

Java Basics

Mindaugas Karpinskas
2017



Java pagrindai (ang. Java Basics)

- Klasė
- Paketai
- Kintamieji
- Metodai

Certified Associate Java® SE 8 Programmer I Study Guide Exam 1Z0-808

- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method; run a Java program from the command line; including console output
- Import other Java packages to make them accessible in your code

Java programos struktūros apžvalga

- Visos Java programos sudarytos iš keturių pagrindinių elementų:
 - Klasių
 - Metodų
 - Kintamųjų
 - Paketų.

Klasė. Pavyzdys

```
class Animal {  
  
}
```

Class <vardas>{}

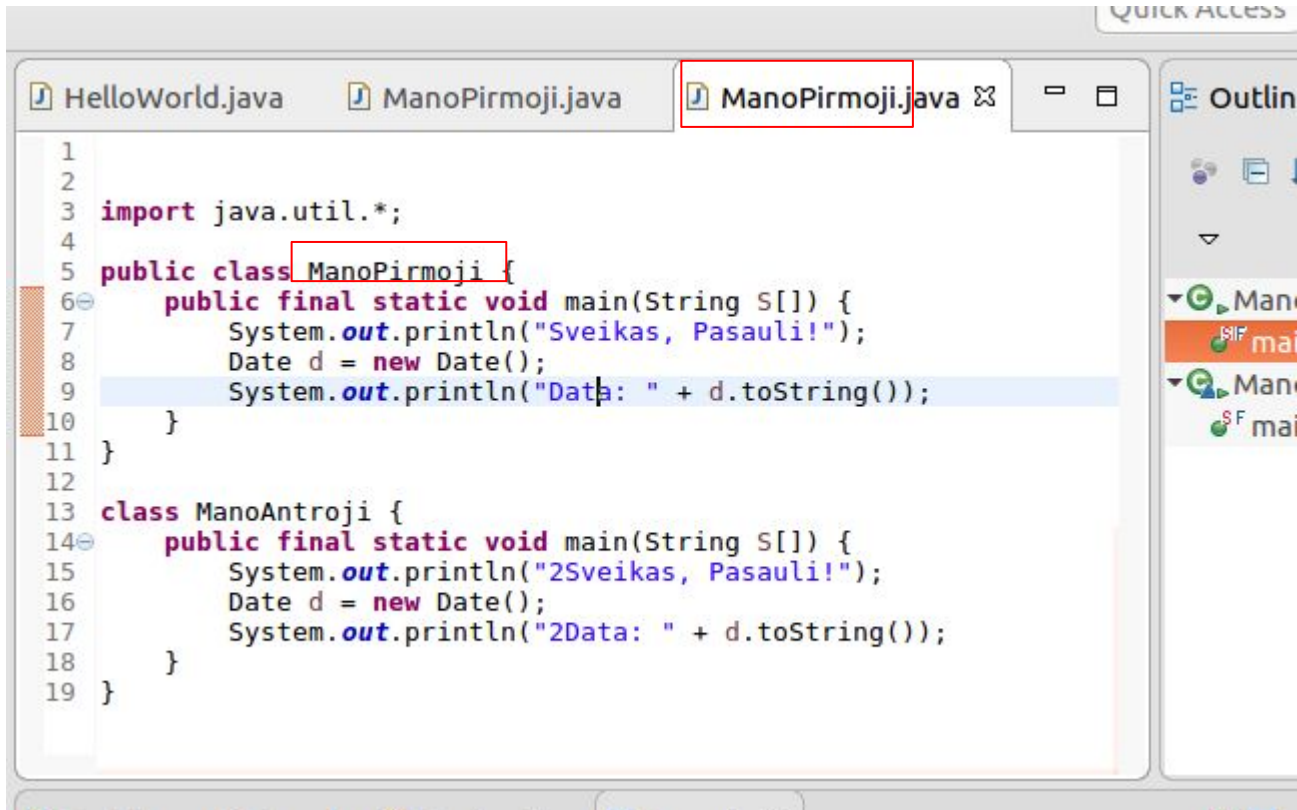
Klasė. Taisyklė NR 1

1. Java classes have **two** primary elements: **methods**, often called functions or procedures in other languages, and **fields**, more generally known as variables
 - a. Kintamieji
 - b. Metodai

```
class Animal {  
    String name;  
    String getName() {  
        return name;  
    }  
    void setName(String newName) {  
        name = newName;  
    }  
}
```

Klasė vs Java byla (*.java)

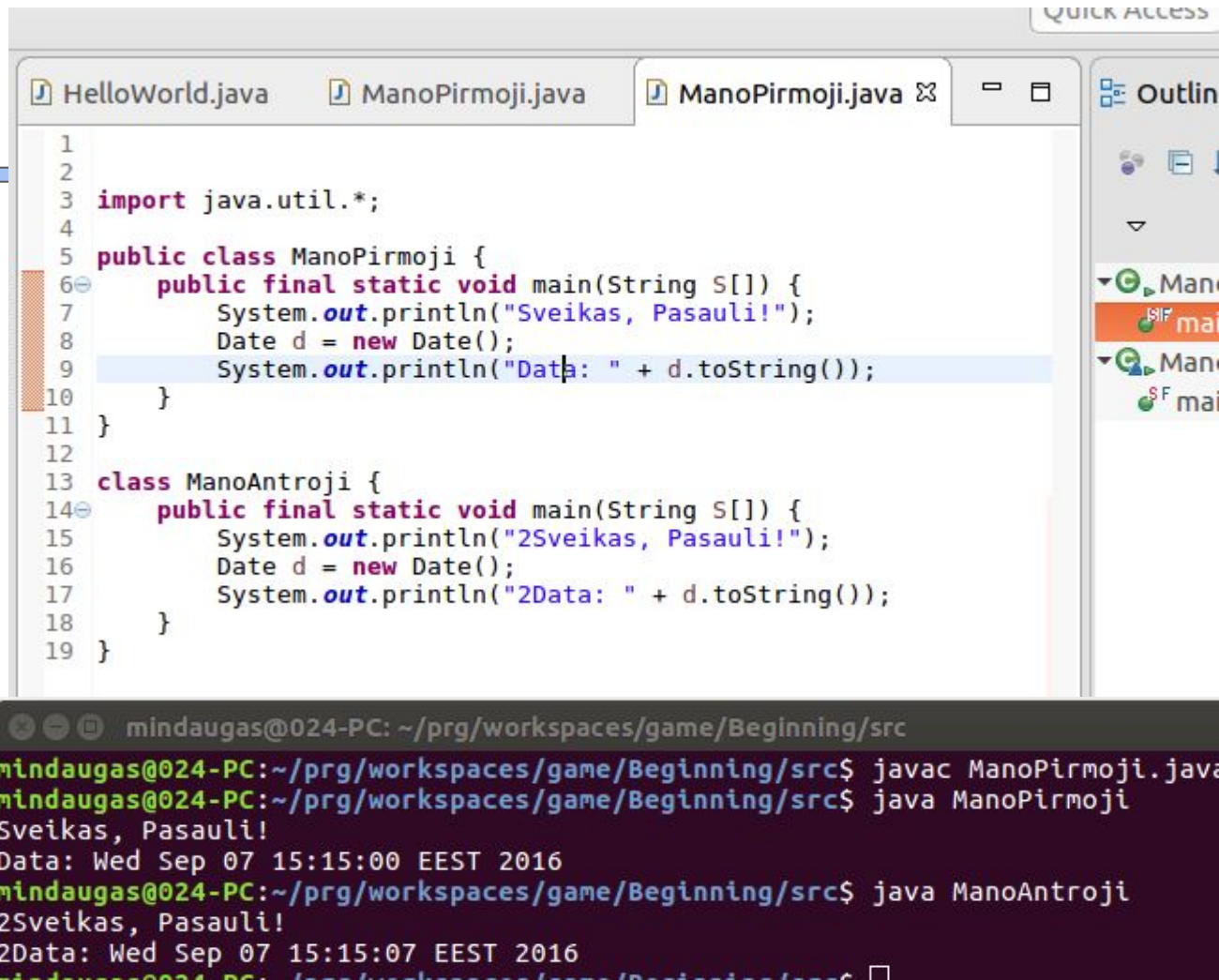
1. Java byla gali turėti keletą kalsių
2. **public** klasė *.java byloje turi būti tik viena
3. **public** klasės pavadinimas turi sutapti su bylos pavadinimu



```
1
2
3 import java.util.*;
4
5 public class ManoPirmoji {
6     public final static void main(String S[]) {
7         System.out.println("Sveikas, Pasauli!");
8         Date d = new Date();
9         System.out.println("Data: " + d.toString());
10    }
11 }
12
13 class ManoAntroji {
14     public final static void main(String S[]) {
15         System.out.println("2Sveikas, Pasauli!");
16         Date d = new Date();
17         System.out.println("2Data: " + d.toString());
18    }
19 }
```

Klasė vs Java byla (*.java)

1. Java byla gali turėti keletą kalsių
2. **public** klasė *.java byloje turi būti tik viena
3. **public** klasės pavadinimas turi sutapti su bylos pavadinimu



The screenshot shows an IDE with two tabs: 'HelloWorld.java' and 'ManoPirmoji.java'. The 'ManoPirmoji.java' tab is active, displaying the following code:

```
1
2
3 import java.util.*;
4
5 public class ManoPirmoji {
6     public final static void main(String S[]) {
7         System.out.println("Sveikas, Pasauli!");
8         Date d = new Date();
9         System.out.println("Data: " + d.toString());
10    }
11 }
12
13 class ManoAntroji {
14     public final static void main(String S[]) {
15         System.out.println("2Sveikas, Pasauli!");
16         Date d = new Date();
17         System.out.println("2Data: " + d.toString());
18    }
19 }
```

