

16-MA'RUZA. FUNKSIYALAR VA ULARNI E'LON QILISH

1. Funksiya haqida tushuncha, e'lon qilish, qaytish qiymati

Funksiya - bu ma'lum bir vazifani bajarish uchun bayonotlar ketma-ketligi. Ko'pincha, dastur kodi funktsiyani bajarishi uchun boshqa operatorlarni to'xtatib turadi.

Haqiqiy hayotda ham shunga o'xshash narsalarni doimo bajarasiz. Masalan, siz kitob o'qiyapsiz va telefon orqali qo'ng'iroq qilishingiz kerakligini eslaysiz. Kitobingizda xatcho'p qoldirasiz, telefonni ko'tarib raqamni terasiz. Suhbatlashgandan so'ng, kitobni o'qishga qaytasiz, ya'ni qoldirgan sahifangizdan kitobni o'qishni davom ettirasiz.

C++ dasturlari shunga o'xshash tarzda ishlaydi. Ba'zan, dastur kodni bajarayotganda, funksiya murojaatiga duch kelishi mumkin.

Funksiyaga murojaat – bu protsessorga joriy funktsiyani bekor qilishni va boshqa funktsiyani ishlatishni aytadigan ifoda. Protsessor joriy bajarilish nuqtasida "xatcho'p qoldiradi" va keyin chaqirilgan funktsiyani ishlatadi. Murojaat qilingan funktsiyani bajarish tugagandan so'ng, protsessor "xatcho'p"ga qaytadi va to'xtatilgan funksiya bajarilishini davom ettiradi.

Funksiyani e'lon qilish. Funksiya dastur bajaradigan harakatlarni belgilaydi. Funktsiyalar sizga ko'rsatmalar to'plamini ajratib ko'rsatish va unga nom berish imkonini beradi. Keyin dasturning turli qismlarida tayinlangan funksiya nomi yordamida bir necha marta murojaat qilish mumkin. Funksiya aslida kodning nomlangan blokidir.

```
tip funksiya_nomi(parametrlar)
{
    ko'rsatmalar
}
```

Birinchi satr funksiya sarlavhasini aks ettiradi. Dastlab, funktsiyani qaytarish turi (tipi) ko'rsatiladi. Agar funksiya hech qanday qiymat qaytarmasa, u holda void turi ishlatiladi.

Keyin ixtiyoriy identifikatorni ifodalovchi funksiya nomi keladi. Funktsiyani nomlashda o'zgaruvchi nomlashda qo'llaniladigan qoidalaridan foydalaniladi.

Funksiya nomidan keyin parametrlar qavs ichida keltirilgan. Funktsiyaning parametrlari bo'lmasligi mumkin, bu holda bo'sh qavslar ko'rsatiladi.

Funksiya sarlavhasidan so'ng, sistemali qavslarda bajariladigan ko'rsatmalar berilgan funksiya tanasini ifodalaydi.

Qaytish qiymati. Natija qaytarish uchun funksiya **return** operatoridan foydalanadi. Agar funksiya qaytarish turi sifatida void tipidan boshqa har qanday

turga ega bo'lsa, u return operatori yordamida qiymatni qaytarishi shart. Masalan, har qanday C++ dasturida bo'lishi kerak bo'lgan va uning bajarilishi boshlanadigan asosiy funksiya ta'rif:

```
int main()  
{  
  return 0;  
}
```

main() bajarilishini tugatgandan so'ng, return operatori yordamida butun sonni operatsion tizimga qaytaradi.

Biz yozadigan funksiyalar qiymatlarni ham qaytarishi mumkin. Buning uchun siz qaytish turini (yoki "qaytish tipini") belgilashingiz kerak. Bu funksiyani e'lon qilishda, uning nomidan oldin ko'rsatiladi. Qaytish turi qaysi qiymat qaytarilishini ko'rsatmasligini unutmang. Bu faqat ushbu qiymat turini bildiradi.

So'ngra, chaqirilgan funksiya ichida, qaytish qiymatini belgilash uchun return operatoridan foydalanamiz - qaysi qiymat murojaat qiluvchiga qaytariladi.

