdagi mavzulardan birini o'qitishda interaktiv metodlardan biri **“Qalin va ingichka savollar”** metodidan foydalanishni ko'rib chiqamiz.

Mavzu: C++ dasturlash tilida o'zgarmlar, o'zgaruvchilar va standart funksiyalar.

C++ algoritmik tilining alifbosi quyidagilardan iborat:

- katta va kichik lotin harflari;
- 0 dan 9 gacha raqamlari;
- maxsus belgilar (+, -, *, /, =, >, <, {, }, [,], ') ni o'z ichiga oladi.

C++ tilida so'z deb bir nechta belgilar ketma – ketligi tushuniladi. Xizmatchi so'z deb C++ tilidagi standart nom tushuniladi. Bu nom maxsus ma'noni anglatadi va uni ma'lumotlarga berib bo'lmaydi. Masalan: int, float, for, while va hokazo.

C++ tilida ma'lumotlarning elementlari bo'lib o'zgaruvchilar, o'zgarmlar, izohlar xizmat qiladi. C++ dasturlash tilida dastur quyidagi tarkibda tashkil topadi:

Direktivalar – funksiyalar kutubxonasini chaqirish. Ular maxsus include katalogida joylashgan va .h kengaytmali fayllar bo'ladi. C++ tilida masalaning qo'yilishiga qarab kerakli kutubxonalar chaqiriladi. Bus esa dasturning xotirada egallaydigan joyini minimallashtiradi. Masalan, ma'lumotlarni kiritish-chiqarish proseduralari uchun: #include <stdio.h> tizimdan chaqirish #include “stdio.h” joriy katalogdan chaqirish. C++ dasturlash tili bilan ishlovchi eng sodda dasturlar Dev C++ va CodeBlocks dasturlaridir. Ularning tarkibida 300 dan ortiq kutubxonalar mavjud. Eng ko'p ishlatiladigan kutubxonalar quyidagilar: #include<iostream.h>,

#include <math.h>

#include <conio.h>

#include <graphics.h>

#include <memory.h> va boshqalar Makrolar (**#define**) – dastur bajarilishi davomida o'zgaruvchi ko'rsatilgan qiymatni qabul qilishi uchun (**const**). Unda makroning nomi va qiymati ko'rsatiladi. Macalan: **#define pi 3.1415 #define x 556 #define s[100] #define M x*x*x** **main ()** funksiyasi– asosiy degan ma'noni anglatadi. Bu funksiya “{“ belgisidan boshlanadi va dasturning asosini tashkil etuvchi o'zgaruvchilarning toifalari ko'rsatiladi. Dastur “}” belgisi bilan yakunlanishi shart. Agar dasturda qism dasturlardan foydalanilayotgan bo'lsa, ularning nomlari va haqiqiy parametrlari keltiriladi. So'ngra dasturning asosiy buyruqlari yoziladi. Agar buyruqlar murakkab bo'lsas, ular alohida “{ }” belgilari orasiga olingan bo'lishi kerak. C++ tilida dasturning asosi bo'lmish buyruqlar kichik harflar bilan yoziladi. Buyruqlar nuqta-verguk bilan (;) yakunlanadi. Buyruqlar bir qator qilib yozilishi ham mumkin. C++ dasturlash tilida dastur funksiya va funksiyalardan tashkil topadi. Agar dastur bir nechta funksiyalardan tashkil topgan bo'lsa, bir funksiyaning nomi **main** deb nomlanishi shart. Dastur aynan **main** funksiyasining birinchi operatoridan boshlab bajariladi. C++ tilidagi dastur ko'rinishini quyidagi misol yordamida keltirib o'tamiz.

```
#include <iostream.h> // sarlavha faylni qo'shish
int main () // bosh funksiya tavsifi
{ // blok boshlanishi
    cout << "Salom Dunyo! \n"; // satrni chop etish
    return 0; // funksiya qaytaradigan qiymat
} // blok tugashi
```

Dasturning 1-satrida **#include** direktivasi bo'lib, dastur kodiga oqimli o'qish/yozish funksiyalari va uning o'zgaruvchilari e'loni joylashgan **iostream.h** sarlavha faylini qo'shadi. Keyingi qatorlarda dasturning yagona, asosiy funksiyasi **main()** funksiyasi tavsifi keltirilgan. Shuni qayd etish kerakki, C++ dasturida albatta **main()** funksiyasi bo'lishi shart va dastur shu funksiyaning bajarish bilan o'z ishini boshlaydi. Dastur tanasida konsol rejimi (**Consol** – rejimi bu MS DOS oynasi ko'rinishiga o'xshash oyna bo'lib, unda foydalanuvchi dastur tuzuishda faqat dastur kodlari bilan ishlaydi. **Graphic interface** –

rejimida esa faqat tilning kodlari bilangina emas muhitning menyulari, komponentalari bilan ham ishlashi mumkin bo'ladi) da belgilar ketma-ketligini oqimga chiqarish amali qo'llanilgan. Ma'lumotlarni standart oqimga (ekranga) chiqarish uchun quyidagi format ishlatilgan:

```
cout << <ifoda>;
```

