

# Uitwerkingen oefenopgaven hoofdstuk 3

## Oefenopgave 1a

```
package com.loi;
import java.util.Scanner;

public class Student {
    private String naam;
    private static double wiskTotaal = 0, aantalCijfers = 0;
    Scanner keyboard;

    public Student() {
        keyboard = new Scanner(System.in);
    }

    public String getNaam() {
        return naam;
    }

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

    public double getGemiddelde() {
        double gemiddelde = wiskTotaal / aantalCijfers;

        return gemiddelde;
    }

    public void leesCijfer() {
        System.out.print("geef cijfer voor wiskunde: ");
        double cijfer = keyboard.nextDouble();
        wiskTotaal += cijfer;
        aantalCijfers++;
    }

    public String toString() {
        return naam + " heeft voor wiskunde gemiddeld het cijfer: " +
getGemiddelde();
    }
}

package com.loi;

public class TestStudent {
```

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Student student = new Student();
    student.setNaam("Karel");
    student.leesCijfer();
    student.leesCijfer();
    System.out.print(student.toString());
}
}

```

Er zijn twee lokale variabelen in dit programma: double gemiddelde en double cijfer.

## Oefenopgave 1b

```
/**
 * p Title: JAV01 /p
 *
 * p Description: Hoofdstuk 1 cursus Java Programmeren LOI Hogeschool /p
 *
 * p Copyright: Copyright (c) 2013 LOI Hogeschool /p
 *
 * p Company: LOI Hogeschool /p
 *
 * author Erik Mols
 * version 1.0
 */
public class Kat {
    String naam;
    String ras;
    int leeftijd;
    public void miauwen(){
        System.out.println("Miaaaaauwwwww miaauuuuwwww ik ben " + naam + "
de " + ras + " ik ben " + leeftijd + " jaar oud");
    }
}

public class testKat {
    public static void main(String[] args) {
        Kat kat1 = new Kat();
        kat1.naam = "Tijger";
        kat1.ras = "Siamees";
        kat1.leeftijd = 4;
        Kat kat2 = new Kat();
        kat2.naam = "Puk";
        kat2.ras = "Bengaal";
        kat2.leeftijd = 10;
        Kat kat3 = new Kat();
        kat3.naam = "Sammie";
        kat3.ras = "Europese Korthaar";
        kat3.leeftijd = 19;
        kat1.miauwen();
        kat2.miauwen();
        kat3.miauwen();
    }
}
```

- Het zijn allebei katten – instanties van de class Kat.
- Ze verschillen in naam, ras en leeftijd – instance-variabelen!
- Ja, Puk miauwt anders dan Sammie – instance behavior!

## Oefenopgave 2

```
package par3_2;

public class Exam {
    private String name;
    static int teller = 0;
    public Exam() {
        System.out.println("de constructor wordt aangeroepen");
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
        System.out.println("setName() is aangeroepen");
    }

    @Override
    protected void finalize() throws Throwable {
        // System.out.println("Het object wordt afgebroken"); //wordt
niet aangeroepen
        ++teller;
        super.finalize();
        ++teller;
    }

    static{ System.out.println("de static initializer wordt
uitgevoerd"); }
    { System.out.println("de initializer wordt uitgevoerd"); }
}

package par3_2;

public class ObjectLifel {
    public static void main(String[] args) {
        Exam myExam = new Exam(); //object creation
        myExam.setName("OCA Java Programmer 1"); //acces method
        myExam = null; //set reference to null
        myExam = new Exam(); //object creation
        myExam.setName("PHP"); //acces method
        System.out.println("de teller = " + Exam.teller);
    }
}
```

### Oefenopgave 3a

```
package par3_3;

public class Student {
    private String naam;
    private static double wiskTotaal = 0, aantalCijfers = 0;

    public String getNaam() {
        return naam;
    }

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

    public void setCijfer(double cijfer){
        wiskTotaal += cijfer;
        aantalCijfers++;
    }

    public double getGemiddelde() {
        double gemiddelde = wiskTotaal / aantalCijfers;

        return gemiddelde;
    }

    public String toString() {
        return naam + " heeft voor wiskunde gemiddeld het cijfer: " +
getGemiddelde();
    }
}

package par3_3;

import java.util.Scanner;

public class TestStudentV2 {
    Scanner keyboard;
    public TestStudentV2() {
        keyboard = new Scanner(System.in);
    }

    public static void main(String[] args) {
        TestStudentV2 test = new TestStudentV2();
        Student student = new Student();
        student.setNaam("Karel");
        student.setCijfer(test.leesCijfer());
        student.setCijfer(test.leesCijfer());
        System.out.print(student.toString());
    }
}
```

