

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

ggoralski.com

# Metody

Metody, parametry, zwracanie wartości

# Metody - co to jest i po co?

- \*Metoda to wydzielona część klasy, mająca za zadanie wykonanie określonych operacji.
- \*Program musi posiadać przynajmniej jedną metodę: **main**.
- \*Inne metody tworzy się w celu wydzielenia fragmentów kodu, które wykonują jakieś konkretne operacje
- \*Dzielenie kodu na wiele metod niesie liczne korzyści, takie jak:
  - \*podzielenie klasy na pewne logiczne części zwiększa przejrzystość i czytelność kodu
  - \*pozwala znacznie ograniczyć powtarzanie takich samych (lub bardzo podobnych) fragmentów kodu
  - \*umożliwia łatwy dostęp do poszczególnych funkcjonalności **obiektów** (będzie na ten temat mowa w dalszej części kursu)

# Dwie podstawowe kategorie metod:

- \* Metody które zwracają wartości
- \* Metody które nie zwracają wartości

# Tworzenie i wywołanie metody

Będzie o tym później

Rodzaj zwracanej wartości, jeśli metoda nie zwraca wartości to: `void`

```
private static int podajSumeLiczby(int[] liczby) {  
    ....  
}
```

Elementy które przekazujemy do metody (parametry, argumenty) jeśli jest ich więcej niż 1, oddzielamy je przecinkami,

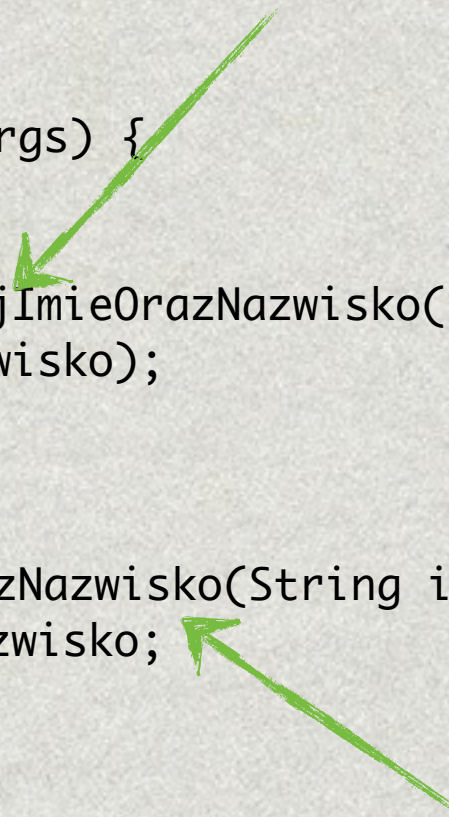
\* Przykład:

```
int[] tablicaLiczby = {1,2,3,4,5};  
int suma = podajSumeLiczby(tablicaLiczby);
```

# Przykład

Wywołujemy metodę, której przekazujemy dwa parametry typu String i odbieramy jeden String

```
public class Słowa {  
  
    public static void main(String[] args) {  
        String imie = "Adam";  
        String nazwisko = "Słodowy";  
        String imieOrazNazwisko = podajImieOrazNazwisko(imie, nazwisko);  
        System.out.println(imieOrazNazwisko);  
    }  
  
    private static String podajImieOrazNazwisko(String imie, String nazwisko) {  
        String polaczone = imie+" "+nazwisko;  
        return polaczone;  
    }  
}
```



Definiujemy metodę, która przyjmuje dwa parametry typu String i zwraca jeden String

# Przykład:

- \* Napisz program, który:
  - \* pobiera od użytkownika określoną liczbę liczb - użytkownik podaje ile to liczb, może też podać nowy zestaw liczb
  - \* dla podanych liczb użytkownik może uzyskać (menu):
    - \* największą liczbę
    - \* najmniejszą liczbę
    - \* średnią arytmetyczną
    - \* sumę liczb
    - \* wariancję
    - \* od razu wszystkie powyższe wartości
  - \* program działa dopóki użytkownik nie wybierze z menu opcji zakończenia

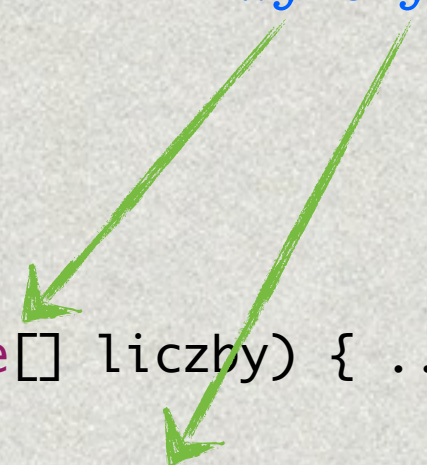
# Jak to zrobić?