O'zgaruvchi. Xotiraning nomlangan qismi bolib, o'zida ma'lum bir toifadagi qiymatlarni saqlaydi. O'zgaruvchining nomi va qiymati bo'ladi. O'zgaruvchining nomi orqali qiymat saqlanayotgan xotira qismiga murojaat qilinadi. Programma ishlashi jarayonida o'zgaruvchining qiymatini o'zgartirish mumkin. Har qanday o'zgaruvchini ishlatishdan oldin, uni e'lon qilish lozim.

Quyida butun sonlardan foydalanish uchun b, haqiqiy sonlardan foydalanish uchun h.k. o'zgaruvchisi e'lon qilingan: `int b; float h;`

Operator. Tilning yakunlangan jumlasini hisoblanadi va ma'lumolar taxlilining tugallangan bosqichini ifodalaydi. Operatorlar nuqtali vergul “;” bilan ajratiladi. Ya'ni “;” operatorning tugallanganligini bildiradi. C++ da operatorlar programmada keltirilgan ketma - ketlikda bajariladi.

Identifikator. Programmist tomonidan programma elementlari (funksiya, o'zgaruvchilar, o'zgaruvchilar ...) uchun ixtiyoriy tanlangan nom.

Identifikator tanlaganda quyidagilarga ahamiyat berish kerak:

- Identifikator lotin harflaridan boshlanishi shart;
- Ikkinchi simvoldan boshlab raqamlardan foydalanish mumkin;
- C++ da katta kichik harflar farq qiladi. Ya'ni quyidagilarning har biri alohida identifikator hisoblanadi: KATTA, katta, KaTTa, kAttA, Katta, KattA, ...
- Probel C++ da so'zlarni ajratish uchun ishlatiladi. Shuning uchun identifikatorda probel dan foydalanib bo'lmaydi;
- Xizmatchi (int, float, for, while kabi) so'zlardan identifikator sifatida foydalanib bo'lmaydi.

O‘zgaruvchilarni e’lon qilish. Programmada ishlatilgan barcha o‘zgaruvchilarni qaysi toifaga tegishli ekanligini e’lon qilish kerak. Ma’lulotlarni e’lon qilishning umumiy ko‘rinishi quyidagicha:

toifa_nomi o‘zgaruvchi;

Agar bir nechta o‘zgaruvchi bir toifaga mansub bo‘lsa, ularni vergul bilan ajratib berish mumkin. [10-11;22]

Butun sonlarni ifodalash uchun int va haqiqiy sonlarni ifodalash uchun float xizmatchi so‘zlaridan foydalaniladi. Bu ma'ruzada shu 2 tasini bilish bizga kifoya qiladi. Keyingi mavzuda butun va haqiqiy sonlar haqida batafsil gaplashamiz.

int x,y; // butun toifadagi o‘zgaruvchilarni e’lon qilish float a,b,c; // haqiaiy toifadagi o‘zgaruvchilar e’lon qilish

Standart funksiyalar.

T/r	Matematik ko‘rinishi	C++ dasturida yozilishi
1.	$ x $	Abs(x)
2.	x^2	Pow(x,2)
3.	a^k	Pow(a,k)
4.	\sqrt{x}	Sqrt(x)
5.	$\sqrt[n]{x}$	Exp(x*log(x)/n)
6.	$\ln x$	Log(x)
7.	$\cos x$	Cos(x)
8.	$\tan x$	Sin(x)/cos(x)
9.	$\log_a x$	log(x)/log(a)
10.	2,3	2.3

1-misol. Quyidagi berilgan matematik masalani C++ dasturlash tilida yozing:

$$y = \frac{x^2}{2,3} - \sqrt[n]{x} + \sin x$$

Yechilishi: pow(x,2)/2.3 –exp(log(x)/n)+sin(x).

“Qalin va ingichka savollar” metodida savollar uch bosqichda bo‘lishi mumkin:
1-bosqich boshlang’ich bosqich(BB)

2-bosqich anglab yetish bosqichi(AYB)

3-bosqich mavzuni mustahkamlash bosqichi(MMB)

Masalan

BB

1. C++ tilida barcha raqamlarni ifodalash mumkinmi?
2. O'zgarmaslar qanday so'z yordamida e'lon qilinadi?
3. C++ da opertatorlar nima?