Below the code editor, a terminal window shows the execution of the code:

```
mindaugas@024-PC: ~/prg/workspaces/game/Beginning/src
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ javac ManoPirmoji.java
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ java ManoPirmoji
Sveikas, Pasauli!
Data: Wed Sep 07 15:15:00 EEST 2016
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ java ManoAntroji
2Sveikas, Pasauli!
2Data: Wed Sep 07 15:15:07 EEST 2016
```


Klasė vs Java byla (*.java)

prg school T1

Pirmoji.class Pirmoji.java Pirmoji1.class Pirmoji2.class

```
Pirmoji.java (~/prg/school/T1) - gedit
class Pirmoji {
    public final static void main(String S[]) {
        System.out.println("Sveikas, Pasauli!");
    }
}
class Pirmoji1 {
    public final static void main(String S[]) {
        System.out.println("Sveikas, Pasauli!");
    }
}
class Pirmoji2 {
    public final static void main(String S[]) {
        System.out.println("Sveikas, Pasauli!");
    }
}
```

```
mindaugas@024-PC: ~/prg/school/T1
mindaugas@024-PC:~/prg/school/T1$ javac Pirmoji.java
mindaugas@024-PC:~/prg/school/T1$
```

Do it

```
import java.util.*;
```

```
public class ManoPirmoji {  
    public final static void main(String S[]) {  
        System.out.println("1 Sveikas, Pasauli!");  
        Date d = new Date();  
        System.out.println("1 Data: " + d.toString());  
    }  
}
```

```
class ManoAntroji {  
    public final static void main(String S[]) {  
        System.out.println("2 Sveikas, Pasauli!");  
        Date d = new Date();  
        System.out.println("2 Data: " + d.toString());  
    }  
}
```

public static void **main**(String[] args)

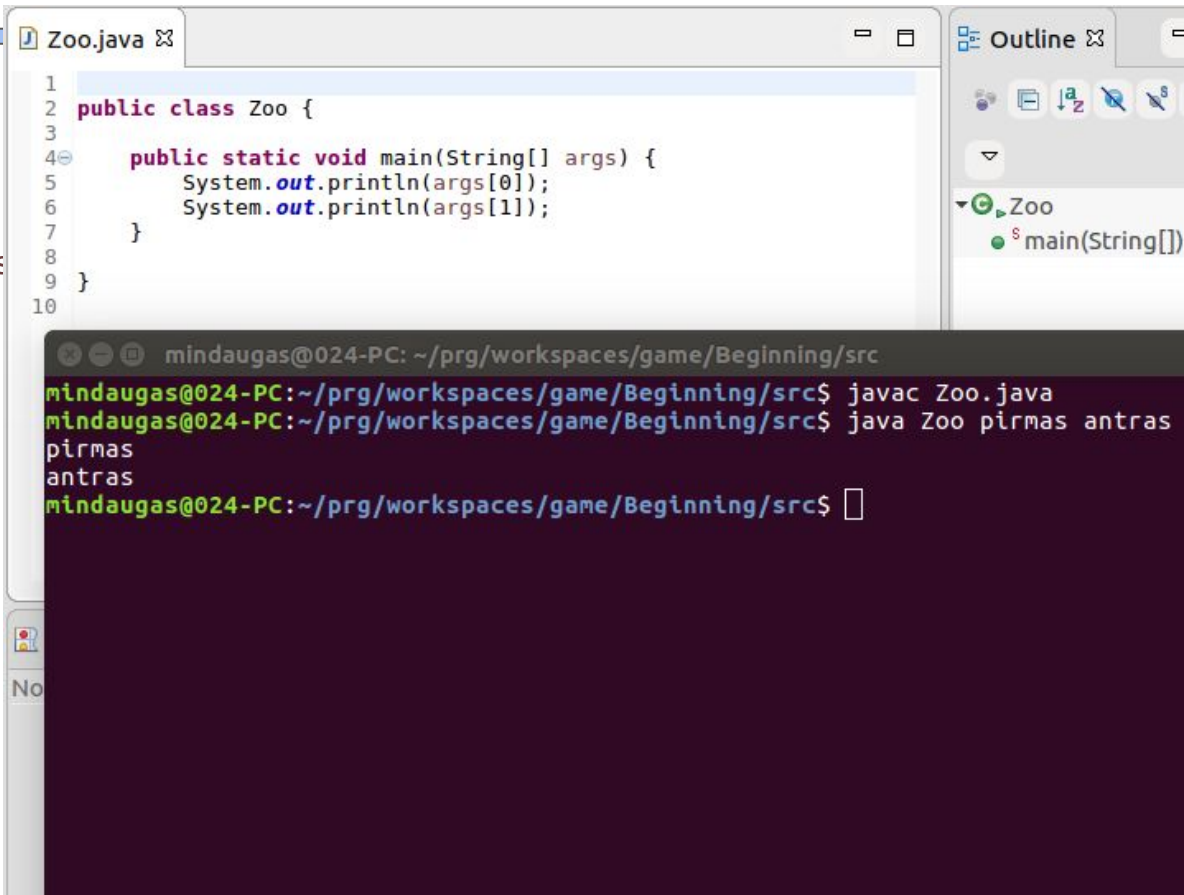
Java aplikacijos startuoja **main** metodu

```
public class Pirmoji {  
    public final static void main(String S[]) {  
        System.out.println("Sveikas, Pasauli!");  
    }  
}
```

Parametrų perdavimas

```
public class Zoo {  
  
    public static void main(String[] args)  
        System.out.println(args[0]);  
        System.out.println(args[1]);  
}  
  
}
```

>Java Zoo pirmas antras



The screenshot shows an IDE with a file named Zoo.java. The code in the file is as follows:

```
1  
2 public class Zoo {  
3  
4     public static void main(String[] args) {  
5         System.out.println(args[0]);  
6         System.out.println(args[1]);  
7     }  
8  
9 }  
10
```

On the right side of the IDE, there is an 'Outline' panel showing the class structure:

- Zoo
 - main(String[])

Below the IDE, a terminal window is open, showing the execution of the Java program:

```
mindaugas@024-PC: ~/prg/workspaces/game/Beginning/src  
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ javac Zoo.java  
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ java Zoo pirmas antras  
pirmas  
antras  
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$
```

Užduotis: Klasė

Sukurit klase su main metodu, kuri į konsolę išvestu jūsų vardą.

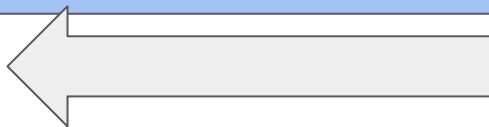
package

Paketai

- Java klases galima apjungti į vieną grupę, vadinamą paketu.
- Į vieną paketą tikslinga įtraukti giminingos paskirties programas.
 - Pavyzdžiui, mes ne kartą naudojome **System.out.println** metodą. **System** yra klasė patalpinta **java.lang** pakete. **java.lang** paketas yra išskirtinis ta prasme, kad kompiliatoriui jo nurodyti nereikia, nes laikoma, kad šis paketas yra tiek svarbus, kad jį pagal nutylėjimą naudoja visos programuotojų rašomos klasės.

Prisiminkime

```
import java.util.*;
```



```
public class ManoPirmoji {  
    public final static void main(String S[]) {  
        System.out.println("1 Sveikas, Pasauli!");  
        Date d = new Date();  
        System.out.println("1 Data: " + d.toString());  
    }  
}
```

```
class ManoAntroji {  
    public final static void main(String S[]) {  
        System.out.println("2 Sveikas, Pasauli!");  
        Date d = new Date();  
        System.out.println("2 Data: " + d.toString());  
    }  
}
```


Paketai (2)

import java.util.*;

Date - yra patalpinta java.util pakete.

Žvaigždutė pažymi, kad reikia importuoti visas java.util paketo klases. Java kompiliatorius pats sugrupuoja visas einamajame kompiliuojamos klasės kataloge esančias programas į vieną paketą ir importuoja jį. Rašant klases tai patogiu, nes kuriant bent kiek sudėtingesnę programą kyla noras ją išskaidyti į atskiras dalis, kurias natūralu laikyti viename kataloge. Nėra jokios problemos nurodyti tiksliai, kokią mums klasę reikia importuoti, jei tik žinome paketo vardą, kuriame patalpinta importuojama klasė.

```
import java.util.*;
```

```
public class ManoPirmoji {  
    public final static void main(String S[]) {  
        System.out.println("Sveikas, Pasauli!");  
        Date d = new Date();  
        System.out.println("Data: " + d.toString());  
    }  
}
```

Paketai (3)

```
package simplepackage;
import java.util.Date;
public class SimpleClass1 {
    public static void main(String[] args) {
        System.out.println("Sveikas, Pasauli!");
        Date d = new Date();
        System.out.println("Data: " + d.toString());
    }
}
```

```
> javac simplepackage/SimpleClass1.java
> java simplepackage.SimpleClass1
```

```
Sveikas, Pasauli!
Data: Mon Sep 05 19:52:13 EEST 2016
```

Užduotis

(klasė **HelloWorld** su **main** metodu)

Sukurti klasę pakete:

<šalis>.<įmonė>.<moduliopavadinimas>.<func>

...



...

