

Objektinis programavimas JAVA

Mindaugas Karpinskas
2018



Objektinis Programavimas OP vs PP

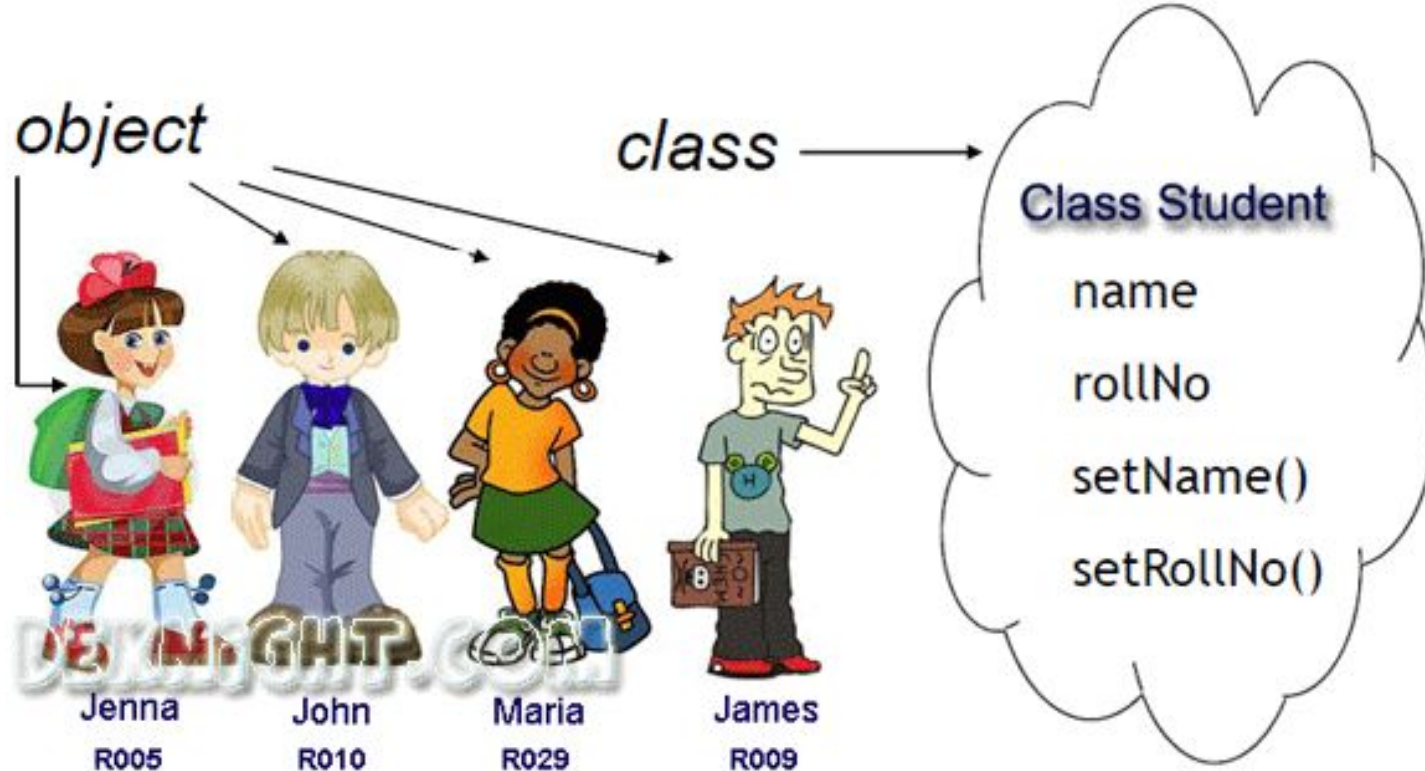
- Procedūrinio **P**rogramavimo paradigma PP – skaidyti programą į procedūras
 - Trūkumas ... d u o m e n y s ?!
- Objektiškai orientuoto programavimo idėja – apjungti procedūras su duomenimis.

Pagrindinės objektinio programavimo (OOP) sąvokos

- **Objektas** – vienareikšmiškai identifikuojamas, savistovis programinis darinys, turintis būseną ir elgesį.
- **Pranešimas** - komunikacijos (bendravimo) tarp objektų būdas (kitais vadinama metodais).
- **Klasė** - prototipas, apibrėžiantis tam tikros rūšies objektų ypatybes. Kiekvienas konkretus objektas yra kurios nors klasės egzempliorius (instance).