AYB

1. C++ tilida ma'lumotlarning elementlari?
2. **Int** va **float** operatorlarining farqi?
3. { } ushbu belgilar nima uchun xizmat qiladi?

MMB

1. C++ tilida **Direktivalar ning** vazifasi qanday?
2. a^k ushbu matematik ifoda C++ tilida qanday yoziladi?
3. C++ tilida bo'lish qoldig'ini olish qanday bajariladi?

Qalin va ingichka savollar metodini qo'llash orqali o'quvchilar bilimini sifatli oshirishga erishish mumkin. Savollarning uch bosqichda bo'lishi, soddalikdan murakkablikka bosqichma-bosqich o'tilishi, bu metodning yutuqli tomonidir. Savollarni bosqichlarga mos tuzish zarur. Shundagina bu metodning samarasi yanada oshadi. [24]

3.3 Dasturlash tillarini o'qitishda multimediy texnologiyalaridan foydalanish.

Ko'pgina mualliflar informatikani o'qitishda yangilarni ishlatishni strategik muhimligini ta'kidlaydilar. Ta'limning kompyuterizatsiyalash muammolariga bag'ishlangan maxsus konferensiyalar tashkil etilmoqda. Ta'limda multimedia vositalarini qo'llash eng muhimdir.

Multimedia - bu multimedia axborot texnologiya vositalari bo'lib (MIT) kompyuter va foydalanuvchi o'rtasida ma'lumot almashishning turli kanallarini bir vaqtda ishlatishga yo'l ochib beradi. Chunki multimedia o'qitish jarayonida ma'lumotlarning barcha kanallarni ishga soladi, xotiraning barcha turlarini maksimal darajada ishlatadi.

Kompyuter o'qitishiga bag'ishlangan mavzular bo'yicha maqolalar soni nashriyotda ortib bormoqda, ilmiy jamoalar axborot texnologiya va ta'lim jarayonini o'zaro ta'sirini ko'rib chiqishmoqda.

Kompyuterni o'qitish vositasi yoki interfaol ekspert sifatida ishlatishga bag'ishlangan tadqiqotlar olib borilmoqda. Ta'limda ko'p maktablar multimedia vositalari bilan jihozlangan kompyuterning oxirgi avlodlarini qo'llashmoqda. Ta'limda kompyuterdan foydalanish qimmatbaholigiga qaramay aslida ma'lumotlarning uzatilish sifati uning yangilanish va qo'shimchalar kiritilishi arzonligi kompyuterni kutubxona, audio-video sifatida qo'llash, shubhasiz boshlang'ich harajatlarni barchasini o'n barobar qoplaydi.[16-17]

Entuziaet - o'qituvchilar multimedia vositalari yordamida informatika darslarni mustaqil ishlab chiqib dars o'tishmoqda ammo maktabda informatikani o'qitishda multimedia vositalariga asoslangan, samarali metodika zarurdir, chunki u zamonaviy informatika sohasidagi bilimlardagi muammoni hal etish kerak.

Dasturlash asoslari fani o'qituvchilari o'z amaliyotida (faoliyatida) umumiy dasturlash tillarini o'qitishda darslarni ko'rgazmalilik va yetkaza olish prinsiplarini amalga oshirish muammolariga duch keladi. Bu muammolar sababi bo'lib, tasviriy ko'rgazmalilikning an'anaviy vositalarini deyarli samarali emasligi ularning past dinamikligi kuzatiladi, bu esa o'quvchilar tomonidan materialni yaxshi o'zlashtirishga xalaqit beradi.

Ta'lim muassasalari rivojlanishi uchun o'qitishning insonparvarlashtirish va differensiallashtirish xususiyatlari xosdir. Multimedia vositalari bilan o'quv jarayoni yetarlicha jihozlanganini o'qitishning individuallashtirishini kuchaytirishga olib keladi.

Boshqa tomondan, dasturlash asoslari fanini uzluksiz rivojlanishi bilan birga ilmiylik prinsipi dasturlash ta'limini ma'zmunini doimiy tahlildan o'tkazishini talab qiladi. Bu masalani tez va oson yechish uchun ma'lumotlarni elektron tashuvchilari va zamonaviy telekommunikatsiya vositalarini qo'llash maqsadga muvofiqdir.

Multimediali o'qitish prinsiplari ko'pincha dasturlashtirilgan o'qitish nazariyasidan o'zlashtirilgan, shuning uchun bu muammoni kengroq ko'lamda ko'rib chiqamiz.

