

JAVA operatoriai

Mindaugas Karpinskas
2017



Komandos

- Bloko sakinyys
- Priskyrimas
- Operatoriaus sąvoka
- Matematiniai operatoriai
- Didinimas/mažinimas vienetu
- Lyginimo operatoriai
- Loginiai operatoriai
- Tipų konvertavimas ir palyginimas
- Sąlygos sakinyys, Switch sakinyys
- Metodo iškviatimas

Links

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

https://www.tutorialspoint.com/java/java_basic_operators.htm

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op1.html>

Bloko sakiny

- Keli sakiniai gali būti apjungti į bloko sakinį
- Blokas išskiriamas riestiniais skliaustais {sakiniai}
- Bloke galima deklaruoti kintamuosius, tačiau jie nebus pasiekiami už bloko ribų

```
class Rectangle {  
  
    private int len, width;  
    public void show() {  
        System.out.println("Stačiakampis(" + x + "," + y + ")");  
        System.out.println("Ilgis=" + len + ", storis=" + width);  
    }  
    public void hide() {  
    }  
}
```

Bodies Body {}

static {

~~Dynamic / object~~{

Constructor (){

method(){

body inside body

{

{

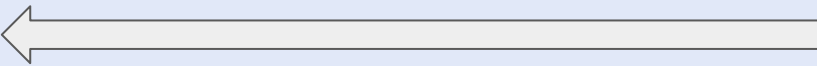
{

}

}

}

```
class Rectangle {  
    static {  
        System.out.println("Class Body");  
    }  
    {  
        System.out.println("Object body");  
    }  
    private int len, width;  
    Rectangle(int x, int y, int Len, int Width) {  
        len = Len;  
        width = Width;  
        System.out.println("Konstruktor");  
    }  
    public void show() {  
        System.out.println("Staciamkampus(" + len + ", " + width + ")");  
        System.out.println("Ilgis=" + len + ", storis=" + width);  
    }  
    public void hide() {  
        {  
        }  
    }  
}
```



Bodies Body {}

static {

~~Dynamic / object~~{

Constructor (){

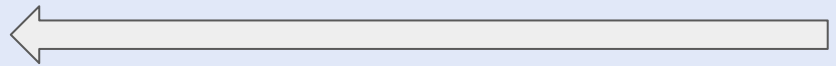
method(){

body inside body

```
{  
    {  
        {  
            int sk = 10;  
        }  
        System.out.println(sk);  
    }  
}
```



```
class Rectangle {  
    static {  
        System.out.println("Class Body");  
    }  
    {  
        System.out.println("Object body");  
    }  
    private int len, width;  
    Rectangle(int x, int y, int Len, int Width) {  
        len = Len;  
        width = Width;  
        System.out.println("Konstruktor");  
    }  
    public void show() {  
        System.out.println("Staciamkampus(" + x + "," + y + ")");  
        System.out.println("Ilgis=" + len + ", storis=" + width);  
    }  
    public void hide() {  
        {  
        }  
    }  
}
```



PVZ

```
public static
    void main(String[] s) {
    new Body();
}
```

```
public class Body {
    static {
        System.out.println("AAA");
    }
    static {
        System.out.println("BBB");
    }
    {
        System.out.println("CCC");
    }
    {
        System.out.println("DDD");
    }
    static String kint1 = "kint1";
    static String kint2 = kint2();
    String kint3 = "kint3";
    String kint4 = kint4();
    private static String kint2() {
        return "kint2";
    }
    private String kint4() {
        return "kint4";
    }
}
```

Užd. blokai

1. static blokas - išveda "Statinis"
2. ~~dinamie~~ blokas - išveda "Objektas"
3. Konstruktorius - "Konstruktorius"
4. Metodas m1(); - "metodas veikia"

Kintamieji

1. Metodo kintamieji

- a. "Vietiniai" (angl. Local variables)
- b. Laikini (angl. Temporary variables)

2. Klasės kintamieji

- a. Gali pasiekti visi klasės metodai ir kai kuriais atvejais kitų klasių metodai
- b. Išsaugo informacija
 - i. Objekto būseną