Butun sonni qaytaradigan oddiy funksiyani ko'rib chiqaylik:

```
int Funk()  
{  
  // Ushbu funksiya butun qiymatni qaytaradi, shuning uchun biz return  
  operatoridan foydalanishimiz kerak  
  return 7;  
}
```

void tipi. Yuqoridagi funksiyaning qaytish turi int, shuning uchun funksiya return operatoridan foydalanishi va int turiga mos keladigan ba'zi bir qiymatni qaytarishi kerak. Qaytish qiymati return bayonotidan keyin joylashtiriladi.

Funksiyalar qiymatlarni qaytarishi yoki qaytarmasligi mumkin. Kompilyatorga funksiya qiymatni qaytarmasligini aytish uchun siz void return turidan foydalanishingiz kerak. Quyida berilgan doPrint() funksiyasini ko'rib chiqamiz:

```
void doPrint()  
{  
  cout<<"Salom";  
}
```

Ushbu funksiya void qaytarish turiga ega, ya'ni funksiya qiymatni qaytarmaydi. Hech qanday qiymat qaytarilmasligi sababli, hech qanday return bayonoti talab qilinmaydi.

2. Funksiyalarga murojaat qilish va undan foydalanish

Funksiyaga murojaat qilish. Funksiyadan foydalanish uchun uni chaqirish kerak. Funksiya quyidagi shaklda chaqiriladi:

funksiya_nomi(argumentlar);

Funksiya nomidan keyin qavslar qo'shiladi, unda argumentlar - funksiya parametrlari uchun qiymatlar keltirilgan.

Masalan, eng oddiy funkسيyani aniqlaymiz va uni chaqiramiz:

```
#include <iostream>
using namespace std;
void hello()
{
    cout << "Salom\n";
}

int main()
{
    hello();
    hello();
    return 0;
}
```

Bu hello funksiyasini belgilaydi, bu asosiy funksiyada ikki marta chaqiriladi. Bu funksiyalarning afzalligi: biz ba'zi bir umumiy harakatlarni alohida funksiyaga o'tkazamiz va keyin ularni dasturning turli joylarida ko'p marta chaqiramiz. Natijada, dastur "Salom" satrini ikki marta bosib chiqaradi.

Funksiyalarni chaqirishda quyidagi xatoliklarga yo'l qo'ymaslik kerak. Quyidagi misolni qarab chiqaylik:

```
#include <iostream>

void Funk1()
{
    cout<<"Salom!"<<endl;
}
int main()
{
```

```

Funk1 ();
cout<< Funk1 ();// bu xatolik
return 0;
}

```

Funk1() ga birinchi murojaat “Salom!” ni bosib chiqaradi, ammo murojaat qiluvchiga hech narsa qaytarilmaydi. Bajarilish nuqtasi **main()** funksiyasiga qaytadi, bu yerda dastur o‘z bajarilishini davom ettiradi.

Funk1() ga ikkinchi murojaat hatto tuzilmaydi. **Funk1()** funksiyasi void qaytarish turiga ega, ya’ni bu funksiya qiymat bermaydi. Biroq, **main()** funksiyasi ushbu qiymatni (qaytarilmagan) **cout** operatoriga chiqish uchun yuborishga harakat qiladi. **cout** bu ishni ko‘rib chiqa olmaydi, chunki chiqish qiymati berilmagan. Shunday qilib, kompilyator xatoga yo‘l qo‘yadi.

main() funksiyasi haqida. Endi siz **main()** funksiyasi qanday ishlashini tushunasiz. Dastur bajarilgach, operatsion tizim **main()** funksiyasini chaqiradi va uning bajarilishini boshlaydi. **main()** dagi ifodalar ketma-ket bajariladi. Nihoyat, **main()** funksiyasi operatsion tizimga butun qiymatni qaytaradi (odatda 0). Shuning uchun **main()** funksiyasi **int main ()** deb e’lon qilinadi.

Nima uchun operatsion tizimga qiymatlarni qaytarish kerak? Gap shundaki, **main()** funksiyasining qaytish qiymati operatsion tizimga dasturning muvaffaqiyati yoki muvaffaqiyatsizligi to‘g‘risida ma’lumot beradigan holat kodidir. Odatda, qaytarish qiymati 0 (nol) har bir narsaning muvaffaqiyatli bo‘lganligini anglatadi, boshqa har qanday qiymat esa xato / xato degan ma’noni anglatadi.

