

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

BUXORO DAVLAT UNIVERSITETI

Fizika-matematika fakulteti

“Axborot texnologiyalari” kafedrası

Jo'raqulov Jasur Javhar o'g'lining

**C++ dasturlash tilida statik ma'lumotlar tuzilmalari mavzusida
interaktiv o'quv materiallar yaratish mavzusida**

5111018- “Kasb ta'limi: Informatika va axborot texnologiyalari” ta'lim yo'nalishi
bo'yicha bakalavr darajasini olish uchun

BITIRUV MALAKAVIY ISHI

“Ish ko'rildi va himoyaga ruxsat berildi” **Ilmiy rahbar** ___ **I.O.Jalolov**

« ___ » _____ **2018 y.**

Kafedra mudiri

Taqrizchi _____

_____ **dots.T.B.Boltayev**

« ___ » _____ **2018 y.**

« ___ » _____ **2018 y.**

«Himoya qilishga ruxsat berildi»

Fakultet dekani **prof.Sh.M.Mirzayev**

« ___ » _____ **2018 y.**

Buxoro-2018

Mundarija.

KIRISH	3
I BOB. C++ DASTURLASH TILIDA STATIK MA'LUMOTLAR TUZILMASI	7
1.1. C++ dasturlash tilida asosiy operatorlar.	7
1.2. C++ dasturlash tilida massivlar.	23
1.3. C++ dasturlash tiliga satrlar.	30
Xulosa	36
II BOB. STATIK MA'LUMOTLAR TUZILMASI BO'YICHA INTERAKTIV O'QUV MATERIALLAR TAYYORLASH	37
2.1. Interaktiv o'quv materiallarini tayyorlashga foydalanilgan dasturlar	37
2.2. Interaktiv o'quv materiallar.	58
Xulosa	65
Xotima	65
Adabiyotlar:	66

KIRISH

“O‘zbekiston Respublikasi birinchi Prezidentining o‘lmas g‘oyalari, fikr, ko‘rsatmalari bizning qalbimiz va ongimizga shunday chuqur singib ketganki, ular bizning kelajak sari qat‘iyat bilan borishimizda barchamizga doimo tayanch va suyanch bo‘lishi muqarrar”.

Shavkat Mirziyoyev.

Bugun prezidentimiz Shavkat Mirziyoyevning sa‘y-harakatlari natijasida jamiyatimizning ko‘plab sohalarida yana-da ko‘proq yuksalishlar, o‘zgarishlar bo‘lmoqda. Sir emaski, ushbu rivojlanishlarning asosida xalq manfaati, el-u yurt ravnaqi yotadi. Ma‘lumki, jamiyatimizning rivojlanish strategiyalarining amalda keng ko‘lamli bajarilishini ta‘minlovchi shaxslar – bu rahbarlardir. O‘zbekiston Respublikasi Prezidenti Shavkat Mirziyoyevning ilk «Tanqidiy tahlil, qat‘iy tartib-intizom va shaxsiy javobgarlik – har bir rahbar faoliyatining kundalik qoidasi bo‘lishi kerak» nomidagi kitobi ham aynan shunday rahbarlar haqidadir.

Hozirgi kunda respublikamizdagi barcha oliy o‘quv yurtlarida “Informatika va axborot texnologiyalari” yo‘nalishi va mutaxassisliklariga turli xil dasturlash tillarini o‘rgatish mo‘ljallangan. Bizga ma‘lumki, dasturlash tillarining o‘ndan ortiq ko‘rinishlari mavjud, lekin qo‘llanilishi ko‘lamiga qarab C++ va C# dasturlash tillari yuqori dasturlash sinfiga mansubdir. Mutaxassislarning fikriga ko‘ra C++ dasturlash tili Assembler dasturlash tiliga eng yaqin bo‘lib, tezlik jihatidan 10 % ortda qolar ekan. Keyingi yillarda amaliy dasturchilarga juda ko‘p integratsion dastur tuzish muhitlari taklif etilmoqda. Bu muhitlar u yoki bu imkoniyatlari bilan bir–biridan farq qiladi. Aksariyat dasturlashtirish muhitlarining fundamental asosi C++ tiliga borib taqaladi. Ushbu kurs ishida hozirgi kunda kompyuterda berilgan funksiyalarni grafigini va har xil ko‘rinishdagi shakllarni chizish va ularni aniq koordinatalarini aniqlash, ekranni grafik rejimga o‘tkazish va undagi mavjud piksel va ranglardan foydalanish kabi vazifalarni o‘rganishga olib keladi.

Hozirgi vaqtga kelib komp`yuter olamida ko‘plab dasturlash tillari mavjud. Paskal, C++ va boshqa dasturlash tillaridir. C++ dasturlash tili universal

tildir. U UNIX sistemasi bilan bogʻlangan boʻlib, bu sistemada ishlatiladigan bir qancha dasturlar C++ tilida yozilgan. Paskal tili 1969 yil N. Virt tomonidan yaratilgan boʻlib, keyinchalik amerikaning Borland firmasi tomonidan qayta ishlandi va uni Turbo Pascal deb nomlangan. C++ Denis Ritchi tomonidan 1972 -yili UNIX tipidagi operatsion sistemalarini yaratish uchun loyihalashtirilgan. Turbo Pascal ni qayta ishlash natijasida ob'ektli dasturlash yoʻlga qoʻyildi va 1995 yilda Borland kompaniyasi guruxi dastur tuzuvchilari Chuck va Denny tomonidan Windows uchun moʻljallangan dasturlash muxiti Borland Delphi dasturlash vositasi yaratildi. Borland C++ va Delphi dasturlash tili Windows uchun moʻljallangan boʻlib, uning birinchi versiyasi Windows operatsion sistema qobigʻida ishlagan. Borland C++ va Delphi dasturlash tili – bu dasturlarni qayta ishlash muxiti boʻlib, Windows operatsion sistemasida ishlaydi. Unda ob'ektli dasturlash tillari boʻlgan Object mujassamlashgan. Borland C++ va Delphi vizual proektlar, turli xolat protseduralarini qayta ishlash va dasturlarni qayta ishlashda vaqtdan yutish va boshqalarni oʻz ichiga oladi.

Men ham ushbu BMI davomida C++ tilida matn muxarriri bilan ishlovchi obektlarga moʻljallangan dasturni tuzib chiqmoqchiman. Ushbu dastur kichik dastur boʻlsada kelajakda boshqa dasturlar tuzishim uchun fundament vazifasini bajarishga harakat qilaman. Chunki hozirgi kunda matnlar bilan ishlash uchun katta dasturlar paketini oʻrnatish, agar oʻzida ishlaydigan dasturni ishlatsak oʻzimizni sof tilda emas boshqa davlat tillarida ishlashi noqulayliklar tugdiradi. Shuning uchun kichik matn muxarrirlari bilan ishlovchi dastur ishlab chiqaramiz

Bu dasturni tuzib nafaqat yozishni amalga oshirish balki kichik matnlar ustida taxrirlash va dasturni ishlashni bilmaydigan foydalanuvchi ham bir koʻrinishda tushinadigan tushinarli, bejirm koʻrinishga ega dastur tuzmoqchiman.

Men bu dasturni tuzib obektlar yaratishni va ular orqali shunga oʻxshash yana boshqa obektlar yaratmoqchiman bu dasturda ishlashimga yana bir sabab boshqa obektlarga moʻljallanmagan dasturlash tillarida faqat masalani matematik dasturi tuzib natija olinadi bunda yani C++ Builderda esa boshqa imkoniyatlar va kampanentalar bilan ham ishlashni oʻrganmochimiz .

Mavzuning dolzarbligi. C++ dasturlash tilida statik ma'lumotlar tuzilmalari bo'yicha ma'lumotlarni to'plash va ulardan qanday foydalanish bo'yicha mustaqil o'rganuvchilarga interaktiv o'quv materiallari orqali o'rgatish. Bunday interaktiv o'quv materiallar mavzuni o'rganishda mustaqil o'rganuvchilarga juda katta yordam beradi. Bunday jarayonda mustaqil o'rganuvchilar kerakli mavzuni o'rganib, bu mavzu bo'yicha topshiriqlarni interaktiv bajarib boradilar.

Bitiruv malakaviy ishimizning maqsad va vazifalari. C++ dasturlash tilida statik ma'lumotlar tuzilmalarini o'rganish, interaktiv o'quv materiallarini tayyorlashda foydalaniladigan dasturiy taminotlarni o'rganish. Shular asosida interaktiv o'quv materiallarini tayyorlash.

Bu maqsadga erishish uchun quyidagi masalalarni yechish kerak:

- C++ dasturlash tilini o'rganish;
- Statik ma'lumotlar tuzilmalarini o'rganish;
- Interaktiv o'quv materiallarini tayyorlashni o'rganish;

Bitiruv malakaviy ishimizning o'rganilganlik darajasi. Interaktiv o'quv materiallarini tayyorlashdan oldin C++ dasturlash tili, statik ma'lumotlar tuzilmalari va interaktiv o'quv materiallarini tayyorlashda foydalaniladigan dasturiy taminotlar o'rganib chiqildi.

Bitiruv malakaviy ishimizning predmeti. C++ dasturlash tili, statik ma'lumotlar tuzilmalari va interaktiv o'quv materiallarini tayyorlashda foydalaniladigan dasturiy taminotlar asosiy predmet hisoblanadi.

Bitiruv malakaviy ishimizning obykti sifatida "Obyektga yo'naltirilgan dasturlash", "Dasturlash asoslari" fanlari olindi.

Bitiruv malakaviy ishimizning ilmiy farazi ta'lim jarayonini yanada takomillashtirishning yangi usul va vositalarini yaratish, o'quv jarayonini samaradorligini oshirish. Interaktiv o'quv materiallarini tayyorlash usul va metodlari.

Bitiruv malakaviy ishimizning yangiligi. C++ dasturlash tilida statik ma'lumotlar tuzilmalari o'rganish bo'yicha interaktiv o'quv materiallarini

tayyorlash va o'quv jarayoniga qo'llash bitiruv malakaviy ishimning yangiligi hisoblanadi.

Bitiruv malakaviy ishimizning amaliy ahamiyati shundan iboratki, bu interaktiv o'quv materiallaridan C++ dasturlash tilini o'rganishda va o'quv jarayonida foydalanilsa, ta'lim oluvchilar mustaqil, erkin va ixtiyoriy bilim olish imkoniyatiga ega bo'ladi, bu esa o'zining yuqori samarasini beradi deb o'ylayman.

Bitiruv malakaviy ishimizning metodologik asosini uning tarkibiy qismi, undagi o'quv va didaktik vositalarning jamlanganligidir. Bu dars jarayonini interfaol tashkil etishda ham amaliy ahamiyat kasb etadi.

Bitiruv malakaviy ishimizning metodlari. C++ dasturlash tilida statik ma'lumotlar tuzilmalarini o'rganishda foydalanilgan interaktiv o'quv materiallarini yaratishda ishlab chiqilgan metodlar.

Bitiruv malakaviy ishining tarkibi va hajmi quyidagicha. Kirish, 2 ta bob, har bir bobning qisqacha xulosasi, adabiyotlar ro'yxati, xotimadan iborat bo'lib jami 65 betga bayon qilingan. I- bobda C++ dasturlash tilida statik ma'lumotlar, II- bobda Statik ma'lumotlar tuzilmasi bo'yicha interaktiv o'quv materiallar tayyorlash Kirish qismi 6 betdan iborat , tushuntirish qismi 57 betdan iborat. Bitiruv malakaviy ishida 12 ta adabiyotdan foydalanildi.

I BOB. C++ DASTURLASH TILIDA STATIK MA'LUMOTLAR TUZILMASI.

1.1. C++ dasturlash tilida asosiy operatorlar.

Birinchi elektron hisoblash mashinalari paydo bo`lishi bilan dasturlash tillari evolyutsiyasi boshlanadi. Dastlabki kompyuterlar ikkinchi jahon urushi vaqtida artilleriya snaryadlarining harakat traektoriyasini hisob-kitob qilish maqsadida qurilgan edi. Oldin dasturchilar eng sodda mashina tilini o`zida ifodalovchi kompyuter komandalari bilan ishlaganlar. Bu komandalar nol va birlardan tashkil topgan uzun qatorlardan iborat bo`lar edi. Keyinchalik, insonlar uchun tushunarli bo`lgan mashina komandalarini o`zida saqlovchi (masalan, ADD va MOV komandalari) assembler tili yaratildi. Shu vaqtlarda BASIC va COBOL singari yuqori sathli tillar ham paydo bo`ldiki, bu tillar tufayli so`z va gaplarning mantiqiy konstruktsiyasidan foydalanib dasturlash imkoniyati yaratildi. Bu komandalarni mashina tiliga interpretatorlar va kompilyatorlar ko`chirar edi. Interpretator dasturni o`qish jarayonida uning komandalarini ketma - ket mashina tiliga o`tkazadi. Kompilyator esa yaxlit programma kodini biror bir oraliq forma - obyekt fayliga o`tkazadi. Bu bosqich kompilyatsiya bosqichi deyiladi. Bundan so`ng kompilyator obyektli faylni bajariluvchi faylga aylantiradigan kompanovka dasturini chaqiradi.

Interpretatorlar bilan ishlash osonroq, chunki dastur komandalari qanday ketma - ketlikda yozilgan bo`lsa shu tarzda bajariladi. Bu esa dastur bajarilishini nazorat qilishni osonlashtiradi. Faqatgina kompilyatsiya qilingan fayl tezroq bajariladi, chunki bundagi komandalar kompilyatsiya jarayonida mashina tiliga o`tkazilgan bo`ladi.

C++ kabi kompilyatsiya qiluvchi dasturlash tillarini yana bir afzalligi hosil bo`lgan dastur kompyuterda kompilyatorsiz ham bajarilaveradi. Interpretatsiya qiluvchi tillarda esa tayyor dasturni ishlatish uchun albatta mos interpretator dasturi talab qilinadi.

Ayrim tillarda (masalan, VISUAL BASIC) interpretator rolini dinamik kutibxonalar bajaradi. Java tilining interpretatori esa virtual mashinadir (Virtual Machine, yoki VM). Virtual mashinalar sifatida odatda brouzer (Internet Explorer yoki Netscape) lar qo`llaniladi.

Ko`p yillar davomida dasturlarning asosiy imkoniyati uning qisqaligi va tez bajarilishi bilan belgilanar edi. Dasturni kichikroq qilishga intilish kompyuter xotirasini juda qimmatligi bilan bog`liq bo`lsa, uning tez bajarilishiga qiziqish protsessor vaqtining qimmatbaholigiga bog`liq edi. Lekin kompyuterlarning narxi tushishi bilan dastur imkoniyatini baholash mezoni o`zgardi. Ekspluatatsiyaning oddiyligi, konkret masalani yechish bilan bog`liq bo`lgan talabni ozroq o`zgarishiga, dasturni ortiqcha chiqimlarsiz oson moslashtirish bilan izohlanadi.

C++ tilida ifodalar biror bir hisoblash natijasini qaytaruvchi boshqa ifodalar ketma-ketligini boshqaradi yoki hech nima qilmaydi (nol ifodalar).

C++ tilida barcha ifodalar nuqtali vergul bilan yakunlanadi. Ifodaga misol qilib o`zlashtirish amalini olish mumkin.

```
x=a+b;
```

Algebradan farqli ravishda bu ifoda $x = a + v$ ga teng ekanligini anglatmaydi. Bu ifodani quyidagicha tushinish kerak:

a va v o`zgaruvchilarni qiymatlarini yig`ib natijasini x o`zgaruvchiga beramiz yoki x o`zgaruvchiga $a+v$ qiymatni o`zlashtiramiz. Bu ifoda birdaniga ikkita amalni bajaradi, yig`indini hisoblaydi va natijani o`zlashtiradi. Ifodadan so`ng nuqtali vergul qo`yiladi. (=) operatori o`zidan chap tomondagi operandga o`ng tomondagi operandlar ustida bajarilgan amallar natijasini o`zlashtiradi.

Bo`sh joy (probel) belgisi.

Bo`sh joy belgilariga nafaqat probel, balki yangi satrga o`tish va tabulyatsiya belgilari ham kiradi. Yuqorida keltirilgan ifodani quyidagicha ham yozish mumkin:

```
x = a+b;
```

Bu variantda keltirilgan ifoda ko`rimsiz va tushunarsiz bo`lsa ham to`g`ridir.

Bo'sh joy belgilari dasturning o'qilishligini ta'minlaydi.

Bloklar va kompleks ifodalar.

Ba'zan dastur tushunarli bo'lishi uchun o'zaro mantiqiy bog'langan ifodalarni blok deb ataluvchi komplekslarga birlashtirish qulaydir. Blok ochiluvchi figurali qavs ({) bilan boshlanadi va yopiluvchi figurali qavs (}) bilan tugaydi. Blok ochilganda va yopilganda nuqtali vergul qo'yilmaydi.

```
{  
    temp= a;  
    a = b;  
    b = temp;  
}
```

Bu blok xuddi bir ifodadek bajariladi, u a va b o'zgaruvchilar qiymatlarini almashtiradi.

Amallar.

Bajarilishi natijasida biror bir qiymat qaytaradigan barcha ifodalar C++ tilida amallar deyiladi. Amallar albatta biror bir qiymat qaytaradi. Masalan, 3+2 amali 5 qiymatni qaytaradi.

Operatorlar.

Operator - bu qandaydir amalni bajarish tug'risida kompilyatorga uzatiladigan literaldir. Operatorlar operandlarga ta'sir qiladi. C++ da operandlar deb alohida literallar va butun ifodalar tushuniladi.

