

Z tej lekcji dowiesz się:

- co to są i do czego służą składowe statyczne

Nauczysz się

- Stosować pola i metody statyczne

Składowe statyczne to pola i metody, które **mogą istnieć i działać wtedy, kiedy nie istnieje obiekt danej klasy**. W takim przypadku, takie **pole lub metoda jest wspólne dla wszystkich obiektów danej klasy**.

Jedyną zaś różnicą w zapisie, pomiędzy metodą czy polem statycznym a "zwykłym" jest wyraz ***static***, który znajduje się na początku, ale po specyfikatorze dostępu.

np.:

```
public static void main
```

lub

```
static double metoda(double arg1, double arg1, double arg1){}
```

---

## Metody statyczne

Aby to zobrazować posłużę się przykładem.

Zad. 1.: Napisz program, w którym:

1) stworzysz program główny **Statyczne**

2) stworzysz publiczną klasę **algebra** w oddzielnym pliku, w której:

a) stworzysz metodę statyczną **dodawanie**, która będzie dodawała do siebie dwie wartości liczbowe przekazane jej przez argument **I1** i **I2**.  
Metoda zwraca wynik typu

***float***

.

b) stworzysz metodę statyczną ***odejmowanie***, która będzie dodawała do siebie dwie wartości liczbowe przekazane jej przez argument ***I1*** i ***I2***.

Metoda zwraca wynik typu

***float***

.

3) Z poziomu programu

a) utworzysz klasyczny obiekt ***algebra1***, w którym użyjesz obydwu metod

b) użyjesz metod bez tworzenia obiektu klasy algebra

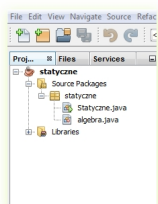
Jeśli nie pamiętasz, w jaki sposób tworzymy klasy publiczne przypomnij sobie lekcję nt. [Specyfikatory dostępu i pakiety](#)

(otworzy się w nowym oknie przeglądarki)

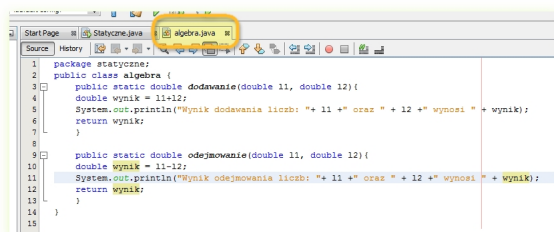
**Bierne i bezmyślne przepisywanie kodu z listingu nie jest dobrą metodą nauki. Postaraj się do punktu 2.b napisać program samodzielnie, w ramach powtórki.**

Oto listingi tego programu:

1) Tworzymy klasę główną **Statyczne** oraz 2) klasę publiczną **algebra**



2a) oraz 2b) Piszemy kod statycznych metod **dodawanie** i **odejmowanie** w klasie publicznej **algebra**



3) W kodzie klasy Statyczne następuje wywołanie metod: przez odwołanie do obiektu oraz przez odwołanie do metody statycznej klasy algebra.

# Składowe statyczne

Wpisany przez Administrator

```
1 package statyczne;
2 public class Statyczne {
3     public static void main(String[] args) {
4         /* sposób używany do tej pory: tworzymy obiekt danej klasy
5          * po czym przez odwołanie do tego obiektu wywołujemy metody tej klasy
6          */
7         algebra algebra1 = new algebra(); //stworzenie obiektu klasy algebra
8         algebra1.dodawanie(4,67); //wywołanie metody klasy algebra
9         //poprzez odwołanie do obiektu tej klasy: dodawanie
10        algebra1.odejmowanie(6784, 976.345); //wywołanie metody klasy algebra
11        //poprzez odwołanie do obiektu tej klasy: odejmowanie
12
13        /* sposób nowy: użycie metod statycznych. Dzięki temu mamy dostęp do metod
14         * bez potrzeby tworzenia obiektu danej klasy
15         */
16        algebra.dodawanie(432.7, 76784.89); //wywołanie metody statycznej
17        algebra.odejmowanie(67533.78, 7895.89); //wywołanie metody statycznej
18    }
19 }
20
```

