

Klasa to typ zdefiniowany przez użytkownika zbudowany z dwóch rodzajów składowych:

- pól,
- metod.

```
class NazwaKlasy {  
    public:  
        //składowe publiczne – dostępne dla wszystkich  
        //np. interfejs użytkownika  
    private:  
        //składowe prywatne – dostępne tylko dla składowych tej klasy  
        //nie ma do nich dostępu na zewnątrz klasy  
        //szczegóły implementacyjne  
};
```

Domyślnie składowe klasy są **prywatne**.

Deklaracja metody może się odbyć wewnątrz klasy, albo poza jej obszarem i wtedy ma postać:

```
typMetody NazwaKlasy::nazwaMetody()  
{ //deklaracja metody  
}
```

Zdefiniowanie obiektu danej klasy ma postać:

```
NazwaKlasy nazwaObiektu;
```

Natomiast wywołanie konkretnej metody na stworzonym obiekcie odbywa się poprzez odwołanie do niej za pomocą kropki:

```
nazwaObiektu.nazwaMetody(parametry_aktualne_metody);
```

Taką szczególną metodą w klasie jest konstruktor. Konstruktor jest funkcją, która jest wywoływana zawsze, gdy tworzymy obiekt danej klasy. Jeśli klasa nie zawiera konstruktora, kompilator automatycznie go tworzy i nic on nie robi. Konstruktor nie pojawi się nigdzie w kodzie, jednak będzie on istniał w skompilowanej wersji programu i będzie wywoływany za każdym razem, gdy będzie tworzony obiekt klasy. Jeśli chcemy zmienić domyślne własności konstruktora i nadać polom w obiekcie wartości początkowe, należy utworzyć własny konstruktor dla klasy. Konstruktor nie ma typu i jego nazwa jest identyczna z nazwą klasy. Można tworzyć kilka konstruktorów dla jednej klasy (muszą się one jednak różnić parametrami wejściowymi – ilością lub typem argumentów). Przykład:

```
Class Osoba{  
    int wiek;  
    char imie[];  
    Public:  
        Osoba();
```

}

Zadanie 1

Napisać klasę o nazwie **punkt**, która będzie przechowywała współrzędne punktu na płaszczyźnie. Współrzędne mają być polami prywatnymi klasy. Należy umieścić w klasie następujące metody publiczne:

- konstruktor z dwoma parametrami;
- wektor – wyświetla wektor wyznaczony przez początek układu współrzędnych i ten punkt (np. [3,7]);
- odleglosc – oblicza odległość tego punktu od początku układu współrzędnych;
- osX – sprawdza, czy podany punkt leży na osi OX;
- osY – sprawdza, czy podany punkt leży na osi OY;
- cwiartka – zwraca numer ćwiartki, w której znajduje się punkt (chodzi o wnętrze ćwiartki, a jeżeli punkt leży na którejś z osi wtedy ta metoda zwraca 0).

Zaimplementować powyższe metody. Następnie utworzyć cztery obiekty typu punkt i przetestować dla nich wszystkie metody klasy.

Zadanie 2

Rozszerz klasę punkt o takie metody, aby w programie głównym możliwe było zdefiniowanie metody, która jako parametry wejściowe dostaje dwa punkty i oblicza odległość między nimi. Zdefiniuj taką metodę i przetestuj ją.

Zadanie 3

Zdefiniuj klasę **kolo**, która powinna zawierać:

- pola prywatne:
 - promień koła (typ float);
 - kolor wypełnienia (typ int);
- konstruktor bezparametrowy inicjujący wartość początkową pól w klasie (dowolne ustalone);
- konstruktor z jednym parametrem (promień, a kolor wypełnienia ustalony);
- konstruktor z dwoma parametrami (promień i kolor wypełnienia);
- metody publiczne:
 - Oblicz_Pole - obliczająca pole powierzchni koła;
 - Oblicz_Obwod - obliczająca obwód koła;
 - Ustaw_Promien - zmieniająca promień koła;
 - Ustaw_Kolor - zmieniającą kolor koła;
 - Zwroc_Promien – podającą wartość promienia;
 - Zwroc_Kolor – podającą wartość koloru.

Uruchom w programie głównym sekwencję instrukcji dla dwóch obiektów K1, K2: konstruktor bezparametrowy, porównanie pól, ustawienie tego samego koloru, porównanie pól, ustawienie różnych kolorów, porównanie pól. Użyj na końcu konstruktora z parametrami dla obiektów K3, K4 i pokaż różne warianty ich porównania. Spróbuj wykonać to samo z udziałem wskaźników na obiekty.