Pvz

```
public class Kintamieji {  
    static String informacija;  
    public static void main(String[] args) {  
        m1();  
        m2();  
    }  
    private static void m1() {  
        informacija = "Reikšmė";  
        String informacija2 = "Labas";  
    }  
    private static void m2() {  
        System.out.println(informacija);  
        System.out.println(informacija2);  
    }  
}
```



Operatoriai: Užduotis0

1. Apsirašyti static klasės kintamąjį String **tekstas** ir 4 metodus:
 - a. metode **ivestiReiksme()** paprašyti vartotojo įvesti sakinį ir priskirti **tekstas**
 - b. metode **didziosios()** pakeisti **tekstas** kintamojo visas raides į didžiasias
 - c. metode **spaudinit()** išspausdinti kintamojo reikšmę
 - d. metode **main(String[] args)** iškviesti:
 - i. `ivestiReiksme();`
 - ii. `didziosios();`
 - iii. `spaudinit();`

Operatoriai: Užduotis0

```
import java.util.Scanner;  
public class Uzduotis0 {  
    private static String tekstas;  
    public static void main(String[] args) {  
        ivestiPaiksmo();  
    }
```

```
}
```

Operatoriai: Užduotis0

```
import java.util.Scanner;
public class Uzduotis0 {
    private static String tekstas;
    public static void main(String[] args) {
        iverstiReiksme();
        didziosios();
        spausdinti();
    }
    private static void iverstiReiksme() {
        System.out.println("Iveskite sakinį:");
        Scanner s = new Scanner(System.in);
        tekstas = s.nextLine();
    }
    private static void didziosios() {
        tekstas = tekstas.toUpperCase();
    }
    private static void spausdinti() {
        System.out.println(tekstas);
    }
}
```

Operatoriaus sąvoka

- Operatoriai kartu su argumentais sudaro išraiškas
- Operatorius ima vieną ar kelis argumentus ir sukuria naują reikšmę
- Kai kurie operatoriai gali pakeisti ir pačius argumentus
- Beveik visi operatoriai skirti darbui tik su primityviaisiais tipais
- Tik operatoriai `=` `==` `!=` dirba ir su objektais, o `+` ir `+=` su `String` objektais

Priskyrimo operatorius

- Žymimas “=”
- Dešinėje pusėje gali būti konstanta, kintamasis, išraiška
- Kairėje pusėje – kintamasis
 - Pvz., a=4;
- Priskiriant primityviojo tipo kintamuosius kopijuojama reikšmė
- Priskiriant objektinio tipo kintamuosius kopijuojama tik nuoroda

Priskyrimo operatorius PVZ

```
int cadence = 0;  
int speed = 0;  
int gear = 1;  
int masyvas[];  
masyvas = new int[5];  
int j = 10;  
int z = j++ % 5; // z = 0  
int c = 10;  
int x = ++c % 5; // x = 1
```


Matematiniai operatoriai

- Sudėtis (+)
 - atimtis ir *unarinis* minusas (-)
 - daugyba (*)
 - dalyba (/)
 - sveikų skaičių dalybos liekana (%)
-
- Matematiniai (ir visi kiti) operatoriai gali būti jungiami su priskyrimu, pvz.,
`a=a+4`; galima pakeisti `a+=4`;

Matematiniai operatoriai

```
class ArithmeticDemo {
    public static void main(String[] args) {
        int result = 1 + 2; // result is now 3
        System.out.println(result);
        result = result - 1; // result is now 2 // same as: result-=1;
                           //same as: result--; // same as: --result;
        System.out.println(result);
        result = result * 2; // result is now 4 // same as: result*=2;
        System.out.println(result);
        result = result / 2; // result is now 2 // same as: result/=2;
        System.out.println(result);
        result = result + 8; // result is now 10 // same as: result+=8;
        result = result % 7; // result is now 3 // same as: result%=7;
        System.out.println(result);
    }
}
```

