Podstawy programowania

Laboratorium 8 - wprowadzenie do metod

mgr inż. Krzysztof Szwarc

krzysztof@szwarc.net.pl

Sosnowiec, 10 maja 2021

Metody

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 1

```
public static void main(String[] args) {
    ourMethodName(); // wywolanie metody
}

// typ void - nic nie zostanie zwrocone
public static void ourMethodName() {
    System.out.println("Wywolano metode");
}
```

Przykład 1 - interpretacja

```
public static void main(String[] args) {
    System.out.println("Wywolano metode");
}
```

Przykład 2

```
public static void main(String[] args) {
    int result = addNumbers(2, 7);
    System.out.println(result); // 9
}

public static int addNumbers(int nm, int nm2) {
    return nm + nm2;
}
```

Przykład 2 - interpretacja

```
public static void main(String[] args) {
    int nm = 2;
    int nm2 = 7;
    int result = nm + nm2;
    System.out.println(result); // 9
}
```

Przykład 3

```
public static void main(String[] args) {
    int result = addNumbers(2, 7);
    System.out.println(result); // 9
}

public static int addNumbers(int... numbers) {
    return numbers[0] + numbers[1];
}
```

Przykład 4

```
public static void main(String[] args) {
    int result = addNumbers("tekst", 2, 7);
    System.out.println(result); // 9
}

public static int addNumbers(String text,
    int... numbers) {
    System.out.println(text); // tekst
    return numbers[0] + numbers[1];
}
```

Po co nam to wszystko?

```
public static void main(String[] args) {
    Scanner reader = new Scanner(System.in);
    int intOne = readInt(reader);
    int intTwo = readInt(reader);
    System.out.println(intOne);
    System.out.println(intTwo);
}

public static int readInt(Scanner reader) {
    System.out.println("Podaj inta");
    while (!reader.hasNextInt()) {
        System.out.println("Zla wartosc");
        reader.nextLine();
    }
    return Integer.parseInt(reader.nextLine());
}
```

Parametry, a argumenty

```
public static void main(String[] args) {
    int result = addNumbers(2); // 2 to argument
}

// number to parametr
public static int addNumbers(int number) {
    return number;
}
```

Zadania

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 złotówek 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 number = 5;
    changeNumber(number);
    System.out.println(number); // 5
}

public static void changeNumber(int number) {
    number = 10;
}
```

Przykład 2

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

/* wartoscia tablicy jest jej adres, wiec
   przekazujemy adres i na nim dzialamy (zmiany
   sa widoczne poza metoda) */
public static void changeArray(int[] array) {
    array[0] = 2;
}
```

Przykład 3

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

public static void changeArray(int[] array) {
    array = new int[1];
    array[0] = 2;
}
```

Zadania

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