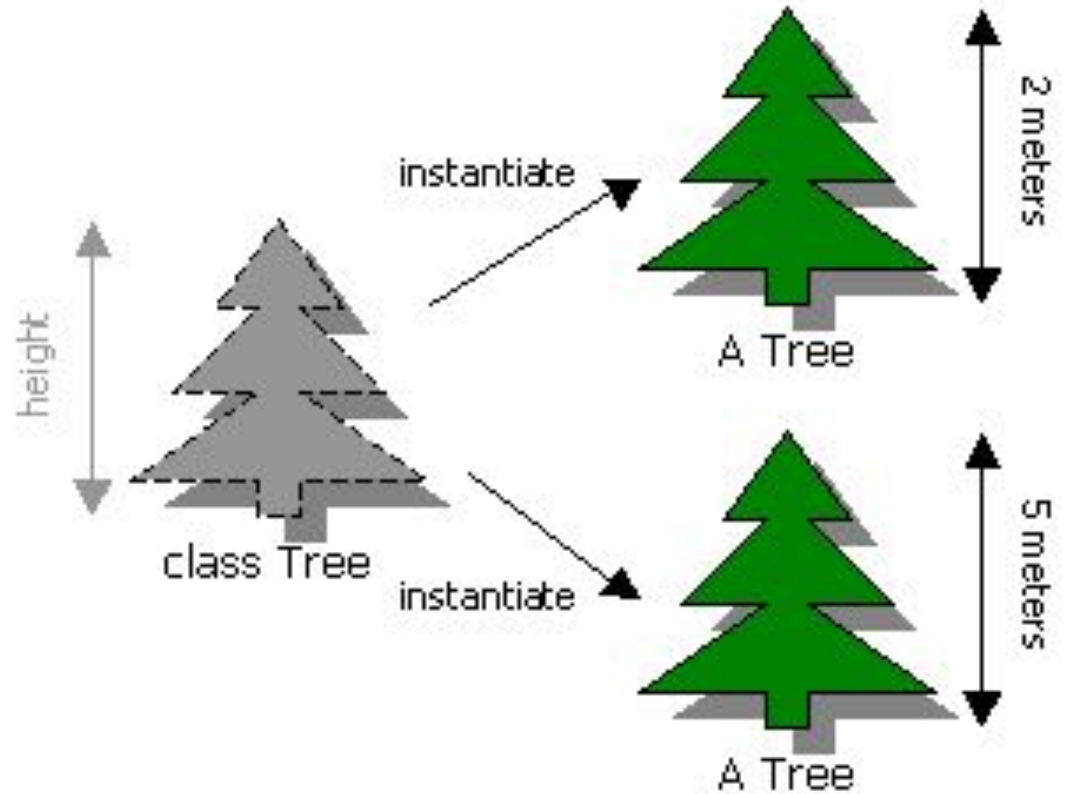
Class vs Object

(šaltinis: <http://www.c4learn.com/>)



Class vs Object

Šaltinis: <http://www.ustudy.in/>



OOP sąvokos

- **Inkapsuliacija** (encapsulation) - objekto elgesio ir būsenos kaip visumos traktavimas. Susijęs su objekto realizacijos paslėpimu, leidžiant manipuluoti objektu tik naudojant viešąją sąsają (interface).
- **Klasių hierarchija, paveldėjimas** - klasių sistemos organizavimo būdas, specializuojant elgesį išvestinėse klasėse. Įgalina programinių sistemų pakartotinį panaudojamumą.
- **Polimorfizmas** – objektų sugebėjimas reaguoti į metodo iškvietimą priklausomai nuo objekto klasės.

OOP sąvokos

- OOP:
 - tai programavimo **metodologija**,
 - kai programa sukomponuojama iš objektų,
 - kiekvienas objektas priklauso tam tikrai klasei,
 - klasės sudaro paveldėjimo hierarchiją (Pagal Booch).
- OOP susijęs taip pat su OO analize ir projektavimu.
- OO programavimas realizuojamas naudojant OO principus įgyvendinčiaja programavimo kalba. Pvz. JAVA

Java programavimo kalba

- objektiškai orientuota, bet paprastesnė nei C++
- interpretuojama,
- architektūriškai neutrali,
- dinaminė,
- saugi,
- daugiagijė,
- turinti turtingą klasių biblioteką,
- turinti išvystytus IDE (Eclipse, NetBeans, IDEA).