C++ tilida ikki ko'rinishdagi operatorlar bor:

o'zlashtirish operatorlari

matematik operatorlar

O'zlashtirish operatori.

O'zlashtirish operatori (=) o'zidan chap tomonda turgan operand qiymatini tenglik belgisidan o'ng tomondagilarni hisoblangan qiymatiga almashtiradi. Masalan,

```
x = a+b;
```

ifodasi x operandga a va b o'zgaruvchilarni qiymatlarini qo'shishdan hosil bo'lgan natijani o'zlashtiradi.

O'zlashtirish operatoridan chapda joylashgan operand adresli operand yoki l-qiymat (chap-chap so'zidan olingan) deyiladi. O'zlashtirish operatoridan o'ngda joylashgan operand operatsion operand yoki r-qiymat deyiladi.

O'zgarmlar faqatgina r-qiymat bo'lishi mumkin va hech qachon adresli operand bo'la olmaydi, chunki dasturning bajarilishi jarayonida o'zgarmlar qiymatini o'zgartirib bo'lmaydi.

```
35 = x // noto'g'ri!
```

l-qiymat esa r-qiymat bo'lishi mumkin.

Matematik operatorlar.

C++ tilida 5 ta asosiy matematik operatorlar qo'llaniladi: qo'shish (+), ayirish (-), ko'paytirish (*), butun songa bo'lish (\) va modul bo'yicha bo'lish (%) (qoldiqni olish).

Ishorasiz butun sonlarni ayirishda, agarda natija manfiy son bo'lsa g'ayrioddiy natija beradi. Buni misoldan ko'rishimiz mumkin.

Ayirish natijasida butun sonni to'lib qolishiga misol

```
#include <iostream.h >
int main()
{
    unsigned int ayirma
    unsigned int kattaSon = 100;
    unsigned int kichikSon = 50;
    ayirma = kattaSon - kichikSon;
    cout << "Ayirma" << ayirma << " ga teng\n";
    ayirma = kichikSon - kattaSon ;
    cout << "Ayirma" << ayirma << " ga teng\n";
    endl;
    return 0;
}
```

HATIJA:

Ayirma: 50 ga teng.

Shartli operator. Shartli operator ikki ko'rinishda ishlatilishi mumkin:

If (ifoda)

1- operator

else

2- operator

Shartli operator bajarilganda avval ifoda hisoblanadi ; agar qiymat rost ya'ni nol dan farqli bo'lsa 1- operator bajariladi. Agar qiymat yolg'on ya'ni nol bo'lsa va **else** ishlatilsa 2-operator bajariladi. **else** qism har doim eng yaqin **if** ga mos qo'yiladi.

if(n>0)

if(a>b)

 Z=a;

else

 Z=b;

Agar **else** qismni yuqori **if** ga mos qo'yish lozim bo'lsa, figurali qavslar ishlatish lozim.

if(n>0) {

if(a>b)

 z=a;

}

else

 z=b;

Misol tariqasida uchta berilgan sonning eng kattasini aniqlash dasturini ko'ramiz:

```
#include <iostream.h>
```

```
void( )
```

```
{ float a,b,c,max);
```

```
    Cout <<"\n a="; Cin>>a;
```

```
    Cout <<"\n b="; Cin>>b;
```

```

    Cout <<“\n c=”; Cin>>c;
if (a>b)
if (a>c) max=a else max=c;
else
if b>c then max=b else max=c;
    Cout <<“\n” <<max;
}

```

Keyingi misolda kiritilgan ball va maksimal ball asosida baho aniqlanadi:

```

#include <iostream.h>
void main( )
{
    float ball,max_ball,baho);
    Cout<< “\n ball=”; Cin>>(“%f”,&ball);
    Cout<<“\n max_ball=”; Cin>>max_ball;
    d=ball/max_ball;
if (d>0.85) baho=5 else
if (d>75) baho=4 else
if (d>0.55) then baho=3 else baho=2;
    Cout<<“\n baho;
}

```

Kalit bo'yicha tanlash operatori. Kalit bo'yicha o'tish **switch** operatori umumiy ko'rinishi quyidagicha

```

Switch(<ifoda>) {
    Case <1-qiymat>:<1-operator>
        ...
break;
    ...
default: <operator>
    ...
case: <n-operator>;
}

```

Oldin qavs ichidagi butun ifoda hisoblanadi va uning qiymati hamma variantlar bilan solishtiriladi. Biror variantga qiymat mos kelsa shu variantda ko'rsatilgan operator bajariladi. Agar biror variant mos kelmasa **default** orqali ko'rsatilgan operator bajariladi. **Break** operatori ishlatilmasa shartga mos kelgan variantdan tashqari keyingi variantdagi operatorlar ham avtomatik bajariladi. **Default; break** va belgilangan variantlar ixtiyoriy tartibda kelishi mumkin. **Default** yoki **break** operatorlarini ishlatish shart emas. Belgilangan operatorlar bo'sh bo'lishi ham mumkin.

Misol tariqasida bahoni son miqdoriga qarab aniqlash dasturini ko'ramiz.

```
Include <iostream.h>
Int baho;
Cin>> baho;
Switch(baho)
{case 2:Cout <<“\n emon”;break;
case 3:Cout <<“\n urta”;break;
case 4:Cout <<“\n yahshi”;break;
case 5:Cout <<“\n a'lo”;break;
default: Cout <<“\n baho notugri kiritilgan”;
};
}
```

Keyingi misolimizda kiritilgan simvol unli harf ekanligi aniqlanadi:

```
Include <iostream.h>
Int baho; Char c; Cin >> c;
Switch(c)
{case 'a':
case 'u':
case 'o':
case 'i':
Cout <<“\n Kiritilgan simvol unli harf”;break;
default: Cout <<“\n Kiritilgan simvol unli harf emas”;
```

```
};  
}
```

Break operatori. Ba'zihollarda sikl bajarilishini ixtiyoriy joyda to'xtatishga to'g'ri keladi. Bu vazifani break operatori bajarishga imkon beradi. Bu operator darhol sikl bajarilishini to'xtatadi va boshqaruvni sikldan keyingi operatorlarga uzatadi.

Misol uchun o'quvchining n ta olgan baholariga qarab uning o'qish sifatini aniqlovchi dasturini ko'ramiz. Buning uchun dasturda o'quvchining olgan minimal bahosi aniqlanadi

```
#include <iostream.h>  
void main()  
{  
    int i,n,min,p;  
while (1)  
    {Cout<<"Baholar soni="; Cin>>n;};  
if (n>0) break;  
Cout<<"Hato! n>0 bulishi lozim ! \n";  
for (i=1,min=5; i<=n; i++)  
{ cin>>p;  
if (p<2)||(p>5) {min=0; break};  
if (min>p) min=p;  
}  
if (p<2)||(p>5) cout break;  
switch(min)  
case 0:cout<<"Baho noto'g'ri kiritilgan";break;  
case 2:cout<<"Talaba yomon o'qiydi";break;  
case 3:cout<<"Talaba o'rtacha o'qiydi";break;  
case 4:cout<<"Talaba yaxshi o'qiydi";break;  
case 5:cout<<"Talaba a'lo o'qiydi";break;  
}
```

Biz misolda xato kiritilgan n qiymatdan saqlanish uchun `while(1)` sikl kiritilgan. Agar $n > 0$ bo'lsa `Break` operatori siklni tuxtatadi va dastur bajarilishi davom etadi. Agar kiritilayotgan baholar chegarada yotmasa n ga 0 qiymat berilib darhol sikldan chiqiladi.

O'tish operatori GO TO.

O'tish operatorining ko'rinishi:

`Go to <identifikator>`. Bu operator identifikator bilan belgilangan operatorga o'tish kerakligini ko'rsatadi.

Misol uchun `goto A1;...;A1:y=5;`

Strukturali dasturlashda `Go to` operatoridan foydalanmaslik maslahat beriladi. Lekin ba'zi hollarda o'tish operatoridan foydalanish dasturlashni osonlashtiradi.

Misol uchun bir necha sikldan birdan chiqish kerak bo'lib qolganda, to'g'ridan-to'g'ri `break` operatorini qo'llab bo'lmaydi, chunki u faqat eng ichki sikldan chiqishga imkon beradi.

Quyidagi misolda n ta qatorga n tadan musbat son kiritiladi. Agar n yoki sonlardan biri manfiy bo'lsa, kiritish qaytariladi:

```
# include <iostream.h>
int n, I, j, k;
M1: Cout<<"\n n="; Cin>>n;
If (n<=0) { Cout<<"\n xato! n>0 bo'lishi kerak";
Go to M1; } ;
M: Cout<<"x sonlarni kiriting \n";
For (I=1; I<=10; I++) {Cout<<"\n I="<< i;
For (j=1 ; j<=10; j++) {Cin>> k;
if (k<=0) goto M;}
}
```

Bu masalani GOTO operatorisiz hal qilish uchun qo'shimcha o'zgaruvchi kiritish lozimdir.

```
# include <iostream.h>
int n, I, j, k;
```

```

while 1 {
    Cout<<“\n n=”; Cin>>n;
if (n>0) break;
    Cout<<“\n hatto! n>0 bo`lishi kerak”;
    };
int M=0;
    While M
{ M=0;
    Cout<<“x sonlarni kiriting \n”;
    For (I=1; I<=10; I++) {
If (M) break;
    Cout<< (“\n I=%, i);
    For (j=1 ;j<=10; j++) {Cin>> (“%f”, k);
if (k<=0) {M=1;break;}
    }
}

```

FOR TAKRORLASH OPERATORI.

For strukturasi sanovchi (counter) bilan bajariladigan takrorlashni bajaradi. Boshqa takrorlash bloklarida (while, do/while) takrorlash sonini control qilish uchun ham sanovchini qo'llasa bo'lardi, bu holda takrorlanish sonini oldindan bilsa bo'lardi, ham boshqa bir holatning vujudga kelish-kelmasligi orqali boshqarish mumkin edi. Ikkinchi holda ehtimol miqdori katta bo'ladi. Masalan qo'llanuvchi belgilangan sonni kiritmaguncha takrorlashni bajarish kerak bo'lsa biz while ni ifodalar-ni ishlatamiz. for da esa sanovchi ifodaning qiymati oshirilib (kamaytirilib) borilaveradi va chegaraviy qiymatni olganda takrorlanish tugatiladi. for ifodasidan keyingi bitta ifoda qaytariladi. Agar bir necha ifoda takrorlanishi kerak bo'lsa, ifodalar bloki {} qavs ichiga olinadi.

//Ekranda o'zgaruvching qiymatini yozuvchi dastur, for ni ishlatadi.

```
# include <iostream.h>
```

```
int main()
```

```
{
```



```
for (int i = 0; i < 5; i++){  
cout<< i << endl;  
}  
return (0);  
}
```

Ekkranda:

```
0  
1  
2  
3  
4
```

for strukturasi uch qismdan iboratdir. Ular nuqtali vergul bilan bir-biridan ajratiladi.for ning ko'rinishi:

```
for( 1. qism ; 2. qism ; 3. qism ){  
takror etiladigan blok}
```

1. qism - e'lon va initsalizatsiya.

2. qism - shartni tekshirish (o`zgaruvchini chegaraviy qiymat bilan solishtirish).

3.qism - o'zgaruvchining qiymatini bajarilish ketma-ketligi quyidagichadir:

1. qism bajariladi (faqat bir marta), keyin

2. qismdagi shart tekshiriladi va agar u true bo'lsa takrorlanish bloki ijro ko'radi

3. qismda o'zgaruvchilar o'zgartiriladi, keyin yana ikkinchi qismga

o'tiladi.for strukturamizni while struktura bilan almashtirib ko'raylik:

```
for (int i = 0; i < 10 ; i++)  
cout<< "Hello!"<<endl;
```

Ekkranga 10 marta Hello! so'zi bosib chiqariladi. i o'zgaruvchisi 0 dan 9 gacha o'zgaradi. i 10 ga teng bo'lganda esa i < 10 sharti noto'g'ri (**false**) bo'lib chiqadi va for strukturasi nihoyasiga yetadi. Buni while bilan yozsak:

```
int i = 0;
```

```

while ( i<10 ){
cout<< "Hello!" <<endl;
i++;
}

```

Endi for ni tashkil etuvchi uchta qismning har birini alohida ko'rib chiqsak. Birinchi qismda asosan takrorlashni boshqaradigan sanovchi (counter) o'zgaruvchilar e'lon qilinadi va ularga boshlang'ich qiymatlar beriladi (initsializatsiya). Yuqoridagi dastur misolida buni `int i = 0;` deb berganmiz. Ushbu qismda bir necha o'zgaruvchilarni e'lon qilishimiz mumkin, ular vergul bilan ajratiladi. Ayni shu kabi uchinchi qismda ham bir nechta o'zgaruvchilarning qiymatini o'zgartirishimiz mumkin. Undan tashqari birinchi qismda for dan oldin e'lon qilingan o'zgaruvchilarni qo'llasak bo'ladi.

Masalan:

```

int k = 10;
int l;
for (int m = 2, l = 0 ; k <= 30 ; k++, l++, ++m) {
cout<< k + m + l;
}

```

Albatta bu ancha sun'iy misol, lekin u bizga for ifodasining naqadar moslashuvchanligi ko'rsatadi. for ning qismlari tushrib qoldirilishi mumkin.

Masalan:

```

for(;;) {}

```

ifodasi cheksiz marta qaytariladi. Bu for dan chiqish uchun break operatorini beramiz. Yoki agar sanovchi sonni takrorlanish bloki ichida o'zgartirsak, for ning 3-qismi kerak emas. Misol:

```

for(int g = 0; g < 10; ){
cout<< g;
g++;
}

```

Yana qo'shimcha misollar beraylik.

```
for (int y = 100; y >= 0; y-=5){
```

```
...
```

```
ifoda(lar);
```

```
...
```

```
}
```

Bu yerda 100 dan 0 gacha 5 lik qadam bilan tushiladi.

```
for(int d = -30; d<=30; d++){
```

```
...
```

```
ifoda(lar);
```

```
...
```

```
}
```

60 marta qaytariladi. For strukurasi bilan dasturlarimizda yanada yaqinroq tanishamiz. Endi

1-qismda e'lon qilinadigan o'zgaruvchilarning xususiyati haqida bir og'iz aytib o'taylik. Standartga ko'ra bu qismda e'lon qilingan o'zgaruvchilarning qo'llanilish sohasi faqat o'sha for strukturasi bilan chegaralanadi. Ya'ni bitta blokda joylashgan for strukturalari mavjud bo'lsa, ular ayni ismli o'zgaruvchilarni qo'llay olmaydilar. Masalan quyidagilar xatodir:

```
for(int j = 0; j<20 ; j++){...}
```

```
...
```

```
for(int j = 1; j<10 ; j++){...} //hato!
```

j o'zgaruvchisi birinchi for da e'lon qilinib bo'lindi. Ikkinchi for da ishlatish mumkin emas. Bu masalani yechish uchun ikki xil yo'l tutish mumkin.

Birinchisi bitta blokda berilgan for larning har birida farqli o'zgaruvchilarni qo'llashdir. Ikkinchi yo'l for lar guruhidan oldin sanovchi vazifasini bajaruvchi bir o'zgaruvchini e'lon qilishdir. Va for larda bu o'zgaruvchiga faqat kerakli boshlang'ich qiymat beriladi xolos.

for ning ko'rinishlaridan biri bo'sh tanali for dir.

```
for(int i = 0 ; i < 1000 ; i++);
```

Buning yordamida biz dastur ishlashini sekinlashtirishimiz mumkin.

Switch operatori.

`if-else-if` yordami bilan bir necha shartni test qilishimiz mumkin. Lekin bunday yozuv nisbatan o'qishga qiyin va ko'rinishi qo'pol bo'ladi. Agar shart ifoda butun son tipida bo'lsa yoki bu tipga keltirilishi mumkin bo'lsa, biz `switch` (tanlash) ifodalarini ishlata olamiz.

`switch` strukturasi bir necha `case` etiketlaridan (label) va majburiy bo'lmagan `default` etiketidan iboratdir. Etiket bu bir nomdir. U dasturnig bir nuqtasidaga qo'yiladi. Programmaning boshqa yeridan ushbu etiketga o'tishni bajarish mumkin. O'tish yoki sakrash `goto` bilan amalga oshiriladi, `switch` blokida ham qo'llaniladi.

5 lik sistemadagi bahoni so'zlik bahoga o'tqizadigan blokni yozaylik.

```
int baho;  
baho = 4;  
switch (baho) {  
case 5: cout << "A'lo";  
break;  
case 4: cout << "Yahshi";  
break;  
case 3: cout << "Qoniqarli";  
break;  
case 2:  
case 1: cout << "A'lo";  
break;  
default: cout << "Baho hat o kiritildi!";  
break;  
}
```

`Switch` ga kirgan o'zgaruvchi (yuqorigi misolda `baho`) har bir `case` etiketlarining qiymatlari bilan solishtirilib chiqiladi. Solishtirish yuqoridan pastga bajariladi. Shartdagi qiymat etiketdagi qiymat bilan teng bo'lib chiqqanda ushbu

`case` ga tegishli ifoda yoki ifodalar bloki bajariladi. So'ng `break` sakrash buyrug'i bilan `switch` ning tanasidan chiqiladi. Agar `break` qo'yilmasa, keyingi etiketlar qiymatlari bilan solishtirish bajarilmasdan ularga tegishli ifodalar ijro ko'raveradi. Bu albatta biz istamaydigan narsa. default etiketi majburiy emas. Lekin shart chegaradan tashqarida bo'lgan qiymatda ega bo'lgan hollarni diagnostika qilish uchun kerak bo'ladi. `case` va etiket orasida bo'sh joy qoldirish shartdir. Chunki, masalan, `case 4: ni case4:` deb yozish oddiy etiketni vujudga keltiradi, bunda sharti test qilinayotgan ifoda 4 bilan solishtirilmay o'tiladi.