Paketai (4) Do it

```
package lt.codeacademy.sdudy.main;
```

```
public class HelloWorld {
```

```
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }
```

```
}
```

Kintamieji

- Kintamieji - tai “dėžutės”, į kurias galima sudėti ir saugoti duomenis.
- Kintamojo tipas nurodo, kokios rūšies duomenis galima saugoti kintamajame.
- Pvz. int tipo kintamajame saugojami sveikieji skaičiai. Pasižiūrėkime į int tipo kintamojo panaudojimo pavyzdį

```
public class UsesInt {  
    public static void main(String S[]) {  
        int i = 4;  
        System.out.println("Kintamojo i reiksme: i=" + i);  
    }  
}
```

Apie pavyzdį

Čia buvo panaudotas priskyrimo operatorius "=", kuris kintamajam i priskyrė reikšmę 4. Šią reikšmę atspausdinome su System.out.println. int kintamųjų tipas priklauso vienai iš dviejų pagrindinių grupių, vadinamai Java pirminiais tipais (primitive types). Kita pagrindinė kintamųjų grupė vadinama nuorodomis. Šiai grupei priklauso: **programoje apibrėžti tipai, masyvai**.

Pirminių tipų kintamieji naudojami saugoti skaičius, pavienes raides, logines (ture/false) reikšmes.

Nuorodos skiriasi tuo, kad jos yra dinaminės.

Pirminiai tipai

Tipas	Dydis	Kitimo ribos	Reikšmių pavyzdžiai
int	4 baitai	nuo -2147483648 iki 2147483648	2003, -2003
short	2 baitai	nuo -32768 iki 32767	1999, -1999
byte	1 baitas	nuo -128 iki 127	100,-100
long	8 baitai	nuo -922372036854775808 iki 922372036854775807	1000000000, -1000000000
float	4 baitai	priklauso nuo tikslumo	3.142
double	8 baitai	priklauso nuo tikslumo	3.141592654
boolean	1 bitas	true, false	true, false
char	2 baitai	visi unicode simboliai	'a','b'

Primitive-Data-Types-in-Java-Language

Type	Contains	Default	Size	Range
boolean	true or false	false	1 bit	NA
char	Unicode character	\u0000	16 bits	\u0000 to \uFFFF
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
long	Signed integer	0	64 bits	-9223372036854775808 to 9223372036854775807
float	IEEE 754 floating point	0.0	32 bits	$\pm 1.4\text{E-}45$ to $\pm 3.4028235\text{E+}38$
double	IEEE 754 floating point	0.0	64 bits	$\pm 4.9\text{E-}324$ to $\pm 1.7976931348623157\text{E+}308$

Kintamųjų pavyzdys (primityvūs tipai) 1

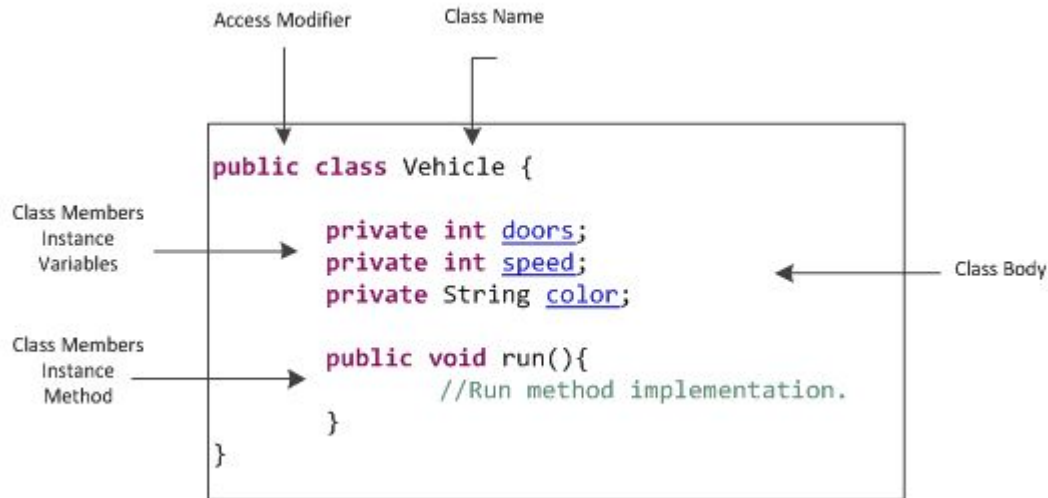
```
class Car {
```

```
    int numberOfDoors = 2;
```

```
    double engineSize = 2.4;
```

```
}
```

Kintamųjų pavyzdys (primityvūs tipai) 2



Užduotis

1. Sukurti klasę pavadinimu “Person” arba “Asmuo”. Nauja klasė - tai mūsų apibrėžtas duomenų tipas. Duomenų tipas (klasė) turi turėti keletą primitivių kintamųjų ir/ar tekstinių kintamųjų
 - a. Kintamojo varas = pradinė reikšmė
 - b. 3-5 kintamieji

Give a time



Class: Person

```
class Person {  
    String name = "Mindaugas";  
    boolean hasDrivingLicense = true;  
    String cityWhereLives = "Vilnius";  
    byte age = 31;  
    char firstSurnameLetter = 'K';  
}
```

Užduotis

Write a Java program to declare two integer variables, one float variable, and one string variable and assign 10, 12.5, and "Java programming" to them respectively. Then display their values on the screen.

JavaExercises

```
class JavaExercises {  
    public static void main(String[] args) {  
        int x;  
        float y;  
        String s;  
        x = 10;  
        y = 12.5f;  
        s = "Java programming";  
        System.out.println(x);  
        System.out.println(y);  
        System.out.println(s);  
    }  
}
```


Primitive vs reference

To Read

<https://javarevisited.blogspot.com/2015/09/difference-between-primitive-and-reference-variable-java.html>

These are the primitive types in Java:

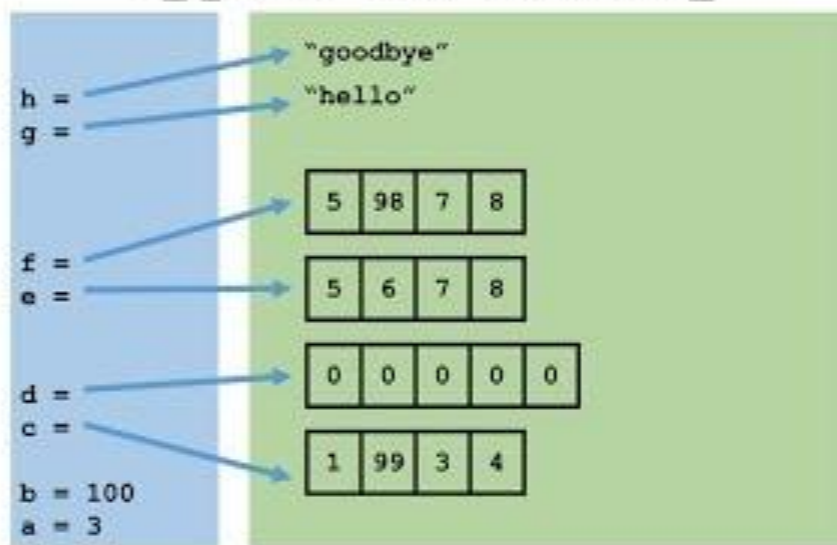
- boolean
- byte
- short
- char
- int
- long
- float
- double

All the other types are reference types: they reference objects.

This is the first part of the [Java tutorial about the basics of the language](#).

Java Tutorial

Primitive vs Reference Types in Memory



Links

<https://www.youtube.com/watch?v=2lhKbHhKfNk>

Šitas geras:

<https://www.youtube.com/watch?v=UcPuWY0wn3w>

Pirminiai ir nuorodų kintamieji

```
public class Variables {  
    public static void main(String ARGV[]) {  
  
        int myPrimitive;           //pirminio tipo kintamasis  
        int myReference[];         //nuorodos tipo kintamasis  
        myPrimitive=1;             //primityviam galime iš karto kažką priskirti  
  
        myReference = new int[3];  //o nuorodai pirma reikia išskirti atminties, kad vėliau ką nors priskirti  
  
        myReference[0]=0;  
        myReference[1]=1;  
        myReference[2]=2;  
  
    }  
}
```

Atminties išskyrimas (why new)

Pirminių kintamųjų atveju programos terpė žino, kiek tiksliai reikia išskirti kintamajam atminties, pvz. sveikajam skaičiui `myPrimitive` reikia 4 baitų. Tačiau sveikųjų skaičių masyvui iš anksto neaišku, kiek reikia išskirti atminties, ir todėl `myReference` kintamajam reikia papildomos komandos **`new`** `int[3]`, kuri nurodo, kad šiuo atveju mums reikės masyvo iš 3-ų sveikųjų skaičių. Taigi nuorodų kintamiesiems atmintis taip pat išskiriama programos metu, tik skirtumas tas, kad reikiamos atminties kiekis iš anksto nežinomas ir tam naudojamas **`new`** operatorius.

Objektų nuorodos

Nuorodų kintamieji tik nurodo atminties vietą, kurioje talpinami kintamieji (pvz. masyvo elementai), o pirminiai kintamieji naudoja fiksuoto dydžio atminties fragmentus.

***Nuorodos primena C/C++ rodykles, tačiau tarp šių sąvokų yra ir skirtumų. Nuorodos nenaudoja atminties adresų aritmetikos ir negalima sužinoti, koks tiksliai fizinis adresas išskirtas nuorodai. Tai padaryta vardan Java programų naudojimo saugumo.