Shuni bilib olingki, C++ standartlari bo‘yicha **main()** funksiyasi butun sonni qaytarishi kerakligini unutmang. Ammo, **main()** funksiyasi oxirida qaytishni ko‘rsatmasangiz, xatolar bo‘lmasa kompilyator avtomatik ravishda 0 ni qaytaradi. Lekin **main()** funksiyasi oxirida **return** ni ko‘rsatish va **main()** funksiyasi uchun **int return** turidan foydalanish tavsiya etiladi.

Funksiyaning qaytuvchi tipi haqida quyidagi ma’lumotlarni bilib olish kerak:

Birinchidan, funksiyani qaytarish turi void bo‘lmasa, u belgilangan turdagi qiymatni qaytarishi kerak (**return** operatoridan foydalaning). Faqatgina istisno **main ()** funksiyasi bo‘lib, u boshqa qiymat berilmasa 0 qiymatini beradi.

Ikkinchidan, protsessor funksiyada **return** operatoriga duch kelganda, darhol qiymatni murojaat qiluvchiga qaytaradi va ijro etish nuqtasi ham murojaat qiluvchiga o‘tadi. Funksiyada qaytarilish orqasida turgan har qanday kod e’tiborga olinmaydi.

Funksiya murojaat qiluvchiga **return** orqali faqat bitta qiymatni qaytarishi mumkin. Bu raqam yoki (masalan, 7), yoki o‘zgaruvchining qiymati, yoki ifoda (natijaga ega) yoki mumkin bo‘lgan qiymatlar to‘plamidan ma’lum bir qiymat bo‘lishi mumkin.

Ammo bir vaqtning o'zida bir nechta qiymatlarni qaytarish orqali bitta qiymatni qaytarish qoidasidan o'tish usullari mavjud, ammo bu haqida boshqa darsda batafsil gaplashamiz.

Nihoyat, funksiya muallifi uning qaytish qiymati nimani anglatishini hal qiladi. Ba'zi funksiyalar funksiyalarni bajarish natijasini (muvaffaqiyatga erishdimi yoki yo'qmi) ko'rsatish uchun qaytarish qiymatlarini holat kodlari sifatida ishlatadilar. Boshqa funksiyalar mumkin bo'lgan qiymatlar to'plamidan ma'lum bir qiymatni qaytaradi. Bundan tashqari, umuman hech narsa qaytarmaydigan funksiyalar mavjud.

Funksiyalardan bir necha marta foydalanish. Xuddi shu funksiyani hatto turli xil dasturlarda ham bir necha marta chaqirish mumkin, bu juda foydali:

```
#include <iostream>
using namespace std;
//getFunk() funksiyasi foydalanuvchidan qiymatni oladi va keyin uni
murojaat qiluvchiga qaytaradi
int getFunk()
{
    cout<<"Raqam kiriting:";
    int x;
    cin>>x;
    return x;
}

int main()
{
    int a = getFunk();
    int b = getFunk();
    cout<<a<<"+"<<b<<"="<<a+b<<endl;
    return 0;
}
```

Bu yerda main() 2 marta to'xtatiladi. E'tibor bering, har ikkala holatda ham natija x qiymati o'zgaruvchida saqlanadi va keyin return yordamida main () ga qaytariladi va u yerda a va b o'zgaruvchisi navbat bilan qiymat oladi.

Bundan tashqari main() boshqa funksiyalarni chaqira oladigan yagona funksiya emas. Har qanday funksiya boshqa har qanday funksiyani chaqira oladi.

```
#include <iostream>
using namespace std;
```

```
void printO()
{
    cout << "O" << endl;
}
```

```
void printK()
{
    cout << "K" << endl;
}
```

// PrintOK() funksiyasi printO() va printK() ni chaqiradi

```
void printOK()
{
    printO();
    printK();
}
```

```
int main()
{
    cout << "Starting main()" << endl;
    printOK();
    cout << "Ending main()" << endl;
    return 0;
}
```

