

# 1 Konstruktor

## 1.1 Deklaracja konstruktora

```
public class Klasa{  
    public Klasa(){  
    }  
    public Klasa(int p1, int p2){  
    }  
}
```

## 1.2 Wywołanie konstruktora

Wywołanie konstruktora tworzy nową instancję klasy (obiekt)

```
Klasa k = new Klasa();  
Klasa k1 = new Klasa(1,2);
```

## 1.3 this

Słowo kluczowe this określa aktualny obiekt

```
public class Klasa{  
    int i;  
    public Klasa(){  
        this.i = 6;  
    }  
}
```

# 2 Inicjalizacja

## 2.1 Inicjalizacja obiektu

Kod wykonywany podczas inicjalizacji obiektu

```
public class Klasa{
```

```
{  
  // Inicjalizacja  
}  
  
}
```

## 2.2 Statyczna inicjalizacja

Kod wykonywany podczas inicjalizacji klasy

```
public class Klasa{  
  static {  
    // Inicjalizacja statyczna  
  }  
}
```

## 3 Usuwanie obiektów

### 3.1 Garbage Collector

Życie obiektów zarządzane jest przez Garbage Collector. Nie istnieją destruktory, a obiekt usuwany jest w sytuacji gdy nie ma miejsca na nowe obiekty. Zostają usuwane obiekty, które nie posiadają żadnych odwołań.

### 3.2 Wymuszenie uruchomienia Garbage Collectora

```
System.gc();
```

W typowym programie nigdy nie powinno się sterować pracą Garbage Collectora

### 3.3 Finalizacja obiektów

W sytuacji gdy obiekty rezerwują pamięć w niestandardowy sposób należy ją zwolnić w metodzie finalize. Metoda ta jest wywoływana przez Garbage Collector przed usunięciem

```
protected void finalize() throws Throwable{  
  
}
```

## 4 Tablice

- `int[] t = new int[10];` //tablica typu int na 10 elementów
- `int[][] t = new int[4][4];` //dwuwymiarowa tablica
- `Klasa[] t = new Klasa[10];` //tablica na obiekty, zawiera puste referencje
- `int[] t = {1, 2, 3};` //inicjalizacja tablicy

## 5 Zadania do wykonania

1. Przetestować działanie konstruktorów z parametrami i bez
2. Określić kiedy jest dostępny konstruktor bezparametrowy
3. Przetestować działanie inicjalizacji i inicjalizacji statycznej
4. Określić kolejność wykonywania konstruktorów i inicjalizacji
5. Przetestować finalizację
6. Ustalić kiedy zostaje uruchomiony Garbage Collector
7. Ustalić co dzieje się przy ręcznym wywołaniu Garbage Collectora
8. Zapoznać się z dokumentacją klasy Object (`clone()`, `equals()`, `finalize()`, `toString()`)
9. Przetestować podstawowe metody klasy Object w stworzonej klasie
10. Stworzyć tablicę typów prostych i odwołać się do jej elementów
11. Stworzyć tablicę obiektów i odwołać się do jej elementów
12. Skopiować zawartość tablicy
13. Określić różnicę pomiędzy płytkim i głębokim porównywaniem tablic
14. Wyszukać element z tablicy obiektów
15. Posortować elementy w tablicy obiektów
16. Zapoznać się z metodami klasy Arrays