Dasturlashtirilgan o'qitish g'oyasi AQSHda 50-yillarda paydo bo'lgan uning muallifi professor B.F. Skinner o'quv jarayonini boshqarish samaradorligini o'rgatish jarayoni haqidagi ilmiy ma'lumotlarga to'liq muvofiqlikda ko'rish orqali taklifni kiritgan. Bizning mamlakatimizda dasturlashtirilgan o'qitish bo'yicha birinchi ilmiy ishlar A.A.Abduqodirov, N.I.Taylakov, Q.T.Olimovlar tomonidan ma'lum qilingan.[15;22]

Dasturlashtirilgan o'qitish g'oyalari sobiq ittifoq pedagoglari va olimlarini jalb qilgan. Bu g'oyalarda ular ittifoq maktabining oldida mavjud bo'lgan dolzarb muammolarni yechishda potensial imkoniyatlarni ko'rishgan. Bahorda 1962 yili YUNESKO yig'ilishida dasturlashtirilgan o'qitish g'oyalari variantida amerikalik psixolog va pedagoglarning g'oyalari, qarashlari va tajribasi ustun bo'lgan sobiq ittifoq mamlakatida dasturlashtirilgan o'qitishning yangi pedagogik g'oyalari paydo bo'lishi sovet kibernetikasi rivojlanishi boshlanishi bilan bir vaqtda bo'lib qoldi va u pedagogikaga kibernetika, mantiq, statistika g'oyalari va metodlari kirib kelishi bilan bog'liq bo'lgan. Ular butun pedagogik jarayonni boshqara oladigan tizim nuqtai nazaridan qarashga yo'l ochib berdilar va shu bilan birga uning eng nozik tomonlarini aniqlashga yordam berdi. O'qitishning optimallashtirish muammolari bo'yicha tadqiqotlar o'quv jarayoni tahlilining aniq statistik metodlari pedagogikada keng qo'llanishiga olib keldi. Bu metodlar o'qituvchi va o'quvchi mehnati qo'yilgan maqsadga qay darajada erishilishiga nisbatan ob'ektiv qarashlarga imkon berdi. Erishilgan natijalar tahlili va o'quv materialini va uning o'zlashtirilishi bo'yicha eng samarali xabar berish metodlarni ishlab chiqishga yordam beradi.

Dasturlashtirilgan o'qitishga bir qator yutuqlarni keltirishgan: individualizatsiyalash (individuallashtirish) va differentsiatsiyalash o'quv jarayonini boshqarishning ilmiy tashkil etishi va bundan kelib chiqib, - uning samaradorligini oshirish. Multimediali o'qitish ham shunday yutuqlarga olib kelishini taxmin qilishimizga yo'l beradi. Dasturlashtirilgan o'qitishni mazmunmohiyatini ko'rib chiqib, o'quv materialining mazmuni va strukturasini yaxshilash bo'yicha bir qancha savollarni ko'rib chiqish zarurdir. Bu savollar o'quv jarayonini optimallashtirishga, uni boshqarish uchun dastur

tuzish bo'yicha, uni o'rganish va o'zlashtirish usullarini optimallashtirishga yordam beradi.

Dasturlashtirilgan o'qitishda o'quv jarayonining barcha elementlari ilmiy asoslanganlikni taqozo etadi. Chunki o'qitish – bu o'quvchi va o'qituvchining o'zaro bog'liq bo'lgan faoliyati jarayoni, unda bevosita bo'lgan aloqani aniqlash mumkin. Bevosita aloqa o'quv jarayoni payti va o'quvchilarning tarbiyasida o'rnatiladi. Qayta bo'lgan aloqa o'quv mazmunini sifati, chuqurligi, mustahkamligi haqidagi ma'lumotlarni olishga yordam beradi.

Dasturlash tillarini multimedia texnologiyalaridan o'qidarlarni asosan Ashampoo Snap va AutoPlay Media Studio dasturidan foydalanish maqsadga muvofiqdir.

Ashampoo Snap – bu dastur keng miqyosdagi audio-visual qo'llanma fayllarni: namoyishlar “презентация”, ko'rgazmali materiallar, audio-visual qo'llanma darslarning slaydlarini yaratish uchun xizmat qiladi. Asosiy funkciyalari:

- ekrandan audio va audio-visual qo'llanma fayllarini yozib olish;
- taxrirlash, loyiha yaratish;
- diskga yozish;
- AVI, FLV, PPT fayllari formatlarini qollash.

