

Извикване на подпрограми в C++ с предаване на параметри

Цел на упражнението:

Преговор на знанията, придобити в дисциплината Програмиране2 за предаване на параметри при извикване на подпрограми.

1. Указатели и псевдоними

а) Указатели

Указателите са променливи, в които могат да бъдат записвани адреси на променливи от същия тип като на указателя.

Примерен кодов фрагмент с използване на указатели:

```
...
int x=7;
int *p= &x; //в указателя p е записан адреса на променливата x
cout<<"Стойността на x е "<<x<<endl; // извежда се 7
cout<<"Стойността на променливата, чийто адрес е записан в p
е"<<*p<<endl; // извежда се 7
...
```

В горния кодов фрагмент ***p** е името на променливата, чийто адрес е записан в указател p (името на променливата, към която указател p сочи)

б) Псевдоними (reference)

Псевдонимът е друго име на променлива от същия тип.

Примерен кодов фрагмент с използване на псевдоними:

```
...
int x=3;
int &y=x; // y е псевдоним на променлива и е инициализиран с
// променливата x, която е дефинирана преди това
// след тази декларация x и y са две различни имена на
// една и съща област от паметта със съдържание 3
y++;
cout<<"Стойността на x е "<< x<<endl; // извежда се 4
cout<<"Стойността на y е "<< y<<endl; // извежда се 4
...
```

При деклариране на псевдоним, той трябва задължително да се инициализира.

2. Подпрограми и предаване на параметри

Действия в програмата, повтарящи се повече от един път могат да бъдат обособени в подпрограма. В програмния език C++ подпрограмите са функции. Изпълнението на едно и също действие, реализирано от функцията, за различни променливи се постига посредством предаване на параметри. При декларирането и дефинирането на функции се използват формални параметри, които показват как функцията да работи с действително предаваните ѝ при нейното извикване параметри. Параметрите на функциите могат да бъдат входни (тези параметри се предават от извикващата към извикваната функция, но евентуални техни промени след изпълнението на извикваната функция **не са видими** за извикващата) и входно-изходни (евентуални техни промени след изпълнението на извикваната функция **са видими** за извикващата).

Указателите и псевдонимите се използват при предаване на входно-изходни параметри (параметри предавани по адрес към подпрограми). При предаване на параметри по адрес извикващата функция съобщава на извикваната такава адреса на паметта, където извикваната да запише очакван резултат. На този адрес резултата от работата на извикваната функция е видим за извикващата – т.е. този резултат е изходен за извикваната функция.

Масивите се предават като входно-изходни параметри чрез указател към началото си, за да не заемат допълнително място в стека. Името на масива е указател към първия му елемент с индекс 0.

Пример:

```
void f1(int a, int b){    // а и b са формални входни параметри
cout<< "Сумата на въведените числа за f1 е: "<< a+b << endl;
}                        // резултатът от сумирането не се предава към
                        // извикващата main функция и затова се извежда
                        // в тялото на извикваната функция f1

int f2(int a, int b){
return a+b;             // функцията връща на извикващата я main функция
}                       // стойността на израза след оператора return

void f3(int a, int b, int *c){// с е формален вх.-изходен параметър
*c=a+b;                // с е указател към int променлива
}                      // в тялото на f3 резултатът от събирането
                      // се записва в int променливата *c,
                      // към която сочи указателят c

void f4(int a, int b, int& c){// с е формален вх.-изходен параметър
c=a+b;                 // с е псевдоним на int променлива
}                      // в тялото на f4 резултатът от събирането
                      // се записва в променливата c

void main(){           // main е извикващата функция
int s1, s2, r;
cout<< "Въведете цяло число за първото събираемо: "; cin>>s1;
cout<< "Въведете цяло число за второто събираемо: "; cin>>s2;

f1(s1, s2);            // s1 и s2 са действителни параметри

f1(44, 89);           // извикване с други действителни параметри

r=f2(s1, s2);          // функцията f2 връща чрез името си
                      // стойност на извикващата я main
                      // в променливата r
cout<< "Сумата на въведените числа за f2 е: "<<r<< endl;

f3(s1, s2, &r);        // при извикването съгласно дефиницията на
                      // f3 се получава
                      // int *c=&r т.е. в указател c се записва
                      // адреса на r и затова във функцията
```

```

// main резултатът е видим в
// декларираната в нея променлива r
cout<< "Сумата на въведените числа за f3 е: "<<r<< endl;

f4(s1, s2, r);           // при извикването съгласно дефиницията на
                        // f4 се получава
                        // int &c=r (инициализация на псевдонима c)
                        // в главната програма резултатът е видим в
                        // декларираната в нея променлива r
cout<< "Сумата на въведените числа за f4 е: "<<r<< endl;
}

```

Задача 1

Напишете функция `f1` за въвеждане от клавиатура на елементите на масив от цели числа. Използвайте два параметъра – един входен за броя на елементите и един изходен – указател към началото на масива. В тялото на функцията използвайте `for` цикъл.

Напишете функция `f2` за извеждане на елементите на масив от цели числа със същите по брой и тип параметри като на функцията `f1`. В тялото на функцията използвайте `for` цикъл.

В главната програма декларирайте два масива от цели числа с не повече от 3 или 4 елемента и за всеки от тях извикайте функциите `f1` и `f2` съответно за попълване и извеждане на елементите му.

Задача 2

В тялото на `f1` от задача 1 заменете `for` цикъла с `while` цикъл, а в тялото на `f2` – с `do while` цикъл.

Задача 3

Допълнете кода на задача 1 с функция, която да връща чрез името си сумата от елементите на масив с целочислени елементи. Извикайте я за двата масива и предвидете в главната програма извеждане на резултата на екран.

Задача 4

Допълнете кода на задача 1 с функция, която да връща чрез входно-изходен параметър **псевдоним** (reference) най-малкия от елементите на масив с целочислени елементи. Извикайте я за двата масива и предвидете в главната програма извеждане на резултата на екран.

Задача 5

Допълнете кода на задача 1 с функция, която да връща чрез входно-изходен параметър **указател** най-малкия от елементите на масив с целочислени елементи. Извикайте я за двата масива и предвидете в главната програма извеждане на резултата на екран.