Oto wynik działania programu dla przykładowych danych:

```
Output - statyczne (run) | Tasks
run
Wynik dodawania liczb: 4.0 oraz 67.0 wynosi 71.0
Wynik odejmowania liczb: 6784.0 oraz 976.345 wynosi 6807.655
Wynik dodawania liczb: 432.7 oraz 76784.89 wynosi 77217.59
Wynik odejmowania liczb: 67533.78 oraz 7895.89 wynosi 66637.89
BUILD SUCCESSFUL (total time: 0 seconds)
```

Uklaszczenie obiektów i zmiennej statycznej, oraz użycie metod statycznych, aby zobaczyć, jak to działa, kliknij tutaj

```
1 package statyczne;
2 public class algebra {
3     public static double dodawanie(double l1, double l2){
4         double wynik = l1+l2;
5         System.out.println("Wynik dodawania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
6         return wynik;
7     }
8
9     public static double odejmowanie(double l1, double l2){
10        double wynik = l1-l2;
11        System.out.println("Wynik odejmowania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
12        return wynik;
13    }
14 }
15
```



```
1 package statyczne;
2 public class algebra {
3     public static double dodawanie(double l1, double l2){
4         double wynik = l1+l2;
5         System.out.println("Wynik dodawania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
6         return wynik;
7     }
8
9     public static double odejmowanie(double l1, double l2){
10        double wynik = l1-l2;
11        System.out.println("Wynik odejmowania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
12        return wynik;
13    }
14 }
15
```

Klasa statyczna. Wróćmy teraz do klasy Statyczne, aby zobaczyć, jakie są tego

```
1 package statyczne;
2 public class Statyczne {
3     public static void main(String[] args) {
4         /* sposób używany do tej pory: tworzymy obiekt danej klasy
5          * po czym przez odwołanie do tego obiektu wywołujemy metody tej klasy
6          */
7         algebra algebra1 = new algebra(); //stworzenie obiektu klasy algebra
8         algebra1.dodawanie(4,67); //wywołanie metody klasy algebra
9         //poprzez odwołanie do obiektu tej klasy: dodawanie
10        algebra1.odejmowanie(6784, 976.345); //wywołanie metody klasy algebra
11        //poprzez odwołanie do obiektu tej klasy: odejmowanie
12
13        /* sposób nowy: użycie metod statycznych. Dzięki temu mamy dostęp do metod
14         * bez potrzeby tworzenia obiektu danej klasy
15         */
16        algebra.dodawanie(432.7, 76784.89); //wywołanie metody statycznej
17        algebra.odejmowanie(67533.78, 7895.89); //wywołanie metody statycznej
18    }
19 }
20
```

Notepad informuje nas o błędzie. Przyjrzyjmy się bliżej temu komunikatowi:

```
10 algebra.odejmowanie(6784, 976.345); //wywołanie metody klasy algebra
11 //poprzez odwołanie do obiektu tej klasy: odejmowanie
12
13 /* sposób nowy: użycie metod statycznych. Dzięki temu mamy dostęp do metod
14 * bez potrzeby tworzenia obiektu danej klasy
15 */
16 algebra.dodawanie(432.7, 76784.89); //wywołanie metody statycznej
17 algebra.odejmowanie(67533.78, 7895.89); //wywołanie metody statycznej
18 }
19 }
20
```

Non-static method odejmowanie(double,double) cannot be referenced from a static context

```
1 package statyczne;
2 public class algebra {
3     static double pi = 3.14;
4     public static double dodawanie(double l1, double l2){
5         double wynik = l1+l2;
6         System.out.println("Wynik dodawania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
7         return wynik;
8     }
9
10    public static double odejmowanie(double l1, double l2){
11        double wynik = l1-l2;
12        System.out.println("Wynik odejmowania liczb: "+ l1 +" oraz " + l2 +" wynosi " + wynik);
13        return wynik;
14    }
15 }
16 }
17
```