Do while takrorlash operatori.

`Do while` ifodasi `while` strukturasi bilan o'xshashdir. Bitta farqi shundaki `while` da shart boshiga tekshiriladi. `Do while` da esa takrorlanish tanasi eng kamida bir marta ijro ko'radi va shart strukturaning so'ngida test qilinadi. Shart true bo'lsa blok yana takrorlanadi. Shart false bo'lsa `do while` ifodasidan chiqiladi. Agar `do while` ichida qaytarilishi kerak bo'lgan ifoda bir dona bo'lsa `{ }` qavslarning keragi yo'qdir. Quyidagicha bo'ladi:

```
do
```

```
ifoda;
```

```
while (shart);
```

Lekin `{ }` qavslarning yo'qligi dasturchini adashtirishi mumkin. Chunki qavssiz `do while` oddiy `while` ning boshlanishiga o'xshaydi. Buni oldini olish uchun `{ }` qavslarni har doim qo'yishni tavsiya etamiz.

```
int k = 1;
```

```
do {
```

```
    k = k * 5;
```

```
} while ( !(k>1000) );
```

Bu blokda 1000 dan kichik yoki teng bo'lgan eng katta 5 ga karrali son topilmoqda. `while` shartini ozroq o'zgartirib berdik, `!` (not - inkor) operatorining ishlashini misolda ko'rsatish uchun. Agar oddiy qilib yozadigan

bo'lsak, while shartining ko'rinishi bunday bo'lardi: while (k<=1000); Cheksiz takrorlanishni oldini olish uchun shart ifodasining ko'rinishiga katta e'tibor berish kerak. Bir nuqtaga kelib shart true dan false qiymatiga o'tishi shart.

Mantiqiy operatorlari.

Operator

Belgi

Misol

VA

&&

1ifoda && 2ifoda

YOKI

||

1ifoda||2ifoda

INKOR

!

!ifoda

Mantiqiy ko'paytirish operatori.

Mantiqiy ko'paytirish operatori ikkita ifodani hisoblaydi, agar ikkala ifoda true qiyma tqaytarsa VA operatori ham true qiymat qaytardi. Agarda sizning qorningiz ochligi rost bo'lsa VA sizda pul borligi ham rost bo'lsa siz supermarketga borishingiz va u yerdan o'zingizga tushlik qilish uchun biror bir narsa xarid qilishingiz mumkin. Yoki yana bir misol

masalan

if(x==5)&&(y==5)

mantiqiy ifodasi agarda x va y o'zgaruvchilarini ikkalasining ham qiymatlari 5 ga teng bo'lsagina true qiymat qaytaradi. Bu ifoda agarda o'zgaruvchilardan birortasi 5 ga teng bo'lmagan qiymat qabul qilsa false qiymatini qaytaradi. Mantiqiy ko'paytirish operatori faqatgina o'zining ikkala ifodasi ham rost bo'lsagina true qiymat qaytaradi.

Mantiqiy ko'paytirish operatori && belgi orqali belgilanadi.

Mantiqiy qo'shish operatori

Mantiqiy qo'shish operatori ham ikkita ifoda orqali hisoblanadi. Agarda ulardan birortasi rost bo'lsa mantiqiy qo'shish operatori true qiymat qaytaradi. Agarda sizda pul YOKI kredit kartochkasi bo'lsa, siz schyotni to'lay olasiz. Bu holda ikkita shartning birdaniga bajarilishi: pulga ham va kredit kartochkasiga ham ega bo'lishingiz shart emas. Sizga ulardan birini bajarilishi yetarli. Bu operatorga oid yana bir misolni qaraymiz.

Masalan,

```
if(x==5)||(y==5)
```

ifodasi yoki x o'zgaruvchi qiymati, yoki u o'zgaruvchi qiymati, yoki ikkala o'zgaruvchining qiymati ham 5 ga teng bo'lsa rost qiymat qaytaradi.

Mantiqiy inkor operatori

Mantiqiy inkor operatori tekshirilayotgan ifoda yolg'on bo'lsa true qiymat qaytaradi. Agarda tekshirilayotgan ifoda rost bo'lsa inkor operatori falseqiymat qaytaradi.

Masalan,

```
(if!(x==5))
```

ifodasining qiymati, agarda x o'zgaruvchisi 5 ga teng bo'lmasa trueqiymat qaytaradi. Bu ifodani boshqacha ham yozish mumkin:

```
if(x!=5)
```

1.2. C++ DASTURLASH TILIDA MASSIVLAR.

C++ funksiya va obyektning 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 obyekt funksiyalarni qo'llashni o'rganishdir.

Massiv – bu bir toifali, chekli qiymatlarning tartiblangan to'plamidir. Massivlarga misol qilib matematika kursidan ma'lum bo'lgan vektorlar, matritsalarini ko'rsatish mumkin.

Massivlar odatda bir o'lchovli va ko'p o'lchovli turlarga bo'linadi.

Massiv bir o'lchamli deyiladi, agar uning elementiga bir indeks orqali murojat qilish mumkin bo'lsa.

C++ dasturlash tillaridagi massiv elementlar indekslari har doim noldan boshlanadi (birdan emas). Bizga char tipidagi m nomli massiv berilgan bo'lsin. Va u 3 ta elementdan tashkil topgan bo'lsin.

m[0] → -9;

m[1] → 15;

m[2] → 3;

Demak, elementga murojat qilish uchun massiv nomi va [] qavslar ichida element indeksi yoziladi.

Bu yerda birinchi element qiymati -9, ikkinchi element – 1 nomerli indeksda -15 qiymati bor ekan. Oxirgi element indeksi n-1 bo'ladi (n-massiv elementlari soni). [] qavs ichidagi indeks butun son yoki butun songa olib keluvchi ifoda bo'lmog'i lozim. Masalan:

```
int n=6, m=4;
```

```
L[n-m]=33; // L[2]=33;
```

```
Cout<<m[2]; // ekranda : 3;
```

Massiv elementlariga murojat qilish oddiy o'zgaruvchilarga murojat qilishdan biroz farq qiladi. Massiv elementiga murojat qilish indeksi orqali bo'ladi.

```
a[1] = 5; a massivning indeksi 1 bo'lgan elementi 5 qiymat o'zlashtirilsin.
```

```
cin>>a[2]; a massivning elementi 2 bo'lgan elementi kiritilsin;
```

```
cout<<a[3]; a massivning indeksi 3 bo'lgan elementi ekranga chiqarilsin;
```

Bir o'lchamli massivlarni e'lon quyidagicha bo'ladi :

```
<Toifa> <massiv_nomi> [elementlar_soni] = { boshlang'ich qiymatlar };
```

```
1)float a[5], 2) int b[6], 3) boll c[7];
```


1) a elementi haqiqiy sondan iborat bo`lgan , 4 ta elementdan tashkil topgan massiv. Indeksleri esa 0 dan 3 gacha bo`lgan sonlar.

Float a[5]					
Massiv elementlari	a [0]	a [1]	a [2]	a [3]	a [4]
Qiymati	4	11	-8	12	122

2) b elementi butun sondan iborat bo`lgan , 6 ta elementdan tashkil topgan massiv. Indeksleri esa 0 dan 5 gacha bo`lgan sonlar.

int a[6]						
Massiv elementlari	a [0]	a [1]	a [2]	a [3]	a [4]	a [5]
Qiymati	2	99	-5	28	112	54

3) c elementlari mantiqiy qiymatlardan (true, false) iborat bo`lgan 7 ta elementdan tashkil topgan massiv. Indeksleri esa 0 dan 6 gacha bo`lgan sonlardir.

Massivni e`lon qilishda uning elementlariga boshlang`ich qiymat berish mumkin va buning bir necha usuli mavjud.

1) O`lchami ko`ratilgan massivni to`liq initsializatsiyalash.

```
int k[5] = {2, 15 , -9, 45, 3 , 7};
```

Bu yerda 5 ta elementdan iborat k massivi e`lon qilingan va massivning barcha elementlariga boshlang`ich qiymat berilgan.

2) O`lchami ko`rsatilgan massivni to`liqmas to`liqmas initsializatsiyalash.

```
int k[5] = {2, 15, -9 };
```

Bu yerda 5 ta elementdan iborat bo'lgan k massivi e'lon qilingan va dastlabki 3 ta elementlariga boshlang'ich qiymat berilgan.

3) O'lchami ko'rsatilmagan massivni to'liq initsializatsiyalash.

```
int k[] = {2, 15, -9, 45, 3, 7};
```

Shuni takidlash lozimki, agar massiv o'lchami ko'rsatilmasa, uni to'liq initsializatsiyalash shart. Bu xolda massiv o'lchami kompilyatsiya jarayonida massiv elementlar soniga qarab aniqlanadi. Bu yerda massiv o'lchami 5 ga teng.

4) O'lchami ko'rsatilgan massivning barcha elementlariga boshlang'ich qiymat 0 berish

```
int k[5] = {0};
```

Masalan:

O'lchami ko'rsatilgan massivning barcha elementlariga boshlang'ich qiymat 0 berish.

```
#include<iostream.h>
```

```
int main ()
```

```
{
```

```
    int k[5]={0}; // massivning barcha elementlariga 0 qiymat berish.
```

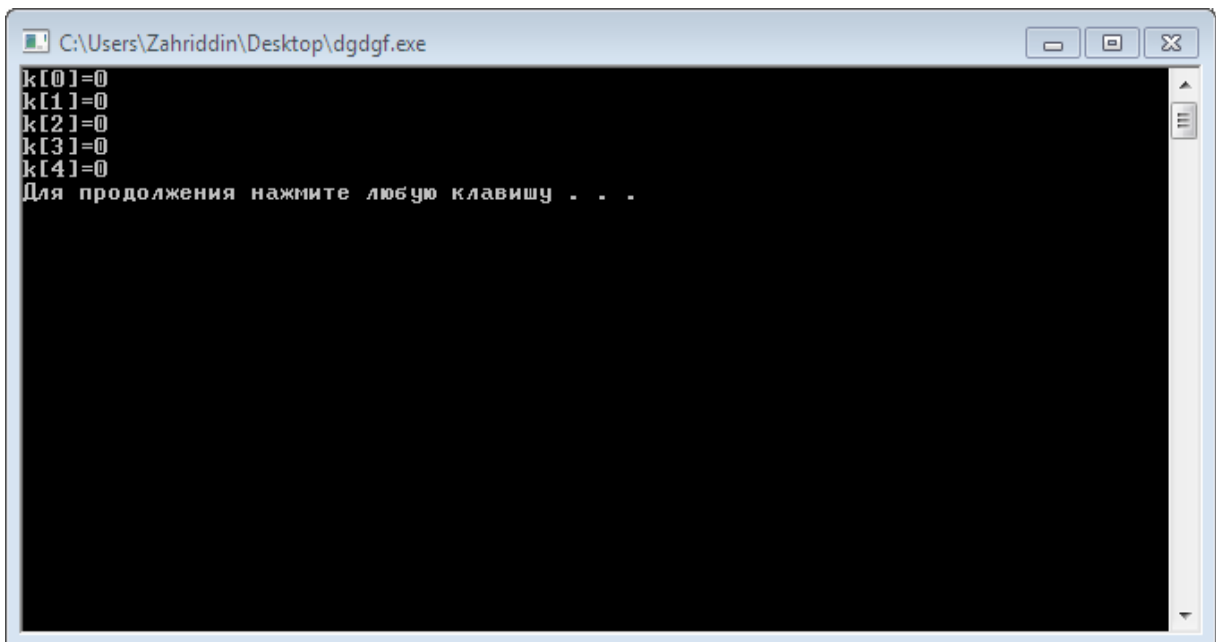
```
    for (int i=0; i<5; i++ )
```

```
        cout<<"k["<<i<<"]="<<k[i]<<endl;
```

```
    return 0;
```

```
}
```

Ekkranga quyidagicha natija chiqadi:



```
C:\Users\Zahriddin\Desktop\dgdgf.exe
k[0]=0
k[1]=0
k[2]=0
k[3]=0
k[4]=0
Для продолжения нажмите любую клавишу . . .
```

Ko`p o`lchovli statik massivlar

C++ tilida massivlar elementining turiga cheklovlar qo`yilmaydi , lekin bu turlar chekli o`lchamdagi obyektlarning turi bo`lishi kerak.

CHunki kompiyator massivning hotiradan qancha joy (bayt) egallashini xisoblay olish kerak. Xususan , massiv komponentasi massiv bo`lish mumkin

(“vektorlar - vektori”) , natijada **matritsa** deb nomlanuvchi ikki o`lchamli massiv xosil bo`ladi.

Agar matritsaning elementi xam vektor bo`lsa , uch o`lchamli massivlar - **kub** xosil bo`ladi. Shu yo`l bilan yechilayotgan masalaga bog`liq ravishda ixtiyoriy o`lchamdagi massivlarni yaratish mumkin.

Ikki o`lchamli massivda birinchi indeks satrlar sonini , ikkinchisi esa ustunlar sonini bildiradi.

Birinchi satrning dastlabki elementi a_{10} – a biri nol element deb o`qiladi .
a o`n deyilmaydi.

M ta satr n ta ustunga ega bo`lgan massivga (mxn)o`lchamli massiv deyiladi. Agar $m=n$ (satrlar va ustunlar soni teng) bo`lsa **kvadrat massiv** deyiladi.

Ikki o`lchamli massivning sintaksi quyidagi ko`rinishda bo`ladi:

`<tur><nom>[<uzunlik>][<uzunlik>]`

Masalan, 10X20 o`lchamli xaqiqiy sonlar massivning e`loni:

`Float a[10][20];`

E`lon qilingan a matritsa ko`rinishi quyidagicha ko`rinishda bo`ladi.

J

$a_{[0]}: (a_{[0][0]}, a_{[0][2]}, \dots, \dots a_{[0][18]}, a_{[0][19]},)$

$a_{[1]}: (a_{[1][0]}, a_{[1][1]}, \dots, \dots a_{[1][18]}, a_{[1][19]},)$

....

i $a_{[i]}: (\dots, \dots , \dots , \dots a_{[i][j]} \dots , \dots \dots)$

....

$a_{[9]}: (a_{[9][0]}, a_{[9][1]}, \dots, \dots a_{[9][18]}, a_{[9][19]},.)$

Ikki o`lchamli massivning hotirada joylashuvi

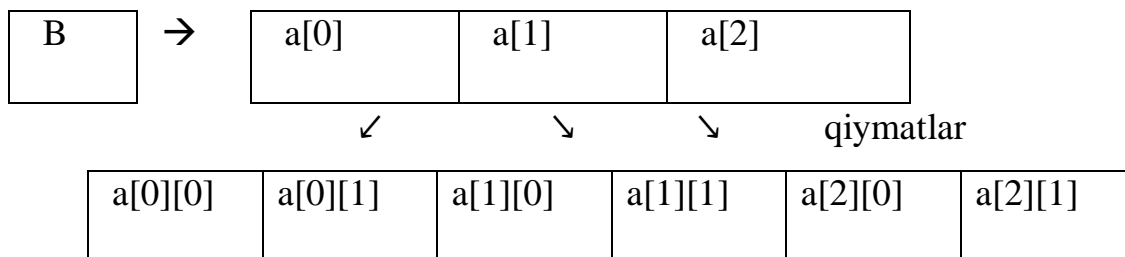
Endi adres nuqtayi - nazaridan ko`p o`lchamli massiv elementlariga murojat qilishni ko`raylik. Quyidagi elonlar berilgan bo`lsin:

Int a[3][2];

Float b[2][2][2];

Birinchi elonda ikki o'lchamli massiv, yani 2 ta satr va 3 ustundan iborat matritsa e'lon qilingan, ikkinchisida uch o'lchamli - 3 ta 2x2 matritsadan iborat bo'lgan massiv e'lon qilingan. Uning elementlariga murojat sxemasi:

Adres ko'rsatkichlar massivi

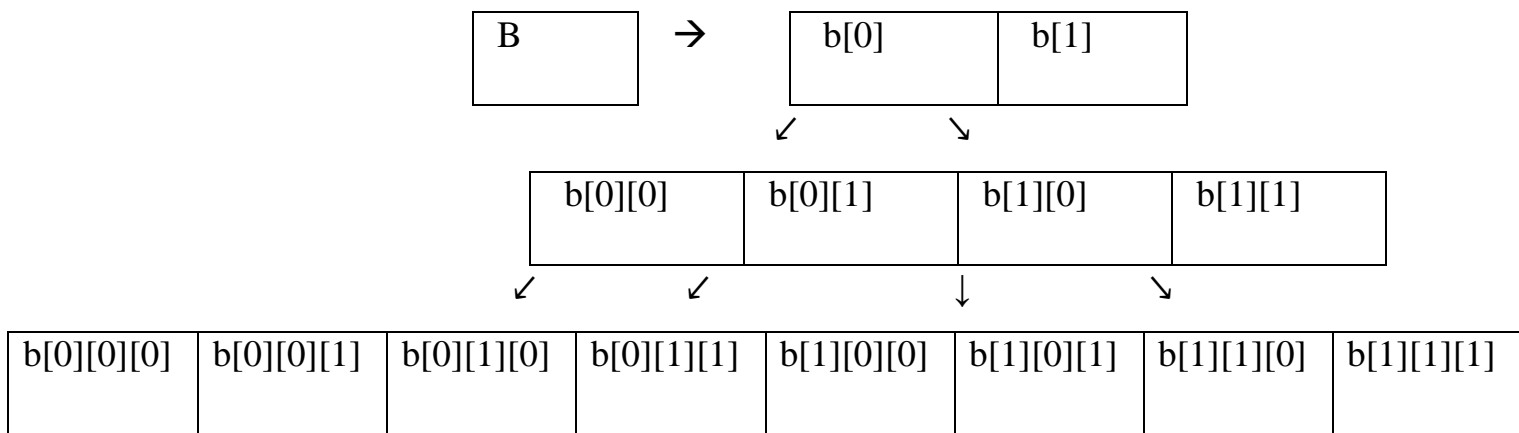


Ikki o'lchamli massiv elementlariga murojat;

Bu yerda $a[i]$ ko'rsatkichida i -chi satrning boshlang'ich adresi joylashadi, massiv elementiga $a[i][j]$ ko'rinishidagi asosiy murojatdan tashqari vositali murojat qilish mumkin: $*(*(a+i)+j)$ yoki $*(a[i]+j)$.