Reikšmių priskyrimas boolean tipo kintamajam

```
public class TestBoolean {  
  
    public static void main(String[] args) {  
        boolean truth = true;  
        System.out.println(truth);  
        boolean fallacy = false;  
        System.out.println(fallacy);  
        truth = (1 == 1);  
        fallacy = (1 == 0);  
        System.out.println(truth);  
        System.out.println(fallacy);  
    }  
}
```

true
false
true
false

Boolean

```
package lt.codeacademy.paskaitos.antra;
public class Uzduotis {
    public static void main(String[] args) {
        boolean tiesa = true; //----- pasiratyti ir priskirti reikšmę, deklaruoti,
                                sukurti kintamąjį

        System.out.println(tiesa); //----- išspausdinti, išvesti į konsolę, parodyti
    }
}
```

Uzduotis0

- Sukurit klase Uzduotis0 ir **main** metode apsirašyti visu 8 primitius tipus:
 - boolean
 - char
 - byte
 - short
 - int
 - long
 - float
 - double
- ir išvesti jų reikšmes į konsolę panaudoti

Užduotis 0

```
package lt.codeacademy.paskaitos.antra;
public class Uzduotis0 {
    public static void main(String[] args) {
        boolean tiesa = true;
        char raide = 'A';
        byte automobilioDuruSkaicius = 4;
        short zmogausAmzius = 136;
        int sveikasSkaicius = 123456789;
        long didelsiSveikasSkacius = 1234567890123456789L;
        float suKableliu = 3.14F;
        double pi = 3.14159265358979323846264338327950288;
        System.out.println(tiesa);
    }
}
```

Arithmetic operators

Operator	Result
+	Addition
-	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
--	Decrement

Compound assignment operators

Užduotis 1

1. Sukurti naują klasę su main metodu (Uzduotis1)
 - a. Metode sukurti du **int** tipo kintamuosius *pirmas* ir *antras su pradinėmis reikšmėmis -19 ir 40*.
 - b. Metode atlikti veiksmus su kintamaisiais: sudėtis, atimtis, daugyba, dalyba, dalybos liekana %.
 - c. Atliekant veiksmus su kitamaisiais rezultata priskirti naujam kintamajam: rezultatas1, rezultatas2, rezultatas3, ... ir
 - i. Išspausdinti skaičių sumos, atimties, daugybos, dalybos ir dalybos liekanos % rezultato kintamųjų reikšmes.
 - ii. Pvz: "Sudėję du skaičius -19 ir 40 gauname: " + rezultatas1
 - d. Pabandyti atlikti veiksmus be *rezultatas* kintamojo, iškart išvedant į konsolę tekstą:
 - i. `System.out.println("-19 + 40 = " + -19 + 40);`
 - ii. Atliekant aritmetinius veiksmus galima naudoti (skliaustelius) pvz: $(a+b) * 5$
 - iii. arba "rezultatas" = "a + b = " + (a+b);

Užduotis 1

```
package lt.codeacademy.paskaitos.antra;  
  
public class Uzduotis1 {  
  
    public static void main(String[] args) {  
  
    }  
}
```

Užduotis 1

```
package lt.codeacademy.paskaitos.antra;  
  
public class Uzduotis1 {  
  
    public static void main(String[] args) {  
        int pirmas = -19;  
        int antras = 40;  
    }  
}
```


Užduotis 1

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
  
    int rezultatas1 = pirmas + antras;  
    int rezultatas2 = pirmas - antras;  
    int rezultatas3 = pirmas * antras;  
    int rezultatas4 = pirmas / antras;  
    int rezultatas5 = pirmas % antras;  
  
    System.out.println("Sudėję du skaičius -19 ir 40 gauname: " + rezultatas1);  
    System.out.println("Atėmę iš -19 skaičių 40 gauname: " + rezultatas2);  
    System.out.println("Padauginę -19 iš 40 gauname: " + rezultatas3);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + rezultatas4);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + rezultatas5);  
  
}
```

Uzduotis1

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
    int rezultatas1 = pirmas + antras;  
    int rezultatas2 = pirmas - antras;  
    int rezultatas3 = pirmas * antras;  
    int rezultatas4 = pirmas / antras;  
    int rezultatas5 = pirmas % antras;  
    System.out.println("Sudėję du skaičius -19 ir 40 gauname: " + rezultatas1);  
    System.out.println("Atėmę iš -19 skaičių 40 gauname: " + rezultatas2);  
    System.out.println("Padauginę -19 iš 40 gauname: " + rezultatas3);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + rezultatas4);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + rezultatas5);  
  
    System.out.println("Sudėję du skaičius -19 ir 40 gauname: " + pirmas + antras); ----- rezultatas ?  
    System.out.println("Atėmę iš -19 skaičių 40 gauname: " + pirmas - antras);  
    System.out.println("Padauginę -19 iš 40 gauname: " + pirmas * antras);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + pirmas / antras);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + pirmas % antras);  
}
```

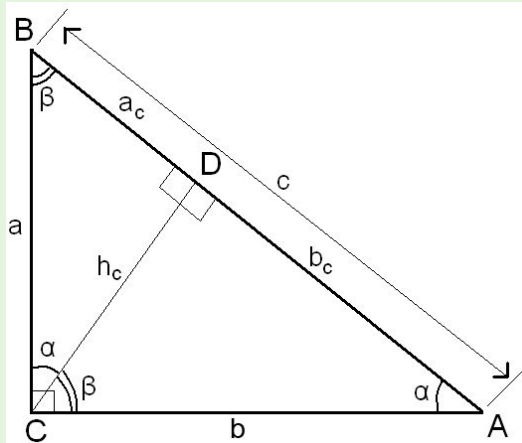
Užduotis 1

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
    int rezultatas1 = pirmas + antras;  
    int rezultatas2 = pirmas - antras;  
    int rezultatas3 = pirmas * antras;  
    int rezultatas4 = pirmas / antras;  
    int rezultatas5 = pirmas % antras;  
    System.out.println("Sudėję du skaičius -19 ir 40 gauname: " + rezultatas1);  
    System.out.println("Atėmę iš -19 skaičių 40 gauname: " + rezultatas2);  
    System.out.println("Padauginę -19 iš 40 gauname: " + rezultatas3);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + rezultatas4);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + rezultatas5);  
  
    System.out.println("Sudėję du skaičius -19 ir 40 gauname: " + (pirmas + antras));  
    System.out.println("Atėmę iš -19 skaičių 40 gauname: " + (pirmas - antras));  
    System.out.println("Padauginę -19 iš 40 gauname: " + pirmas * antras);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + pirmas / antras);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + pirmas % antras);  
}
```

Užduotis 2

- Klasė Uzduotis2 turi **main** metode apskaičiuoti stačiojo trikampio
 - plotą $S = (a * b) / 2$
 - perimetrą $P = a + b + c$, kur $c = \text{šaknis}(a*a + b*b)$;
 - pasinaudoti šaknies funkcija: **double** cq = Math.sqrt(**double** x);
- Žinomomos dvi kraštinės: $a=5$, $b=2.5$
- Naudokime double tipą

$$c^2 = a^2 + b^2.$$

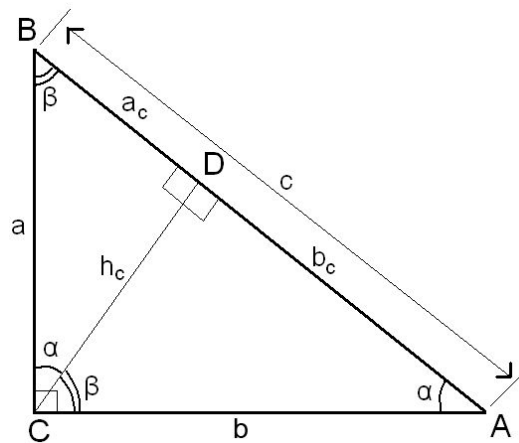


Užduotis 2

```
package lt.codeacademy.paskaitos.antra;  
public class Uzduotis2 {  
    public static void main(String[] args) {  
        double a = 5;  
        double b = 2.5;
```

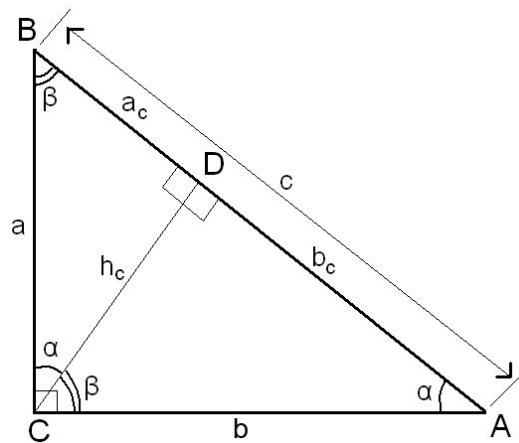
??????

```
}
```



Užduotis 2

```
package lt.codeacademy.paskaitos.antra;  
public class Uzduotis2 {  
    public static void main(String[] args) {  
        double a = 5;  
        double b = 2.5;  
  
        double plotas = (a * b) / 2;  
        double c = Math.sqrt(a * a + b * b);  
        double perimetras = a + b + c;  
  
        System.out.println("Trikampio plotas: " + plotas);  
        System.out.println("Trikampio įžambinė: " + c);  
        System.out.println("Trikampio perimetras: " + perimetras);  
    }  
}
```



Užduotis 3

- Klasė Uzduotis3 turi main metode apskaičiuoti stačiakampio
 - Plotą
 - Perimetrą
- Stačiakampio kraštinės dydžius a ir b įvedame pasinaudoję:
- `Scanner scanner = new Scanner(System.in);`
 - pvz.: `a = scanner.nextDouble();`
-

Užduotis 3

```
package lt.codeacademy.paskaitos.antra;  
import java.util.Scanner;  
public class Uzduotis3 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.println("Įveskite kraštinės a reikšmę:");  
        double a = scanner.nextDouble();  
  
    }  
}
```


Užduotis 3

```
package lt.codeacademy.paskaitos.antra;
import java.util.Scanner;
public class Uzduotis3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Įveskite kraštinės a reikšmę:");
        double a = scanner.nextDouble();

        System.out.println("Įveskite kraštinės b reikšmę:");
        double b = scanner.nextDouble();

        System.out.println("Plotas: " + (a * b));
    }
}
```

Klasės tipo kintamasis

```
package lt.codeacademy.paskaitos.pirma;
```

```
public class Kintamieji2 {  
    public static void main(String[] args) {
```