Tekstinės eilutės sudėtis

```
class ConcatDemo {  
    public static void main(String[] args) {  
        String firstString = "This is";  
        String secondString = " a concatenated string.";  
        String thirdString = firstString + secondString;  
        System.out.println(thirdString);  
    }  
}
```

Didinimas/mažinimas vienetu

- ++a
- a++
- --a
- a--
- Pvz.,
 - `int i=0, j=0;`
 - `j=++i; //=> i=i+1; j=i; => j==1`
 - `j=i++; //=> j=i; i=i+1; => j==1`

Unary operators

- +** **Unary plus operator; indicates positive value (numbers are positive without this, however)**
- **Unary minus operator; negates an expression**
- ++** **Increment operator; increments a value by 1**
- **Decrement operator; decrements a value by 1**
- !** **Logical complement operator; inverts the value of a boolean**

Unariniai operatoriai

```
class Demo {  
    public static void main(String[] args) {  
        int i = 3;  
        i++;  
        System.out.println(i); // "4"  
        ++i;  
        System.out.println(i); // "5"  
        System.out.println(++i); // "6"  
        System.out.println(i++); // "6"  
        System.out.println(i); // "7"  
    }  
}
```

Operatoriai: Užduotis1

- ++x
- x++

```
System.out.println("++x: " + (++x));
```

```
System.out.println("reikšmė po ++x: " + x);
```

```
System.out.println("x++: " + (x++));
```

```
System.out.println("reikšmė po x++: " + x);
```

Operatoriai: Užduotis2

1. `grazinklrPadidink(int i);`
2. `padidinklrGrazink(int i);`

Operatoriai: Užduotis2

```
public static void main(String[] args) {  
    int i = 10;  
    int i1 = grazinklrPadidink(i);  
    int i2 = padidinklrGrazink(i);  
    System.out.println("Pradinė reikšmė: " + i);  
    System.out.println("grazinklrPadidink: " + i1);  
    System.out.println("padidinklrGrazink: " + i2);  
}
```

Operatoriai: Užduotis2

```
public static void main(String[] args) {  
    int i = 10;  
    int i1 = grazinklrPadidink(i);  
    int i2 = padidinklrGrazink(i);  
    System.out.println("Pradinė reikšmė: " + i);  
    System.out.println("grazinklrPadidink: " + i1);  
    System.out.println("padidinklrGrazink: " + i2);  
}  
private static int grazinklrPadidink(int i) {  
    return i++;  
}  
private static int padidinklrGrazink(int i) {  
    return ++i;  
}
```

Lyginimo operatoriai

- Lyginimo operatoriaus rezultatas – loginė reikšmė true arba false
 - Lygu (==)
 - nelygu(!=)
 - mažiau (<)
 - mažiau arba lygu (<=)
 - daugiau (>)
 - daugiau arba lygu (>=)
- Loginio (boolean) tipo kintamiesiems galima taikyti tik operatorius == ir !=

Lygybės ir palyginimo operatoriai

`==` equal to

`!=` not equal to

`>` greater than

`>=` greater than or equal to

`<` less than

`<=` less than or equal to

Operatoriai: Užduotis3

- Sukurit klase Uzduotis3 ir **main** metode apsirašyti visu primitivius tipus (išskyrus boolean) po du kartus:
 - ~~boolean~~
 - char
 - byte
 - short
 - int
 - long
 - float
 - double
- Palyginti jų reikšmes, ir rezultatą priskirti boolean kintamiesiams: intTest, longTest,...
- Išspausdinti rezultatus;

PVZ

```
class Palyginimas {  
    public static void main(String[] args) {  
        int value1 = 1;  
        int value2 = 2;  
        boolean testInt = value1 == value2;  
        if (testInt)  
            System.out.println("value1 == value2");  
        if (value1 != value2)  
            System.out.println("value1 != value2");  
        if (value1 > value2)  
            System.out.println("value1 > value2");  
        if (value1 < value2)  
            System.out.println("value1 < value2");  
        if (value1 <= value2)  
            System.out.println("value1 <= value2");  
    }  
}
```

