

# Uitwerkingen oefenopgaven hoofdstuk 1

## Code Oefenopgave 1

```
package javastructuur;  
  
public class JavaStructuur {  
  
    public static void main(String[] args) {  
        String s = "loi wenst u veel succes met deze cursus";  
        System.out.println(s);  
        System.out.println(args[0]);  
        System.out.println(args[1]);  
    }  
}
```

## Code Oefenopgave 2

```
package myPackage;

/**
 * @author xxxxx    dit is Javadoc commentaar
 * @version          dit is Javadoc commentaar
 * Oefening 2
 * deze code is geschreven door: xxxxx
 */
public class Oef2 {

    public static void main(String[] args) { //commentaar
        System.out.print("ik geef Java ");
        for(int i=0;i<5;i++) System.out.print("*");
        System.out.println(" sterren");
    }
}
```

Het commentaar is herkenbaar aan: /\* /\*\* \*/ en //

Javadoc commentaar begint met /\*\*.

**Code Oefenopgave 3**

```
com.loi.student.Persoon pers = new Persoon(12,"Klaas", "Lutjebroek");
```

### Code Oefenopgave 4b

```
package com.loi.student;

public class Student extends Persoon{
    private String opleiding;

    public Student(int id, String naam, String woonplaats, String
opleiding) {
        super(id, naam, woonplaats);
        this.opleiding = opleiding;
    }

    @Override
    public String toString() {
        return "id = " + id + ", naam = " + naam + ", woonplaats
= " + woonplaats + " opleiding = " + opleiding;
    }
}
```

Veranderingen in de class Persoon:

```
public class Persoon {
    protected int id;
    protected String naam;
    protected String woonplaats;
```

In de class JavaStructuur is het dikgedrukt toegevoegd. Omdat er in de package javastructuur al een class student stond, moet u bij het maken van een nieuwe student uit de package com.loi.student de volledige package naam meegeven! Probeer ook eens System.out.println("id = " + student1.id enz. te schrijven. Kan dat? Waarom wel of niet?

```
public class JavaStructuur {

    public static void main(String[] args) {

        Student student = new Student(5,"Jan","Leiden");
        System.out.println("id = " + student.id + ", naam = " +
student.naam + ", woonplaats = " + student.woonplaats);

        com.loi.student.Student student1 = new
com.loi.student.Student(123,"Jan","Scheveningen", "Java Programmeren");
        System.out.println(student1.toString());
    }
}
```

## Code Oefenopgave 5

```
package dieren;

public class TestDier {

    public static void main(String[] args) {
        //Dier dier = new Dier("Tortelduif", "mais"); //kan niet want dier is
abstract!
        Dier dier = new Vogel(true, "duif", "Tortelduif", "mais"); //abstracte
class als type gebruikt.
        System.out.println(dier.kenmerken());
        System.out.println("het aantal = " + dier.geefAantal());
        Vogel diera = new Vogel(false, "vink", "StreepVink",
"insecten"); //abstracte class als type gebruikt.
        System.out.println(diera.kenmerken());
        System.out.println("het aantal = " + diera.geefAantal());

    }

}
```

```
package dieren;

public abstract class Dier {
    String Naam;
    String Voedsel;
    static int aantal = 0; //static variabele

    public Dier() { //default constructor
        aantal++;
    }

    public Dier(String Naam, String Voedsel) { //constructor
        this();
        this.Naam = Naam;
        this.Voedsel = Voedsel;
    }

    public String getNaam() { //concrete methodes
        return Naam;
    }

    public void setNaam(String Naam) {
        this.Naam = Naam;
    }

    public String getVoedsel() {
        return Voedsel;
    }

    public void setVoedsel(String Voedsel) {
        this.Voedsel = Voedsel;
    }
}
```

```

        final int geefAantal() { //final methode
            return aantal;
        }
        public abstract String kenmerken(); //abstracte methode zonder body
    }

package dieren;

public class Vogel extends Dier { //concrete subclass
    private boolean tam;
    private String soort;

    public Vogel(boolean tam, String soort, String Naam, String Voedsel)
    {
        super(Naam, Voedsel);
        this.tam = tam;
        this.soort = soort;
    }

    public boolean isTam() {
        return tam;
    }

    public void setTam(boolean tam) {
        this.tam = tam;
    }

    public String getSoort() {
        return soort;
    }

    public void setSoort(String soort) {
        this.soort = soort;
    }

    @Override
    public String kenmerken() { //Abstracte methode geïmplementeerd in concrete
subclass
        if (tam)
            return getNaam() + " behoort tot de soort: " + soort + " eet " +
getVoedsel() + " en is tam";
        return getNaam() + " behoort tot de soort: " + soort + " eet " +
getVoedsel() + " leeft in het wild";
    }

}

```