Ashampoo Snap dasturi kompyuter ekranidagi harakatlar, mikrofondan ovozlarni yozadi hamda kompyuterlar tushunadigan audio-visual qo'llanma fayllar formatiga o'girib beradi. Bu vositalardan Ashampoo Snap dasturi o'zining interfeysi, juda ko'plab formatlari, audio-visual qo'llanma fayllarga turli xil belgilar va izohlar qo'yilishi hamda dastur yordamida audio fayllarni ham yaratish mumkin. Dars yozish davomida ekranning kerakli joyini alohida ajratib ko'rsatish imkoni ham mavjud. Ana shu afzalliklari tufayli audio-visual qo'llanma darslar yozuvchi dastur sifatida Ashampoo Snap dasturi tanlandi.

Ashampoo Snap dasturi tasnifi. Ashampoo Snap dasturi ekranda bo'layotgan jarayonlarni saqlab qoyish uchun ishlatiladi. Dastur to'rtta yordamchi qismlardan iborat:

- MenuMaker;
- Player;
- Theater;
- Recorder.

Dasturning asosiy qismi, shubhasiz, Ashampoo Snap Recorder hisoblanadi. Barcha darslar aynan shu dastur yordamida yaratildi. Ashampoo Snap dasturi oynasining pastki qismida timeline deb nomlanuvchi ishchi stoli mavjud bo'lib, u orqali audio va audio-visual qo'llanma fayllar ustida turli xil amallarni bajarish mumkin. Bular jumlasiga fayllarni bir- biriga bog'lash, keraksiz qismlarni qirqib tashlash kabilar kiradi. Asosiy oynaning markazida dastur ishlashi mumkin bo'lgan fayllar ro'yxatini ko'rsatuvchi «Корзина клипов» (Clip Bin) qismi joylashgan. Shu fayllarni o'ng tarafdagi audio-visual qo'llanma pleyerda ko'rish mumkin. Bu kichkinagina Ashampoo Snap Player dasturi faqatgina bitta vazifani bajaradi-AVI fayllarini namoyish etadi. Ta'lim tizimida turli darslar orasidan keraklisini topish qiyinchilik tug'dirishi mumkin, buning uchun foydalanuvchi uchun navigatsiya menyusini yaratish kerak bo'ladi. Bunday navigatsiya interfeysini AutoPlay Media Studio dasturi yordamida hosil qilish mumkin. Mazkur dastur yordamida vizual ob'ektlarni qo'llagan holda mukammal navigatsiyani yaratish mumkin. Interfeysda hosil bo'ladigan har bir sahifa xuddi veb sahifalar kabi ko'rinishga ega bo'lib, sahifalardagi ob'ektlarda turli-xil amallarni belgilash mumkin. [12;18;21]

Autoplay Media Studio - Istalgan fayl yoki fayllar to'plamini bitta muhitga birlashtirish, qolaversa, CD yoki DVD disklar uchun Autorun-menjusini hosil qilishda Autoplay Media Studio eng kuchli vizual paket hisoblanadi. Multimedia texnologiyalariga asoslangan amaliy dasturlarni yaratish uchun Autoplay Media Studio dasturidan foydalanish foydalanuvchilar uchun juda oson va qulay interfeysni taqdim etadi.

Autoplay Media Studio bilan ishlashda deyarli dasturlash ishlari talab qilin- maydi. Foydalanuvchi faqat turli dizaynli dasturiy muhitni tanlash uchun bir nechta tayyor shakllardagi loyiha shablonlaridan foydalanishi mumkin. Bunda amaliy dastur muhitini dizaynga boy holatga tashkil etish uchun Autoplay dasturiy vositasi tarkibida tayyor ob'ektlar mavjud bo'lib, ular tarkibiga buyruq tugmasi, tovush kuchaytirgichi, fayllarni printerdan bosmaga chiqarishni ta'minlovchi vosita, Web-saytlarni ochuvchi va ularga murojaatni amalga oshirib beruvchi qator funksional ob'ektlarni kiritish mumkin. Amaliy dasturlarga oid grafik qobiqlarni yaratish va uni avtomatik ishga tushirish uchun Autoplay

Media Studio barcha kerakli fayllarni o'zi yaratadi. Foydalanuvchilar zimmasiga esa faqat qattiq disk va kompakt diskni yozish uchun tayyor loyihalarni shakllantirish vazifasi qoladi. 2008-yilning 12-martida Indigo Rose Corporation kompaniyasi Autoplay

Media Studio 7.1 versiyasini iste'molga chiqardi.