- \* W programie wydzielimy metody które będą realizować poszczególne rodzaje obliczeń
- \* Program powinien posiadać główną pętlę wykonującą się aż do decyzji użytkownika o zakończenia. Znajdzie się tam:
  - \* wywołanie wydruku menu,
  - \* pobranie decyzji od użytkownika
  - \* zestaw instrukcji (`if`, `else if`) wywołujących odpowiednie metody
- \* Wartości, które są obliczane przez niektóre metody mogą być wykorzystane przez inne metody - np. przy obliczaniu wariancji potrzebujemy średniej arytmetycznej.

# szkielet

```
public class Statystyka {  
    public static void main(String[] args) {  
        .... (deklaracje)  
        while (....) { ... }  
    }  
  
    ... wyswietlMenu() { ... }  
  
    ... wprowadzNoweDane() { ... }  
  
    ... podajNajwiekszaLiczbe(double[] liczby) { ... }  
  
    ... podajNajmniejszaLiczbe(double[] liczby) { ... }  
  
    ... podajSumeLiczby(double[] liczby) { ... }  
  
    ... podajSredniaArytmetyczna(double[] liczby) { ... }  
  
    ... podajWariancje(double[] liczby) { ... }  
  
    ... podajWszystko(double[] liczby) { ... }  
}
```

Przekazujemy do metod  
wartości na których będą  
wykonywane obliczenia





# metoda main

```
public static void main(String[] args) {
    String opcja = "";
    double[] liczby = null;
    Scanner skaner = new Scanner(System.in);

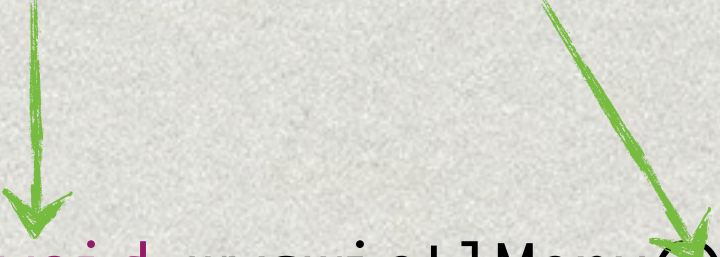
    while (!opcja.equalsIgnoreCase("k")) {
        wyswietlMenu();
        opcja = skaner.nextLine();
        if (liczby==null && !opcja.equalsIgnoreCase("k")) liczby=wprowadzNoweDane();
        else if (opcja.equalsIgnoreCase("d")) liczby=wprowadzNoweDane();
        else if (opcja.equalsIgnoreCase("m")) podajNajmniejszaLiczbe(liczby);
        else if (opcja.equalsIgnoreCase("n")) podajNajwiekszaLiczbe(liczby);
        else if (opcja.equalsIgnoreCase("s"))
            System.out.println("Średnia arytmetyczna liczb to: "+
                podajSredniaArytmetyczna(liczby));
        else if (opcja.equalsIgnoreCase("r"))
            System.out.println("Suma liczb to: "+podajSumeLiczb(liczby));
        else if (opcja.equalsIgnoreCase("w"))
            System.out.println("Wariancja dla podanych liczb to: "+
                podajWariancje(liczby));
        else if (opcja.equalsIgnoreCase("a")) podajWszystko(liczby);
        else if (opcja.equalsIgnoreCase("k")) System.out.println("Koniec!");
        else System.out.println("Nie rozumiem, spróbuj ponownie");
    }
}
```

Komunikat można  
zaimplementować  
w metodzie lub nie\*

\* Zależy to np. od tego czy wartość obliczana w metodzie będzie jeszcze do czegoś wykorzystywana, jeśli tak to komunikat mógłby być zbędny, a nawet niepożądany.

