Увод в програмирането

с Java

Методи

Съдържание

- Какво е това метод и защо е важно?
- Деклариране
- Параметри
- Имплементация
- Извикване
- Връщане на стойност

Методи

Методи

Методите са, най-просто казано, ***именувана*** последователност от действия. Те ни позволяват да разделим програмата на подпрограми, всяка от които решава отделен проблем/задача.

Пример

Метод, който принтира на екрана Hello World!:

```
public static void sayHello() {
        System.out.println(“Hello World”);
}
```

Пример

Метод, който принтира сумата на две цели числа:

```
public static void sum(int a, int b){
	int result = a + b;
	System.out.print(result);
}
```

Пример

Метод, който сортира масив по метода на мехурчето и го извежда:

```
public void bubbleSort(int[] array){
        boolean swapped = true;
                while (swapped){
                        swapped = false;
                        for (int i = 1; i < size; i++){
                                int temp;
                                if (array[i - 1] > array[i]){
                                        temp = array[i - 1];
                                        array[i - 1] = array[i];
                                        array[i] = temp;
                                        swapped = true;
                                }
                        }
                }
        // принтиране на сортирания масив
}
```

Защо са важни методите?

- за да не повтаряме код
- за да бъде кодът ни по-ясен и по-добре структуриран
- за да бъде лесно достъпен, без да се налага да пишем много

Какво правим с методите?

- деклариране - създаване на метода
- имплементация - какво прави методът ни?
- извикване - използване на метода за решаване на даден проблем

Деклариране

Деклариране

За да използваме един метод, трябва първо да го създадем. Методите се създават **само в рамките на някакъв клас** и извън други методи.

```
class Example {

	public static void sum(int a, int b){
		System.out.println(a + b);
	}

	public static void main(String[] args){
		...
	}
}
```

Деклариране

Декларацията на метод се състои в следните три задължителни неща:

<тип> <име> (<параметър1>,<параметър2>…<параметърN>),

където:

- <тип> е типът на променливата, която ни връща като резултат методът, независимо дали примитивен (int, long, double, etc.) или референтен (масиви, стрингове и др.)
- <име> е името на метода
- <параметър1>...<параметърN> са променливите, върху които трябва да извършим определени действия, за да решим задачата

Засега в декларацията на метода ще слагаме и public static, като по-натам ще видим какво правят те.

Пример

```
class Example {

    public static int sum(int a, int b){
        ...
    }

    public static double product(double a, double b){
        ...
    }

}
```

Пример - нещо познато

```
class Example {

        public static void sayHello(){
                System.out.println(“Hello world!”);
        }

        public static void main(String[] args){
                ...
        }

}
```

Когато методът не връща стойност, пишем void. public и static ги игнорираме засега.

Имена на методи - конвенции, конвенции...

Имената на методите трябва да започват с малка буква и да бъдат с т.нар. CamelCase - всяка следваща дума в името на метода започва с главна буква, а също така и добре да описват какво прави методът ни.

```
class Example {

	public static void thisIsTheMostFuckingAwesomeMethod(){
		System.out.println(“CamelCase, baby”);
	}
}
```

Параметри

Параметри

Параметрите са променливите, върху които ще извършваме действия, за да постигнем даден резултат. Типовете и имената на променливите се записват между кръгли скоби след името на метода, както видяхме в примерите. Някои методи нямат параметри - тогава пишем просто отваряща и затваряща скоба.

```
class Example {
	public static void giveMePI(){
		System.out.println(Math.PI);
	}

	public static void sum(int a, int b){
		System.out.println(a + b);
	}

}
```

Параметри

Параметрите могат да бъдат променливи от всякакъв тип - както примитивни типове (като int, double, float, long, etc.), така и от референтни типове като масиви, стрингове и др.

```
class Example {

	public static int[] sort(int[] array){
		// some sorting algorithm;
	}

	public static void main(String[] args){
	…
	}
}
```