Java Language Keywords

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

Complete List of Java Keywords

| | | | | | |
|----------|-----------|---------|--------------|----------|------------|
| abstract | boolean | break | byte | case | catch |
| char | class | const | continue | default | do |
| double | else | extends | final | finally | float |
| for | goto | if | implements | import | instanceof |
| int | interface | long | native | new | package |
| private | protected | public | return | short | static |
| strictfp | super | switch | synchronized | this | throw |
| throws | transient | try | void | volatile | while |
| assert | | | | | |

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 $\pm 1.7976931348623157\text{E+}308$ |

Java kalbos 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 | 922372036854775808 - 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 | |

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

Logical operators

Relational Operators

| Operator | Use | Description |
|----------|----------------------------|--|
| > | <code>op1 > op2</code> | Returns <code>true</code> if <code>op1</code> is greater than <code>op2</code> |
| >= | <code>op1 >= op2</code> | Returns <code>true</code> if <code>op1</code> is greater than or equal to <code>op2</code> |
| < | <code>op1 < op2</code> | Returns <code>true</code> if <code>op1</code> is less than <code>op2</code> |
| <= | <code>op1 <= op2</code> | Returns <code>true</code> if <code>op1</code> is less than or equal to <code>op2</code> |
| == | <code>op1 == op2</code> | Returns <code>true</code> if <code>op1</code> and <code>op2</code> are equal |
| != | <code>op1 != op2</code> | Returns <code>true</code> if <code>op1</code> and <code>op2</code> are not equal |

Operators 3

| Operator | Purpose | Example | Equivalent |
|-----------------|----------------|---------------------|------------------------|
| | | | |
| <code>+=</code> | Addition | <code>x += 2</code> | <code>x = x + 2</code> |
| <code>-=</code> | Subtraction | <code>x -= 2</code> | <code>x = x - 2</code> |
| <code>/=</code> | Division | <code>x /= 2</code> | <code>x = x / 2</code> |
| <code>*=</code> | Multiplication | <code>x *= 2</code> | <code>x = x * 2</code> |
| <code>%=</code> | Modulus | <code>x %= 2</code> | <code>x = x % 2</code> |

Paprasčiausia programa

```
/**
 * Hello console
 *
 * @author anonymous
 */
public class HelloConsole {
    /**
     * Main function.
     *
     * @param args - command line arguments
     * @return Nothing
     */
    public static void main(String[] args) {
        // System - class name (java.lang.System)
        // out statical field being java.io.PrintStream
        System.out.println("Hello, world");
    }
}
```

Programos vykdymo etapai

- Kompiliavimas komandinės eilutės aplinkoje (sukuriamas *HelloWorld.class* failas) :
 - *javac HelloWorld.java*
- Vykdymas komandinės eilutės aplinkoje:
 - *java HelloWorld*