Autoplay Media Studio dasturi muhitida Visual Basic, Visual C++, Java, Macromedia Flash kabi qator tizimlarda yaratilgan hujjatlarni ham bimalol qayta ishlash mumkin. Dastur yordamida animatsiyalanuvchi menyuni, kataloglar daraxtini, ma'lumotlar bazasini va shunga o'xshash ob'ektlarni nafaqat tez yaratish, balki ularni boshqarish ham mumkin. Avtomatik ishga tushuvchi oynalarni o'zining kutubxonasidagi "niqob"lardan foydalangan holda ixtiyoriy shaklda (formada) yaratish mumkin. Bunday "niqob" sifatida .jpg, .bmp va .png kabi formatdagi fayllardan foydalanilsa ham bo'ladi. Qolaversa, ma'lumotlarni CD uchun tayyorlagan holda uni dasturning o'zidan turib, CD yoki DVDga yoza olishi Autoplay Media Studio dasturi naqadar keng imkoniyatlarga ega ekanligini ko'rsatadi. Tayyor loyiha bunda .exe kengaytmali fayl sifatida o'zi ochiluvchi arxiv ko'rinishda yoki qattiq diskdagi alohida papkada shakllantirilishi mumkin.

Bundan tashqari, dasturga matnni orfografik tekshirish imkoniyati ham kiritilgan.

Dasturning bu xossasi uning "Label", "Paragraph" va "Button" kabi ob'ektlari bilan birga ishlaydi. Agar dastur kompyuterga to'liq versiya bilan o'rnatilgan bo'lsa, matnni orfografik tekshirish uchun uning kutubxonasida juda katta hajmdagi lug'atlar bo'lishi mumkin. Shunday qilib, AutoPlay Media Studio 7.0 ning yangi versiyasi quyidagi imkoniyatlarga ega holda iste'molga chiqarilgan:

1. Avtomatik ishga tushuvchi xususiy menyu, interfaol taqdimotlar, multimediai lovalar, sanoqli daqiqalarda dasturiy ta'minotlarni yaratish;
2. Loyihaga turli-tuman fotografiya, musiqa, video, animatsiya, matn va boshqalarni biriktira olish xususiyati;
3. Web-ilovalarni yaratishga mo'ljallangan mukammallashgan instrumentlar;
4. XML, SQL va shifrlash mexanizmalari bilan ishlay olishi;
5. RTF-formatli hujjatlar bilan ishlay olishi;
6. Slide-show bilan ishlash imkoniyati;

7. Matn rangini o'zgartirish uchun "RadioButton" ob'ekting mavjudligi;
8. Bosmaga chiqarishning kengaytirilgan funksiyasi;
9. Ob'ektlarni formatlash imkoniyati;
10. Kalit so'zlar yordamida qidiruv tizimining mavjudligi;
11. CD, DVD kabi kompakt disklarga yozish imkoniyati va hk.

AutoPlayda ssenariylar yozish. Agar ishchi sohadagi (aniqrog'i sahifadagi) biror ob'ekt bajarishi kerak bo'lgan funksiyani xossalar panelidan foydalanib belgilamoqchi bo'lsak, u holda bitta ob'ekt uchun faqat bitta funksiya biriktiriladi, xolos. Ammo real sharoitda bitta ob'ekt bir nechta funksiyani bajarishiga zaruriyat tug'ilib qolishi mumkin. Bunday sharoitda vaziyatdan chiqish uchun xossalar panelidan tanlangan funksiyani bekor qilish va har bir funksiya uchun alohidaalohida ssenariy (script)lar yozish maqsadga muvofiq bo'ladi. Masalan, bir sahifaga audio musiqani qo'yib, boshqa sahifaga o'tganda musiqani o'chirish uchun uning "On Show" hodisasiga oid quyidagicha ssenariyni yozish mumkin:

```
Audio.Load(0, "AutoPlay//Audio//01.ogg", true, false);
```

Bunda: *true* – sahifa ochilishi bilan audio musiqa ishga tushishini, *false* esa boshqa sahifaga o'tganda bu audio musiqaning davom etmasligi yoki takrorlanmasligini bildiradi.

Bundan tashqari, sahifaga *Tree* ob'ekti joylashtirilgan bo'lishi mumkin. U holda bu *Tree* ob'ektidan foydalanganda uning tarmoqlarini ishga tushirish uchun quyidagi ssenariyni yozish mumkin: *If Tree.GetNode("Tree1", "1.1").Selected then Page.Jump("PgWebMavzu1");*

```
elseif Tree.GetNode("Tree1", "1.2").Selected then
```

```
Page.Jump("PgWebTest");
```

```
End
```

Bu ssenariyga binoan, *Tree* ob'ekting 1-darajali tarmog'ida turgan 1-satr belgilanib, sichqonchanning *DoubleClick* hodisasi ishlatilsa, u holda "PgWeb-

Mavzu1" nomli sahifa yuzaga keladi.

