

Ćwiczenie 4

Klasy i obiekty

1 Wprowadzenie

1.1 Klasy

Klasa (typ klasowy) jest strukturą składającą się ze stałej liczby elementów (składowych), którymi mogą być:

- pola - przechowują stan obiektów, mogą być typu prostego lub typu obiektowego,
- konstruktory - służą do tworzenia obiektów
- metody - służą do wykonywania operacji na obiektach.

1.1.1 Definicja klasy

```
kwalifikator class NazwaKlasy
{
    // ... deklaracja pól ...
    // ... definicja konstruktorów ...
    // ... deklaracja/definicja metod ...
}
```

Klasa z kwalifikatorem **public** jest klasą publiczną dostępną w każdym miejscu programu.

1.1.2 Specyfikatory dostępu do składowych klasy

- **public** – składowa publiczna - składowa klasy dostępna jest w każdym miejscu.
- **private** – składowa prywatna - składowa klasy dostępna jest tylko w danej klasie.
- **protected** – składowa chroniona - składowa klasy dostępna jest w danej klasie oraz we wszystkich klasach dziedziczących z danej klasy.
- *brak specyfikatora* – składowa pakietowa – składowa klasy dostępna jest w danej klasie i we wszystkich klasach danego pakietu.

1.1.3 Deklaracja pola

```
specyfikator_dostępu typPola nazwaPola;
```

1.1.4 Definicja konstruktora

```
specyfikator_dostępu nazwaKlasy(argumenty)
{
    //... treść (ciało) konstruktora...
}
```

1.1.5 Definicja metody

```
specyfikator_dostępu typ_zwracany nazwaMetody(argumenty)
{
    //... treść (ciało) metody ...
}
```

1.1.6 Przekazywanie argumentów

W języku Java argumenty przekazywane są wyłącznie przez wartość.

1.1.7 Przeciążanie konstruktorów i metod

W każdej klasie może zostać zdefiniowane więcej konstruktorów/metod o takiej samej nazwie, ale różniących się argumentami wejściowymi. W przypadku metod, mogą one również, ale nie muszą, różnić się typem zwracanym.

1.2 Obiekty

Obiekt powstaje podczas wykonywania programu i jest konkretnym egzemplarzem danej klasy.

1.2.1 Tworzenie obiektów

1. Deklaracja referencji do obiektu (zmiennej obiektowej):

```
NazwaKlasy referencjaDoObiektu;
```

2. Utworzenie obiektu za pomocą wyrażenia **new** i konstruktora odpowiedniej klasy i przypisanie do referencji wskaźnika do obiektu:

```
referencjaDoObiektu=new nazwaKlasy(argumenty);
```

1.2.2 Odwołania do składowych obiektu

```
referencjaDoObiektu.pole;
referencjaDoObiektu.metoda(argumenty);
```

1.2.3 Słowo kluczowe **this**

Słowo kluczowe **this** to referencja do bieżącego (aktualnego) obiektu, np.

```
this.pole;
this.metoda(argumenty);
```

2 Zadania

2.1

Napisz program wczytujący dane dziesięciu czytelników biblioteki (nazwisko, imię, numer karty bibliotecznej, liczba wypożyczonych książek). Następnie program powinien umożliwiać:

- wyświetlanie informacji o czytelniku z podanym numerem karty bibliotecznej,
- wyświetlanie informacji o wszystkich czytelnikach biblioteki,
- wyświetlenie informacji o czytelnikach, którzy nie wypożyczyli żadnej książki,
- zmianę liczby wypożyczonych książek przy wypożyczeniu kolejnych książek (program powinien uniemożliwić wypożyczenie powyżej 10 książek),
- zmianę liczby wypożyczonych książek przy oddaniu pewnej ilości książek.

Do tego celu wykorzystaj stworzoną przez siebie klasę **Czytelnik** z odpowiednimi składowymi.

2.2

Napisz program wczytujący dane pięciu samochodów (marka pojazdu, pojemność silnika, rok produkcji, numer rejestracyjny, nazwisko właściciela). Następnie program powinien umożliwiać:

- wyświetlanie informacji o samochodzie z podanym numerem rejestracyjnym,
- wyświetlanie informacji o wszystkich samochodach,
- zmianę właściciela pojazdu o podanym numerze rejestracyjnym,
- wyświetlenie informacji o pojazdach wyprodukowanych przed podanym rokiem,
- posortowanie pojazdów względem pojemności silnika (od najmniejszej do największej).

Do tego celu wykorzystaj stworzoną przez siebie klasę **Samochód** z odpowiednimi składowymi.

2.3

Napisz program wczytujący dane piętnastu wektorów w przestrzeni dwuwymiarowej (nazwa wektora, współrzędne x oraz y). Następnie program powinien umożliwiać:

- wyświetlanie informacji o wektorach,
- obliczenie długości wektorów,
- wyświetlenie wektorów, które są równoległe do osi X ,
- wyświetlenie wektorów, które są równoległe do osi Y ,
- sortowanie wektorów względem ich długości (od najkrótszego do najdłuższego),
- obliczenie iloczynu skalarnego dwóch wybranych wektorów.

Do tego celu wykorzystaj stworzoną przez siebie klasę **Wektor** z odpowiednimi składowymi.