```
        Asmuo a = new Asmuo();
```

```
        System.out.println(a.vardas);
```

```
    }
```

```
}
```

```
class Asmuo {  
    String vardas = "Jonas";  
}
```

Kintamieji:

- Metodo

- Klasės

Papildomai **vėliau*

Questions and Exercises: Variables

Questions

1. The term "instance variable" is another name for ____.
2. The term "class variable" is another name for ____.
3. A local variable stores temporary state; it is declared inside a ____.
4. A variable declared within the opening and closing parenthesis of a method signature is called a ____.
5. What are the eight primitive data types supported by the Java programming language?
6. Character strings are represented by the class ____.
7. An ____ is a container object that holds a fixed number of values of a single type.

Exercises

1. Create a small program that defines some fields. Try creating some illegal field names and see what kind of error the compiler produces. Use the naming rules and conventions as a guide.
2. In the program you created in Exercise 1, try leaving the fields uninitialized and print out their values. Try the same with a local variable and see what kind of compiler errors you can produce. Becoming familiar with common compiler errors will make it easier to recognize bugs in your code.

Array (masyvo) tipai

<https://www.w3resource.com/java-exercises/array/index-test.php>

http://www.homeandlearn.co.uk/java/arrays_and_strings.html

Array (masyvo) tipai

- Array tipai apibrėžia sutvarkytą vienatipių elementų masyvą.
- Masyvo elementais gali būti visi pirminiai tipai ir visi programuotojo apibrėžti tipai.
- Galima apibrėžti masyvą masyvų, masyvą masyvų masyvą ir taip toliau.
- Masyvo elementai pasiekiami naudojant indeksą, ar indeksus
- Masyvo pirmo elemento indeksas 0
- *Jei galime sukurti kokį nors tipą, tai galime sukurti ir masyvą su to tipo elementais*

```
int myIntArray[];  
myIntArray=new int[3];
```

```
myType myObjectArray[];  
myObjectArray=new myType[3];
```

Masyvo dydžio nustatymas

```
int myIntArray[];  
myIntArray=new int[3]
```

```
int len = myIntArray.length;
```

```
System.out.println("myIntArray dydis (ilgis) yra "+len);
```

PVZ

```
int myIntArray[];  
myIntArray = new int[3];
```

```
myIntArray[0] = 0;  
myIntArray[1] = 1;  
myIntArray[2] = 2;
```

```
int len = myIntArray.length;
```

```
System.out.println("myIntArray dydis (ilgis) yra "+len);
```

System.out.println("myIntArray dydis (ilgis) yra "+len); ???

Masyvo paskutinio elemento pasiekimas

```
public class Masyvai0 {  
    public static void main(String[] args) {  
        int myIntArray[];  
        myIntArray = new int[3];  
        int lenght = myIntArray.length;  
        myIntArray[0] = 10;  
        myIntArray[1] = 20;  
        myIntArray[2] = 30;  
  
        System.out.println("myIntArray dydis (ilgis) yra " + lenght);  
        System.out.println("paskutinis narys " + myIntArray[lenght - 1]);  
    }  
}
```


Masyvai 1

- Klasė su main metodu, kuriame sukurkime String tipo masyvą iš 5 elementų su pavadinimu **draugai**
- Priskirkime keikvienam elementui po reikšmę
- Išspausdinkit:
 - Kokio ilgio masyvas;
 - Pirmą elementą;
 - Paskutinį elementą;
 - Visus elementus.

Masyvai 1

```
package lt.codeacademy.paskaitos.antra.arrays;  
public class Masyvai1 {  
    public static void main(String[] args) {  
        String[] draugai = new String[5];  
        draugai[0] = "Simonas";  
        draugai[1] = "Luknė";  
        draugai[2] = "Laura";  
        draugai[3] = "Tadas";  
        draugai[4] = "Valdas";  
    }  
}
```

Masyvai 1

```
package lt.codeacademy.paskaitos.antra.arrays;
public class Masyvai1 {
    public static void main(String[] args) {
        String[] draugai = new String[5];
        draugai[0] = "Simonas";
        draugai[1] = "Luknė";
        draugai[2] = "Laura";
        draugai[3] = "Tadas";
        draugai[4] = "Valdas";

        System.out.println("Ilgis: " + draugai.length);
        System.out.println("Pirmas elementas: " + draugai[0]);
        System.out.println("Paskutinis elementas: " + draugai[draugai.length - 1]);
        //....
    }
}
```

Declare And Initialize Java Array In One Statement

```
public class InitArray {  
  
    public static void main(String[] args) {  
        String[] masyvas = { "Laba", "diena"  
                             , "būsimieji", "Java", "programuotojai" };  
  
        System.out.println(masyvas[?]); // Java  
    }  
}
```

<https://docs.oracle.com/javase/specs/jls/se8/html/jls-10.html#jls-10.6>

Declare And Initialize Java Array In One Statement

```
public class InitArray {  
  
    public static void main(String[] args) {  
        String[] masyvas = { "Laba"0, "diena"1  
                             , "būsimieji"2, "Java"3, "programuotojai"4 };  
  
        System.out.println(masyvas[ 3  ]); // Java  
  
        System.out.println(masyvas.length); // ?  
    }  
}
```

<https://docs.oracle.com/javase/specs/jls/se8/html/jls-10.html#jls-10.6>

Masyvai 2

- Klasė su main metodu, kuriame sukurkime float tipo masyvą iš 7 elementų su pavadinimu **laikas**. (kiekvienas elementas užpildomas, kiek savaitės dienoje sugaišote laiko mokydami programuoti)
- Priskirkime masyvo elementams reikšmes inicijavimo metu
- Išspausdinkit:
 - Kokio ilgio masyvas;
 - Pirmą elementą;
 - Paskutinį elementą;
 - Visus elementus.

Masyvai 2

```
class Masyvai2 {  
  
    public static void main(String[] args) {  
        float[] laikas = { 0, 3.5f, 0, 3, 0, 4, 0 };  
  
        System.out.println("Ilgis: " + laikas.length);  
        System.out.println("Paskutinis elementas: "  
                               + laikas[laikas.length - 1]);  
        System.out.println("Pirmas elementas: " + laikas[0]);  
    }  
}
```

Initialize Java Array

Declare And Initialize

```
class InitializeJavaArray {  
    public static void main(String[] args) {  
        int[] testArray;  
        testArray = new int[] { 5, 7, 11, 13, 17 };  
        System.out.println(Arrays.toString(testArray));  
    }  
}
```

Loop

```
class InitializeJavaArrayLoop {  
    public static void main(String[] args) {  
        int[] testArray = new int[10];  
        for (int i = 0; i < testArray.length; i++) {  
            testArray[i] = i + 1;  
        }  
        System.out.println(Arrays.toString(testArray));  
    }  
}
```

Console

```
class InitializeJavaArrayConsole {  
    public static void main(String[] args) {  
        int[] testArray = new int[10];  
        for (int i = 0; i < 10; i++) {  
            String stringValue =  
System.console().readLine("Enter number: ");  
            testArray[i] = Integer.parseInt(stringValue);  
        }  
        System.out.println(Arrays.toString(testArray));  
    }  
}
```

Užduotis: Masyvai 3

Klasė su metodu main. Metode sukurti String tipo masyvą **dienos**. Masyvą užpildyti savaitės dienų pavadinimais: Pirmadienis,...,Sekmadienis.

1. Išspausdinti masyvo dydį: "Masyvas dienos dydis: " + ???
2. Išspausdinti paskutinę masyvo reikšmę: "Paskutinis masyvo elementas: " + ???
3. Paprašyti vartotojo įvesti int skaičių nuo 1 iki 7
 - a. `Scanner scanner = new Scanner(System.in);`
 - b. `int i = scanner.nextInt();`
4. Pagal įvestą reikšmę parinkti masyvo elementą ir išspausdinti jo reikšmę:
 - a. Pvz.: "Pasirinkta diena: " + <Šeštadienis>

Masyvai 2

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



```
length - 1]);
```

Masyvai 2

```
public static void main(String[] args) {  
    String[] dienos = new String[7];  
    dienos[0] = "Pirmadienis";  
    dienos[1] = "Antradienis";
```



```
    );  
    System.out.println(dienos[dienos.length - 1]);
```

```
}
```

Masyvai 2

```
public static void main(String[] args) {  
    String[] dienos = new String[7];  
    dienos[0] = "Pirmadienis";  
    dienos[1] = "Antradienis";  
    dienos[2] = "Trečiadienos";  
    dienos[3] = "Ketvirtadienis";  
    dienos[4] = "Penktadienis";  
    dienos[5] = "Šeštadienis";  
    dienos[6] = "Sekmadienis";  
  
    System.out.println("Masyvas dienos dydis: " + dienos.length);  
    System.out.println("Paskutinis masyvo elementas: " + dienos[dienos.length - 1]);  
    System.out.println("Įveskite reikšmę i:");  
    Scanner scanner = new Scanner(System.in);  
    int i = scanner.nextInt() - 1;  
  
    System.out.println("Pasirinkta diena: " + dienos[i]);  
}
```

Loop&Array

```
class Exercise {  
    public static void main(String[] args) {  
        int[] numbers = { 1, 2, 3, 4, 5 };  
        int sum = 0;  
        for (int i = 0; i < numbers.length;i++){  
            sum = sum + numbers[i];  
        }  
        System.out.println("Sum: " + sum);  
    }  
}
```


Užduotis: Masyvai 3

Klasė su metodu main. Metode sukurti String tipo masyvą **dienos**. Masyvą užpildyti savaitės dienų pavadinimais: Pirmadienis,...,Sekmadienis.

1. Išspausdinti masyvo elementu naudojant ciklą 'for'

Hint:

```
for (int i = 0; i < numbers.length; i++) {  
  
}
```

Methods

Metodai

Metodo sintaksė

Grįžties tipas

Parametrų perdavimas

Metodų perkrova

Metodo sintaksė

<panaudojimo_modifikatorius>

<final> <metodo_modifikatorius>

grįžties_tipas metodo_vardas (<parameters>) {

metodo kamienas

}

public final static int sudek(int a, int b){

return a + b;

}

Pirma rašomas metodo panaudojimo modifikatorius, final modifikatorius (gali būti ir praleistas), toliau metodo tipo modifikatorius (dinaminis modifikatorius yra nutylimas ir yra praleidžiamas, t.y. **static** reikia rašyti, o **dinamic** - niekada nerašoma). Toliau užrašomas grįžties tipas, metodo pavadinimas ir parametrų sąrašas. Metodo kamienas (body) rašomas tarp riestinių skliaustų {...}.

Grįžties tipas

Visi metodai privalo nurodyti grįžties tipą. Jei nereikia grąžinti nieko, privalote nurodyti **void** grįžties tipą.

void tipo metodai analogiški Paskalio procedūroms.

Jei metodas nėra **void** tipo, jame turi būti return operatorius, nurodantis, kokį kintamąjį grąžinti. Pateiksime ir vienos, ir kitos metodų rūšių pavyzdžių.

Metodo pvz. koks rezultatas?