Agar *Tree* ob'ektining 1-darajali tarmog'ida turgan 2-satr belgilanib, ikki marta sichqoncha chap tugmasi bosilsa (ya'ni sichqonchaning *DoubleClick* hodisasi ishlatilsa), u holda "PgWebTest" nomli sahifa yuzaga keladi.

Shuningdek, biror buyruq tugmasi (*Button*)ni bosish orqali biror .exe faylni ishga tushirish kerak bo'lsa, u holda ssenariy quyidagicha yoziladi:

```
Shell.Execute("AutoPlay\\Docs\\MyProg.exe", "", "", SW_NORMAL, true);
```

Ba'zida turli variantlarda turli muloqot oynasini yuzaga keltirishga oid tarmoqlanuvchi algortmni yozishga to'g'ri keladi. Masalan, bosh sahifa ishga tushishidan oldin parol so'ragan kabi "InputBox" dastlab ekranga chiqishini xohlasangiz, u holda bosh sahifaning "Preload" bandi uchun quyidagi scriptni yozish kifoya:

```
1. name = "";
```

```
2. while (name ~= "CANCEL") and (name == "") do
```

```
3. name=Dialog.Input("Diqqat", "Ismi sharifingizni kiriting:");
```

```
4. if name == "" then
```

```
5. result=Dialog.Message("Xato!", "Ma'lumot kiritmadingiz!", MB_OK,
```

```
MB_ICONEXCLAMATION, MB_DEFBUTTON1);
```

```
6. elseif name == "CANCEL" then
```

```
7. name=Dialog.Input("Personal Information", "Bu yerga muallifning ismi  
sharifi kiritiladi:");
```

```
8. end
```

```
9. Paragraph.SetText("Paragraph5", name);
```

```
10. end
```

Ssenariy (Script) yozishda asosiy e'tibor ob'ekt ustida yuzaga keladigan hodisaga qaratiladi.

AutoPlay Media Studio 7.0 da ssenariy yozish birmuncha murakkabroq ish, ammo har qanday takrorlanuvchi vaziyatlarda ssenariy tarkibini kichraytirish yoki soddalashtirish maqsadida golabl ssenariylar yozish imkoniyatini taqdim eta oladi.[17]

Umumiy xulosa va tavsiyalar.

Hozirgi davrda O'zbekistonda ta'lim-tarbiya jarayonini jahon talablari darajasiga yetkazish borasida aql-zakovatli, mustaqil fikrlay oladigan, intellektual salohiyatga ega yoshlarni tarbiyalashga alohida e'tibor berilayapti. Bu o'z navbatida ta'lim jarayoniga katta e'tibor qaratish kerakligi, ta'lim jarayonining samaradorligini oshirish fan va texnika sirlarini chuqur egallashga yo'llaydi.

Bugungi kunda ta'lim muasasalarida «Dasturlash tillarini» o'qitish eng dolzarb muammolardan biri hisoblanadi. Uni hal qilish uchun ushbu bitiruv malakaviy ishida Dasturlash tillarini o'qitish uslubiyoti ishlab chiqildi. Jumladan quyidagi ishlar amalga oshirildi:

- Paskal dasturlash tilini o'qitishning asosiy shakli va mazmuni ko'rsatib berildi;
- Delphi dasturlash tilini o'qitishning asosiy shakli va mazmuni ko'rsatib berildi;
- C++ dasturlash tilini o'qitishning asosiy shakli va mazmuni ko'rsatib berildi;
- Paskal, Delphi, C++ dasturlash tillarini bo'yicha asoslangan tavsiyalar ishlab chiqildi;
- Paskal, Delphi, C++ dasturlash tillarini o'qitish bo'yicha bitta elektron darslik yaratildi.

Bitiruv malakaviy ishida dasturlash texnologiyalarini o'qitishning interaktiv metodlari va uning ahamiyatlari haqida ma'lumotlar hamda tavsiyalar berib o'tdik. Mavzuni o'rganish uchun bitiruv malakaviy ishini uchta bobga ajratdik. Birinchi bobda dasturlash vositalarining zamonaviy texnologiyalari, dasturlash tillarining tarixi, muammolari yechimlari hamda dasturlash texnologiyalari haqida ma'lumot berib o'tdik. Ikkinchi bobda Paskal, Delphi, C++ dasturlash tillarining imkoniyatlari haqida ma'lumot berib o'tdik. Uchinchi bobda Paskal, Delphi, C++ dasturlash tillarini o'qitish bo'yicha tavsiyalar berib o'tildi. Xulosa qilib shuni aytish mumkin bitiruv malakaviy ishida keltirib o'tilgan dasturlash texnologiyalarini o'qitish bo'yicha tavsiyalardan foydalanilsa dasturlash tillarini o'rgatishga yordam bo'lar edi. Zamonaviy interaktiv metodlar o'quvchilarning darsga qiziqishini oshiribgina qolmay bilim saviyalarini oshirishga, zaruruy ko'nikmalarga ham ega bo'lishlarida katta yordam beradi.

