

Z tej lekcji dowiesz się:

*co to jest przeciążanie metod;

*co to jest konstruktor

nauczysz się:

*wykorzystywać przeciążanie metod

*stosować konstruktory

Przypomnienie podstaw...

Aby zrozumieć poprzednią lekcję przypomnijmy sobie podstawy obiektowego programowania w Javie. W tym celu napiszmy prosty program:

Zadanie 1: Napisz program, który używa klas do przechowywania obiektów typu punkt o polach: x oraz y.

(na końcu instrukcji znajdują się rozwiązania)

Zadanie 2: Teraz, zgodnie z tym, czego nauczyliśmy się na poprzedniej lekcji, dodajmy metodę do klasy, która ustawia wartość x na "10", oraz drugą metodę, która ustawia wartość y na "20".

Przeciążanie metod

Przypomnieliśmy sobie w jaki sposób używamy metod. Teraz nauczymy się co to jest przeciążanie metod. Odpowiedź jest krótka: jest to definiowanie dwóch lub więcej metod pod tą samą nazwą, ale z innymi argumentami (parametrami), Np.: dla metody ustawX przekazywaliśmy parametr typu integer (nowy_x). Teoretycznie możemy dodać drugą metodę, która jako argumentu użyje innych parametrów. Pokażę to na przykładzie:

Zadanie 3: Program, który przeciąża metodę ustaw_x używając jako argumentu innego obiektu.

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt {
5             int x;
6             int y;
7             void ustawX(int nowy_x) {
8                 x = nowy_x;
9                 System.out.println("Zmieniono wartość pola 'x' na: " + x);
10            }
11            void ustawY(int nowy_y) {
12                y = nowy_y;
13                System.out.println("Zmieniono wartość pola 'y' na: " + y);
14            }
15            void ustaw(punkt punkt_arg) {
16                x = punkt_arg.x;
17                System.out.println("Zmieniono wartość pola 'x' na: " + x);
18            }
19        }
20        punkt punkt1 = new punkt();
21        punkt1.ustawX(10);
22        punkt1.ustawY(20);
23        punkt punkt2 = new punkt();
24        punkt2.ustaw(punkt1);
25    }
26 }
27 }
```

Zastanówmy się, co stało się w programie:

utworzyliśmy w sumie 2 obiekty klasy punkt (linijki 20 oraz 23)

ustawiliśmy parametry obiektu punkt1 (linijki 21 i 22)

ustawiliśmy parametr x dla obiektu punkt2 używając przeciążonej metody ustawX

Zauważ, że w klasie punkt są 2 metody o nazwie ustawX: jedna jako argument przyjmuje liczbę całkowitą, druga przyjmuje obiekt klasy punkt, z którego "wyciąga" sobie wartość parametru. Tego typu zabiegi (przeciążanie metod) używane są bardzo często, ponieważ programista musi zapamiętać tylko jedną nazwę funkcji, za to jako parametr podaje to, co mu jest wygodnie w tym momencie użyć.

Ćwiczenie do utrwalenia:

Zadanie 3: Napisz program, który analogicznie jak w powyższym przypadku używa przeciążonej metody `ustawY` przyjmującej jako argument obiekt klasy `punkt`.

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt {
5             int x;
6             int y;
7             void ustawX(int nowy_x){
8                 x = nowy_x;
9                 System.out.println("Zmieniono wartość pola 'x' na: " + x);
10            }
11            void ustawY(int nowy_y){
12                y = nowy_y;
13                System.out.println("Zmieniono wartość pola 'y' na: " + y);
14            }
15            void ustawX(punkt punkt_arg){
16                x = punkt_arg.x;
17                System.out.println("Zmieniono wartość pola 'x' na: " + x);
18            }
19            void ustawY(punkt punkt_arg){
20                y = punkt_arg.y;
21                System.out.println("Zmieniono wartość pola 'y' na: " + y);
22            }
23        }
24        punkt punkt1 = new punkt();
25        punkt1.ustawX(10);
26        punkt1.ustawY(20);
27        punkt punkt2 = new punkt();
28        punkt2.ustawX(punkt1);
29    }
30 }
```

Zadanie 4: Zmodyfikuj powyższy program tak, aby za pomocą jednej metody można było ustawić obydwa parametry (x oraz y) korzystając z gotowych metod (`ustawX` oraz `ustawY`).

