



VILNIAUS TECHNOLOGIJŲ IR VERSLO
PROFESINIO MOKYMO CENTRAS

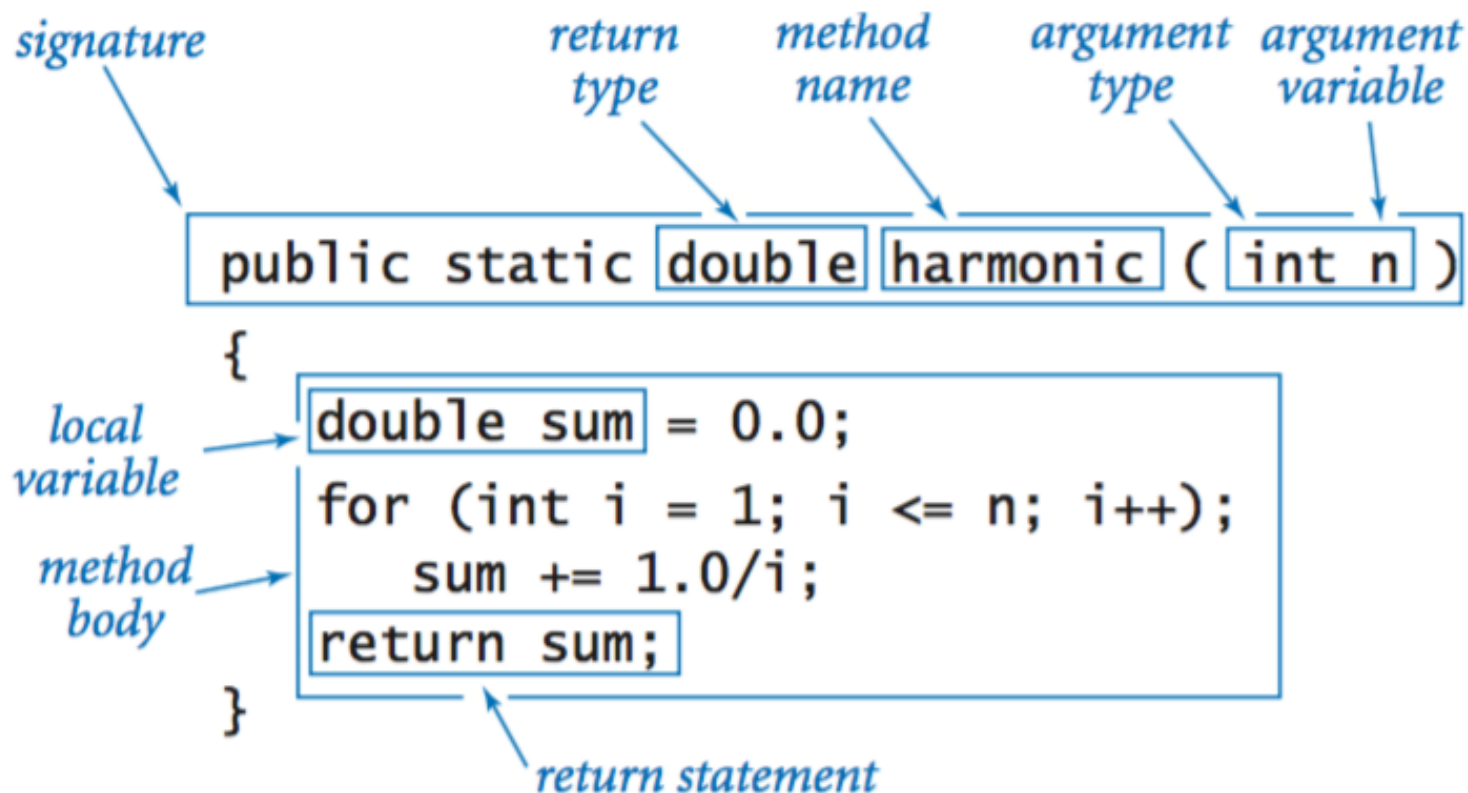
4 - METODAI

Jaroslav Grablevski / Justina Balsė

Turinys

- Metodai grąžinantys reikšmę
- Metodai negrąžinantys reikšmės
- Uždavinių sprendimo pavyzdžiai

Metodai



Method modifiers (modifikatoriai)*

Access control modifiers (matomumo modifikatoriai):

<code>public</code>	matomumo sritis neribojama (jei matoma jo klasė)
<i>(pagal nutylėjimą)</i>	matomi savo paketo klasėse
<code>protected</code>	matomi savo paketo ir išvestinėse klasėse
<code>private</code>	matomi savo klasėje

<code>static</code>	galima taikyti klasei (be objekto)
<code>abstract</code>	metodas neturi kūno
<code>final</code>	metodas negali būti užklotas
<code>native</code>	parašytas ne javos kalba
<code>synchronized</code>	tuo pačiu metu metodą gali naudoti tik vienas procesas (thread)
<code>strictfp</code>	apribojimai skaičiavimams su slankiuoju kableliu

*plačiau bus kai nagrinėsime objektinį programavimą

Rezultato tipas (return type)