# metoda `wyswietlMenu`

Ta metoda niczego nie zwraca, ani nie potrzebuje danych wejściowych



```
private static void wyswietlMenu() {  
    System.out.println("Co chcesz zrobić?\n" +  
        "d - wprowadź nowe dane \n" +  
        "n - podaj największą liczbę \n" +  
        "m - podaj najmniejszą liczbę \n" +  
        "s - podaj średnią arytmetyczną \n" +  
        "r - podaj sumę liczb \n" +  
        "w - podaj wariancję \n" +  
        "a - podaj wszystkie powyższe \n" +  
        "k - koniec programu");  
}
```

# metoda wprowadzNoweDane

zwraca tablicę wprowadzonych wartości

nie potrzebuje danych wejściowych

```
private static double[] wprowadzNoweDane() {
    Scanner skaner = new Scanner(System.in);
    System.out.println("Podaj ile liczb chcesz wpisać ");
    String liczbaLiczby = skaner.nextLine();
    int dLTabeli = Integer.parseInt(liczbaLiczby);
    double[] liczby = new double[dLTabeli];
    for (int i=0; i<liczby.length; i++) {
        System.out.println("Podaj "+ (i+1) +" liczbę z "+dLTabeli);
        String liczba = skaner.nextLine();
        liczby[i] = Double.parseDouble(liczba);
    }
    return liczby;
}
```

jeśli metoda coś zwraca, to instrukcja `return` musi znaleźć się na końcu - określa ona co metoda zwraca

# metody `podajNajwiekszaLiczbe`, `podajNajmniejszaLiczbe`

przekazujemy metodzie dane potrzebne do obliczeń

```
private static void podajNajwiekszaLiczbe(double[] liczby) {  
    double max = liczby[0];  
    for (double l : liczby) {  
        if (l > max) {  
            max = l;  
        }  
    }  
    System.out.println("Największa liczba to: "+max);  
}
```

obliczana wartość nie będzie nam potrzebna do innych obliczeń, metoda może nic nie zwracać, a komunikat możemy zaimplementować wewnątrz niej

```
private static void podajNajmniejszaLiczbe(double[] liczby) {  
    double min = liczby[0];  
    for (double l : liczby) {  
        if (l < min) {  
            min = l;  
        }  
    }  
    System.out.println("Najmniejsza liczba to: "+min);  
}
```

# metoda `podajSredniaArytmetyczna`

zwraca średnią - będzie nam potrzebna  
np. do liczenia wariancji



```
private static double podajSredniaArytmetyczna(double[] liczby) {  
    double srednia = podajSumeLiczby(liczby)/liczby.length;  
    return srednia;  
}
```



do obliczenia średniej potrzebna nam suma  
liczb - uzyskamy ją z osobnej metody

# metoda podajSumeLiczb

zwraca sumę - będzie nam potrzebna np.  
do liczenia wariancji



```
private static double podajSumeLiczb(double[] liczby) {  
    double sum = 0;  
    for (double l : liczby) {  
        sum=sum+l;  
    }  
    return sum;  
}
```

# Zadanie

- \* Dokończyć program

# metoda podajWariancje

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

```
private static double podajWariancje(double[] liczby) {  
    double wariancja;  
    double srednia = podajSredniaArytmetyczna(liczby);  
    double sum=0.0;  
    for (double l : liczby) {  
        sum=sum+(l-srednia)*(l-srednia);  
    }  
    wariancja = sum/(liczby.length-1);  
    return wariancja;  
}
```



# metoda podajWszystko

```
private static void podajWszystko(double[] liczby) {  
    podajNajmniejszaLiczbe(liczby);  
    podajNajwiekszaLiczbe(liczby);  
    System.out.println("Średnia arytmetyczna liczb to:  
        "+podajSredniaArytmetyczna(liczby));  
    System.out.println("Suma liczb to: "+podajSumeLiczb(liczby));  
    System.out.println("Wariancja dla podanych liczb to:  
        "+podajWariancje(liczby));  
}
```