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt {
5             int x;
6             int y;
7             void ustawX(int nowy_x){
8                 x = nowy_x;
9                 System.out.println("Zmieniono wartość pola 'x' na: " + x);
10            }
11            void ustawY(int nowy_y){
12                y = nowy_y;
13                System.out.println("Zmieniono wartość pola 'y' na: " + y);
14            }
15            void ustawX(punkt punkt_arg){
16                x = punkt_arg.x;
17                System.out.println("Zmieniono wartość pola 'x' na: " + x);
18            }
19            void ustawY(punkt punkt_arg){
20                y = punkt_arg.y;
21                System.out.println("Zmieniono wartość pola 'y' na: " + y);
22            }
23            void ustawXY(int nowy_x, int nowy_y){
24                ustawX(nowy_x);
25                ustawY(nowy_y);
26            }
27        }
28        punkt punkt1 = new punkt();
29        punkt1.ustawX(10);
30        punkt1.ustawY(20);
31        punkt punkt2 = new punkt();
32        punkt2.ustawX(punkt1);
33    }
34 }
```

Konstruktory

Konstruktor jest to metoda, która tworzy nowy obiekt danej klasy. Domyślnie (jeśli nie zdefiniujemy inaczej), dla każdego pola klasy przy tworzeniu nowego obiektu ustawiane są wartości domyślne dla każdego z tych pól. Często jednak potrzebujemy tworzyć nowe obiekty danej klasy i to w taki sposób, aby "już na starcie" wartości tych pól były zdefiniowane. W tym celu użyjemy właśnie konstruktora.

Zadanie 5. Napisz program, który korzystając z konstruktora inicjuje nowe obiekty klasy punkt wartościami: x=20, y=30.

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt {
5             int x;
6             int y;
7         }
8         punkt(int nx, int ny) {
9             x=20;
10            y=30;
11            System.out.print("Zainicjowano obiekt klasy punkt ");
12            System.out.println("x = " + nx + " oraz y= " + ny);
13        }
14    }
15    punkt punkt1 = new punkt(20,30);
16    }
17 }
18 }
19 }
20 }
```

Ćwiczenie dla utrwalenie wiadomości:

Zadanie 6: Napisz program, który używa konstruktora do tworzenia nowych obiektów klasy

punkt z tym, że użytkownik podaje wartości parametrów.

Konstruktory, podobnie jak każde inne metody można przeciążać, czyli może być np.: kilka konstruktorów, z których każdy przyjmuje inne argumenty. Programista, w zależności od tego, czym będzie dysponował, użyje odpowiedniego konstruktora.

Zadanie 7: Napisz program, który używa konstruktora (przeciążonego), który jako argument przyjmuje inny obiekt klasy punkt i tworzy nowy obiekt, który jest kopią tego, który jest argumentem.

Rozwiązania zadań

Zadanie1:

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt{
5             int x;
6             int y;
7         }
8     }
9 }
10
```

Zadanie 2:

Przeciążanie metod i konstruktory

Wpisany przez Administrator

```
1 package punkt2d;
2 public class Punkt2d {
3     public static void main(String[] args) {
4         class punkt{
5             int x;
6             int y;
7             void ustawX(int nowy_x){
8                 x = nowy_x;
9                 System.out.println("Zmieniono wartość pola 'x' na: " + x);
10            }
11            void ustawY(int nowy_y){
12                y = nowy_y;
13                System.out.println("Zmieniono wartość pola 'y' na: " + y);
14            }
15        }
16        punkt punkt1 = new punkt();
17        punkt1.ustawX(10);
18        punkt1.ustawY(20);
19    }
20 }
21
```