

Podstawy i języki programowania

Laboratorium 7 - wprowadzenie do metod

mgr inż. Krzysztof Szwarz

krzysztof@szwarz.net.pl

Sosnowiec, 23 listopada 2018

Funkcja, a metoda

Funkcją określamy pewien wydzielony fragment kodu, który może zostać wykonany z różnych miejsc programu. W paradygmacie programowania obiektowego funkcje są ściśle związane z klasami i nazywają się **metodami**. Przykładem metody jest **println**, wykorzystywana podczas wypisywania tekstu na ekran. Metoda składa się ze specyfikatora dostępu (public, private, protected, default), zwracanego typu danych (void, int, double, ...), nazwy metody, zwracanej wartości (dla typu innego niż void) oraz parametrów w postaci:

```
typParametru nazwaParametru
```

Po co nam metody?

- 1 Pozwalają nam zwiększyć czytelność kodu.
- 2 Ograniczają powielanie się kodu (jego redundancję).
- 3 Ułatwiają utrzymanie aplikacji.
- 4 Zmniejszają prawdopodobieństwo wystąpienia błędów.

Przykład

```
public static void main(String[] args)
{
    naszaMetoda(); // wywołanie metody
}
public static void naszaMetoda() // typ
    void - nic nie zostanie zwrócone
{
    System.out.println("Wywołano
        metoda");
}
```

Przykład 2

```
public static void main(String[] args)
{
    int wynik = dodajLiczby(2, 7);
    System.out.println(wynik); // 9
}
public static int dodajLiczby(int
    liczba1, int liczba2)
{
    return liczba1+liczba2;
}
```

Przykład 3

```
public static void main(String[] args)
{
    int wynik = dodajLiczby(2, 7);
    System.out.println(wynik); // 9
}
public static int dodajLiczby(int...
    liczby)
{
    return liczby[0]+liczby[1];
}
```

Przykład 4

```
public static void main(String[] args)
{
    int wynik = dodajLiczby("tekst", 2,
        7);
    System.out.println(wynik); // 9
}
public static int dodajLiczby(String
    tekstDoWypisania, int... liczby)
{
    System.out.println(tekstDoWypisania);
    // tekst
    return liczby[0]+liczby[1];
}
```

Parametry, a argumenty

```
public static void main(String[] args)
{
    int wynik = dodajLiczby(2); // 2 to
        argument
}
public static int dodajLiczby(int
    liczba) // liczba to parametr
{
    return liczba;
}
```


- 1 Napisz program konwertujący liczbę w systemie dziesiętnym na liczbę w systemie o podstawie 2, 4 oraz 8. Niech korzysta on z metody zawierającej uogólniony algorytm uzależniony od otrzymanego parametru (podstawy).
- 2 Napisz metodę weryfikującą czy wprowadzona przez użytkownika wartość jest typu int i jest mniejsza od 100.
- 3 Napisz metodę proszącą użytkownika o wpisanie liczby typu double do czasu, gdy nie wpisze on poprawnej wartości. Niech zwraca ona pobraną liczbę.
- 4 Napisz przelicznik złotych na euro i dolary. Niech każda opcja będzie osobną metodą zwracającą obiekt klasy **BigDecimal**.

Przekazywanie parametrów

Parametry w Javie są zawsze przekazywane przez wartość.

Przykład 1

```
public static void main(String[] args)
{
    int liczba = 5;
    zmienLiczbe(liczba);
    System.out.println(liczba); // 5
}
public static void zmienLiczbe(int
    liczba)
{
    liczba=10;
}
```

Przykład 2

```
public static void main(String[] args)
{
    int[] tab = {1};
    zmienTablice(tab);
    System.out.println(tab[0]); // 2
}
public static void zmienTablice(int[]
    tab)
{
    tab[0] = 2;
}
```

Przykład 3

```
public static void main(String[] args)
{
    int [] tab = {1};
    zmienTablice(tab);
    System.out.println(tab[0]); // 1
}
public static void zmienTablice(int []
    tab)
{
    tab = new int [1];
    tab[0] = 2;
}
```

- 1 Napisz program zawierający metodę zwracającą tablicę o wymiarach podanych przez użytkownika (niech metoda przyjmuje tę wartość jako parametr) wypełnioną losowymi liczbami z przedziału od 0 do 15.
- 2 Przerób program z zadania 1 tak, aby referencja do tablicy była przekazywana jako parametr, a metoda cechowała się zwracanym typem void (przygotowana tablica ma być dostępna poza metodą).

Dziękuję za uwagę