

## Język C++ zajęcia nr 4

### Zadanie CPP04 – do wykonania podczas zajęć:

Napisz dwie następujące funkcje **przeciążone**, obie o nazwie *rozw*:

1. funkcja wyznacza rozwiązanie równania liniowego postaci  $ax+b=0$
2. funkcja wyznacza rzeczywiste rozwiązania równania kwadratowego postaci  $ax^2+bx+c=0$ , przy założeniu że  $a \neq 0$ ,

Obie funkcje są typu **int** i zwracają **liczbę istniejących rozwiązań** w liczbach rzeczywistych dla danego równania (0, 1 lub 2). Współczynniki równania są argumentami funkcji przekazywanymi przez wartość, a **wartości rozwiązań są argumentami przekazywanymi przez referencję**. Przykładowe deklaracje funkcji:

1. `int rozw (double a, double b, double &x1);`
2. `int rozw (double a, double b, double c, double &x1, double &x2);`

Następnie napisz funkcję **main** wywołującą obie te funkcje z przykładowymi parametrami aktualnymi i drukującą otrzymane rozwiązania. Uruchom program.

---

### Operator zasięgu

Nazwa zadeklarowana w bloku zaślania identyczną nazwę globalną lub zadeklarowaną w otaczającym bloku zewnętrznym. Do zasłoniętej nazwy globalnej można odwoływać się używając **operatora zasięgu** `::` umieszczonego przed nazwą:

#### Dostęp do obiektu globalnego (zinterpretuj program):

```
#include <iostream>
using namespace std;
int x=2;                // obiekt globalny x
int main()
{
    int x=3;            // obiekt lokalny x
    cout << "globalne x=" << ::x << endl << " lokalne x=" << x;
}
```

```
globalne x=2
lokalne x=3
```

**Uwaga:** Poprzez użycie operatora zasięgu nie można uzyskać dostępu do zasłoniętej nazwy lokalnej.

# Klasy

**Pojęcie klasy.** Definicja klasy wprowadza nowy typ danych, który może być wykorzystywany w programowaniu na równi z klasycznymi typami wbudowanymi w składnię języka C++.

## Definicja klasy:

```
class nowy_typ
{
    // ... ciało klasy ...
};
```

Po zdefiniowaniu klasy można deklarować obiekty jej typu, podobnie jak w przypadku typów standardowych:

```
char x;                // x jest typu char
nowy_typ y;            // y jest typu nowy_typ
nowy_typ *p;           // p jest typu wskaźnik na nowy_typ
nowy_typ t[12];        // t jest tablicą elementów typu nowy_typ
```

Zasięg definicji klasy obejmuje najmniejszy blok lub blok funkcyjny, w którym klasa została zdefiniowana. Zasięg globalny mają klasy zdefiniowane na zewnątrz funkcji.

## Zasięg definicji klas:

```
class P                                // definicja klasy P o zasięgu globalnym
{
    int a;
};
int main()
{
    class N                            // definicja klasy N o zasięgu funkcji
    {
        int a;
    };
    {                                  // początek bloku -----
        class K                        // definicja klasy K o zasięgu bloku
        {
            int a;
        };
        P x1;
        N y1;
        K z1;
    }                                  // koniec bloku -----
    P x2;
    N y2;
    K z2;                             // błąd, deklaracja poza zasięgiem definicji klasy K
}
P x3;
N y3;                                // błąd, deklaracja poza zasięgiem definicji klasy N
K z3;                                // błąd, deklaracja poza zasięgiem definicji klasy K
```

## Składowe klasy

Ciało klasy zawiera składniki (**składowe** lub **komponenty**) klasy, którymi mogą być:

**Dane składowe** będące danymi różnych typów deklarowanymi wewnątrz definicji klasy. Występujące w klasie dane są niezależnie umieszczane w każdym obiekcie tej klasy. W deklaracji klasy nie mogą być inicjalizowane.

**Funkcje składowe**, którymi są funkcje deklarowane wewnątrz definicji klasy. Funkcje takie są umieszczane tylko jednokrotnie bez względu na liczbę obiektów klasy. Mają one bezpośredni dostęp poprzez nazwy do wszystkich składowych klasy.

## Dostęp do składowych klasy

Składową klasy można osiągnąć poprzez 1) ***nazwę obiektu***, 2) ***wskaźnik*** na obiekt lub 3) ***referencję*** do obiektu. W przypadkach 1) i 3) dostęp do składowych obiektu danej klasy uzyskiwany jest poprzez użycie operatora składowej, a w przypadku 2) poprzez operator wskaźnikowy składowej – podobnie jak w przypadku struktur lub unii. W wymienionych przypadkach stosowana jest składnia:

***obiekt . składowa***

***wskaźnik -> składowa***

***referencja . składowa***

### Użycie w ciele funkcji składowej:

- nazwy danej składowej bez operatora . lub -> oznacza odwołanie do danej składowej tego obiektu, dla którego wywołana została funkcja składowa,
- nazwy pewnej funkcji składowej bez pośrednictwa operatora . lub -> oznacza wywołanie tej funkcji dla tego obiektu, dla którego wywołana została funkcja, w której ciele znajduje się wspomniana nazwa funkcji.