ad. 2) i 3)

```

1 package statyczne;
2 public class Statyczne {
3     public static void main(String[] args) {
4         algebra algebra1 = new algebra(); //stworzenie obiektu klasy algebra
5     }
6     /*
7     * pod spodem sprawdzenie wartości pola pi dla obiektu Algebra1
8     */
9     System.out.print("Wartość pola statycznego pi dla obiektu algebra1 ");
10    System.out.println("klasy algebra " + algebra1.pi);
11
12    double promien = 100; //zmienia przechowujaca długość promienia
13
14    /*
15    * pod spodem znajduje się odwołanie do statycznego pola pi
16    * klasy Algebra, które ustalone jest na 3,14. Odwołanie wygląda tak:
17    * pobieramy wartość tego pola (statycznego) i mnożymy jego wartość
18    * przez kwadrat promienia
19    */
20    double pole_kola = algebra.pi * promien * promien;
21    System.out.print("Pole kola dla promienia równego: ");
22    System.out.println(promien + " wynosi " + pole_kola);
23
24    }
25 }

```

Oto wynik wywołania programu:

```

Output - statyczne (run) | Tasks
run:
Wartość pola statycznego pi dla obiektu algebra1 klasy algebra 3.14
Pole kola dla promienia równego: 100.0 wynosi 31400.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

1 package statyczne;
2 public class Statyczne {
3     public static void main(String[] args) {
4         algebra algebra1 = new algebra(); //stworzenie obiektu klasy algebra
5     }
6     /*
7     * pod spodem sprawdzenie wartości pola pi dla obiektu Algebra1
8     */
9     System.out.print("Wartość pola statycznego pi dla obiektu algebra1 ");
10    System.out.println("klasy algebra " + algebra1.pi);
11
12    double promien = 100; //zmienia przechowujaca długość promienia
13
14    double pole_kola = algebra.pi * promien * promien;
15    System.out.print("Pole kola dla promienia równego: ");
16    System.out.println(promien + " wynosi: " + pole_kola);
17
18    algebra.pi = 15; //zmiana wartości pola statycznego przez odwołanie "statyczne"
19
20    /*
21    * pod spodem sprawdzenie wartości pola pi dla obiektu Algebra1
22    */
23    System.out.print("Wartość pola statycznego pi dla obiektu algebra1 ");
24    System.out.println("klasy algebra " + algebra1.pi);
25
26    /*
27    * pod spodem sprawdzenie wartości pola pi na podstawie wyniku działania
28    */
29    pole_kola = algebra.pi * promien * promien;
30    System.out.print("Pole kola dla promienia równego: ");
31    System.out.println(promien + " wynosi: " + pole_kola);
32
33    //zmiana wartości pola statycznego przez odwołanie poprzez obiekt klasy
34    algebra1.pi = 92;
35
36    /*
37    * pod spodem sprawdzenie wartości pola pi dla obiektu Algebra1
38    */
39    System.out.print("Wartość pola statycznego pi dla obiektu algebra1 ");
40    System.out.println("klasy algebra " + algebra1.pi);
41
42    /*
43    * pod spodem sprawdzenie wartości pola pi na podstawie wyniku działania
44    */
45    pole_kola = algebra.pi * promien * promien;
46    System.out.print("Pole kola dla promienia równego: ");
47    System.out.println(promien + " wynosi: " + pole_kola);
48
49    }
50 }