Uch o'lchamli massivning xotirada tashkil bo'lishi:

Adres ko'rsatkichlar massivi



Massiv elementlariga murojat qilish uchun nomdan keyin kvadrat qavsda xar bir o`lcham uchun indeks yozilishi kerak , masalan `b[i][j][k]`. Bu elementga vositali murojat xam qilish mumkin va uning variantlari:

`*(*(b+i)+j)+k` yoki `*(*(b[i]+j)+k)` yoki `*(b[i][j]+k)`;

Ko`p o`lchovli massivlarni initsializatsiyalash

```
Int a[2][3] = {2, 6, 8, 7, 12, 5};
```

```
Int b[3][3] = {{2, 6, 8}, {7, 12, 5}, {20, 21, 22}}
```

Birinchi operatorida boshlang`ich qiymatlar ketma – ket yozilgan, Ikkinchi operatorida qiymatlar guruxlangan.

1.3. C++ DASTURLASH TILIGA SATRLAR.

Satrlar bilan ishlashda unda birorta belgini izlash uchun «string.h» kutubxonasida bir qator standart funksiyalar mavjud.

Birorta belgini berilgan satrda bor yoki yo`qligini aniqlab beruvchi `strchr()` funksiyasining prototipi

```
char* strchr(const char* string, int c);
```

ko`rinishida bo`lib, u `s` belginining `string` satrida izlaydi. Agar izlash muvofaqiyatli bo`lsa, funksiya shu belgining satrdagi o`rnini (adresini) funksiya natijasi sifatida qaytaradi, aks holda, ya'ni belgi satrda uchramasa funksiya `NULL` qiymatini qaytaradi. Belgini izlash satr boshidan boshlanadi.

Quyida keltirilgan programma bo`lagi belgini satrdan izlash bilan bog`liq.

```
char satr[]="0123456789";
```

```
char* pSatr;
```

```
pSatr=strchr(satr,'6');
```

Programma ishlashi natijasida `pSatr` ko`rsatkichi `satr` satrining '6' belgisi joylashgan o`rni adresini ko`rsatadi.

`strchr()` funksiyasi berilgan belgini (`s`) berilgan satr (`string`) oxiridan boshlab izlaydi. Agar izlash muvaffaqiyatli bo`lsa, belgini satrga oxirgi kirishining o`rnini qaytaradi, aks holda `NULL`.

Misol uchun

```
char satr[]="0123456789101112";
```

```
char* pSatr;
```

```
pSatr=strchr(satr,'0');
```

ammallarini bajarilishida pSatr ko'rsatkichi satr satrining "01112" satr qismining boshlanishiga ko'rsatadi.

strspn() funksiyasi ikkita satrni belgilarni solishtiradi. Funksiya quyidagi

```
size_t strspn(const char* str1, const char* str2);
```

ko'rinishga ega bo'lib, u str1 satrdagi str2 satrga kiruvchi birorta belgini izlaydi va agar bunday element topilsa, uning indeksi funksiya qiymati sifatida qaytariladi, aks holda funksiya satr uzunligidan bitta ortiq qiymatni qaytaradi.

Misol:

```
char satr1[]="0123ab6789012345678";
```

```
char satr2[]="a32156789012345678";
```

```
int farqli_belgi;
```

```
farqli_belgi=strspn(satr1,satr2);
```

```
cout<<"Satr1 satridagi Satr2 satrga kirmaydigan\birinchi belgi indeksi =  
"<<farqli_belgi;
```

```
cout<<"va u "<<satr1[farqli_belgi]<<" belgisi.";
```

amallar bajarilishi natijasida ekranga

```
Satrlardagi mos tushmagan belgi indeksi = 5
```

satri chop etiladi.

strcspn() funksiyasining prototipi

```
size_t strcspn(const char* str1, const char* str2);
```

ko'rinishida bo'lib, u str1 va str2 satrlarni solishtiradi va str1 satrining str2 satriga kirgan birinchi belgani indeksini qaytaradi.

Masalan.

```
char satr[]="Birinchi satr";
```

```
int index;
```

```
index=strcspn(satr,"sanoq tizimi");
```

amallari bajarilgandan keyin index o'zgaruvchisi 1 qiymatini qabul qiladi, chunki birinchi satrning birinchi o'rindagi belgisi ikkinchi satrda uchraydi.

strpbrk() funksiyasining prototipi

```
char strpbrk(const char* str1, const char* str2);
```

ko'rinishga ega bo'lib, str1 satrdagi str2 satrga kiruvchi birorta belgini izlaydi va agar bunday element topilsa, uning adresi funksiya qiymati sifatida qaytariladi, aks holda funksiya NULL qiymati qaytaradi. Quyidagi misol funksiyani qanday ishlashini ko'rsatadi.

```
char satr1[]="0123456789ABCDEF";
```

```
char satr2[]="ZXYabcdefABC";
```

```
char* element;
```

```
element = strpbrk(satr1,satr2);
```

```
cout<<element<<'\n';
```

Programma ishlashi natijasida ekranga str1 satrining

ABCDEF

satr ostisi chop etiladi.

C++ da satrlar bilan ishlashni qulaylashtirish uchun string sinfi kiritilgan. string sinfi satrlarida satr oxirini '\0' belgisi belgilamaydi.

Standart kutubxonadagi string sinfidan foydalanish uchun <string> sarlavha faylini dasturga qo'shish kerak.

C++ tilida kompilyatorlarda <cstring.h> yoki <bstring.h> sarlavha faylini qo'shish kerak bo'ladi. Oddiy usuldagi satrlar bilan ishlash uchun esa, <string.h> sarlavha fayli qo'shiladi.

Eng afzali, o'zingiz ishlatayotgan kompilyator bilan yaxshilab tanishib chiqing. Satrlar bilan ishlovchi asosiy funksiyalar bilan tanishib chiqamiz.

Satr xususiyatlarini aniqlash uchun quyidagi funksiyalardan foydalanish mumkin:

```
unsigned int size() const; // satr o'lchami
```

```
unsigned int length() const; // satr elementlar soni
```



```
unsigned int max_size() const; // satrning maksimal uzunligi
```

```
unsigned int capacity() const; // satr egallagan xotira hajmi
```

```
bool empty() const; // satrning bo'shligini aniqlash
```

Satrning uzunligini aniqlash uchun `length()` yoki `size()` funksiyalaridan foydalanish mumkin.

Misol: Satr uzunligini aniqlash dasturi

```
#include <iostream.h>
```

```
#include <string>
```

```
int main()
```

```
{
```

```
string s;
```

```
cout << "Satr kiriting" << endl;
```

```
getline(cin, s);
```

```
cout << "Siz kiritgan satr " << s.length() << " ta belgidan iborat";
```

```
cout << "Siz kiritgan satr " << s.size() << " ta belgidan iborat";
```

```
system ("pause");
```

```
return 0;
```

```
}
```

Belgilar. Belgili o'zgaruvchilar odatda bir bayt joyini egallaydi va bu 256 xil belgini saqlash uchun etarlidir. Char tipi qiymatlarini 0..255 sonlar to'plamiga yoki ASCII belgilar to'plamiga interpretatsiya qilish mumkin. `<string.h>` C uslubida satrlar bilan ishlovchi funksiyalar e'loni berilgan.

Maxsus belgilar.

C++ kompilyatori tekstlarni formatlovchi bir nechta maxsus belgilardan tashkil topgan. (Ulardan eng ko‘p tarqalgani 3.2. - jadvalda keltirilgan). Bu belgilarni dasturda ishlatishda «teskari slesh»dan foydalanamiz. Teskari sleshdan keyin boshqaruvchi belgi yoziladi. Masalan, tabulyasiya belgini dasturga qo‘yish uchun quyidagicha yozuvni yozish kerak.

```
Char tab = '\t';
```

Bu misoldagi char tipidagi o‘zgaruvchi \t qiymatini qabul qiladi. Maxsus belgilar axborotlarni ekranga, faylga va boshqa chiqarish qurilmalariga chiqarishda formatlash uchun qo‘llaniladi.

C++ da belgili ma’lumotlar uchun char turi qabul qilingan. Belgili axborotni taqdim etishda belgilar, simvulli o‘zgaruvchilar va matniy konstantalar qabul qilingan. C++ da satrlar (string) qo‘shirnoqlar (") orasida bo‘ladi. Bitta harfli literalar esa bitta tirnoq - apostrof (‘) ichiga olinadi. Misol uchun: 'A', '\$'. Bitta harf yoki belgini qo‘shirnoq ichiga olsa u satr kabi qabul qilinadi.

Misollar:

```
sonst char c='c'; //belgi - bir baytni egallaydi, uning qiymati o‘zgarmaydi.
```

```
char a,b; //belgili o‘zgaruvchilar, bir baytdan joy egallaydi, qiymatlari o‘zgaradi.
```

```
const char *s= "\n satrining misoli"; //matniy konstanta
```

C++ dagi satr - bu nul-belgi - '\0' (nul-terminator) - bilan tugallanuvchi belgilar massivi. Nul-terminatorning holatiga qarab satrning amaldagi uzunligi aniqlanadi. Bunday massivdagi elementlar soni, satr tasviriga qaraganda, bittaga ko‘p.

Qiymat berish operatori yordamida satrga qiymat berish mumkin emas. Satrni massivga yoki kiritish paytida yoki nomlantirish yordamida joylashtirish mumkin.

Misol:

```
void main()
```

```

{
    char s1[10]="string1";
    int k=sizeof (s1);
    cout<<s1<<"\t"<<k<<endl;
    char s2[]="string2";
    k=sizeof(s2);
    cout<<s2<<"\t"<<k<<endl;
    char s3[]={ 's', 't', 'r', 'i', 'n', 'g', '3' };
    k=sizeof(s3);
    cout<<s3<<"\t"<<k<<endl;
    char *s4="string4";//satr ko'rsatkichi, uni o'zgartirib bo'lmaydi
    k=sizeof(s4);
    cout<<s4<<"\t"<<k<<endl;
}

```

Natijalar:

string1 10 - 10 bayt ajratilgan, shu jumladan \0 ga

string2 8 - 8 bayt ajratilgan (7+1 bayt /0 ga)

string3 8 - 8 bayt ajratilgan (7+1 bayt /0 ga)

string4 4 - ko'rsatkichning o'lchamlari

Satrlar, yani harflar ketma-ketligi ("Toshkent", "Yangi yilingiz bilan!"...)

C/C++ da char tipidagi massivlar yordamida beriladi. Bunday satrlar bilan islovlar juda tez bajariladi. Chunki ortiqcha tekshirishlar bajarilmaydi. Bundan tashqari C++ da ancha rivojlangan String klasi mavjuddir, u oddiy char bilan berilgan satrlardan ko'ra qulayroqdir. Lekin ushbu klas ko'proq joy egallaydi va massivli satrlardan ko'ra sekinroq ishlaydi. String klasini keyingi qismlarda o'tamiz. Qolaversa, satrlar bilan ishlash uchun biz o'zimiz klas yoki struktura yozishimiz mumkin. C dan meros bo'lib qolgan satrlar ustida amallar bajarish uchun biz dasturimizga **<string.h>** (yangi ismi **<cstring>**) e'lon faylini kiritishimiz kerak. Ushbu e'lon faylida berilgan funksiyalar bilan keyingi bo'limda ishlaymiz. Harflar, yani literalar, aytib o'tganimizdek, C++ da char tipi orqali beriladi. Literalar

apostroflarga ('S', '*' ...) olinadi. Satrlar esa qo'shtirnoqlarga olinadi. Satrlar e'loniga misol beraylik.

```
char string[] = "Malibu";
```

```
char *p = "Ikkinchi!?!";
```

Satrlarni yuqoridagi ikkita yo'l bilan initsalizatsiya qilsa bo'ladi. Satrlar ikkinchi uslubda e'lon qilinganda, yani pointer mehanizmi qo'llanilganda, ba'zi bir kompilyatorlar satrlarni hotiraning konstantalar saqlanadigan qismiga qo'yadi. YAni ushbu satrlarga o'zgartirish kiritilishi ta'qiqlanadi. SHu sababli satrlarni hardoim o'zgartirish imkoni bo'lishi uchun ularni char tipidagi massivlar ko'rinishida e'lon qilish afzaldir. Satrlar massiv yordamida berilganda, ularning uzunligi noma'lumdir. SHu sababli satrning tugaganligini bildirish uchun satr ohiriga mahsus belgi nol literasi qo'yiladi. Uning dastursa belgilanishi '\0' ko'rinishga ega. Demak, yuqorida berilgan "Malibu" satriga ohiriga '\0' belgisi qo'shiladi, yani massivning umumiy uzunligi "Malibu":6 + '\0':1 = 7 = 7 ta char tipidagi elementga teng bo'ladi. Satrlarni massiv initsalizatsiya ro'yhati ko'rinishida ham bersak bo'ladi:

```
char c[6] = {'A', 'B', 'C', 'D', 'E', '\0'};
```

```
...
```

```
cout << c;
```

```
...
```

Ekranda:

```
ABCDE
```

Xulosa.

Bitiruv malakaviy ishning I-bobi 3 ta bo'limdan iborat bo'lib, bunda statik ma'lumotlar tuzilmalari bo'yicha ma'lumotlar keltirilgan. Ma'lumotlar C++ dasturlash tilida bo'lganligi uchun. I-bobning 1- bo'limida C++ dasturlash tilining asosiy operatorlari keltirilgan. I-bobning 2- bo'limida C++ dasturlash tilida massivlar va ular ustida bajariladigan amallar misollar bilan tushuntirib berilgan.

I-bobning 3- bo'limida C++ dasturlash tilida satr tipi va ular ustida bajariladiga funksiyalar batafsil yoritib berilgan.

II BOB. STATIK MA'LUMOTLAR TUZILMASI BO'YICHA INTERAKTIV O'QUV MATERIALLAR TAYYORLASH.

2.1. Interaktiv o'quv materiallarini tayyorlashga foydalanilgan dasturlar .

Flash texnologiyasiga - ShockWave Flash (SWF) formatli vektorli grafikdan foydalanishga asoslangan texnologiyadir. Bu format eng samarali grafik formatlardan bo'lmisada, SWF formati foydalanuvchilarga grafik imkoniyatlari cheklanmagan grafiklar bilan ishlovchi vositalar va natijani Web- brouzerlarda, kerakli muxarrirlarda foydalanish imkoniyatlari mavjud. Flash texnologiyasining imkoniyatlardan yana biri bu uning moslashuvchanligidir, ya'ni bu format barcha platformalarda (MacOS tizimli Macintosh kompyuterlari yoki Windows tizimli

kompyuterlarida) ishlatilishi mumkin. Yana bir qulay imkoniyati uning yordamida yaratilgan tasvirlar nafaqat animatsiyali bo`lishi, balki interfaol elementlar va tovush bilan boyitilishi hamda dasturlash orqali boshqarilishi mumkin.

Flash texnologiyasining mosalashuvchanlik va interfaol multimediya dasturlar yaratish imkoniyati ko`pchilik Web-dizaynerlar o`rtasidagi bahslarga sabab bo`lib, uni mashhurligini oshishiga imkoniyat berdi. Shuning uchun bu texnologiyaning yaratilishi bilan bir vaqtda Macromedia kompaniyasi tomonidan ikki asosiy web-brouzerlari, Internet Explorer va Netscape Communicatorlar uchun elementlar Plug-In yaratildi. Bu esa, o`z navbatida Flash texnologiyasini Internetda yana ham keng tarqalishiga olib keldi. Natijada ushbu web-brouzerlar yaratuvchilari swf formatini o`z dasturlarini asosiy formatlar bazasiga qabul qildi. Bunday usulni boshqa yirik dasturiy ta`minot yaratuvchilar (masalan, Adobe firmasi) ham qo`llay boshladi. Macromedia kompaniyasi swf formatini juda oddiy va qulay uskunalar bilan ta`minlaganligi bu formatdan ko`p muxlislarning foydalanishiga olib keldi. Shuni aytish kerakki, hozirgi vaqtda ushbu uskunalarni bir qancha to`liq to`plamlari ham mavjud. Ushbu uskunalarni bir turi Macromedia Director Shockwave Studio - multimediya taqdimotlarni yaratish, Macromedia FreeHand va Macromedia Fireworks - grafik tasvirlar muharriri, Macromedia Authorware va Macromedia CourseBuilder - interfaol o`rgatuvchi kurslarni yaratish muharriri va boshqalarni misol qilib olish mumkin. Web-sahifa yaratuvchilar orasida eng ko`p ishlatiladigani bu Macromedia Flash dasturidir, chunki ushbu dastur ixtiyoriy Web-sahifaga mashhurlik olib keluvchi banner va animatsiya, interfaol lavhalar yaratish imkonini beardi. Balki shuning uchundir swf formatini oddiy qilib Flash deb atalish odatga kirib kolgan.