```
public class MethodExamples {  
  
    static void voidMethod() {  
        System.out.println("As esu nieko negražinantis metodas :(");  
    }  
    static int returnInt() {  
        int i = 4;  
        System.out.println("returning 4");  
        return i;  
    }  
    static public final void main(String S[]) {  
        System.out.println("Sveiki, metodai!");  
        System.out.println("Kviečiu void tipo metodą");  
        voidMethod();  
        int ans = returnInt();  
  
        System.out.print("Metodas sako -");  
        System.out.println(ans);  
    }  
}
```

Metodo pvz.

```
public class MethodExamples {  
  
    static void voidMethod() {  
        System.out.println("As esu nieko negražinantis metodas :(");  
    }  
    static int returnInt() {  
        int i = 4;  
        System.out.println("returning 4");  
        return i;  
    }  
    static public final void main(String S[]) {  
        System.out.println("Sveiki, metodai!");  
        System.out.println("Kviečiu void tipo metodą");  
        voidMethod();  
        int ans = returnInt();  
  
        System.out.print("Metodas sako -");  
        System.out.println(ans);  
    }  
}
```

Sveiki, metodai!
Kviečiu void tipo metodą
As esu nieko negražinantis metodas :(
returning 4
Metodas sako -4

Metodo kvietimas

....

```
voidMethod();  
int ans = returnInt();
```

...

Java kalboje metodos gali būti sukurtas tik kaip klasės dalis

Metodo kvietimo sintaksė: `objektoVardas.metodoVardas(arg1, ...);`

Užduotis: Metodai1

- Klasė turi main metodą, kuris kviečia jūsų sukurta static metodą spausdink();
- spausdink() - metode išvedame į konsolę pranešimą “Metodas veikia”

Užduotis: Metodai1

```
package lt.codeacademy.paskaitos.antra.metodai;
```

```
public class Metodai1 {
```

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

```
          
    }
```

```
      
}
```

Užduotis: Metodai1

```
package lt.codeacademy.paskaitos.antra.metodai;  
public class Metodai1 {  
  
    public static void main(String[] args) {  
        spausdink();  
    }  
  
    static void spausdink() {  
        System.out.println("Metodas sako labas!");  
    }  
  
}
```

Metodo kvietimas

....

```
voidMethod();  
int ans = returnInt();
```

...

Java kalboje metodos gali būti sukurtas tik kaip klasės dalis

Metodo kvietimo sintaksė: `objektoVardas.metodoVardas(arg1, ...);`

Užduotis: Metodai2

- Klasė turi **main** metodą, kuris kviečia jūsų sukurtą static metodą reksme();
- reksme() - metodas grąžina int tipo reikšmę 105.
- **main** metode grąžintą reikšmę priskyriame kintamajam a
- **main** metode išspausdinam a reikšmę

Užduotis: Metodai2

```
package lt.codeacademy.paskaitos.antra.metodai;  
public class Metodai2 {
```

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

```
        int a;
```

```
        a = 10;
```

```
    }
```

Užduotis: Metodai2

```
package lt.codeacademy.paskaitos.antra.metodai;  
public class Metodai2 {  
  
    public static void main(String[] args) {  
        int a;  
        a = reiksme();  
        System.out.println(a);  
    }  
    private static int reiksme() {  
        return 105;  
    }  
}
```

Užduotis: Metodai3

- Klasė turi **main** metodą, kuris kviečia jūsų sukurtą static metodą m1();
 - m1() - metodas išveda į konsolę pranešimą “Pirmas metodas” ir po to kviečia jūsų sukurtą static metodą m2();
 - m2() - metodas išveda į konsolę pranešimą “Antras metodas” ir po to kviečia jūsų sukurtą static metodą m3();
 - m3() - metode išvedame į konsolę pranešimą “Trečias metodas”
-
- Rezultatas
 - Pirmas metodas
 - Antras metodas
 - Trečias metodas

Užduotis: Metodai3

```
package lt.codeacademy.paskaitos.antra.metodai;  
public class Metodai3 {  
    public static void main(String[] args) {
```

```
}
```

Užduotis: Metodai3

```
package lt.codeacademy.paskaitos.antra.metodai;
public class Metodai3 {
    public static void main(String[] args) {
        m1();
    }
    private static void m1() {
        System.out.println("Pirmas metodas");
        m2();
    }
    private static void m2() {
        System.out.println("Antras metodas");
        m3();
    }
    private static void m3() {
        System.out.println("Trečias metodas");
    }
}
```



Prisimenamė

- Klasė
- **main** metodas
- Paketai
- Kintamieji
- Primityvūs tipai
- Masyvai
- **new**

Užduotis

1. Programa, kuri išveda į ekraną suformatuotą informaciją

Šablonas:

“Aš, [vardas] [pavade], tikrai išmoksiu programuoti [kalba] kalba per [savaiciuSkaicius] savaites”

Programos paleidimas

>*javac Informacija.java*

>*java Informacija* **Mindaugas Karpinskas Java 9**

Programos rezultatas

>*javac Informacija*

>*java Informacija* **Mindaugas Karpinskas Java 9**

```
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ javac Informacija.java
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ java Informacija Mindaugas Karpinskas Java 9
Aš, Mindaugas Karpinskas, tikrai išmoksiu programuoti Java kalba per 9 savaites
```

Give time



Programos rezultatas

>*javac Informacija*

>*java Informacija Mindaugas Karpinskas Java 9*

```
public class Informacija {
```

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

```
String vardas = args[0];
```

```
String pavadre = args[1];
```

```
String kalba = args[2];
```

```
String savaitciuSkaicius = args[3];
```

```
System.out.println("Aš, " + vardas + " " + pavadre + ", tikrai išmoksiu programuoti " + kalba + " kalba per " +  
savaiteciuSkaicius + " savaites");
```

```
}
```

```
}
```

```
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ javac Informacija.java  
mindaugas@024-PC:~/prg/workspaces/game/Beginning/src$ java Informacija Mindaugas Karpinskas Java 9  
Aš, Mindaugas Karpinskas, tikrai išmoksiu programuoti Java kalba per 9 savaites
```

Užduotis: Metodai3 (debugging)

- Klasė turi **main** metodą, kuris kviečia jūsų sukurtą static metodą m1();
 - m1() - metodas išveda į konsolę pranešimą “Pirmas metodas” ir po to kviečia jūsų sukurtą static metodą m2();
 - m2() - metodas išveda į konsolę pranešimą “Antras metodas” ir po to kviečia jūsų sukurtą static metodą m3();
 - m3() - metode išvedame į konsolę pranešimą “Trečias metodas”
-
- Rezultatas
 - Pirmas metodas
 - Antras metodas
 - Trečias metodas

Užduotis: Metodai3

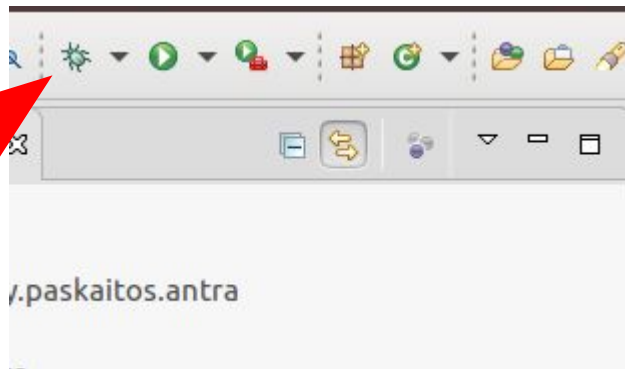
```
package lt.codeacademy.paskaitos.antra.metodai;  
public class Metodai3 {  
    public static void main(String[] args) {
```

```
}
```

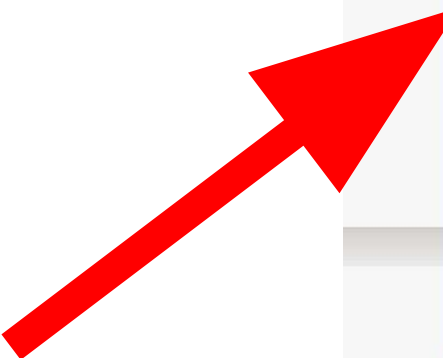
Užduotis: Metodai3

```
package lt.codeacademy.paskaitos.antra.metodai;
public class Metodai3 {
    public static void main(String[] args) {
        m1();
    }
    private static void m1() {
        System.out.println("Pirmas metodas");
        m2();
    }
    private static void m2() {
        System.out.println("Antras metodas");
        m3();
    }
    private static void m3() {
        System.out.println("Trečias metodas");
    }
}
```

(DE)BUG



Breakpoint