Mano pirmoji

```
package lt.mycompany.examples.main;
```

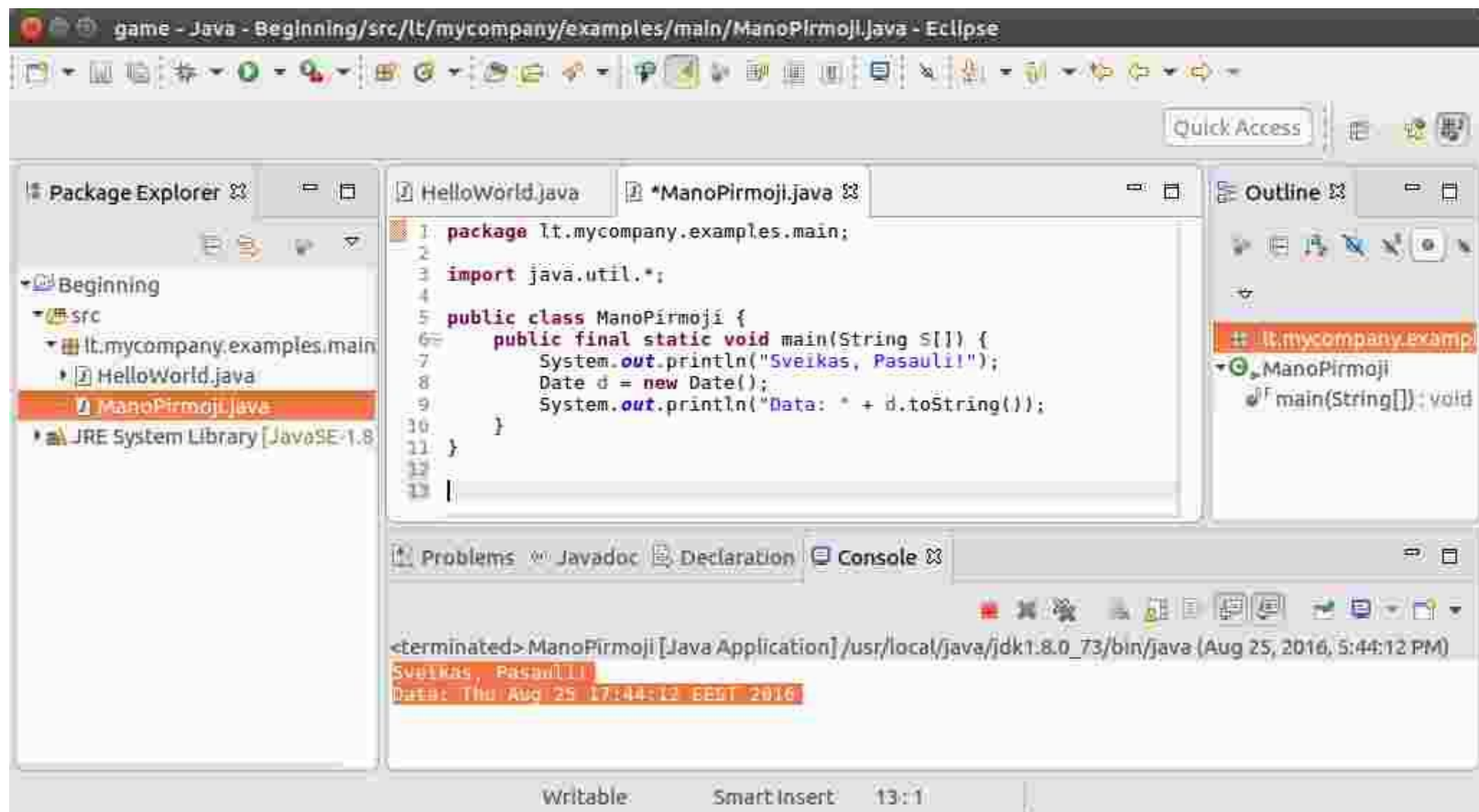
```
import java.util.*;
```

```
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());  
    }  
}
```

```
> Sveikas, Pasauli!
```

```
> Data: Thu Aug 25 17:44:12 EEST 2016
```

Eclipse



Komandinē eilutē



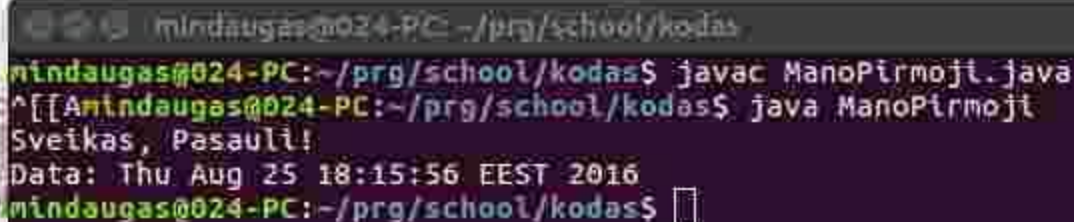
```
1 import java.util.*;
2
3 public class ManoPirmoji {
4     public final static void main(String S[]) {
5         System.out.println("Sveikas, Pasauli!");
6         Date d = new Date();
7         System.out.println("Data: " + d.toString());
8     }
9 }
10
```

Java

Ln 10, c

Problems Javadoc

<terminated> ManoPirmoji
Sveikas, Pasauli!
Data: Thu Aug 25 17:44



```
mindaugas@024-PC: ~/prg/school/kodas
mindaugas@024-PC:~/prg/school/kodas$ javac ManoPirmoji.java
mindaugas@024-PC:~/prg/school/kodas$ java ManoPirmoji
Sveikas, Pasauli!
Data: Thu Aug 25 18:15:56 EEST 2016
mindaugas@024-PC:~/prg/school/kodas$
```

Pabaiga

Klausimai ? ? ?

Kita:

-> aplinkos paruošimas

Klausimai