Flash texnologiyalar tarkibining elementlari:

vektorli grafika;

animatsiyani bir qancha usullarda ishlash;

interfeysda interfaol elementlarini yaratish;

sinxron ovoz qo`shish;

HTML formati va boshqa internetda foydalaniladigan barcha formatlarga o'tkazishni ta'minlash; mustaqil platformali;

Flash-roliklarni avto rejimda ham, Web - browser yordamida ham ko'rish imkoniyati mavjud;

vizual uskunalari mavjudligi Flash-rolik yaratuvchilarini ko'plab murakkab amallardan xalos etadi, shuningdek Flash-texnologiyalarning texnik asnektlarni o'rganishni talab etmaydi.

Flash dasturida ishlashni o'rganish uchun avvalo uning interfeysi bilan tanishish

lozim. Agar Macromedia firmasi tomonidan yaratilgan biror-bir dasturiy ta'minotlar

(masalan, Dreamweaver muxarriri) bilan tanish bo'lsangiz Flashni interfeysini o'zlashtirish unchalik murakkab bo'lmaydi. Agar Macromedia tomonidan yaratilgan

uskunalarni o'zlashtirishni aynan Flash dasturidan boshlasangiz, u holda uning menyulari, uskunalari va ob'ekt xususiyatlar paneli va boshqalarning ko'rinishi Windowsning amaliy dasturlarinikidan fraq qiladi. Masalan, aksariyat Windowsdagi amaliy dasturlarda (ofis majmuasi) uskunalari paneli tarkibini boshqarish buyruqlari

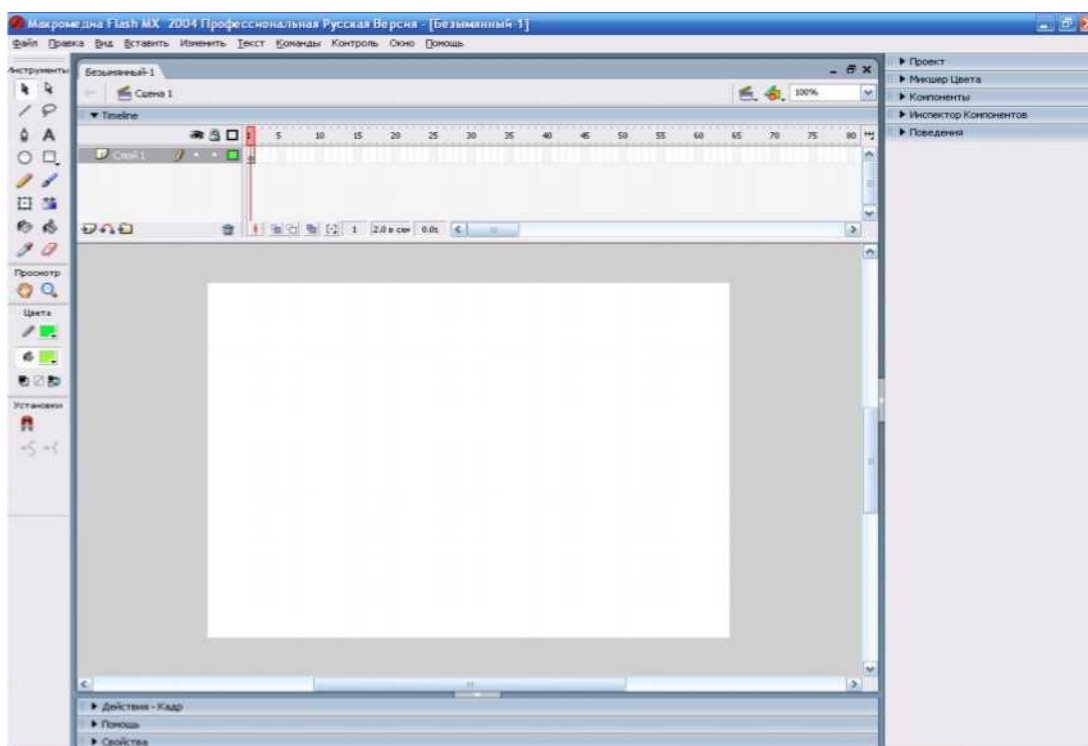
“View” menyusiga kiradi. Flash dasturida esa bu buyruqlar “Window” menyusida joylashtirilgan. Tahrirlanayotgan obyekt xususiyatlar panelining ko'rinishi ham Flash dasturiga xosdir. Lekin bu turdagi yangiliklar unchalik ham qiyinlik tug'dirmaydi.

Muharrirning foydalanuvchi interfeysini tashkil etilishi umumiy holda Windows dasturlar kabi standart ko'rinishga egadir, oynaning yuqori qismida sarlavha, menyular to'plami, asosiy buyruqlar joylashtirilgan uskunalari paneli va boshqalar. Muharrirda yangi vaqt diagrammaning ko'rinishi mavjud.

Muxarrirni ilk bor ishga tushirganda asosiy oyna ustidan qo'shimcha muloqot oynasi naydo bo'ladi. Bu oyna sizga ish boshlashni o'zingiz xohishingiz bo'yicha sozlash imkonini beradi. Bu muloqot oynasida 3 ta Open a Recent Item

(oxirgi flash faylni ochish), Create NEW (yangi fayl yaratish), Create form Template (shablondan foydalanib fayl yaratish) bo`limlari bor. Bu muloqot oynasidan keraklisini tanlab olish mumkin. Aksiyat holatlarda Create NEW -> Flash document bandi tanlanadi vaekranga Flash dasturining asosiy oynasi hosil bo`ladi.

Flash dasturining asosiy oynasi menyular, uskunalar, obekt xususiyatlar va qo'shimchalar, ishchi maydon kabi 4 ta asosiy sohalardan iborat.



(2.1.1)-chizma. Flash dasturining asosiy interfeysi

Flash dasturining menyusi 10 bo'limdan iborat bo'lib, ular quyidagi jadvalga keltirilgan vazifalarni bajaradi.

Flash dasturining menyu bo'limlarining vazifalari

1-jadval

T.r.	Nomi	Vazifasi
1	Файл	Yangi Flash hujjat yaratish, ochish, yopish, xotiraga olish usullari, tasvirlarni import va eskport usullari, publikatsiya,

2	Правка	Flash hujjatdagi tahrirlash ishlarini amalga oshirish (bekor qilish, qaytarish, nusxalash, qirqish, qo'yish usullari,
3	Вид	Flash hujjatning ko'rinishini o'zgartirish amallarini bajaradi (masalan, o'lchamini kattalashtirish yoki kichiklashtirish,
4	Вставить	Flash hujjatga qo'yiladigan obyektlar qo'yish (yangi belgi - rolik, tugma, tasvir va boshqalari)
5	Изменить	Flash hujjatdagi obyektlarni tahrirlash, o'zgartirish amallarini bajaradi.
6	Текст	Flash hujjatdagi matnlarning yozuvi, o'lchami, stili va boshqa amallarni bajarish uchun mo'ljallangan.
7	Команды	Flash hujjat uchun turli buyruqlarni bajartirish va natijalar olish uchun mo'ljallangan.
8	Контроль	Flash hujjatni boshqarish amallari uchun mo'ljallangan.
9	Окно	Flash interfeysining ko'rinishi ustida amallarni bajarish uchun mo'ljallangan.
10	Помощь	Flash hujjat bo'yicha yordam tizimi.

Flashning uskunalar paneli Flash dasturining uskunalar paneli tasvirlarni yaratish va tahrirlash kabi amallarni bajarishga yordam beradi. Uning tarkibiga quyidagi jadvalda keltirilgan uskunalar kiradi. Tahrirlash uskunalar paneli asosiy ekranni chan tomonida joylashadi. Unda asosan grafik obyektlar yaratish va tahrirlash uchun mo'ljallangan uskunalar joylashtirilgan. Bu uskunalarning ko'chiligi foydalanuvchiga grafik muxarrirlari (masalan, Paint) bilan ishlash jarayonidan tanishdir. Foydalanuvchiga qulaylik tug'dirish uchun uskunalar paneli to'rt qismga bo'lingan.

Tools (Инструменты) – bu qismda aniq uskunalar joylashtirilgan. Bu uskunalarni tanlash va chizish uskunalar tashkil etadi.

View (Просмотр) – ish stolidagi tasvirni ko`rishni boshqarish vositalari joylashtirilgan. View ikki uskunadan iborat: Hand Tool- bu uskuna tanlanganda maxsus rejim yoqiladi, bu rejimda sichqoncha yordamida ish maydonini turli yo`nalish bo`yicha siljitish mumkin; Zoom Tool – bu uskuna tanlanganda ish maydonidagi tasvirni tez masshtablash rejimi yoqiladi. Bu rejim yoqilganda Options maydonida ikki qo`shimcha uskunasi naydo bo`ladi, bu uskunalar masshtablash yo`nalishini (kattalashtirish yoki kichiklashtirish) o`zgartirish imkonini beradi.

Colors (Цвета) – bu erdagi uskunalar chegara va maydon asosining rangini bir-biridan alohida o`zgartirish imkonini beradi.

Options (Установки) – tanlangan uskuna uchun qo`shimcha parametrlarni o`rnatish elementlari joylashtirilgan. Qo`shimcha parametrlari bo`lmagan uskunalar uchun Options maydoni bo`sh qoladi.

Flashni avvalgi variantlariga nisbatan, chizish uskunalarini ayrim parametrlarini o`rnatish xususiyatlar inspektori paneli yordamida ham amalga oshirish mumkin. Har bir uskunani sozlashni o`ziga xos tomonlari keyingi mavzularda ko`rib o`tiladi. Agar zaruriyat tug`ilsa uskunalar panelini oynadan «uzish» va istalgan joyga joylashtirish mumkin.

Kompyuter texnologiyasining rivojlanishi hozirgi zamonda ixcham va bejirim, foydalanuvchi uchun qulay bo`lgan harakatli dasturlar yaratilish imkoniyatini yaratmoqda. Bu yaratilgan dasturlar Netscape Navigatir (NN) bilan birga ishlashdan tashqari, Internet Explorer (IE) bilan ham ishlaydi. Micromedia kompaniyasi tomonidan shunday programmalardan biri Flash paket programmasi yaratilgan bo`lib, bu programma texnik WEB-dizayn vositalarining to`liq imkoniyatidan foydalanish imkoniyatini beradi.

Bu programmaning imkoniyatlari juda keng bo`lib, bunda harakatlar va tovushlar 100 kb fayl hajmnigina egallaydi.

Flash quyidagi imkoniyatlarga ega:

- Yaratilayotgan faylni hajmi kichikligi va Flash dasturining tarmoqdan tez yuklanuvchanligi. Flash da vektorli format qo'llanganligi sababli, unda fayllar siqiladi va shuning uchun fayl hajmi kamayadi;

- Brauzerlar orasidagi bog'liqlik, ya'ni Flash IE, NN lar bilan birga ishlaydi;

- Boshqaruv tilining qudratliligi. Micromedia Flashda maxsus dasturlash tili ishlatililib, bunda bajaruvchi o'z saxifasi uchun qulay imkoniyatlardan foydalanishi mumkin, ya'ni massivlar, takrorlanish, formulalar va shartlardan to'liq foydalanish mumkin;

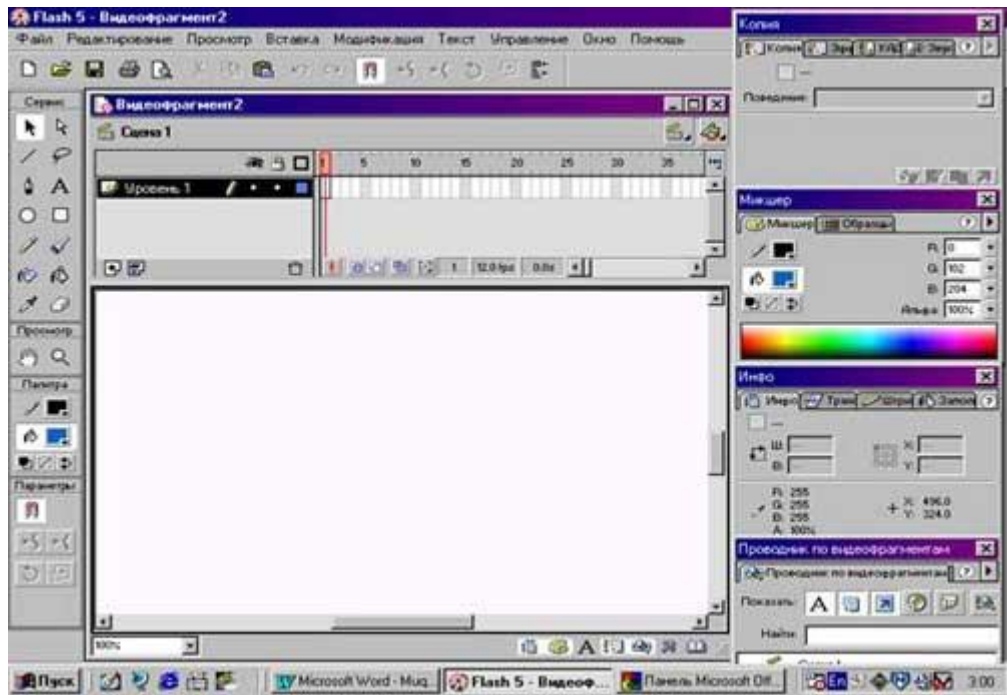
- Go'zalligi. Flashda oddiy shar yoki ixtiyoriy shakl ham juda chiroyli ranglar bilan tasvirlanishi mumkin.

- Qulayliligi. Flashdan oddiy rasm chizishni bilgan har qanday o'quvchi foydalanishi mumkin;

- Bajaruvchilarning ko'pligi. Agar foydalanuv-chiga grafikli, tovushli va kichik hajmli fayllar kerak bo'lsa, unda Flashning tengi yo'q. Flash dasturi Windows 95/98/NT/2000 uchun ishlaydi.

- Flash interfeysi juda qulay va onson.

Micromedia kompaniyasi tomonidan Flashning bir qancha versiyalari yaratildi. Biz Flash 5 misolida Flashda shlash va uning bir qancha imkoniyatlari haqida gapirib o'tmoqchimiz. Micromedia kompaniyasi Flashni yanada takomillashib uning yangi imkoniyatlarini yaratib berishmoqda.



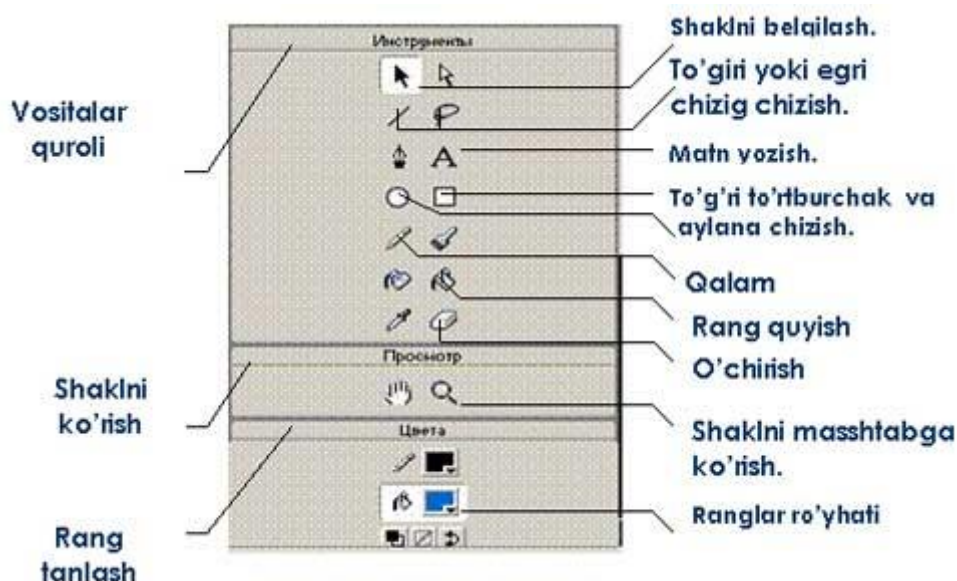
(2.1.2-чизма) **Micromedia Flash 5 oynasi.**

Windows operitsion tizimida birir programma o'rnatgan foydalanuvchi uchun, Flash dasturini o'rnatish qiyinchilik yaratmaydi. Flash dasturi ishga tushirilgach quyidagi oyna paydo bo'ladi (2.1.2- chizma). **FLASH 5** oynasining umumiy ko'rinishi. (2.1.2-чизма) da Flash 5 oynasining umumiy ko'rinishi keltirilgan. Flash interfeysi Adobe dasturining interfeysiga juda o'xshashdir. Flash bilan islaganda foydalanuvchi uning qulayligi va soddaligidan Micromedia firmasining bu dasturga juda ko'p mehnati singanligini bilishi mumkin.

