

1. Definiowanie klas.

Definicja klasy wprowadza do programu nowy typ, a co bardziej istotne, daje programiście do dyspozycji nowe pojęcie, którego może używać przy budowanie swojej aplikacji. W Javie można definiować klasy na najwyższym poziomie zagnieżdżeń struktury programu, można też deklarować klasy zagnieżdżone, czyli takie, których definicja jest zawarta w innej klasie lub interfejsie (mogą to być klasy składowe, lokalne lub anonimowe).

2. Składnia deklaracji.

Najprostsza deklaracja klasy przyjmuje następującą postać

```
class Pusta{  
  
}
```

Taka postać oczywiście nie jest zbyt użyteczna. Jak widać deklaracja klasy zaczyna się od słowa **class** (poprzedzonego być może pewnymi modyfikatorami), po którym znajduje się nazwa klasy oraz treść klasy ujęta w nawiasy klamrowe.

Ponieważ zwykle chcemy, by klasy definiowane na najwyższym poziomie struktury programu były dostępne w całym programie, zwykle przed deklaracją takiej klasy umieszczamy modyfikator **public** (nazwa klasy musi się wtedy zgadzać z nazwą pliku w którym jest umieszczona).

W treści klasy umieszcza się deklaracje składowych klasy, to znaczy atrybutów, metod i konstruktorów. Te deklaracje mogą być poprzedzone modyfikatorami.

Deklaracja atrybutu składa się z (podanych w takiej właśnie kolejności) typu, nazwy i nieobowiązkowej części inicjującej. Deklaracja atrybutu kończy się średnikiem. Na przykład definicja klasy przechowującej imię i nazwisko osoby może wyglądać tak:

```
class Osoba {  
    String imie;  
    String nazwisko;  
}
```

Deklaracja metody zaczyna się od podania typu wyniku (lub słowa kluczowego **void**), identyfikatora metody, następnie otoczonej okrągłymi nawiasami listy parametrów formalnych (może być puste) oraz treści metody. Treść metody to po prostu instrukcja bloku. Załóżmy, że chcemy dodać do naszej klasy metody do odczytywania imienia i nazwiska:

```
class Osoba {
    String imie;
    String nazwisko;

    String imie() {
        return imie;
    }

    String nazwisko() {
        return nazwisko;
    }
}
```

Reguły składni Javy ogólnie zabraniają deklarowania dwu rzeczy w tym samym zasięgu z tą samą nazwą, ale zadeklarowanie metody i atrybutu o tej samej nazwie nie prowadzi do żadnych niejednoznaczności (przy wywołaniu metody zawsze trzeba podać po jej nazwie otoczoną nawiasami listę parametrów).

3. Przeciążanie nazw metod.

Przeciążaniem metod (ang. **Overloading**). Pozwala to najprościej mówiąc na tworzenie metod o takich samych nazwach, ale różnych parametrach. Analogicznie nieprawidłowe jest utworzenie w jednej klasie dwóch metod o identycznej nazwie i przyjmującej takie same parametry, a także metody o takiej samej nazwie i parametrach, ale różniące się **tylko** zwracanym typem. Poprawna jest więc poniższa klasa:

```
class Test {
    int dodaj(int a, int b) {
        return a + b;
    }

    double dodaj(double a, double b) {
        return a + b;
    }

    double dodaj(double a, int b) {
        return a + b;
    }
}
```

Zadanie:

Zaimplementuj aplikację wykorzystującą klasy, wykonaj przeciążenie kilku dowolnych metod.