```

Wynik wywołania programu:

```

Output - statyczne (run) | Tasks
run:
Wartość pola statycznego pi dla obiektu algebra1 klasy algebra: 3.14
Pole kola dla promienia równego: 100.0 wynosi: 31400.0
Wartość pola statycznego pi dla obiektu algebra1 klasy algebra: 15.0
Pole kola dla promienia równego: 100.0 wynosi: 150000.0
Wartość pola statycznego pi dla obiektu algebra1 klasy algebra: 92.0
Pole kola dla promienia równego: 100.0 wynosi: 920000.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

**Có zyskujemy i có "tracimy" używając statycznych metod i pól?**

Jeśli przypomnisz sobie pierwszy program z zadania 1., to łatwo zauważysz, że łatwiej i wygodniej było nam wykonać dodawanie i odejmowanie "w locie" niż gdybyśmy musieli tworzyć obiekt tej klasy, który jest niepotrzebny, (jedynie pewna metoda klasy jest potrzebna). "Odchudza" to nasz program z niepotrzebnie zaalokowanej pamięci (na obiekt klasy) oraz odciąża procesor w wykonywaniu instrukcji konstruktora.

Elementy statyczne lokowane są w oddzielnym bloku pamięci nawet wtedy, kiedy nie istnieje

obiekt danej klasy. W normalnych warunkach, komputer "zużywał" pamięć dopiero, kiedy trzeba w niej było ulokować obiekt danej klasy. jeśli w trakcie działania programu taki obiekt tej klasy nie powstał, pamięć nie była obciążona.

Drugą kwestią jest to, że statyczne metody i pola są wspólne dla ewentualnych wszystkich obiektów danej klasy. Oznacza to, że zmieniając np. wartość statycznego pola, zmienia się ono dla wszystkich obiektów tej klasy. Nie jest to pole prywatne, ale wspólne. Może to być bardzo dużym plusem i ułatwieniem (np.: w bazie danych przedsiębiorstwa wartość podatku VAT jako pola statycznego zmienia się z 23 na 18% dla wszystkich klas, nie trzeba zmieniać tego ręcznie oddzielnie w każdej klasie).

Czyli statyczne składowe mogą być bardzo dużym ułatwieniem, ale trzeba pamiętać, że ich używanie wiąże się z pewnymi konsekwencjami w strukturze kodu oraz w działaniu programu.

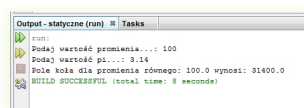
---

Zadania do utrwalenia wiadomości:

Zad. 6.: Zmodyfikuj powyższy program. Niech użytkownik będzie mógł:

- 1) zmienić wartość pola **promien** tak, aby program liczył pole kół o różnych promieniach
- 2) zmienić wartość pola **pi**

3) Zastanów się, czy można coś zrobić aby uniemożliwić lub ograniczyć możliwość ingerencji (zarówno z poziomu kodu jak również z poziomu użytkownika) w wartość pola statycznego  $\pi$  tak, aby np. przez pomyłkę nikt nie zmienił wartości pola  $\pi$  i nie doszedł do błędnych wyników licząc pole koła.



Zad. 7 Zmodyfikuj powyższy program: dodaj do programu możliwość wyliczenia objętości walca. Zastanów się: jakie dane będą potrzebne? Jakie mogą być statyczne a jakie może podać użytkownik?

Zad. 8: Zmodyfikuj powyższy program:

1) przenieś funkcje geometryczne do klasy publicznej **geometria**, znajdującej się w osobnym pliku tworząc z nich statyczne metody

2) w klasie publicznej **algebra** pozostaw cztery stworzone metody statyczne: **dodawanie**, **odejmowanie**, **iloczyn**, **iloraz**.



Zad. 9.: Dla **ambitnych**: zmodyfikuj powyższy program:

1) dodaj do programu możliwość obliczania silni danej liczby.

2) zastanów się, w której klasie będzie ta metoda? Może trzeba dodać kolejną klasę?

3) zastanów się, jakiego typu będzie zarówno zmienna jak również metoda (czyli jakiego typu wynik metoda będzie zwracała)

Podpowiedź: **silnia**:

$$5! = 1*2*3*4*5$$

$$10! = 1*2*3*4*5*6*7*8*9*10$$

## Rozwiązania zadań:

### Zad. 6.:

```
1 package statyczne;
2 import java.io.*;
3 public class Statyczne {
4     public static void main(String[] args) {
5         algebra algebra1 = new algebra();
6
7         double promien=0;//zmienna przechowująca długość promienia
8         double pole_kola=0;
9
10        Reader r = new BufferedReader (new InputStreamReader (System.in));
11        StreamTokenizer wpro = new StreamTokenizer(r);
12
13        try{
14            System.out.print ("Podaj wartość promienia...: ");
15            wpro.nextToken();
16            promien = wpro.nval;
17        }catch (IOException e){}
18        try{
19            System.out.print ("Podaj wartość pi...: ");
20            wpro.nextToken();
21            algebra.pi = wpro.nval;
22        }catch (IOException e){}
23
24        pole_kola = algebra.pi * promien * promien;
25        System.out.print("Pole kola dla promienia równego: ");
26        System.out.println( promien + " wynosi: " + pole_kola);
27
28    }
29
30 }
```