Loginiai operatoriai

- Loginio operatoriaus rezultatas – loginė reikšmė true arba false
 - Ir (&&)
 - arba (||)
 - ne (!)
- Loginius operatorius galima taikyti tik loginiams (boolean) kintamiesiems ir reiškiniams
 - `boolean t1 = false && true; // t1 - reikšmė ?`
 - `boolean t2 = false || true; // t2 - reikšmė ?`

Another conditional operator is `?:`, which can be thought of as shorthand for an if-then-else statement

Loginiai operatoriai PVZ

```
class Demo1 {  
    public static void main(String[] args) {  
        int value1 = 1;  
        int value2 = 2;  
        if ((value1 == 1) && (value2 == 2))  
            System.out.println("value1 is 1 AND value2 is 2");  
        if ((value1 == 1) || (value2 == 1))  
            System.out.println("value1 is 1 OR value2 is 1");  
    }  
}
```


Operatoriai: Loginiai1

Main metode paprašyti vartotojo įvesti tris **int** skaičius ir juos priskirti local kintamiesiems: sk1, sk2, sk3;

1. Sukurti **boolean** tipo kintamąją **arReiksmeslygios**
 - a. nustatyti ar įvestos reikšmės yra lygios ir rezultatą priskirti **arReiksmeslygios**;
 - b. kintamojo “**arReiksmeslygios**” reikšmė **true** tik tuo atveju jei visos įvestos reikšmės sk1, sk2, sk3 yra lygios
 - c. išspausdinti “**arReiksmeslygios**” reikšmę
2. Sukurti **boolean** tipo kintamąją **arIvesta10**
 - a. nustatyti ar bent viena iš įvestų reikšmių yra lygi 10
 - b. kintamasis “**arIvesta10**” turi įgauti *true* reikšmę tik tuo atveju jei bent vienas skaičius lygus 10
 - c. išspausdinti rezultatą

Loginiai1 .. main(String[] args) {

```
int sk1, sk2, sk3;  
Scanner s = new Scanner(System.in);  
System.out.println("Įveskite pirmą skaičių:");  
sk1 = s.nextInt();  
System.out.println("Įveskite antrą skaičių:");  
sk2 = s.nextInt();  
System.out.println("Įveskite trečią skaičių:");  
sk3 = s.nextInt();
```

0 0 10 = sk3.

???

System.out.println("Pirmas skaičius: " + sk1);

Loginiai1 .. main(String[] args) {

```
int sk1, sk2, sk3;  
Scanner s = new Scanner(System.in);  
System.out.println("Įveskite pirmą skaičių:");  
sk1 = s.nextInt();  
System.out.println("Įveskite antrą skaičių:");  
sk2 = s.nextInt();  
System.out.println("Įveskite trečią skaičių:");  
sk3 = s.nextInt();
```

```
boolean arReiksmeslygios = sk1 == sk2 && sk2 == sk3;  
System.out.println("Kintamojo arReiksmeslygios reikšmė: " + arReiksmeslygios);  
  
boolean arIvesta10 = sk1 == 10 || sk2 == 10 || sk3 == 10;  
System.out.println("Kintamojo arIvesta10 reikšmė: " + arIvesta10);
```

Loginiai operatoriai

- Loginio operatoriaus rezultatas – loginė reikšmė true arba false
 - Ir (&&)
 - arba (||)
 - ne (!)
- Loginis operatorius galima taikyti tik loginiams (boolean) kintamiesiems ir reiškiniams
 - `boolean t1 = false && true; // t1 - reikšmė ?`
 - `boolean t2 = false || true; // t2 - reikšmė ?`

Another conditional operator is ?:, which can be thought of as shorthand for an if-then-else statement

`result = someCondition ? value1 : value2;`

Loginiai operatoriai PVZ 2

```
class Demo2{
```

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

```
        int value1 = 1;
```

```
        int value2 = 2;
```

```
        int result;
```

```
        boolean someCondition = true;
```

```
        result = someCondition ? value1 : value2;
```