Ichki funksiyalar. C ++ da ba'zi funksiyalarni boshqa funksiyalar ichida e'lon qilish mumkin emas (ya'ni ularni joylashtirish mumkin emas). Quyidagi kod kompilyatsiya xatoligini keltirib chiqaradi:

```
#include <iostream>
using namespace std;
int main()
{
    int boo() // bu funksiya main() funksiyasi ichida joylashgan, bu esa
    ta'qiqlangan
    {
        cout << "boo!";
        return 0;
    }
    boo();
}
```

```
    return 0;
}
```

Bu aslida quyidagicha bo'ladi:

```
#include <iostream>
using namespace std;
int boo()
{
    cout << "boo!";
    return 0;
}
```

```
int main()
{

    boo();
    return 0;
}
```

3. Funksiya parametrlari va argumentlari

Ko'p hollarda, biz chaqirilgan funksiyaga qandaydir tarzda ta'sir o'tkazishi uchun ma'lumotlarni uzatishimiz kerak bo'ladi. Masalan, agar biz ikkita sonni ko'paytirish uchun funksiya yozmoqchi bo'lsak, unda qanday qilib funksiyaga ular qaysi raqamlar bo'lishini aytib berishimiz kerak. Aks holda, u nimani nimaga ko'paytirishni qayerdan biladi? Bu yerda parametrlar va argumentlar tushunchasi kerak bo'ladi.

Funksiya parametri - bu funksiyada ishlatiladigan va qiymatini chaqiruvchi (murojaat qiluvchi) tomonidan ta'minlanadigan o'zgaruvchidir. Parametrlar funktsiyani e'lon qilishda qavs ichida ko'rsatiladi. Agar ularning soni ko'p bo'lsa, ular vergul bilan ajratiladi, masalan:

```
// Bu funksiya hech qanday parametrغا ega emas
void doPrint()
{
    cout << "doPrint() funksiyasi" << endl;
}
```

```
// Bu funksiya bitta parametrغا ega va u int tipiga tegishli
void printValue(int a)
{
```

```

    cout << a << endl;
}

// Bu funksiya ikkita int tipidagi a va b parametrlariga ega
int add(int a, int b)
{
    return a + b;
}

```

Har bir funksiyaning parametrlari faqat shu funksiya doirasida amal qiladi. Shuning uchun, agar printValue() va add() parametrlari **a** deb nomlangan bo'lsa, unda bu nom ziddiyatli bo'lishini anglatmaydi. Ushbu parametrlar mustaqil deb hisoblanadi va bir-birlari bilan hech qanday ta'sir qilmaydi.

Funksiya argumenti - bu murojaat qiluvchidan funksiyaga uzatiladigan va murojaat qiluvchida funksiyani chaqirishda qavs ichida ko'rsatilgan qiymat:

```

printValue(7);
add(4,5);

```

E'tibor bering, argumentlar vergul bilan ham ajratilgan. Argumentlar soni parametrlar soniga to'g'ri kelishi kerak, aks holda kompilyator xato xabari chiqaradi.

Funksiya parametrlari va argumentlari qanday ishlaydi? Funksiya chaqirilganda uning barcha parametrlari lokal o'zgaruvchilar sifatida yaratiladi va har bir argumentning qiymati mos keladigan parametrga (lokal o'zgaruvchiga) ko'chiriladi. Ushbu jarayon **qiymat bo'yicha uzatish** deyiladi. Masalan:

```

#include <iostream>
using namespace std;
void printValues(int a, int b)
{
    cout<<a<<endl;
    cout<<b<<endl;
}
int main()
{
    printValues(8,9);
    return 0;
}

```


printValues() funksiyasi chaqirilganda, 8 va 9 argumentlari a va b parametrlariga ko‘chiriladi. a parametrga 8, b parametrga esa 9 qiymat beriladi.

Parametrlar va funksiyalarning qaytish qiymatlari qanday ishlaydi? Parametrlar va qaytish qiymatlari yordamida biz ma’lumotlarni qabul qiladigan va ishlov beradigan funksiyalarni yaratib, so‘ngra murojaat qiluvchiga natijani qaytaramiz.

Masalan, ikkita butun sonni oladigan va ularning yig‘indisini qaytaradigan oddiy funksiya:

```
#include <iostream>
using namespace std;
int add(int a, int b)
{
    return a+b;
}
int main()
{
    cout<<add(7,8)<<endl;
    return 0;
}
```