```

    public double leesCijfer() {
        System.out.print("geef cijfer voor wiskunde: ");
        double cijfer = keyboard.nextDouble();
        return cijfer;
    }
}

```

### Vraag

U hebt waarschijnlijk in de class een static variabele opgenomen voor wiskTotaal en een aantalCijfers. Is dat terecht of moeten deze twee variabelen gewoon instance variabelen zijn?

De variabelen `private static double wiskTotaal = 0, aantalCijfers = 0;` moeten gewoon instance variabelen zijn dus: `private double wiskTotaal = 0, aantalCijfers = 0;`

Stel het volgende senario voor: Welk gemiddelde cijfer wordt afgedrukt bij Karel en Piet?

```

student1.setNaam("Karel");
student1.setCijfer(test.leesCijfer());           //4
Student student2 = new Student("Piet");
student1.setCijfer(test.leesCijfer());           //6
student2.setCijfer(test.leesCijfer());           //7
System.out.println(student1.toString());
student2.setCijfer(test.leesCijfer());           //8
System.out.println(student2.toString());

```

	wiskTotaal	aantalCijfers	output
Student student1 = new Student();	0	0	
student1.setCijfer(test.leesCijfer());	4	1	
Student student2 = new Student("Piet");	0	0	
student1.setCijfer(test.leesCijfer())	6	1	
student1.setCijfer(test.leesCijfer())	7	2	
System.out.println(student1.toString());			6.5
student1.setCijfer(test.leesCijfer())	8	3	
System.out.println(student1.toString());			7

Sommige studenten zullen blij zijn wanneer we static zo toepassen, maar de meeste niet!

Static variabelen kunnen we gebruiken wanneer we voor alle objecten van die class één variabele willen bijhouden. U zou deze twee variabelen wel kunnen gebruiken voor bijvoorbeeld het bijhouden van het gemiddelde van de klas.

### Oefenopgave 3b

```
public class Student {  
    private String naam;  
    private double wiskTotaal = 0, aantalCijfers = 0;  
    static private double klasWiskTotaal = 0, klasAantalCijfers = 0;
```

De methode die aangepast moet worden:

```
    public void setCijfer(double cijfer){  
        wiskTotaal += cijfer;  
        aantalCijfers++;  
        klasAantalCijfers++;  
        klasWiskTotaal += cijfer;  
    }
```

```
static String getKlasGemiddelde() {  
    double gemiddelde = klasWiskTotaal / klasAantalCijfers;  
    return "De klas heeft voor wiskunde gemiddeld het cijfer: " +  
gemiddelde;  
}
```

of

```
static double getKlasGemiddelde() {  
    return klasWiskTotaal / klasAantalCijfers;  
}
```

In de testclass wordt toegevoegd:

```
    System.out.println(Student.getKlasGemiddelde());
```

of

```
System.out.println("De klas heeft voor wiskunde gemiddeld het cijfer: "  
+ Student.getKlasGemiddelde());
```

#### Oefenopgave 4

```
package par3_4;

import java.lang.Math;

public class Converter {

    static int convertValue(double number){
        return (int) number;
    }

    static double convertValue(int number){
        return (double) number;
    }

    static double convertValue(double number, int aantalDecimalen){
        double deel = Math.pow(10.0, aantalDecimalen);
        double result = number * deel;
        result += 0.5;
        result = (int)result;
        result /= deel;
        return result;
    }
}

package par3_4;

public class TestConverter {

    public static void main(String[] args) {
        int getal1 = 150;
        double getal2 = 12.34567;
        double getal3 = 12.344444;
        System.out.println(Converter.convertValue(getal1));
        System.out.println(Converter.convertValue(getal2) );
        System.out.println(Converter.convertValue(getal2, 3) );
        System.out.println(Converter.convertValue(getal3, 3) );