Flash 5 oynasi quyidagi ko'rinishga ega:

- Oynaning chap qismida vositalar paneli joylashgan. (2.1.3-чизма) Vositalar paneli yordamida kerakli vosita quroli olinib, ishchi muxitini boshqarish va kerakli ranglarni tanlash mumkin;
- Oynaning o'ng tomonida esa, kerakli panel, ranglar, ob'ektni sozlash va dialog oynasi joylashgan;
- Oynaning o'rta qismida ishchi muxiti bo'lib, unda bajaruvchi o'z ijodini yaratishi mumkin;
- Panelning yuqori qismida esa vaqtlar shkalasi joylashgan (Time line). (2.1.4-чизма).

Flashda chizish juda onson. Buning uchun vositalar panelidagi qurollardan foydalanish kerak. (2.1.3-chizma)da vositalar panelining umumiy ko'rinishi keltirilgan.



(2.1.3-chizma). Vositalar paneli.

Flash fayllarini film (Movie) yoki «multik» deb atash mumkin. Chunki Flash da vaqt shkalasi mavjud bo'lib, undagi cheksiz imkoniyatlardan foydalanish mumkin va bu imkoniyatlar albatta vaqt bilan bog'liq bo'ladi. Bajaruvchi yaratilgan multikni ishga tushirishi, to'xtatishi, bir kadrda ikkinchi kadrda o'tishi, boshqa multiklarni ishga tushirishi va hakazolarni bajarishi mumkin.

Flashda fayl quyidagicha yaratiladi. Flashda yangi yaratilgan fayllar **.fla** kengaytmali bo'lib, bu fayllarni translatsiya qilishi natijasida yakuniy **.shf** fayli hosil bo'ladi. Shundan so'ng uni brauzerda ko'rish mumkin. Undan tashqari Flashda yaratilgan faylni bajariluvchi **.exe** fayli ko'rinishida ham tashkil qilish mumkin. Buning uchun yaratilgan Flashning faylini jack-kodi bilan generatsiya qilish kerak. Undan tashqari Flash faylini **GIF** tasviri ko'rinishida ham saqlash mumkin.

Masalan: Eng oddiy multik qilish uchun biz «**Овал**» (O) va «**Выделение**» () dan foydalanishimiz mumkin. Buning uchun quyidagi harakatlar ketma-ketligini bajarilish kerak: «**Овал**» olinib, oynaning chap qismida aylana yoki oval

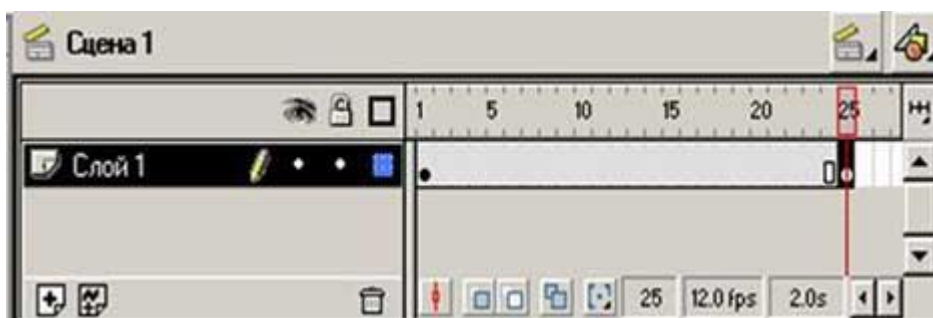
chiziladi;

«**Выделение**» tanlanib, chizilgan shakl chegarasi bilan ajratiladi. Buning uchun shaklni to'g'ri to'rtburchak ichiga olinadi. Bunda shakl () yordamida to'g'ri to'rtburchak ichiga olinib ajratiladi (sichqonchanning chap tugmasi bosib turiladi) yoki shaklga sichqoncha ko'rsatkichi keltirilib, chap tugma ikki marta bosiladi. Shaklning hamma qismi ajratiladi; INSERT (**вставка**) menyusidagi CONVERT to symbol (**Конвертировать в символ**) belgilanib, dialog oynasidagi grafik tanlanadi va **ОК (Да)** tugmasi bosiladi (yoki F8 tugma bosiladi);



(2.1.4- chizma). Belgi (**СИМВОЛ**) turini tanlash oynasi

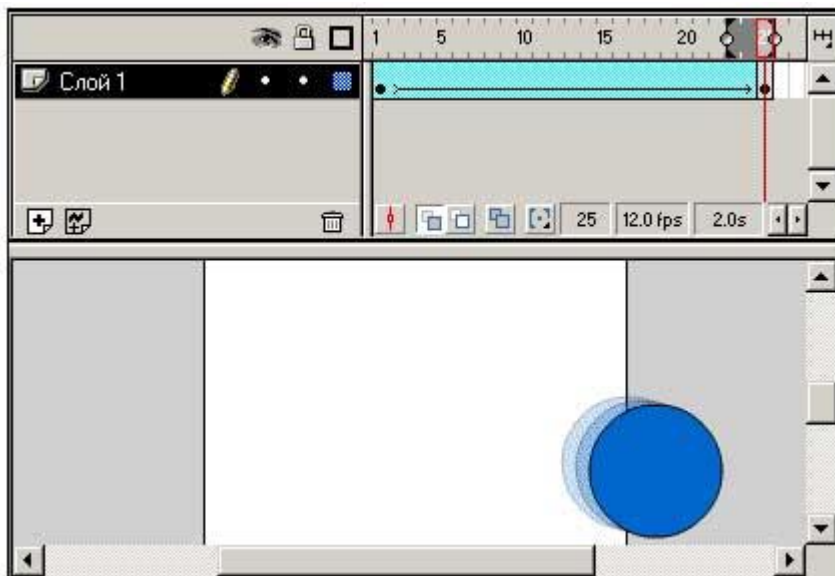
Vaqt shkalasidagi biror kadr (masalan 25-kadr) tanlanib (2.1.4-chizma), (shu kadrda sichqoncha tugmasi bosiladi) menyusidagi INSERT@keyframe (**Ключевой фрейм**) bosiladi (yoki F6 tugma bosiladi). Bu bilan biz 25-kadrda asosiy kadrni (**Ключевой**) hosil qildik; Shakl ajratiladi, (shakl atrofida havo rang ramka paydo bo'lishi kerak) va uni ishchi oynasining o'ng tomoniga surib o'tkaziladi;



(2.1.5-chizma). 25-kadrda (**ключевой кадр**) asosiy kadrni hosil qilish.

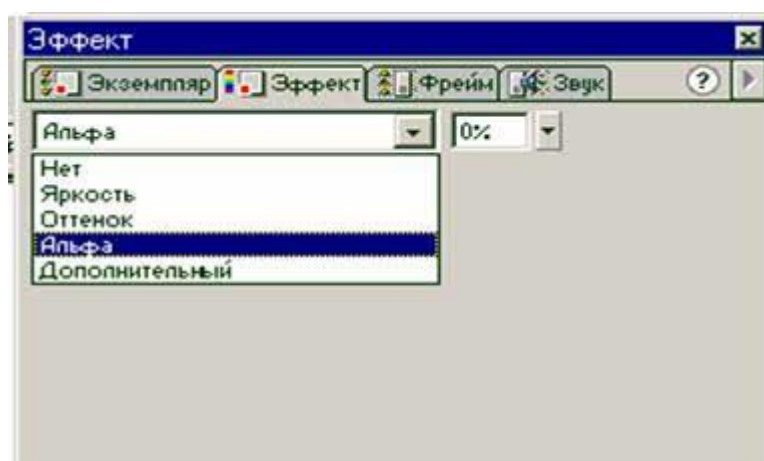
Birinchi kadrqa qaytiladi (shaklni yana chap tomonda ko'rishi kerak). INSERT menyusidan (yoki sichqonchanning o'ng tugmasi bosilganda paydo bo'ladigan

kontekst menyusidan) Creute montion tween (**Создание движения**) tanlanadi va Enter bosiladi. Mana oddiy multik hosil bo'ldi. (2.1.6-chizma).



(2.1.6-chizma). **Aylananing haraktlanish multigi.**

Multikka yana bir element qo'shish mumkin. Buning uchun 25-kadr tanlanib, shakl ajratiladi. EFFECT dialogiga kiriladi. Agar dialogda kerakli buyrug'ini topish qiyin bo'lsa menudagi Windows®Panels®Effect ning o'ng tomonidan tanlash mumkin. Bu dialogdagi Alfa (**Прозрачность**) parametrini 0% ga tushuring.



(2.1.7-chizma). Alfa (**прозрачность**) ga keltirish.

Hosil qilingan filmni boshqattan ishlatish. Flash da faqat harakat uchun asosiy nuqtalarni berish kerak, oraliq kadrlardagi harakatlarni programmaning o'zi bajaradi. Ammo kadr oralig'idagi harakatlarni ham hosil qilish mumkin. Oxirgi qadam bu .swf fayliga translyatsiya qilinib, NTML fayli bilan generatsiya qilishdir. Buni File®Publish (**Файл публикация**) yoki Shift+F12 tugmani bosish bilan hosil qilinadi. Fayl brauzerga yuklanadi. Faylni ochish uchun File®Publish Preview yoki F12 tugmani bosish kerak.

Macromedia Flashda harakatlanish.(анимация)

Harakatlanish bu kadrlarning bir-biri bilan almashinuvi natijasida hosil bo'ladigan harakatlardir. Macromedia Flashda harakatni hosil qilishning ikki xil usuli mavjud:

1. Har bir kadrni bajaruvchining o'zi yaratadi. Flash esa yaratilgan sahifalarni tez-tez varaqlash uchun xizmat qiladi;
2. Bu usulda oraliq kadrlarni Flash yordamida avtomatik tarzda o'qishni tashkil qilish.

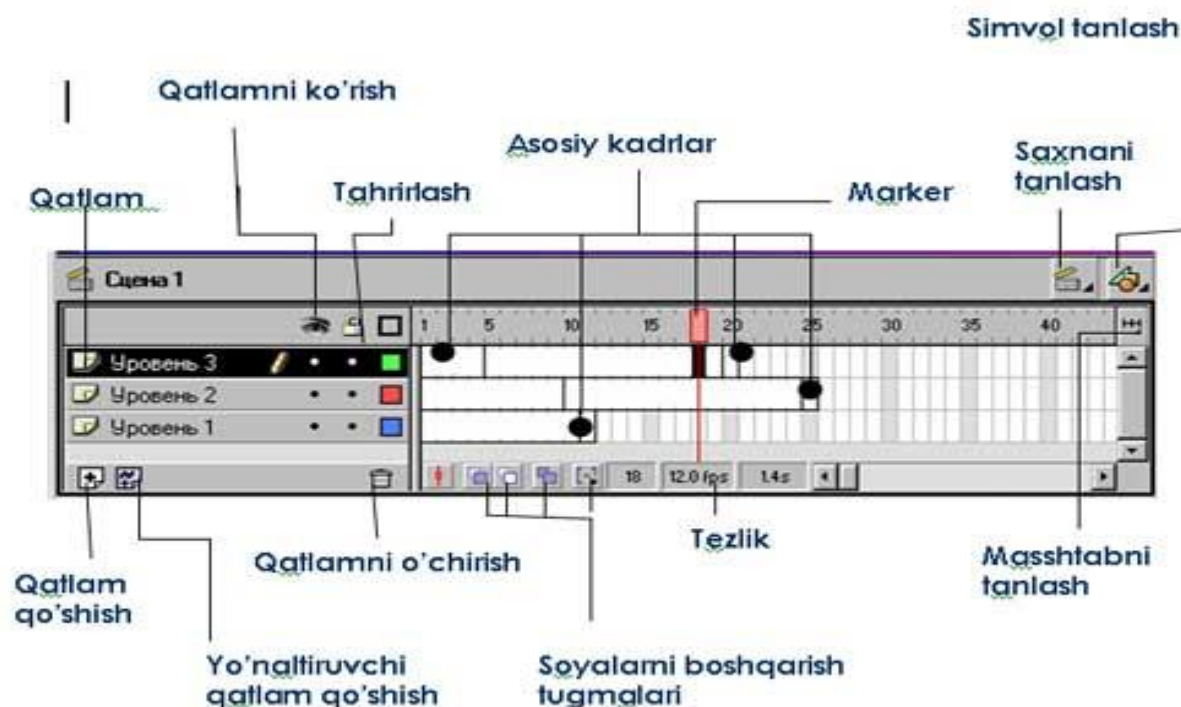
Masalan: yashil rangli sharni oynaning chap qismidan o'ng qismiga 25 ta kadr ichida o'tkazish kerak. Birinchi usulda har bir kadr uchun shar chizish va har bir yangi chizilgan kadrda shar avvalgi kadrda nisbatan bir oz o'ng tomonga surilgan holda bo'lishi kerak. Agar sharni o'ng tomonga surilgan holda asta-sekin yo'qolishini ko'rsatmoqchi bo'lsak, har bir kadrda harakatda **прозрачность** qiymatini o'rnatib borish kerak. Shuning uchun multikdagi harakatni hosil qilish uchun ikkinchi usuldan foydalangan ma'qul. Bu usulda harakatlanayotgan sharni ko'zga ko'rinmaydigan holga keltirish uchun, har bir harakatni tasvirlashda tweening animationdan foydalanish kerak. Bunda faqat asosiy kadrlar (**ключевые кадры**) Keyframes ya'ni boshlang'ich va ohirgi kadrlar beriladi. Oraliq kadrlarni esa Flashning o'zi hisoblaydi. Bunda oraliq qiymat chiziqli qonuniyatga asoslanib o'rnatiladi. Ammo o'sib yoki kamayib boruvchi qiymatlarni ham berish mumkin. Bu albatta biror qonuniyatga asoslangan bo'lishi kerak. Masalan: koptokning sakrab harakatlanishi.

Kadrlar, qatlamlar, simvollar, vaqt shkalasi.

Flash dasturidagi harakatlarning barcha usullarini ko'rib o'tmoqchimiz. Ammo avval boshlang'ich ma'lumotlar bilan tanishamiz. Bu tushunchalarga kadrlar(кадры) frames, simvollar (СИМВОЛЫ) symbols, qatlamlar (слои) layers va vaqt shkalasi (шкала времени) timelinelar kiradi. Har birini ko'rib o'tamiz.

Vaqt shkalasi.(Шкала времени)

Vaqt shkalasi (шкала времени) Flashning asosiy quolidir. Unda qanday qatlamlar mavjudligi qaysi kadrlar asosiy ekanligi hamda Flash qaysilarini generatsiya qilishi haqida ma'lumot bo'ladi. Vaqt shkalasida shuningdek qaysi kadrlar harakatdan yoki belgidan iborat ekanligini ko'rish mumkin. U asosiy kadrlarni harakatlar qismi bilan birlashtirish imkoniyatiga ega. Flash interfeysining qulayligi shundaki, birinchi bor ishlagan bajaruvchi ham, vaqt shkalasini tez topishi mumkin.



2.1.8-Chizma Vaqt shkalasi.

2.1.8-chizmada vaqt shkalasining umumiy ko'rinishi keltirilgan. Vaqt shkalasidagi barcha elementlarni sanab o'tish ko'p vaqtni talab etadi. Shuning uchun ba'zilar ustida to'xtalib o'tamiz. 2.1.8- chizmadagi vaqt shkalasida keltirilgan ba'zi bir qurollar quyidagi vazifalarni bajaradi: Marker – oynadagi mavjud bo'lgan kadrlarni ko'rsatadi. Vaqt shkalasidagi biror kadrda murojat qilinsa, marker shu kadrda avtomatik tarzda o'tadi.

Qatlam (**слой**) – vaqt shkalasining chap tomonida qatlamlar ro'yxati joylashgan. Uning pastki qismida qatlamni qo'shish va olish tugmachalari mavjud. Har bir qatlamni ko'rinmas holga keltirish va taxrirlashni (**редактирование**) taqiqlashi mumkin. Kadrlar shkalasi (**шкала кадров**) – bu shunday maydonki, unda oddiy va asosiy kadrlarni qo'shish va olish mumkin. Agar biror kadrda sichqoncha ko'rsatkichining o'ng tugmasini bosilsa, kontekstniy meny (**контекстное меню**) paydo bo'ladi. Kontekstniy menuda bajariladigan harakatlar ro'yxatini ko'rish mumkin. Shkalada asosiy kadrlar haqida axborot kadrlar qora aylanalar bilan ko'rsatiladi. Harakat mavjud bo'lib, ular qora aylanada “a” harfi bilan yoki qizil bayroqchadan keyin belgi (**метка**) nomi ko'rsatiladi. Ranglar ham kadr turini bildiradi. Masalan, kulrang – bu kadr asosiy (**ключевой**) (Keyfrem) kadrlarni takrorlovchi kadrlar. Ko'kishroq yoki yashilroq rang Flash bilan generatsiya qilinganligini ko'rsati. Oq yoki “bo'sh” chizikli rang esa kadrda hech qanday ma'lumot yo'qligini bildiradi.

Soyalarni boshqarish tugmalari (**кнопки управления тенями**)– bu tugmalar qo'shni kadrni ko'rish, ya'ni avvalgi kadr bilan keyingi kadrlarni juda yuqqa qog'oz yordamida ko'rsatiladi. Bunday ko'rinishlarni ko'rish chuqurligini markerning ikki tomonida ham berish mumkin.

Qatlamlar.(Слой).

Kompyuter grafikasida qatlamlar juda ko'p ishlatiladi. Masalan, siz yuqqa qog'ozga rasm chizib, ularni ustma-ust qo'yapsiz. Ustki qog'oz pastki qog'ozni

berkitadi. Qatlamlarni ko'rinmas holatga yoki ularga murojat qilishni man etilgan holatga keltirish mumkin, chunki bu o'z navbatida sahnani taxrirlashda osonlik yaratadi. Flash da ikki xil qatlamlar mavjud. Bu harakatni ko'rsatuvchi traektoriya qatlami va maska qatlami. Qqatlamlarsiz Flash da umuman ishlab bo'lmaydi, chunki har bir vaqt uchun, har bir harakatlanuvchi ob'ektga alohida qatlam yaratilishi kerak. Harakatlanuvchi ob'ekt shakl (shape) yoki simvol (**СИМВОЛ**) bo'lishi mumkin.