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1 O'zbekiston Respublikasi Prezidentining 2018-yil 13 dekabrda "O'zbekiston Respublikasi davlat boshqaruviga raqamli iqtisodiyot, elektron hukumat hamda axborot tizimlarini joriy etish bo'yicha qo'shimcha chora-tadbirlar to'g'risida"gi PF-5598-son Farmoni.

2 O'zbekiston Respublikasi Prezidentining 2018-yil 21-noyabrda "Raqamli iqtisodiyotni rivojlantirish maqsadida raqamli infratuzilmani yanada modernizatsiya qilish chora-tadbirlari to'g'risida"gi PQ-4022-son Qarori.

3 Axborot sohasi va ommaviy kommunikatsiyalarni yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida O'zbekiston Respublikasi Prezidentining Farmoni, 02.02.2019 yildagi PF-5653-son

4 Nazirov Sh., Musayev M.M., Ne'matov A., Qobulov R.V. "Delphi tilida dasturlash asoslari". O'quv qo'llanma. Toshkent 2006 y.

5 Aripov M. M., Otaxanov N. A. DELPHI dasturlash tili. /O'quv qo'llanma. -N.: Namangan, 2007. -544 b.

6 Nazirov Sh.A., Divald G. Dasturlash asoslari. Axborot-kommunikatsiya texnologiyalari sohasidagi kasb-hunar kollejlarning "Axborot-kommunikatsiya tizimlari mutaxassisligi talabalari uchun o'quv qo'llanma Toshkent - 2007 y.

7 Nazirov Sh.A., Qobulov R.V. Ob'ektga mo'ljallangan dasturlash. Kasb-hunar kollejlari uchun o'quv qo'llanma. G'afur G'ulom nomidagi nashriyot-matbaa ijodiy uyi Toshkent -2007 y.

8 Немнюгин А. Turbo Pascal, Москва Харьков-Минск 2000 г.

9 Бёме, д-р. Р., «Programmiersprachen C/C++» (Язык программирования C/C++), конспект лекций Института информатики Университета им. Мартина-Лютера, Халле-Виттенберг, 1996

10 ОТМ uchun C++ dasturlash tilini o'rganish bo'yicha uslubiy qo'llanma. Andijon 2015

11 Madraximov Sh.F, Gaynazarov S.M C++ tilida programmalash asoslari Toshkent 2009

12 Informatika va programmalsh.O'quv qo'llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, U.E.Adamboev, O'zMU, 2005 yil

13 N. A. Otaxanov. DELPHI tilini o'qitishdagi ayrim muammolar va ularni bartaraf qilish uslubiyoti «FARG'ONA» nashriyoti 2008 yil

14 M.Aripov, B.Begalov, U.Begimqulov, M.Mamarajabov. Axborot texnologiyalari. O'quv qo'llanma. Toshkent —Noshir 2009.

15 H.Raximov, T.Dexqonov. Zamonaviy dasturlash tillari. O'quv qo'llanma Toshkent 2009.

16 Xodiyev B.Y., Golish L.V. Mustaqil o'quv faoliyatini tashkil etish usul va vositalari: O'quv uslubiy qo'llanma– Toshkent TDIU, 2010.

17 Zakirova F., Muxamedxanov U., Sharipov Sh., Isyanov R., Esanboyev F. Elektron o'quv-metodik majmualar va ta'lim resurslarini yaratish metodikasi. Metodik qo'llanma.– Toshkent 2010.

18 Aripov M. “Programmalashga kirish”. O'quv qo'llanma. Toshkent 2008 y.

19. Otaxanov N. A. Ob'ektga yo'naltirilgan dasturlash texnologiyalarin o'qitishda turli tipdagi ma'lumotlar bilan ishlash uslubiyoti. //“Ta'lim jarayonini sifatli tashkil etishda ilg'or pedagogik texnologiyalarning roli”. Respublika ilmiy-amaliy konferensiya materiallari. -Toshkent, 2008

Internet manbaalari:

20.<http://cs.usu.edu.ru/>. Мультимедиа.

21.<http://websound.ru/>. MIDI-технология.

22.Dastur.uz

23.Ziyonet.uz

24.Shuhratbek.uz

25. ITportal.uz