```
1 package lt.codeacademy.paskaitos.antra.metodai;
2
3 public class Metodai3 {
4
5     public static void main(String[] args) {
6         m1();
7     }
8
9     private static void m1() {
10         System.out.println("Pirmas metodas");
11         m2();
12     }
13
14     private static void m2() {
15         System.out.println("Antras metodas");
16         m3();
17     }
18
19
20     private static void m3() {
21         System.out.println("Trečias metodas");
22     }
23 }
24
25 }
26
```

Debugging

Masyvai0.java Testas.java Metodai3a.java Masyvai2.java Metodai3.java

```
1 package lt.codeacademy.paskaitos.antra.metodai;
2
3 public class Metodai3 {
4
5     public static void main(String[] args) {
6         m1();
7     }
8
9     private static void m1() {
10         System.out.println("Pirmas metodas");
11         m2();
12     }
13
14     private static void m2() {
15         System.out.println("Antras metodas");
16         m3();
17     }
18
19
20     private static void m3() {
21         System.out.println("Trečias metodas");
22     }
23
24
25 }
26
```


Užduotis: Metodai3a

- Klasė turi **main** metodą, kuris kviečia jūsų sukurtus du static metodus:
 - m1(); ir po to m2();
 - metodai turi gražinti **double** rezultata
 - main metode: metodų rezultatai priskieiami kintamiesiems d1, d2
 - main metodas išspausdina jų sumą
- m1() - metodas gražina double tipo reikšmę 123.5;
- m2() - metodas gražina double tipo reikšmę kurią įvedame į programą naudodami:
 - Scanner scanner = new Scanner(System.*in*);
 - System.*out*.println("Įveskite skaičių:");
 - **double** a = scanner.nextDouble();
 -

Užduotis: Metodai3a

```
import java.util.Scanner;  
public class Metodai3a {  
    public static void main(String[] args) {  
        double d1 = m1();  
        double d2 = m2();  
        System.out.println("Suma: " + (d1 + d2));  
    }  
}
```

Užduotis: Metodai3a

```
import java.util.Scanner;
public class Metodai3a {
    public static void main(String[] args) {
        double d1 = m1();
        double d2 = m2();
        ));
    }
    static double m1() {
        return 123.5;
    }
    static double m2() {
        m.in);
    }
}
```

Užduotis: Metodai3a

```
import java.util.Scanner;
public class Metodai3a {
    public static void main(String[] args) {
        double d1 = m1();
        double d2 = m2();
        ));
    }
    static double m1() {
        return 123.5;
    }
    static double m2() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Įveskite skaičių:");
        double a = scanner.nextDouble();
        return a;
    }
}
```

Užduotis: Metodai3a

```
import java.util.Scanner;
public class Metodai3a {
    public static void main(String[] args) {
        double d1 = m1();
        double d2 = m2();
        System.out.println("Suma: " + (d1 + d2));
    }
    static double m1() {
        return 123.5;
    }
    static double m2() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Įveskite skaičių:");
        double a = scanner.nextDouble();
        return a;
    }
}
```



Parametrų perdavimas

Parametrais gali būti visų tipų kintamieji, tame tarpe apibrėžti klasėse ir masyvai. Tačiau pirminių tipų ir nuorodų tipų parametrai perduodami skirtingai.

- Visi **pirminiai** kintamieji perduodami savo reikšme:
 - prieš darant perdavimą, yra pasidaroma kintamojo **kopija** ir perduodama kopijos reikšmė
 - jei metodo viduje yra keičiama parametro reikšmė, tai po grįžties parametro reikšmė nepakinta
 - metodo viduje yra manipuluojama su **kintamojo** kopija

Pvz.

```
public class ParameterExample {  
    static int addFour(int i) {  
        i = i + 4;  
        System.out.println("Vietine kopija: i=" + i);  
        return i;  
    }  
    public final static void main(String S[]) {  
        System.out.println("Sveikas, parametru siuntejau!");  
        int i = 10;  
        System.out.print("Pradinė reikšmė: i=" + i);  
  
        int j = addFour(i);  
        System.out.println("Tarpinė reikšmė: j=" + j);  
  
        System.out.println("Einamoji reikšmė: i=" + i);  
    }  
}
```

Sveikas, parametru siuntejau!
Pradinė reikšmė: i=10
Tarpinė reikšmė: j=14
Einamoji reikšmė: i=10

Užduotis: Metodai4

1. Sukurti naują klasę su main metodu (Metodai4)
 - a. Metode sukurti du **int** tipo kintamuosius *pirmas* ir *antras su pradinėmis reikšmėmis -19 ir 40*.
 - b. Sukurti naują metodą, kuris turi du parametrus: **darbas(int a, int b);**
 - c. Metode atlikti veiksmus su kintamaisiais: sudėtis, atimtis, daugyba, dalyba, dalybos liekana %.
 - d. Atliekant veiksmus su kitamaisiais rezultata priskirti naujam kintamajam: rezultatas1, rezultatas2, rezultatas3, ... ir
 - i. Išsapausdinti skaičių sumos, atimties, daugybos, dalybos ir dalybos liekanos % rezultato kintamųjų reikšmes.
 - ii. Pvz: "Sudėję du skaičius -19 ir 40 gauname: " + rezultatas1
 - e. Pabandyti atlikti veiksmus be *rezultatas* kitamojo, iškart išvedant į konsolę tekstą:
 - i. `System.out.println("-19 + 40 = " + -19 + 40);`
 - ii. Atliekant aritmetinius veiksmus galima naudoti (skliaustelius): $(a+b) * 5$
 - iii. arba "rezultatas" = "a + b = " + (a+b);

Užduotis: Metodai4

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;
```

```
}
```

Užduotis: Metodai4

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
    darbas(pirmas, antras);}  
private static void darbas(int pirmas, int antras) {  
    int rezultatas1 = pirmas + antras;
```

```
}
```

Užduotis: Metodai4

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
    darbas(pirmas, antras);} // darbas(-19, 40)  
private static void darbas(int pirmas, int antras) {  
    int rezultatas1 = pirmas + antras;  
    int rezultatas2 = pirmas - antras;  
    int rezultatas3 = pirmas * antras;  
    int rezultatas4 = pirmas / antras;  
    int rezultatas5 = pirmas % antras;  
    System.out.println("Rezultatai: 10, 19, 40, -59, -760");  
}
```

Užduotis: Metodai4

```
public static void main(String[] args) {  
    int pirmas = -19;  
    int antras = 40;  
    darbas(pirmas, antras);}   
private static void darbas(int pirmas, int antras) {  
    int rezultatas1 = pirmas + antras;  
    int rezultatas2 = pirmas - antras;  
    int rezultatas3 = pirmas * antras;  
    int rezultatas4 = pirmas / antras;  
    int rezultatas5 = pirmas % antras;  
    System.out.println("Sudėje du skaičius -19 ir 40 gauname: " + rezultatas1);  
    System.out.println("Atėme iš -19 skaičių 40 gauname: " + rezultatas2);  
    System.out.println("Padaugine -19 iš 40 gauname: " + rezultatas3);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + rezultatas4);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + rezultatas5);  
    System.out.println("Sudėje du skaičius -19 ir 40 gauname: " + (pirmas + antras));  
    System.out.println("Atėme iš -19 skaičių 40 gauname: " + (pirmas - antras));  
    System.out.println("Padaugine -19 iš 40 gauname: " + pirmas * antras);  
    System.out.println("Dalybos rezultatas skaičių -19 ir 40 yra: " + pirmas / antras);  
    System.out.println("Dalybos skaičių -19 ir 40 liekana yra: " + pirmas % antras);  
}
```

Užduotis: Metodai4

```
public static void main(String[] args) {
    int pirmas = 101;
    int antras = 10;
    darbas(pirmas, antras);
}

private static void darbas(int pirmas, int antras) {
    int rezultatas1 = pirmas + antras;
    int rezultatas2 = pirmas - antras;
    int rezultatas3 = pirmas * antras;
    int rezultatas4 = pirmas / antras;
    int rezultatas5 = pirmas % antras;

    System.out.println("Sudėje du skaičius " + pirmas + " ir " + antras + " gauname: " + rezultatas1);
    System.out.println("Atėme iš " + pirmas + " skaičių " + antras + " gauname: " + rezultatas2);
    System.out.println("Padaugine " + pirmas + " iš " + antras + " gauname: " + rezultatas3);
    System.out.println("Dalybos rezultatas skaičių " + pirmas + " ir " + antras + " yra: " + rezultatas4);
    System.out.println("Dalybos skaičių " + pirmas + " ir " + antras + " liekana yra: " + rezultatas5);
    System.out.println("Sudėje du skaičius " + pirmas + " ir " + antras + " gauname: " + (pirmas + antras));
    System.out.println("Atėme iš " + pirmas + " skaičių " + antras + " gauname: " + (pirmas - antras));
    System.out.println("Padaugine " + pirmas + " iš " + antras + " gauname: " + pirmas * antras);
    System.out.println("Dalybos rezultatas skaičių " + pirmas + " ir " + antras + " yra: " + pirmas / antras);
    System.out.println("Dalybos skaičių " + pirmas + " ir " + antras + " liekana yra: " + pirmas % antras);
}
```