Kadrlar.

Harakat (**анимация**) – kadrlar ketma-ketligidan iboratdir. Kadrlar bajaruvchi tomonidan yaratilishi yoki Flash tomonidan generatsiya qilingan bo'lishi mumkin. Bu bitta qatlamga tegishlidir. Flashdagi sahna (**сцена**) bir necha qatlamlardan iborat bo'lganligi uchun, yakuniy ko'p qatlamli kalrlar generatsiya qilingan yoki bajaruvchi tomonidan yaratilgan bo'lishi mumkin. Harakatda asosiy (**Ключевой**) kadr degan tushuncha bor (keyframes). Bu kadrlarni harakat davomida Flash o'zgartira olmaydi. Bu asosiy kadrlarni bajaruvchi beradi, oraliqlarini esa Flash hosil qiladi. Ikki xil oraliq kadrlar mavjud:

1-geometrik shakllarni o'zgartirish natijasida hosil bo'lgan kadrlar (shape tweening);

2- simvollarning o'zgarishi natijasida hosil bo'lgan kadrlar (motion tweening).

Shuningdek, kadrlar bo'sh bo'lib, hach qanday ma'lumot bo'lmasligi mumkin.

Kadrlar bilan oddiy amallar bajarish.

Bo'sh asosiy kadrni qo'yish – INSERT->BLANK->keyframe, F7.

Oldingi kadrni takrorlovchi asosiy kadr – INSERT -> keyframe, F6.

Asosiy kadrni tozalash – INSERT->CLEAR->keyframe, shift+F6.

Oddiy kadrni qo'yish – INSERT->FRAME, F5.

Kadrni olib tashlash (o'chirish) – INSERT->REMOVE->FRAMES, shift+F5.

Simvollar. (СИМВОЛЫ)

Simvollar bu Flash dagi asosiy tushunchalardan biridir. Oddiy geometrik shakl yoki bir necha shakllarning birlashmasidan, yoki harakatlar (move) simvol bo'lishi mumkin. Bu bilan simvollarni Flash- ning eng asosiy mexanizmi deb atash mumkin. Masalan, g'ildirak, avtomobil korpusi, oyna, antenna simvollarini yaratib, ularni birlashtirish natijasida avtomobil ko'rinishidagi simvolni yaratish mumkin. Shundan so'ng avtomobilning harakatlanishini ko'rsatuvchi sahnani yaratish mumkin. Yana boshqa bir misol. Yog'ayotgan qorni hosil qilmoqchimiz Buning uchun bitta qor simvolini yaratib, bir nechta harakatlanayotgan qor parchalar va ustun shaklidagi yog'ayotgan qorlarni hosil qilinadi. So'ngra bu ustunlar ko'paytirilib, sahnada yog'ayotgan qor tasvirini hosil qilish mumkin. Simvollar hosil qilinayotgan sahnaga egiluvchanlik kiritadi, ya'ni avtomobildagi g'ildirakni aylantirilsa, avtomobil yurib ketayotgandek tuyuladi yoki avtomobil eshiklaridagi tugmalar bilan boshqarish mumkin. Buning uchun tugmaga sichqoncha ko'rsatkichi keltirilib bosilsa eshik ochiladi yoki yopiladi. Istalgan vaqtda simvolni o'zgartirish mumkin. Bu esa Flash da yaratilgan sahnalarga ketayotgan mablag'ni kamaytiradi.

Uch xil simvollar mavjud. Harakat (**анимация**) move clip, tugma (**кнопка**) button hamda tasvir (**изображение**) graphic.

Har birini quyidagicha izohlash mumkin:

1. Tasvir (**изображение**) graphic – bu bitta kadrda iborat simvol. Agar, simvol harakatlanmayotgan bo'lsa unda simvolni tasvir graphic ko'rinishida tashkil qilinishi mumkin;

2. Tugma (**кнопка**) button – Flashda maxsus funktsiyalar tugmalar ko'rinishidagi simvollar mavjuddir. Tugmada 4 ta kadr mavjud. Up, Over, Down, Hit. Bu tugmalar quyidagi holatini bildiradi:

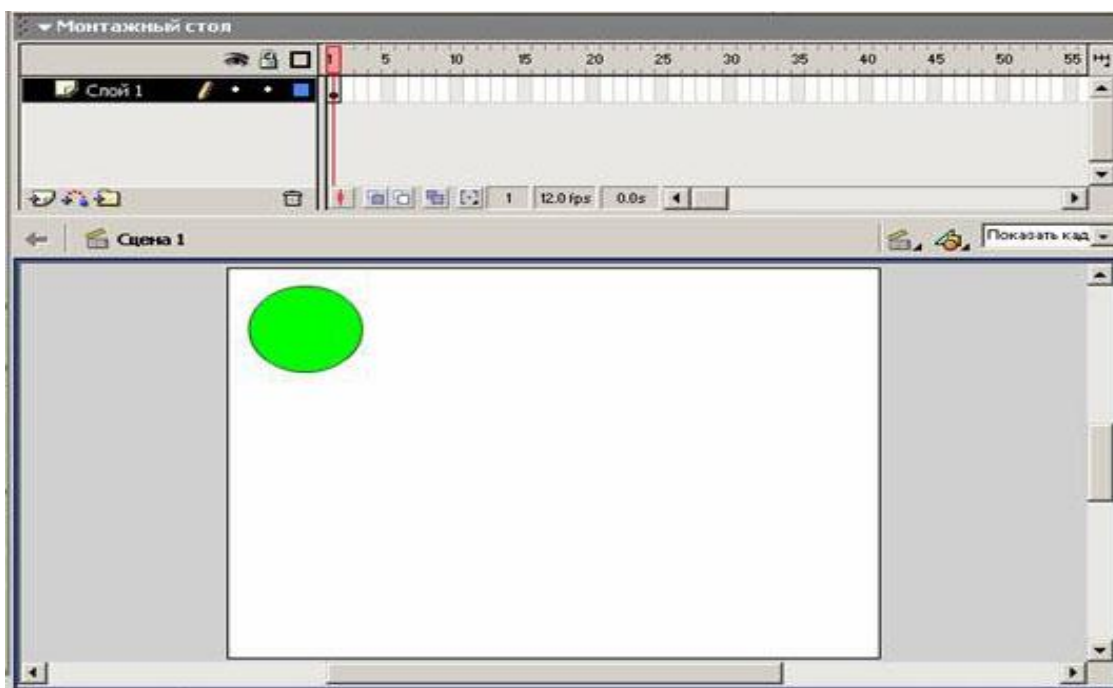
- Up tugmaning oddiy holati;
- Over sichqoncha ko'rsatkichi tugmaning yuqorisida turgan holati;
- Down sichqoncha ko'rsatkichi tugmaning pastki qismida bo'lib, sichqonchanning tugmasi bosilgan holatini bildiradi;

- Hit tugmaning oddiy holati bo'lib, tugmaga avval ham murojaat etilganligi ko'rsatiladi.

3. Harakat (**анимация**) move clip – bu simvollarning eng to'la ko'rinishidir. Bunda istalgancha kadrlar soni bo'lishi mumkin. Bu turdagi simvollar Action scriptdagi Move (Flash tili) ob'ekt deb qaralishi mumkin.

Simvollar o'z turiga bog'liq bo'lmagan holda joylashishi mumkin. Bu ularning eng asosiy xususiyatlaridan biri. Masalan, shunday tugma yaratish mumkinki, sichqoncha ko'rsatkichi shu tugmaga tegishi bilan harakatlansin. Kadrda Over simvolini yoki mushukni simvol o'rnida kiritish mumkin. Simvolni yangidan hosil qilish uchun INSERT->New symbol, ctrl+F6 yoki ajratilgan shaklni simvolga keltirish mumkin INSERT-> Convert to symbol, F8. Simvollar haqida ma'lumotni Library bibliotekasidan olib, ko'rish mumkin. Biblioteka menudagi Windows-> Library (yoki ctrl+F6)da joylashgan.

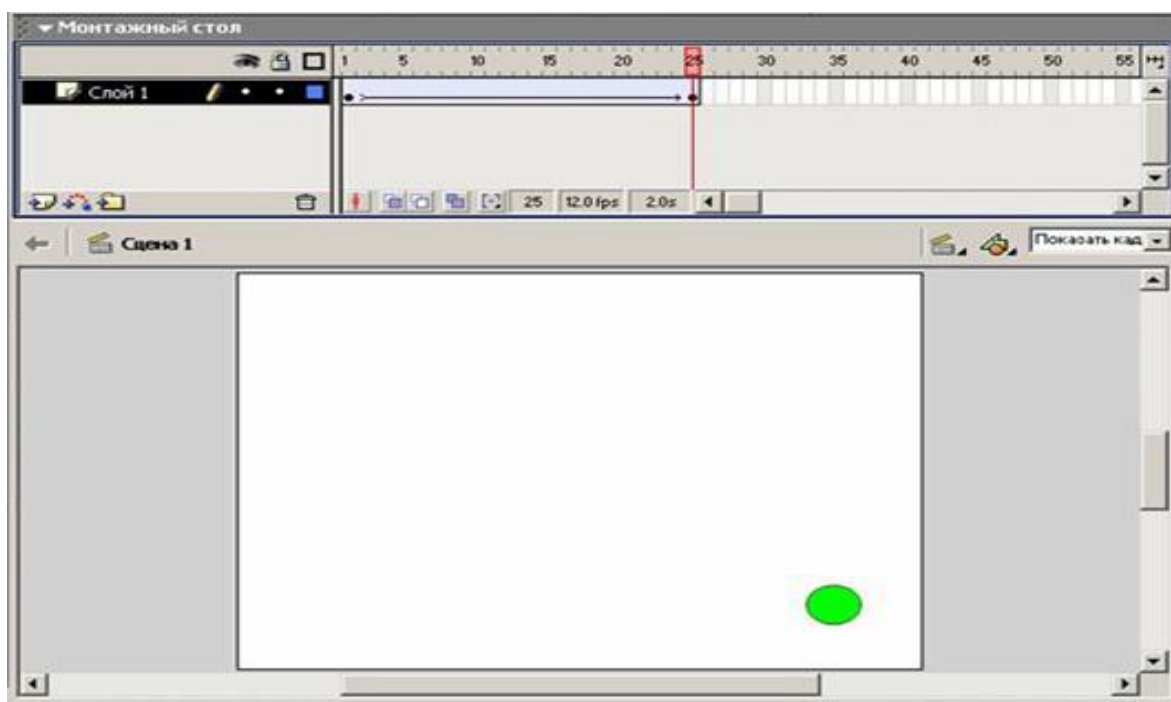
Flash animatsiya ikki xil buladi: kadrli (pokadrovoe sozдание) va avtomatik(avtomaticheskoe sozдание promejutochnqx kadrov). Avtomatik animatsiya shakllar geometriyasini uzgarishi (sha`e tweening) yoki boshkaruv kadrlar uzgarishi (motion tweening) asosidagi animatsiya turlarga bulinadi.



2
 .1.9-
 chizm
 a
 Shu
 turdag
 i
 animat
 siyani

yaratish uchun biz bitta boshkaruv kadrni yaratamiz va unga belgi kushamiz.

Masalan boshkaruv kadrda aylana chiziladi va u **grafik tasvir** belgi turiga **F8** yoki Vstavka menyusida Preobrazovatg` v simbol (**Convert to Symbol**) buyrigi yordamida utkaziladi. Yoki **Ctrl+F8** yoki Vstavka menyusida Novqy simbol (**New symbol**) buyrigini tanlab yangi belgi yaratamiz va Belgilar kutubxonasi yordamida uni boshkaruv kadrga kushamiz.



2
 .1.
 10
 -
 chi
 zm
 a
 En

di belgi joylashgan boshkaruv kadrni sichkonchanning ung tomondagi tugma yordamida tanlab **Creat motion tweening** yoki Vstavitg` menyusining shu nomli buyrugini tanlamiz. SHu xarakterlar natijasida boshkaruv kadr rangi kuk rangga uzgaradi. Endi s

ichkoncha bilan yangi kadrni tanlaymiz, (masalan 25-chi kadrni) va **F6** yoki Vstavka menyusida Klyuchevoy kadr (**Insert keyframe**) aktiv katlamda keyingi boshkaruv kadrini yaratish buyrugini tanlaymiz. Natijada 25-chi kadrda kuk rangli boshkaruv kadr xosil kilinadi va shu kadrgacha birinchi boshkaruv kadr dan strelka xosil kilinadi. Birinchi boshkarish kadr dan ikkinchi boshkarish kadrgacha kadrlar kuk rangda avtomatik xosil kilinadi. Oxirgi xarakterimiz - bu ikkinchi boshkarish kadrdagi belgini uzgartirish (**chuzish, aylantirish, kattalashtirish, kichkinalashtirish yoki kadrdagi joylanishini uzgartirish**). Endi


klaviaturadagi Enter tugmasini bosamiz va biz yaratgan animatsiyani kurishimiz mumkin.

2. Shakllar geometriyasini o`zgarishi (sha`e tweening)asosidagi yaratilgan animatsiya.

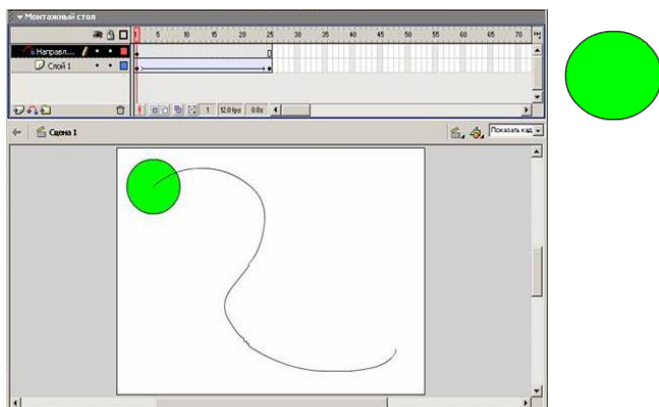
SHu turdagi animatsiyani yaratish uchun biz boshkaruv kadrlar uzgarishi (motion tweening) asosidagi yaratilgan animatsiya xosil kilamiz. Fakat endi oxirida ikkinchi katlamdagi belgini butunlay uchirib uning urniga kvadrat chizamiz. SHu xarakatimizdan keyin kadrlar rangi normal rangga kaytadi. Keyin birinchi va ikkinchi boshkaruv kadrlardagi **grafik tasvir** belgini CtrlKB yoki Izmenitg` menyusidagi Razdelitg` otdeg`no (Breack a`artack) buyrugi yordamida aloxida shakllarga bulib chikamiz.

Endi avval birinchi boshkaruv sichkoncha chap tugmasi bilan tanlab **Svoystva (`ro`rties)** yoki **CtrlKF3** yoki Okno menyusining shu nomli buyrugini tanlamiz. Natijada mulokot oynasi xosil kilinadi va unda Tweening soxasida Motions urniga Sha`e xolatini tanlaymiz. SHu natijasida boshkaruv kadr rangi yashil rangga uzgaradi. Endi ikinchi boshkaru kadrni xam sichkoncha bilan tanlab animatsiya turini Motions dan Sha`e ga uzgartiramiz va oxirida klaviaturadagi Enter tugmasini bosamiz va biz yaratgan animatsiyani kurishimiz mumkin.




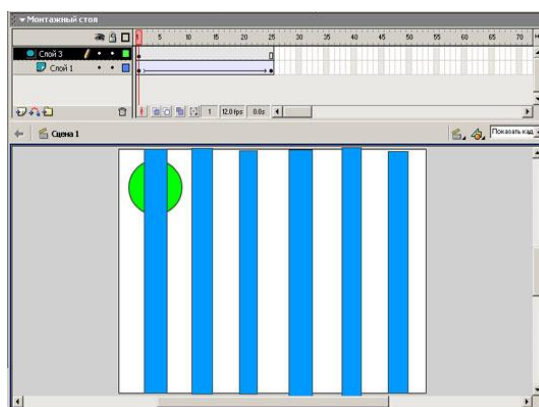
Kurishimiz mumkin ikkala animatsiyalarda xam ikkita boshkarish kadrlar urtasidagi masofa kadrlarni kompg`yuter uzi avtomatik yaratgan va animatsiya tugri chizik buyicha xarakatlanyapti. Ammo agar bizga xarakat traektoriya buyicha bajarilishi kerak bulsa, u xolda nima kilish kerak? Bunday animatsiyalarni xosil kilish uchun bizning katlamimiz ustida maxsus  **xarakat traektoriya katlamini**

yaratish kerak buladi. Va shu katlamda  kalam bilan kiyshik traektoriya chizigini chizamiz. Natijani kurish uchun klaviaturadagi Enter tugmasini bosamiz.



2.1.11-chizma

Shu bilan birga avtomatik animatsiyalarda foydalanish mumkin bulgan yana bir effekt mavjud - bu maska katlami. Maska katlamini uchun asosiy katlam ustida yangi bush katlam yaratamiz. Shu katlamni sichkonchanning chap tugmasi bilan bosib **Maska - Mask** buyrugini tanlaymiz. Natijada katlam kuk rangga uzgaradi va ikkala katlamlar  uzgarishlardan ximoyalanadi. Maska katlamdan shu ximoyani uchirib, boshkaruv kadrda bir nechta turtburchaklar chizamiz. Keyin yana maska katlamning ximoyasini yokib klaviaturadagi Enter tugmasini bosamiz va natijani kuramiz.