Имплементация

Имплементация

След като сме декларирали нашия метод, е време да опишем какво искаме да прави. Това се случва в т.нар. “тяло” на метода или пространството между къдравите скоби:

```
class Example {
	public static void max(int a, int b){
		int max = 0;
		if (a > b)
			max = a;
		else
			max = b;
		System.out.println(max);
	}
}
```

Извикване

Извикване

Да извикаме един метод, означава да “задействаме” кода в него с определени стойности, за да решим дадена задача. Извикването се извършва, като напишем името на метода, последвано от скоби и списък с точни стойности на параметрите (ако има такива), за които искаме да решим задачата:

```
sayHello();          // принтира на екрана Hello World
sum(4, 5);           // принтира на екрана 9
product(4, 5);       //принтира на екрана 20
```

Извикване - параметри vs. аргументи

Когато създаваме един метод, посочваме списък с **параметри** - данните, необходими за решаването на задачата. Когато извикваме един метод, посочваме списък с **аргументи** - конкретните стойности, за които искаме да решим задачата.

```
class Example {

    public static void sum(int a, int b){                    // a, b - параметри
        System.out.println(a + b);
    }

    public static void main(String[] args){
        sum(4, 5);                                           // 4, 5 - аргументи
    }
}
```

Извикване

Извикването се извършва само в тялото на някакъв клас. Може да се извикват методи и в рамките на други методи.

```
class Example {

	public static void sayHello(){
		System.out.println(“Hello!”);
	}

	public static void introduce(){
		sayHello();
		System.out.println(“My name is Java”);
	}
}
```

Връщане на стойност

Досега разглеждахме само методи, които извършват някакво действие (напр. извеждат нещо на екрана). В повечето случаи обаче, методите връщат някаква стойност. Това става чрез израза **return**:

```
class Example {

	public static int square(int x){
		return x * x;
	}

}
```

Връщане на стойност - пример

Върнатата стойност трябва да е от същия тип като типа, посочен в декларацията:

```
class Example {

	public static int sum(int a, int b){
		int result = a + b;
		return result;
	}

	public static double product(double a, double b){
		int result = a * b;
		return result;
	}

}
```

Присвояване на стойност

Когато методът ни връща стойност, тази стойност трябва да бъде присвоена, иначе компилаторът ни хвърля грешка. По тази си особеност, извикването на методи много наподобява създаването на изрази:

```
class Example {
	public static int sum(int a, int b){
		return a + b;
	}

	public static void main(String[] args){
		int result1 = sum(4, 5);
		int result2 = 4 + 5;
	}
}
```

Пример

Да напишем два метода - единият конвертира температура в градуси по Целзий към градуси по Фаренхайт, а другият - обратното. Методът да връща получените стойности.

$$F = C \times 9/5 + 32$$
$$C = (F - 32) \times 5/9$$

Пример

Да се напише метод, който разменя стойностите на две променливи.

Пример

Имаме масив от цени на книги. Да се напише метод, който смята общата стойност на всички книги.

Задачи

Задача

Да се напише метод greet с един параметър name, който пази името на даден човек. Методът да отпечатва на екрана “Hello “ и името на човека.

Задача

Да се напише метод concatenateName, който приема два параметъра - firstName и lastName, съответно за първото име и фамилията на даден човек. Методът да връща цялото име.

Задача

Да се напише метод odd, приемащ един параметър цяло число number. Методът да проверява дали числото е нечетно - ако е нечетно, връща true, ако е четно - false.

Задача

Да се напише метод convertToBgn, който приема два параметъра - число, отговарящо на дадена сума в определена валута, и низ, указващ валутата. Методът да връща сумата в левове. Възможните валути са “usd”, “euro” и “gbp”.

Пример:

convertToBgn(1000, “usd”)

1 USD = 1.7408 BGN

1 EUR = 1.9557 BGN

1 GBP = 2.6415 BGN

Задача

Да се напише метод factoriel с един параметър n - цяло число. Методът да връща стойността на факториела на числото.

n! = 1.2.3...n

Задача

Да се напише метод maxElement с един параметър масив от цели числа.
Методът да връща най-големият елемент в масива.

Задача

Да се напише метод sort с един параметър масив от цели числа. Методът да връща сортирания масив.