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

```
    }
```

```
}
```

Operatoriai: Užduotis3a

- Sukurti klase Uzduotis3a ir **main** metode apsirašyti du int tipo kintamuosius ir jiems priskirti reikšmes
- 1 . Palyginti jų reikšmes (pvz ar pirmasis kintamasis didesnis), ir rezultatą priskirti boolean kintamiesiams: intTest
- 2. Sukurti trečia int kintamąjį ir jam priskirti didesnę reikšmę. Palyginimui padautoi sutrumpinta sąlygos operatoriu:
 - **someCondition ? value1 : value2;**
- Išspausdinti rezultatus;

Uzduotis3a

```
package lt.codeacademy.paskaitos.antra.operatoriai;  
public class Uzduotis3a {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
        boolean intTest;  
        // intTest = 10 < 20 ? true : false;  
    }  
}
```

Uzduotis3a

```
package lt.codeacademy.paskaitos.antra.operatoriai;  
public class Uzduotis3a {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
        boolean intTest;  
        // intTest = 10 > 20; //simple  
        intTest = 10 > 20 ? true : false;  
    }  
}
```


Uzduotis3a

```
public class Uzduotis3a {  
  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
        boolean intTest;  
  
        // intTest = 10 > 20; //simple  
        intTest = 10 > 20 ? true : false;  
  
        int c = a > b ? a : b;  
        System.out.println(c);  
    }  
}
```

Sąlygos sakinyys

Sintaksė:

if(sąlyga) {sakiniai}

else if(sąlyga) {sakiniai}

...

else if(sąlyga) {sakiniai}

else {sakiniai}

Sąlygos sakinio pavyzdys

```
int a = 1, b = 2;
```

```
if (a > 0 && b > 0)
```

```
    System.out.println("Abu teig.");
```

```
else if (a > 0 && b < 0)
```

```
    System.out.println("Pirmas teig.");
```

```
else if (a < 0 && b > 0)
```

```
    System.out.println("Antras teig.");
```

```
else if (a < 0 && b < 0)
```

```
    System.out.println("Abu neig.");
```

```
else
```

```
    System.out.println("Bent vienas nulis");
```

Operatoriai: Užduotis4

1. Main metode apsirašyti keleta boolean tipo kintamųjų ir priskirti reikšmes. Kintamiesiems parašyti po IF sakinį, ir jei boolean reikšmė true, išspausdinti tekstą su kintamojo vardu;
2. Panaudoti primityvius skaičių (po du) tipus IF sakiniuose ir išspausdinti ar reikšmės yra:
 - a. Lygios
 - b. Pirma mažesnė
 - c. Pirma didesnė
3. Panaudoti primityvių skaičių tipus IF sakiniuose ir išspausdinti ar reikšmės yra:
 - a. Pirma mažesnė arba lygios
 - b. Pirma didesnė arba lygios

Operatoriai: Užduotis4

```
public static void main(String[] args) {  
    int sk1 = 15;  
    int sk2 = 10;  
    if (sk1 < sk2) {  
        System.out.println("Sk1 mažesnis");  
    }  
    ...  
    if (sk1 <= sk2) {  
        System.out.println("Sk1 mažesnis arba lygus");  
    }  
    ...  
}
```

Switch sakinyys

Sintaksė:

```
switch(išraiška) {
```

```
case reikšm ė1 : sakiniai; break;
```

```
case reikšm ė2 : sakiniai; break;
```

```
case reikšm ė3 : sakiniai; break;
```

```
// ...
```

```
default: sakiniai;
```

```
}
```

Switch sakinyje įvertinama išraiškos reikšmė ir pagal ją nušokama prie **case** sakinio

Switch sakiny (tęsinys)

- Išraiškos reikšmė gali būti sveiko ar simbolinio tipo, bet negali būti realaus
- Sakiny **default** vykdomas, kai nerandama tinkama reikšmė
- Sakiny **default** nėra būtinas
- Sakiny **break** naudojamas nutraukti vykdymą
- Jei sakinyje **break** nėra, vykdomi visi vėlesni atvejai įskaitant ir **default**