2.1.12-chizma

Maska katlamdagi boshkaruv kadrni **motion tweening** animatsiyalashtirilsa xarakatlanish efekti yanada chiroyli buladi.

3. Macromedia Flash dasturining menyusi.

Menyu Fayl

Novy (Ctrl+N) - Yangi fayl yaratish.

Novqy iz shablona - Yangi faylni shablondan yaratish .

Otkrqt (Ctrl+O) - Eski ilgari yaratilgan faylni ochish.

Otkrqt kak biblioteku (Ctrl+Shift+O) - Belgilar kutuvxonasi xolatida ochish.

Zakrqt (Ctrl+W) - Fayln berkitish.

Soxranit (Ctrl+S) - Faylni saklash.

Soxranit kak (Ctrl+Shift+S) - Yangi nom ostida faylni saklash.

Soxranit kak shablon - Faylni shablok kurinishida saklash.

Vernut - Importirovat (Ctrl+R) - Faylga yangi Flash ga mansub bulmagan ob`ektini aktiv boshkaruv kadrga kushish.

Import v biblioteku - Faylga yangi Flash ga mansub bulmagan ob`ektini "Belgilar kutubxonasi"ga kushish.

Eksportirovat rolik (Ctrl+Alt+Shift+S) - Xarakatchan animatsiyani eksport (saklash) kilish

Eksportirovat izobrajenie - Boshkaruv kadrni eksport (saklash) kilish

Obo`ie nastroyki (Ctrl+Shift+F12) - Xarakatchan animatsiya rolikini kurib chikarish xususiyatlari.

Prosmotr - Xarakatchan animatsiya rolikini kurib chikarish.

2.2. Interaktiv o'quv materiallar.

Massiv elementlarini ko'piklar usuli yordamida qadamma-qadam saralash boyicha interaktiv o'quv materiali. Bu dastur Delphi dasturi yordamida tayyorlandi va dastur yordamida bu jarayon interaktiv namoyish qilindi.



2.2.1-chizma. Dastur ko'rinishi.

Ushbu interaktiv dastur quyidagicha ishlaydi. Massiv elementlar soni kiritilib OK tugmasi bosilsa massiv elementlarini kiritish uchun joy ajratib beradi. Massiv elementlarini kiritib SARALASH tugmasi bosilsa, massiv elementlarini saralash jarayoni qadamma-qadam namoyish etiladi. MAX tugmasi bosilsa massivdagi eng kata element joylashgan yacheyka qizil rangda bo'yaladi. MIN tugmasi bosilsa

massivdagi eng kichik element joylashgan yacheyka yashil rangda bo'yaladi.
Ushbu dasturning dastur matni quyida keltirilgan.

```
unit a;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, StdCtrls, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```
Edit1: TEdit;
```

```
Button1: TButton;
```

```
StringGrid1: TStringGrid;
```

```
Memo1: TMemo;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Button4: TButton;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
procedure Button4Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```

{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
StringGrid1.ColCount:=strtoint(Edit1.Text);
end;
procedure TForm1.Button2Click(Sender: TObject);
var
a:array[1..100] of integer;
k, i, ii,buf,SIZE,q:integer;
changed:boolean;
s:string;
begin
Memo1.Lines.Clear;
SIZE:=strtoint(Edit1.Text);
for i:=1 to SIZE do
a[i] := StrToInt(StringGrid1.Cells[i-1, 0] );
q:=0;
repeat q:=q+1;
s:=inttostr(q)+ ' - qadam ==> ';
Changed:=FALSE;
for k:=1 to SIZE-1 do
if a[k] > a[k+1] then
begin
buf := a[k]; a[k] := a[k+1];
a[k+1] := buf;
changed := TRUE; end;
if not(changed) then begin break;end;
for i:=1 to SIZE do
s:=s+' '+IntToStr(a[i]);
Memo1.Lines.Add(s);

```

```

until not (changed);
end;
procedure TForm1.Button3Click(Sender: TObject);
var rect:TRect;
i,k,n:integer;
begin
n:=strtoint(Edit1.Text);
k:=strtoint(StringGrid1.Cells[0,0]);
for i:=0 to n-1 do
if k<strtoint(StringGrid1.Cells[i,0]) then
k:= strtoint(StringGrid1.Cells[i,0]) ;
for i:=0 to n-1 do begin
if strtoint(StringGrid1.Cells[i,0])=k then begin
Rect := StringGrid1.CellRect(i,0 );
StringGrid1.Canvas.Brush.Color := clred;
StringGrid1.Canvas.Rectangle(Rect);
StringGrid1.Canvas.TextOut(Rect.Left + 5, Rect.Top + 5,
StringGrid1.Cells[i, 0]) ;
end; end; end;
procedure TForm1.Button4Click(Sender: TObject);
var rect:TRect;
i,k,n:integer;
begin
n:=strtoint(Edit1.Text);
k:=strtoint(StringGrid1.Cells[0,0]);
for i:=0 to n-1 do
if k>strtoint(StringGrid1.Cells[i,0]) then
k:= strtoint(StringGrid1.Cells[i,0]) ;
for i:=0 to n-1 do begin
if strtoint(StringGrid1.Cells[i,0])=k then begin

```

```

Rect := StringGrid1.CellRect(i,0 );
StringGrid1.Canvas.Brush.Color := clgreen;
StringGrid1.Canvas.Rectangle(Rect);
StringGrid1.Canvas.TextOut(Rect.Left + 5, Rect.Top + 5,
StringGrid1.Cells[i, 0]) ;
end; end;
end; end.

```

Yana bir nechta interaktiv o'quv materiallar FLASH dasturi yordamida tayyorlandi. Ularning umumiy ko'rinishi quyida keltirilgan.

MASSIVLARNI TARTIBLASH BO'YICHA INTERAKTIV O'QUV MATERIALLAR

AJRATISH USULI	3 7 2 11 4 6 16
TANLASH USULI	<pre> for(i=0;i<n;i++){ m=a[i]; k=i; for(j=i;j<n;j++) if(a[j]<a[k]){ m=a[j]; k=j; } a[k]=a[i]; a[i]=m; } </pre>
KO'PIKLAR USULI	

MUALLIF: JO'RAQULOV JASUR

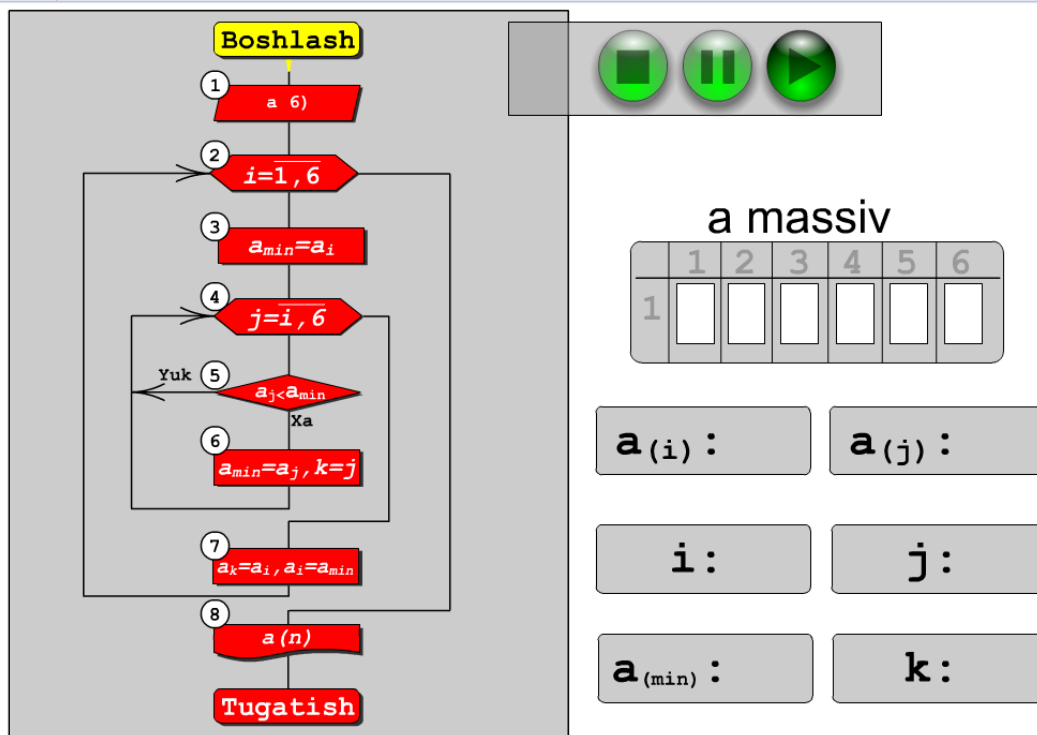
2.2.2-chizma. Dastur ko'rinishi.

K O' P I K L A R U S U L I							
Berilgan massiv	1-qadam	2-qadam	3-qadam	4-qadam	5-qadam	6-qadam	7-qadam
44	06	06	06	06	06	06	06
55	44	12	12	12	12	12	12
12	55	44	18	18	18	18	18
42	12	55	44	42	42	42	42
94	42	18	55	44	44	44	44
18	94	42	42	55	55	55	55
06	18	94	67	67	67	67	67
67	67	67	94	94	94	94	94

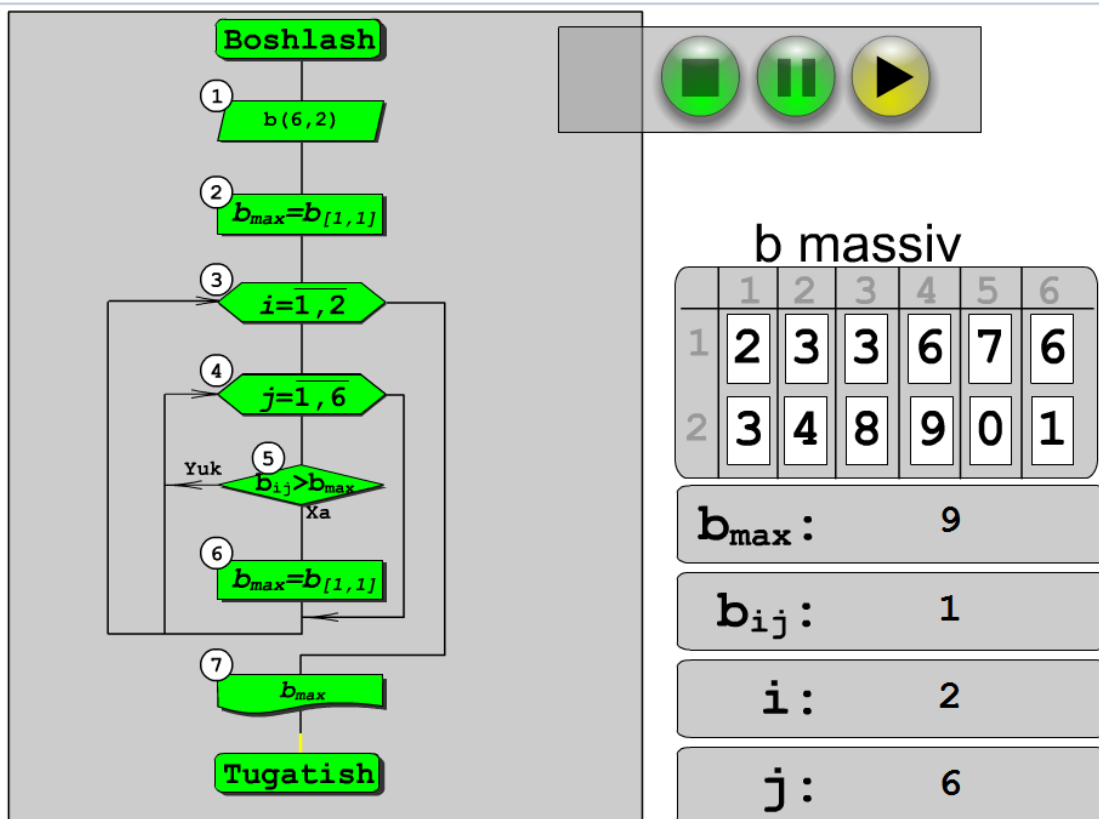
2.2.3-chizma. Ko'piklar usuli dastur ko'rinishi.

A J R A T I S H U S U L I								
Berilgan massiv	44	55	12	42	94	18	06	67
1 - qadam	44	55	12	42	94	18	06	67
2 - qadam	12	44	55	42	94	18	06	67
3 - qadam	12	42	44	55	94	18	06	67
4 - qadam	12	42	44	55	94	18	06	67
5 - qadam	12	18	42	44	55	94	06	67
6 - qadam	06	12	18	42	44	55	94	67
7 - qadam	06	12	18	42	44	55	67	94

2.2.4-chizma. Ajratish usuli dastur ko'rinishi.



2.2.5-chizma. Dastur ko'rinishi.



2.2.6-chizma. Massivdagi eng kata elementni topish.

Xulosa.

II-bobning 1- bo'limida Macromedia Flash dasturidan foydalanib interaktiv o'quv materiallarini tayyorlash jarayonlari keltirilgan. Macromedia Flash dasturi bo'yicha aynan foydalanilgan obyektlar, animatsiyalar yaratish bo'yicha ma'lumotlar misollar orqali tushuntirib berilgan. II-bobning 2- bo'limida C++ dasturlash tilida statik ma'lumotlar tuzilmalarni tasvirlash jarayoni Macromedia Flash dasturi yordamida interaktiv namoyish qiluvchi interaktiv o'quv materiallar tayyorlangan. Aynan ko'proq massivlar va ular ustida bajariladigan amallar bo'yicha materiallar tayyorlangan.

XOTIMA.

Hozirgi kunda C++ dasturlash tili juda ko'p sohalarda qo'llanilib kelinmoqda. Barcha oliy o'quv ta'lim muassasalarida C++ dasturlash tili o'rgatilib kelinadi. Ushbu BMIda C++ dasturlash tilida statik malumotlar tuzilmalari bo'yicha ma'lumotlar keltirilgan va bu jarayonni o'rgatish uchun interaktiv o'quv materiallari yaratilgan.

Bitiruv malakaviy ishning I-bobi 3 ta bo'limdan iborat bo'lib, bunda statik ma'lumotlar tuzilmalari bo'yicha ma'lumotlar keltirilgan. Ma'lumotlar C++ dasturlash tilida bo'lganligi uchun. I-bobning 1- bo'limida C++ dasturlash tilining asosiy operatorlari keltirilgan. I-bobning 2- bo'limida C++ dasturlash tilida massivlar va ular ustida bajariladigan amallar misollar bilan tushuntirib berilgan. I-bobning 3- bo'limida C++ dasturlash tilida satr tipi va ular ustida bajariladiga funksiyalar batafsil yoritib berilgan.

Bitiruv malakaviy ishning 2-bobi 2 ta bo'limdan iborat bo'lib, bunda interaktiv o'quv materiallarini tayyorlash jarayonlari va interaktiv o'quv materiallarini tayyordlashda foydalanilgan dasturiy ta'minotlar keltirilgan. II-bobning 1- bo'limida Macromedia Flash dasturidan foydalanib interaktiv o'quv materiallarini tayyorlash jarayonlari keltirilgan. Macromedia Flash dasturi bo'yicha aynan foydalanilgan obyektlar, animatsiyalar yaratish bo'yicha ma'lumotlar misollar orqali tushuntirib berilgan. II-bobning 2- bo'limida C++ dasturlash tilida statik ma'lumotlar tuzilmalarni tasvirlash jarayoni Macromedia Flash dasturi yordamida interaktiv namoyish qiluvchi interaktiv o'quv materiallar tayyorlangan. Aynan ko'proq massivlar va ular ustida bajariladigan amallar bo'yicha materiallar tayyorlangan.

Ushbu yaratilgan statik ma'lumotlar tuzilmalari bo'yicha interaktiv o'quv materiallardan dasturlash tillari, obyektga yo'naltirilgan dasturlash fanlarini o'tishda foydalanish mumkin.

Adabiyotlar:

1. Karimov I.A. «Yuksak ma'naviyat yengilmas kuch».
1. Павловская Т.А. С++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.
2. Мадрахимов Ш.Ф., Гайназаров С.М. С++ тилида дастурлаш асослари// Тошкент, ЎЗМУ, 2009, 196 бет.
3. Madraximov Sh.F., Ikramov A.M., Babajanov M.R. С++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekiston Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.
4. Bjarne Stroustrup. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1363 page.
5. Bjarne Stroustrup. Programming: Principles and Practice using C++ (Second Edition)" Addison-Wesley, 2014, 1305 page.

6. Никифорова Н. Г., Федоровская Р. А., Никифоров А. В. Работа в среде Macromedia Flash 5; ИВЭСЭП - Москва, 2008. - **899 с.**
7. Пакнелл Шон , Хогг Брайан , Суонн Крейг Macromedia Flash 8 для профессионалов; Вильямс - Москва, 2006. - 672 с.
8. Панфилов, И. Macromedia Flash 8 с нуля!; Лучшие книги - Москва, 2007. - 368 с.

Tavsiya etiladigan web-saytlar va boshqa manbalar.

9. <http://cppstudio.com> – C++ tilida dasturlash bo'yicha namunalar keltirilgan
10. <http://www.compteacher.ru/programming> – dasturlash bo'yicha video darsliklar mavjud.
11. <http://www.ziyonet.uz>.
12. <http://www.elkutubxona.narod.uz>