Užduotis: Metodai5

- Klasė Metodai5, **main** metode kviečiamas naujas metodas **trikampis(int a, int b)** ir jame apskaičiuojame stačiojo trikampio
 - plotą $S = (a * b) / 2$
 - perimetą $P = a + b + c$, kur $c = \text{šaknis}(a*a + b*b)$;
 - pasinaudoti šaknies funkcija: **double** cq = Math.sqrt(**double** x);
- Žinomomos dvi kraštinės: $a=5$, $b=2.5$, parametrai aprašomi main metode ir perduodami į metodą **trikampis(a, b)**
- Naudokime double tipą

Užduotis: Metodai5

```
public static void main(String[] args) {  
    int b = 15;  
    int a = 10;  
    trikampis(a, b);  
}
```

```
}
```


Užduotis: Metodai5

```
public static void main(String[] args) {  
    int b = 15;  
    int a = 10;  
    trikampis(a, b);  
}  
private static void trikampis(int a, int b) {  
    double plotas = (a * b) / 2;  
    double c = Math.sqrt(a * a + b * b);  
    double perimetras = a + b + c;  
    System.out.println("Trikampio plotas: " + plotas);  
    System.out.println("Trikampio istrižainė: " + c);  
    System.out.println("Trikampio perimetras: " + perimetras);  
}
```

Metodo pvz.

```
public class MethodExamples {  
  
    static void voidMethod() {  
        System.out.println("As esu nieko negražinantis metodas :(");  
    }  
    static int returnInt() {  
        int i = 4;  
        System.out.println("returning 4");  
        return i;  
    }  
    static public final void main(String S[]) {  
        System.out.println("Sveiki, metodai!");  
        System.out.println("Kviečiu void tipo metodą");  
        voidMethod();  
        int ans = returnInt();  
  
        System.out.print("Metodas sako -");  
        System.out.println(ans);  
    }  
}
```

Sveiki, metodai!
Kviečiu void tipo metodą
As esu nieko negražinantis metodas :(
returning 4
Metodas sako -4

ParameterExample

```
public class ParameterExample {
```

```
    static int addFour(int i) {  
        i = i + 4;  
        System.out.println("Vietine kopija: i=" + i);  
        return i;  
    }
```

```
    public final static void main(String S[]) {  
        System.out.println("Sveikas, parametru siuntejau!");  
        int i = 10;  
        System.out.println("Pradinė reiksme: i=" + i);  
  
        int j = addFour(i);  
        System.out.println("Tarpine reiksme: j=" + j);  
  
        System.out.println("Einamoji reiksme: i=" + i);  
    }
```

Sveikas, parametru siuntejau!
Pradinė reiksme: i=??
Vietine kopija: i=??
Tarpine reiksme: j=??
Einamoji reiksme: i=??

ParameterExample

```
public class ParameterExample {
```

```
    static int addFour(int i) {  
        i = i + 4;  
        System.out.println("Vietine kopija: i=" + i);  
        return i;  
    }
```

```
    public final static void main(String S[]) {  
        System.out.println("Sveikas, parametru siuntejau!");  
        int i = 10;  
        System.out.println("Pradinė reiksme: i=" + i);  
  
        int j = addFour(i);  
        System.out.println("Tarpine reiksme: j=" + j);  
  
        System.out.println("Einamoji reiksme: i=" + i);  
    }
```

Sveikas, parametru siuntejau!

Pradinė reiksme: i=10

Vietine kopija: i=14

Tarpine reiksme: j=14

Einamoji reiksme: i=10

Pvz.

Taigi i reikšmė nepakito, nors addFour metode prie jos buvo pridėta 4. **Tuo tarpu nuorodų reikšmės**, jei jos tik buvo keičiamos metodo viduje, po grįžties pakinta. Pasižiūrėkime pavyzdį su sveikųjų skaičių masyvu.

```
public class ReferenceParameterExample {  
    static void changeArray(int referencellvariable[]){  
        referencellvariable[2] = 100;  
    }  
    public static void main(String ARGV[]) {  
        int anArray[] = new int[3];  
        anArray[2] = 10;  
        System.out.println("anArray[2]=" + anArray[2]);  
        changeArray(anArray);  
        System.out.println("anArray[2]=" + anArray[2]);  
    }  
}
```

Programos darbo rezultatas bus toks:

```
anArray[2]=10  
anArray[2]=100
```

Parametru perduodame nuorodą ir metodo viduje atliekami pakeitimai tiesiogiai atsiliepia nuoroda pažymėtiems kintamiesiems

Užduotis: Metodai6

- Klasė Metodai6, **main** metode kviečiamas naujas metodas **padidink(int a)**; metodas kuris perduotą parametą padidina vienetu ir gražina gautą reikšmę
 - main metode išspausdinti gautą reikšmę

Užduotis: Metodai6

```
public static void main(String[] args) {  
    int x = 10;  
    int r = padidink(x);  
    System.out.println("x = " + x);  
    System.out.println("r = " + r);  
}  
static int padidink(int a) {  
    a++;  
    return a;  
}
```

Užduotis: Metodai6

```
public static void main(String[] args) {  
    int x = 10;  
    int r = padidink(x);  
    System.out.println("x = " + x);  
    System.out.println("r = " + r);  
}  
static int padidink(int a) {  
    a++;  
    return a;  
}
```

x = 10
r = 11



Metodų perkrova

Java leidžia naudoti tą patį metodo pavadinimą, kai metodai skiriasi naudojamų parametų sąrašu

```
public static String compareNums(int i, int j) {  
    if (i == j) {  
        return "Skaičiai " + i + " ir " + j + " sutampa";  
    }  
    if (i > j) {  
        return "Skaičius " + i + " didesnis už " + j;  
    }  
    return "Skaičius " + j + " didesnis už " + i;  
}
```

```
public static String compareNums(int i, int j, int k) {  
    String S = compareNums(i, j);  
    S = S + "\n";  
    S = S + compareNums(i, k);  
    return S;  
}
```

Metodų perkrova

```
public static String compareNums(int i, int j) {
    if (i == j) {
        return "Skaičiai " + i + " ir " + j + " sutampa";
    }
    if (i > j) {
        return "Skaičius " + i + " didesnis už " + j;
    }
    return "Skaičius " + j + " didesnis už " + i;
}

public static String compareNums(int i, int j, int k) {
    String S = compareNums(i, j);
    S = S + "\n" + compareNums(i, k);
    return S;
}

public static String compareNums(double i, double j) {
    if (i == j) {
        return "Skaičiai " + i + " ir " + j + " sutampa";
    }
    if (i > j) {
        return "Skaičius " + i + " didesnis už " + j;
    }
    return "Skaičius " + j + " didesnis už " + i;
}
```

Metodų perkrova

```
public static void main(String ARGV[]) {
```

```
    int a = 3;
```

```
    int b = 4;
```

```
    int c = 5;
```

```
    double d = 3.3;
```

```
    double e = 4.4;
```

```
    String S = compareNums(a, b);
```

```
    System.out.println(S);
```

```
    S = compareNums(a, b, c);
```

```
    System.out.println(S);
```

```
    S = compareNums(d, e);
```

```
    System.out.println(S);
```

```
}
```

Skaičius 4 didesnis už 3

Skaičius 4 didesnis už 3

Skaičius 5 didesnis už 3

Skaičius 4.4 didesnis už 3.3

Užduotis: Metodai7

- Klasė Metodai6, **main** metode kviečiamas nauji metodai ir išspausdinamos jų gautos reikšmės:
 - ***static int*** sudetis(int a, int b);
 - ***static int*** sudetis(int a, int b, int c);
 - ***static int*** sudetis(int a, int b, int c, int d);
 - ***static int*** sudetis(int a, int b, int c, int d, int e).
- Sukurti atitinkamus metodus ir juose apskaičiuoti perduotų reikšmių sumą

Užduotis: Metodai6

```
public static void main(String[] args) {  
    System.out.println("10 + 11 = " + sudetis(10, 11));  
    System.out.println("10 + 11 + 12 = " + sudetis(10, 11, 12));  
    System.out.println("10 + 11 + 12 + 13 = " + sudetis(10, 11, 12, 13));  
    System.out.println("10 + 11 + 12 + 13 + 14 = " + sudetis(10, 11, 12, 13, 14));  
}
```

Užduotis: Metodai6

```
static int sudetis(int a, int b) {  
    return a + b;  
}  
static int sudetis(int a, int b, int c) {  
    return a + b + c;  
}  
static int sudetis(int a, int b, int c, int d) {  
    return a + b + c + d;  
}  
  
static int sudetis(int a, int b  
                  , int c, int d, int e) {  
    return a + b + c + e;  
}
```

Užduotis: Metodai6

```
static int sudetis(int a, int b) {  
    return a + b;  
}  
static int sudetis(int a, int b, int c) {  
    return sudetis(a, b) + c;  
}  
static int sudetis(int a, int b  
                  , int c, int d) {  
    return sudetis(a, b, c) + d;  
}  
static int sudetis(int a, int b  
                  , int c, int d, int e) {  
    return sudetis(a, b, c, d) + e;  
}
```

```
static int sudetis(int a, int b) {  
    return a + b;  
}  
static int sudetis(int a, int b, int c) {  
    return a + b + c;  
}  
static int sudetis(int a, int b, int c, int d) {  
    return a + b + c + d;  
}  
static int sudetis(int a, int b  
                  , int c, int d, int e) {  
    return a + b + c + e;  
}
```


Klausimai

?

Links:

https://en.wikibooks.org/wiki/Java_Programming/Overloading_Methods_and_Constructors

<https://beginnersbook.com/2013/05/method-overloading/>

<https://www.javatpoint.com/method-overloading-in-java>

<https://www.javatpoint.com/method-overloading-vs-method-overriding-in-java>

<https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>

Prisimename 2

- Klasė
- **main** metodas
- Paketai
- Kintamieji
- Primityvūs tipai
- Nuorodų tipai
- **new**
- String
- Metodai
- Metodų perkrova

Klausimai

-> String