

Laboratorium nr 10

Temat: Funkcje cz.2.

Zakres laboratorium:

- przeciążanie nazw funkcji
- argumenty domyślne funkcji
- przekazywanie danych do funkcji przez wartość, wskaźnik i referencję
- przekazywanie tablic do funkcji
- zadania laboratoryjne

Przeciążanie nazw funkcji

Przeciążanie nazw funkcji

Przeciążenie funkcji – sytuacja, gdy zdefiniowanych jest kilka funkcji o takiej samej nazwie, różniących się zestawem parametrów (argumentów).

Przeciążanie funkcji jest często stosowane do tworzenia różnych funkcji o takiej samej nazwie, które przeprowadzają podobne zadania, ale na różnych typach danych.

Przykład:

```
int kwadrat (int x) { return x*x; }  
double kwadrat (double x) { return x*x; }
```

UWAGA: Możliwe jest **przeciążanie operatorów**, tak aby operowały na **obiektach** typów danych zdefiniowanych przez użytkownika.

Argumenty domyślne funkcji

Argumenty domyślne funkcji

Programista może określić, że dany argument funkcji jest **argumentem domyślnym** i dostarczyć dla niego wartość domyślną.

Argumenty domyślne muszą być położone najdalej z prawej strony na liście parametrów funkcji.

Argumenty domyślne powinny być określone wraz z pierwszym wystąpieniem nazwy funkcji – zazwyczaj w jej prototypie (czyli deklaracji) (w definicji funkcji już nie!!!).

Przykład:

```
int Objetosc_Pudelka(int dlug=1,int szer=1,int wys=1); //prototyp funkcji
```

```
main()
```

```
{
    //przykładowe wywołania funkcji Objetosc_Pudelka
    Objetosc_Pudelka();           //dlug=1,szer=1,wys=1           objetosc=1
    Objetosc_Pudelka(10);        //dlug=10,szer=1,wys=1          objetosc=10
    Objetosc_Pudelka(10,5);      //dlug=10,szer=5,wys=1          objetosc=50
    Objetosc_Pudelka(10,5,2);    //dlug=10,szer=5,wys=2          objetosc=100
}
```

```
int Objetosc_Pudelka(int dlug,int szer,int wys)           //definicja funkcji
```

```
{
    return dlug*szer*wys;
}
```

Przekazywanie danych do funkcji przez
wartość, wskaźnik i referencję

Przekazywanie argumentów do funkcji (wywoływanie funkcji):

- **przez wartość** (funkcja **pracuje na kopii** przekazywanej zmiennej, więc **NIE MA** możliwości jej modyfikowania)
- **przez wskaźnik** (funkcja **pracuje na oryginale** przekazywanej zmiennej, więc **MA** możliwość jej modyfikowania)
- **przez referencję** (funkcja **pracuje na oryginale** przekazywanej zmiennej, więc **MA** możliwość jej modyfikowania)

Domyślnie, zwykle obiekty (zmiennne) przekazywane są do funkcji przez wartość.

Ponieważ każda funkcja za pomocą instrukcji **return** zwraca tylko jedną wartość, więc funkcja otrzymująca argumenty przez wartość może zmodyfikować co najwyżej wartość jakiegoś jednego obiektu.

Chcąc, aby funkcja mogła zmienić więcej obiektów, należy ją wywołać z argumentami przekazanymi przez wskaźnik lub referencję.

Przykład:

```
int zmien_wartosc(int aa, int *bb, int &cc) //definicja funkcji
{
    aa=aa+100;
    *bb=*bb+100;
    cc=cc+100;

    return aa;
}
```

przez wartość przez wskaźnik przez referencję

```
main() //definicja funkcji main
{
    int a=5, b=10, c=15, d; //definicje zmiennych

    cout<<a<<b<<c<<d;

    d=zmien_wartosc(a, &b, c); //wywołanie funkcji

    cout<<a<<b<<c<<d;
}
```

Przed wywołaniem funkcji: a=5, b=10, c=15, d=? (śmieć)

Po wywołaniu funkcji: a=5, b=110, c=115, d=105

Przekazywanie tablic do funkcji

Przekazywanie tablic do funkcji

C++ automatycznie (domyślnie) przekazuje całe tablice do funkcji używając wywołania przez referencję, tj. wywołana funkcja może modyfikować wartości elementów w oryginalnych tablicach.

Pojedyncze elementy tablicy automatycznie są **przekazywane przez wartość**.

Tablicę w funkcji można odebrać na 2 sposoby: 1) jako tablicę, b) jako wskaźnik.

Przykład:

```
int tablica[24];
void modyfikuj_tablice_ref(int tab[], int rozmiar);           //odbieramy tablicę jako tablicę
void modyfikuj_tablice_wsk(int *tab, int rozmiar);           //odbieramy tablicę jako wskaźnik
void modyfikuj_element(int element);
```

```
main()
{
    modyfikuj_tablice_ref(tablica, 24);                       //wywołanie przez referencję
    modyfikuj_tablice_wsk(tablica, 24);                       //wywołanie przez wskaźnik
    modyfikuj_element(tablica[5]);                            //wywołanie przez wartość
}
```

```
void modyfikuj_tablice_ref(int tab[], int rozmiar)
{
    tab[5]=tab[5]+2;     //modyfikacja oryginalnej tablicy!!!
}
```

```
void modyfikuj_tablice_wsk(int *tab, int rozmiar)
{
    tab[5]=tab[5]+2;     //modyfikacja oryginalnej tablicy!!!
    *(tab+5)+=2;         //modyfikacja oryginalnej tablicy!!!
}
```

```
void modyfikuj_element(int element)
{
    element=element+2;   //modyfikacja kopii przekazanego elementu!!!
}
```

Zadania laboratoryjne