- Metodo antraštėje turime nurodyti vieną rezultato tipą (pvz. `double`, `boolean`, `String`, `int[]`, `Person`)
- Metodo apraše (kūne) grąžinamą rezultatą nurodome raktiniu žodžiu `return`
- Jei metodas nieko negražina, turime nurodyti atributą `void`

Parametrų sąrašas (argument list)

- Parametrų sąrašas sudarytas iš kintamųjų tipų ir vardų poru, atskirtų kableliais. Pvz: `(String name, int age)`
- Sąrašas gali būti tuščias: `()`
- Metodo parametrai galioja tik tam metodui

Metodai

```
public static void main(String[] args) {  
    greet();  
    greet();  
    greet();  
}  
  
public static void greet() {  
    System.out.println("Greetings from the world of methods!");  
}
```

```
Greetings from the world of methods!  
Greetings from the world of methods!  
Greetings from the world of methods!
```

Metodai su parametrais

```
public static void main(String[] args) {  
    greet("Jonas");  
    greet("Justina");  
}  
  
public static void greet(String name) {  
    System.out.println("Hi " + name + ", greetings from the world of methods!");  
}
```

```
Hi Jonas, greetings from the world of methods!  
Hi Justina, greetings from the world of methods!
```


Metodai – **negražinantys** reikšmės

```
public class Metodai {  
  
    public static void main(String[] args) {  
  
        // Raskite stac plota  
        int a = 5, b = 6;  
        raskPlota(a, b);  
  
    }  
    public static void raskPlota(int a, int b){  
  
        int S = a * b;  
        System.out.println("S = " + S);  
  
    }  
}
```

Metodai – **negražinantys** reikšmės

```
public static void simpleMethod() {  
    if (Math.random() > 0.5) {  
        System.out.println("Didesnis skaičius");  
        return; // explicit return  
    } else {  
        System.out.println("Mažesnis skaičius");  
    }  
    System.out.println("Pabaiga");  
    //rezultato tipas yra "void", todėl galime  
    //nerašyti raktinio žodžio "return"  
}
```

Metodai – grąžinantys reikšmę

```
public class Metodai {  
  
    public static void main(String[] args) {  
  
        // Raskite stac plotą  
        int a = 5, b = 6;  
        int S;  
  
        S = raskPlota(a, b);  
  
        System.out.println("S = " + S);  
        //System.out.println("S = " + raskPlota(a,b));  
    }  
  
    public static int raskPlota(int a, int b){  
  
        int plotas;  
        plotas = a * b;  
        return plotas;  
        //return a*b;  
    }  
}
```

Metodai – grąžinantys reikšmę

```
public static String returningMethod() {  
    if(Math.random()>0.5) {  
        return "Išėjom ankščiau";  
    }  
    //return 100; - netinka, nes rezultato tipas kitoks  
    return "Metodo pabaiga";  
}
```

```
//return a*b;
```

Pavyzdys (1)

- Duotas sveikųjų skaičių intervalas $[m, n]$.
- Parašykite programą, kuri suskaičiuotų kiek tame intervale yra skaičių, kurie dalinasi iš juos sudarančių skaitmenų sumos.
- **Metodas** – grąžinantis vieno skaičiaus skaitmenų sumą.

Sprendimas (1) Pagrindinis metodas **main**

```
public static void main(String[] args) {  
  
    Scanner rd = new Scanner(System.in);  
  
    int kiek = 0;  
  
    // duomenų įvedimas  
    System.out.print("Iveskite m: ");  
    int m = rd.nextInt();  
    System.out.print("Iveskite n: ");  
    int n = rd.nextInt();  
  
    for(int i=m; i<=n; i++){  
        if(i % skaitmenuSuma(i) == 0)  
            kiek++;  
    }  
  
    System.out.println("Skaiciu kiekis: " + kiek);  
}
```

Sprendimas (2) Metodas **skaitmenuSuma**

```
public static int skaitmenuSuma(int sk){  
    int suma = 0;  
    int psk;  
    while(sk > 0){  
        psk = sk % 10;  
        suma = suma + psk;  
        sk = sk / 10;  
    }  
    return suma;  
}
```

Variable length argument list (varArgs)

- Trys taškai už parametro tipo reiškia, kad argumentų skaičius gali būti bet koks.
- Metode šie argumentai naudojami kaip masyvo elementai.

```
public static int addUpNumbers(int... values) {  
    int sum = 0;  
    for (int v: values) {  
        sum +=v;  
    }  
    return sum;  
}  
  
public static void main(String[] args) {  
    System.out.println(addUpNumbers(1,2,3));  
    System.out.println(addUpNumbers(1,2,3,4,5,6,7));  
    System.out.println(addUpNumbers());  
    System.out.println(addUpNumbers(new int[]{1,2,3})); //masyvas  
}
```


Pavyzdys

```
public static boolean isEven(int sk){  
    boolean is = false;  
  
    if (sk % 2 == 0)  
        is = true;  
    else  
        is = false;  
  
    return is;  
}
```

```
public static boolean isEven(int sk) {  
    return sk % 2 == 0;  
}
```