Switch sakinio pavyzdys

```
class Demo4 {  
    public static void main(String[] args) {  
        int kursas = -1; char k = '?'; //Ka atspausdins, jei k  
reikšmė '1', '2', '4'?  
        switch (k) {  
            case '1':  
                kursas = 1; break;  
            case '2':  
                kursas = 2;  
            case '3':  
                kursas = 3; break;  
            case '4':  
                kursas = 4;  
        }  
        System.out.println(kursas);  
    }  
}
```


Operatoriai: Užduotis5

- Klasės **main** metode kviečiamas jūsų sukurtas naujas void metodas **rodykDiena(int savaitiesDiena)**. Naujame metode panaudoti **switch** sakinį su `int savaitiesDiena` kintamuoju
 - Jei `saavaitiesDiena == 1` išspausdinti Piramadienis
 - ..
 - Jei `saavaitiesDiena == 7` išspausdinti Sekmadienis

Operatoriai: Užduotis4

```
public class Uzduotis4 {  
    public static void main(String[] args) {  
        int diena = 1;
```



```
    }  
}  
}
```

Operatoriai: Užduotis4

```
public class Uzduotis4 {  
    public static void main(String[] args) {  
        int diena = 1;  
        rodykDiena(diena);  
    }  
    static void rodykDiena(int savaitiesDiena) {  
        rodykDiena(savaitiesDiena);  
    }  
}
```

Operatoriai: Užduotis4

```
static void rodykDiena(int savaitiesDiena) {  
    switch (savaitiesDiena) {  
        case 1:  
            System.out.println("Pirmadienis");  
            break;  
        case 2:  
            System.out.println("Antradienis");  
            break;  
        default:  
            System.out.println("NE Pirmadienis ir NE Antradienis");  
            break;  
    }  
}
```

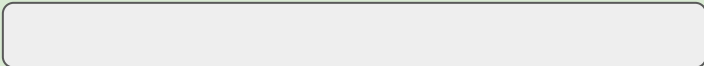
Operatoriai: switch

1. Vartotojo paprašyti įvesti skaičių nuo 1 iki 10.
2. Išvesti įvestą skaičių pvz: vienas, du, trys....
3. Būtinai panaudokime switch sakinį!

Tipų konvertavimas

- Java kalboje naudojamas tipų konvertavimas (angl. casting).
- Kur logiška, kintamojo tipas bus konvertuotas automatiškai.
 - Pvz., slankaus tipo kintamajam priskyrus sveikąją reikšmę, ji bus automatiškai paversta į slankaus kablelio formatą.
- Konvertavimą galima/reikia nurodyti ir programoje, tam prieš išraišką skliausteliuose tereikia nurodyti tipą, pvz., (int)a.
- boolean tipo duomenims negalima taikyti tipų konvertavimo

Tipų konvertavimas PVZ

```
class Demo3{  
  
    public static void main(String[] args) {  
        int value1 = 2;  
        short value2;  
          
        System.out.println(value2);  
    }  
}
```

Tipų konvertavimas PVZ

```
class Demo3{  
  
    public static void main(String[] args) {  
        int value1 = 2;  
        short value2;  
        value2 = (short) value1;  
        System.out.println(value2);  
    }  
}
```


Klausimai pasikartojimui.

Consider the following code snippet.

```
if (aNumber >= 0)
    if (aNumber == 0)
        System.out.println("first string");
else System.out.println("second string");
System.out.println("third string");
```

1. What output do you think the code will produce if aNumber is 3?
2. Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?
3. Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.
4. Use braces, { and }, to further clarify the code.

test1("two"); // result will be ?

```
public static void test1(String question) {  
    switch (question) {  
        case "one":  
            System.out.println("Entered one!");  
            break;  
        case "two":  
            System.out.println("Entered two!");  
            break;  
        default:  
            System.out.println("Error, need one or two ...");  
    }  
}
```

test2("two"); // result will be ?

```
public static void test3(String question) {  
    if (question == "one")  
        System.out.println("Entered one!");  
    else if (question == "two")  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```

test2("one"); // result will be ?