**Dostęp do składowych (wersja 1) - wprowadź i uruchom program, zinterpretuj jego kod:**

```
#include <iostream>
using namespace std;
//----- Definicja klasy
class tramwaj
{
    public:
        //----- dane składowe
        int numer;
        int pasazerowie;
        //----- funkcje składowe
        void wsiada(int liczba_osob){pasazerowie += liczba_osob;}
        void wysiada(int liczba_osob){pasazerowie -= liczba_osob;}
};

int main()
{
    //----- Deklaracje obiektów
    tramwaj niebieski;
    tramwaj czerwony;
    //----- Wykorzystanie składowych

    niebieski.numer = 4;
    niebieski.pasazerowie = 0;
    czerwony.numer = 3;
    czerwony.pasazerowie = 0;
    niebieski.wsiada(15);
    czerwony.wsiada(20);
    niebieski.wysiada(3);
    czerwony.wsiada(5);
    czerwony.wysiada(8);
    cout <<niebieski.numer<<" "<<niebieski.pasazerowie<< endl;
    cout <<czerwony.numer<<" "<<czerwony.pasazerowie<<endl;
}
```

4 12

3 17

**Dostęp do składowych (wersja 2) - wprowadź i uruchom program (jako modyfikację poprzedniego programu), zinterpretuj jego kod:**

```
#include <iostream>
using namespace std;
//----- Definicja klasy
class tramwaj
{
public:
    //----- dane składowe
    int numer;
    int pasazerowie;
    //----- funkcje składowe
    void wsiada(int liczba_osob){pasazerowie += liczba_osob;}
    void wysiada(int liczba_osob){pasazerowie -= liczba_osob;}
};

int main()
{
    //----- Deklaracje obiektów
    tramwaj niebieski;
    tramwaj czerwony;

    tramwaj* p = &czerwony; // p jest wskaźnikiem na obiekt klasy tramwaj
    tramwaj& nowy = niebieski; // nowy jest referencją dla niebieski
    //----- Wykorzystanie składowych

    niebieski.numer = 4;
    niebieski.pasazerowie = 0;
    niebieski.wsiada(15);
    p->numer = 9;
    czerwony.pasazerowie = 0;
    p->wsiada(24);
    nowy.wysiada(5);
    cout << nowy.numer <<" "<< nowy.pasazerowie << endl;
    cout <<niebieski.numer<<" "<<niebieski.pasazerowie<< endl;
    cout <<czerwony.numer<<" "<<czerwony.pasazerowie<<endl;
}
```

```
4 10
4 10
9 24
```

## Kwalifikacja nazw składowych klasy

Jeżeli funkcja składowa zostanie **zdefiniowana** wewnątrz klasy, to kompilator nada jej formę funkcji otwartej **inline** – zobacz oba w/w programy „Dostęp do składowych” wersja 1 i 2.

Funkcja składowa klasy występuje częściej w formie funkcji zamkniętej, co jest uzyskiwane przez umieszczenie wewnątrz klasy jedynie **deklaracji** funkcji składowej, a jej **definicji** na zewnątrz klasy. W takim przypadku nazwa funkcji w definicji musi być **kwalifikowana** nazwą klasy z operatorem zasięgu:

***nazwa\_klasy :: nazwa\_składowej***

### Program „Kwalifikowanie nazw” – zinterpretuj poszczególne elementy kodu źródłowego.

Kwalifikowanie nazw:

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

//----- Definicja klasy
class tramwaj
{
    public:
        int osoby;                // liczba pasażerów wewnątrz
        void wsiada(int osoby);
        void wysiada(int osoby);
};

//----- Definicje funkcji składowych
void tramwaj::wysiada(int osoby)    // kwalifikowana nazwa funkcji
{
    tramwaj::osoby=tramwaj::osoby-osoby;    // kwalifikowana nazwa danej
    return;
}

void tramwaj::wsiada(int osoby)    // kwalifikowana nazwa funkcji
{
    tramwaj::osoby=tramwaj::osoby+osoby;    // kwalifikowana nazwa danej
    return;
}
```

```

int main()
{
    //----- Definicje obiektów
    tramwaj a;
    tramwaj b;
    //----- Wykorzystanie danych składowych
    a.osoby = 25;
    b.osoby = 16;
    cout << a.osoby << " " << b.osoby;
    //----- Wykorzystanie funkcji składowych
    b.wsiada(7);
    cout << "\n" << b.osoby;
    b.wysiada(3);
    cout << "\n" << b.osoby;
    a.wysiada(2);
    cout << "\n" << a.osoby;
    a.wsiada(11);
    cout << "\n" << a.osoby << " " << b.osoby;
}

```

**Oczekiwane wyniki:**

```

25 16

23

20

23

34 20

```

*Jaki charakter mają funkcje składowe klasy tramwaj – otwarty czy zamknięty? Jaka jest różnica pomiędzy funkcją otwartą i zamkniętą?*