    }
}
```



## Oefenopgave 5

Een deel van de code van Student: (Zie oefenopgave 4)

```
public class Student {
    private String naam;
    double wiskTotaal = 0, aantalCijfers = 0;

    public Student() {
        wiskTotaal = 0; aantalCijfers = 0;
    }

    public Student(String naam) {
        this();
        this.naam = naam;
    }

    public Student(String naam, double cijfer) {
        this();
        this.naam = naam;
        setCijfer(cijfer);
    }

    public Student(double cijfer, String naam) {
        this(naam, cijfer);
    }
}
```

Een deel van de code van TestStudent. Zie oefenopgave 4

```
public static void main(String[] args) {
    TestStudentV2 test = new TestStudentV2();
    Student student1 = new Student();
    student1.setNaam("Karel");
    student1.setCijfer(test.leesCijfer());
    student1.setCijfer(test.leesCijfer());
    System.out.println(student1.toString());

    Student student2 = new Student("Piet");
    student2.setCijfer(test.leesCijfer());
    student2.setCijfer(test.leesCijfer());
    System.out.println(student2.toString());

    Student student3 = new Student("Klaas", 5.5);
    student3.setCijfer(test.leesCijfer());
    System.out.println(student3.toString());

    Student student4 = new Student(4.4, "Jan");
    student4.setCijfer(test.leesCijfer());
    System.out.println(student4.toString());
}
```

## Oefenopgave 6

Een deel van de code van Student:

```
public void setCijfer(double... cijfers){
    for (int i = 0; i < cijfers.length; i++)
        setCijfer(cijfers[i] );
}

public Student(String naam, double... cijfers){
    this(naam);
    for (int i = 0; i < cijfers.length; i++)
        setCijfer(cijfers[i] );
}
```

### Een deel van de code van TestStudent

```
Student student5 = new Student("Marie");
student5.setCijfer(4.5 , 5.5, 7);
System.out.println(student5.toString());
Student student6 = new Student("Evelien");
student5.setCijfer(4.5 , 5.5, 7, 3, 9);
S Student student7 = new Student("Jantine", 4, 5 , 6 ,7);
System.out.println(student7.toString());system.out.println(student2.toString());
```

### Vraag

Hoe vaak wordt in de class student een overloaded constructor gemaakt en hoe vaak een overloaded methode?

```
public Student() {
public Student(String naam) {
public Student(String naam, double cijfer) {
public Student(double cijfer, String naam) {
public Student(String naam, double... cijfers){
5 keer wordt een overloaded constructor gemaakt.
```

```
public void setCijfer(double cijfer)
public void setCijfer(double... cijfers)
2 keer wordt een overloaded methode gemaakt.
```

### Vraag

Zou u zelf de class Student zo ontworpen hebben?

Het antwoord zal waarschijnlijk nee luiden. U kunt beter een arraylist toepassen waarin u de cijfers stopt. Dit wordt in een later hoofdstuk behandeld.

### Oefenopgave 7

Hoe wordt encapsulation toegepast in de class Student?

Door de variabelen private te maken.

private String naam;

static private double wiskTotaal = 0, aantalCijfers = 0;

Deze private variabelen kun u in andere classes alleen benaderen met behulp van de public methodes.

Hoe wordt informaton hiding toegepast in de class Student?

Met informaton hiding en encapsulation wordt hetzelfde bedoeld.

## Oefenopgave 8

Wat is het verschil tussen call by reference en call by value?

Bij objecten wordt call by reference toegepast, deze komen in de heap.

Bij primitieve variabelen wordt call by value toegepast die komen op stack.

```
        swap(student1, student2);
        System.out.println(student1.getNaam() + ":" +
student2.getNaam());
        Student tmp = student1;
        student1 = student2;
        student2 = tmp;
        System.out.println(student1.getNaam() + ":" +
student2.getNaam());
    }

    public static void swap(Student p1, Student p2) {
        Student temp = p1;
        p1 = p2;
        p2 = temp;
    }
```

De namen bij de laatste println zijn wel gewisseld.