```
public static void test3(String question) {  
    if (question == "one")  
        System.out.println("Entered one!");  
    if (question == "two")  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```

test3("two"); // result will be ?

```
public static void test2(String question) {  
    if (question.equals("one"))  
        System.out.println("Entered one!");  
    else if (question.equals("two"))  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```

test3("one"); // result will be ?

```
public static void test2(String question) {  
    if (question.equals("one"))  
        System.out.println("Entered one!");  
    if (question.equals("two"))  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```

test3("one"); // result will be ?

```
public static void test2(String question) {  
    if (question.equals("one"))  
        System.out.println("Entered one!");  
  
    if (question.equals("two"))  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```

test3int(2); // rezultatas???

```
public static void test3int(int question) {  
    if (question == 1)  
        System.out.println("Entered one!");  
    else if (question == 2)  
        System.out.println("Entered two!");  
    else  
        System.out.println("Error, need one or two ...");  
}
```


Objektų lyginimas `equals()`

Objektinio tipo kintamiesiems galima taikyti tik operatorius `==` ir `!=`

Lyginant objektinio tipo kintamuosius lyginamos tik nuorodos, t.y. tikrinama, ar nuorodos rodo į tą patį objektą

Norint palyginti pačius objektus, tam objektai turi metodą `equals()`

*Tipų palyginimo operatorius instanceof

```
class InstanceofDemo {  
    public static void main(String[] args) {  
        Parent obj1 = new Parent();  
        Child obj2 = new Child();  
        System.out.println("obj1 instanceof Parent: " + (obj1 instanceof Parent));  
        System.out.println("obj1 instanceof Child: " + (obj1 instanceof Child));  
        System.out.println("obj1 instanceof MyInterface: " + (obj1 instanceof MyInterface));  
        System.out.println("obj2 instanceof Parent: " + (obj2 instanceof Parent));  
        System.out.println("obj2 instanceof Child: " + (obj2 instanceof Child));  
        System.out.println("obj2 instanceof MyInterface: " + (obj2 instanceof MyInterface));  
    }  
}  
  
class Parent {}  
class Child extends Parent implements MyInterface {}  
interface MyInterface {}
```



Prisimename

- Bloko sakiny
- Priskyrimas
- Operatoriaus sąvoka
- Matematiniai operatoriai
- Didinimas/mažinimas vienetu
- Lyginimo operatoriai <>= == !–
- Loginiai operatoriai && ||
- Tipų konvertavimas ir palyginimas
- Sąlygos sakiny, Switch sakiny
- Metodo iškvi

Užduotis6

1. Write a program that accepts user input from the console. The program should take a number and then test for the following age ranges: 0 to 10, 11 to 20, 21 to 30, 30 and over. Display a message in the Output window in the following format: `user_age + " is between 21 and 30"`
2. Write a program that asks a user to choose between four colours: black, white, red, or blue. Use IF ... ELSE IF statements to display one of the following messages, depending on which colour was chosen:
 - a. BLACK "You must be a Goth!"
 - b. WHITE "You are a very pure person"
 - c. RED "You are fun and outgoing"
 - d. BLUE "You're not a Chelsea fan, are you?"
3. Re-write task (2.) to use switch statement.

Klausimai



****Bitų operatoriai*

- Bitų operatoriai taikomi sveiko tipo ar loginiams (boolean) kintamiesiems, rezultatas – taip pat sveiko ar loginio tipo
 - Pabitinis ir (&)
 - pabitinis arba (|)
 - pabitinis išskirtinis arba (angl. XOR) (^)
 - pabitinis ne (~)

****Poslinkio operatoriai*

- Poslinkio operatoriai taikomi tik sveiko tipo kintamiesiems
- Char, byte ir short tipo kintamieji prieš operaciją paverčiami int
 - Poslinkis kairėn (<<)
 - poslinkis dešinėn (>>)
 - poslinkis dešinėn be ženklų (>>>)