

Problem 1, en klass där instansen hanteras som ett postobjekt

```
emacs@LAPTOP-T7EPJS9M
/*
Uppgift1
Skapa en klass för studenter som
hanterar så kallade postobjekt.
Tomas Berggren, tombe691@gmail.com
2019-05-02
*/

public class Probleml {
    // Instance Variables
    int number;
    String name;
    String adress;
    int points;

    // method 1
    public String getInfo() {
        return ("Student is: number "+number+" Name is: "+name+" Adress is: "+adress+" points are: "+points);
    }

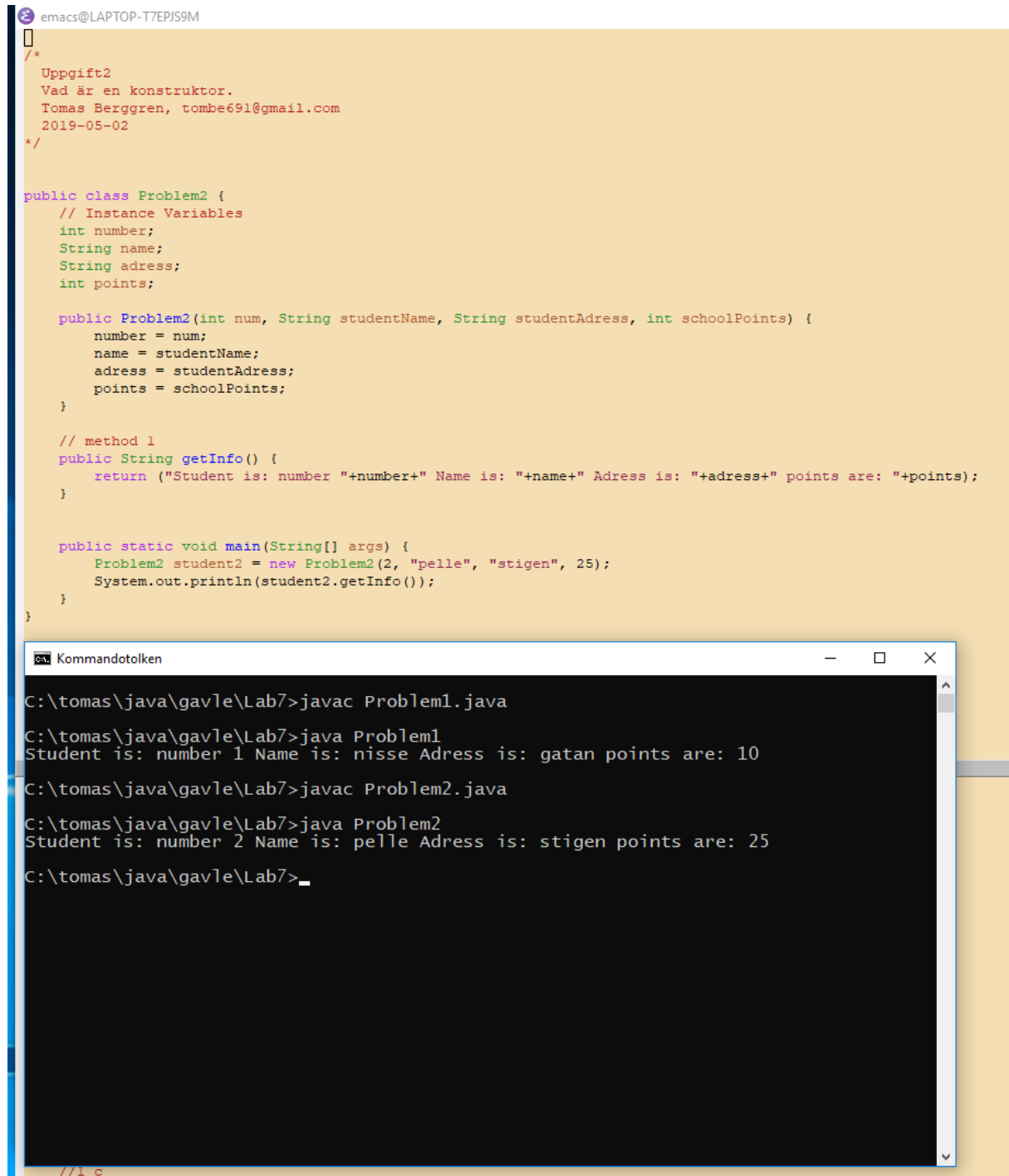
    public static void main(String[] args) {
        Probleml student1 = new Probleml();
        student1.number=1;
        student1.name="nisse";
        student1.adress="gatan";
        student1.points=10;
        System.out.println(student1.getInfo());
    }
}

l\--- Probleml.java|Lab7  All L1  Git-master  (Java//1 Abbrev)

Kommandotolken
C:\tomas\java\gavle\Lab7>javac Probleml.java
C:\tomas\java\gavle\Lab7>java Probleml
Student is: number 1 Name is: nisse Adress is: gatan points are: 10
C:\tomas\java\gavle\Lab7>_
}
```

Problem 2

Jag har utgått från samma uppgift som i problem 1 och har sedan skrivit en konstruktor till denna klass. En konstruktor skapar en instans av klassen och kan skicka med defaultvärden på klassattribut som då sätts på instansen. Använder man alltid en konstruktor när man skapar ett objekt så vet man att de grundläggande attributen är satta. En konstruktor skrivs som en metod med samma namn som klassen. Den kan ha så många parametrar man anser lämpligt och det går skriva flera konstruktörer där endast parameterlistan skiljer, så kallade överlagrade konstruktörer.



```
emac@LAPTOP-T7EPJ59M
/*
Uppgift2
Vad är en konstruktor.
Tomas Berggren, tombe691@gmail.com
2019-05-02
*/

public class Problem2 {
    // Instance Variables
    int number;
    String name;
    String adress;
    int points;

    public Problem2(int num, String studentName, String studentAdress, int schoolPoints) {
        number = num;
        name = studentName;
        adress = studentAdress;
        points = schoolPoints;
    }

    // method 1
    public String getInfo() {
        return ("Student is: number "+number+" Name is: "+name+" Adress is: "+adress+" points are: "+points);
    }

    public static void main(String[] args) {
        Problem2 student2 = new Problem2(2, "pelle", "stigen", 25);
        System.out.println(student2.getInfo());
    }
}

Kommandotolken
C:\tomas\java\gavle\Lab7>javac Problem1.java
C:\tomas\java\gavle\Lab7>java Problem1
Student is: number 1 Name is: nisse Adress is: gatan points are: 10
C:\tomas\java\gavle\Lab7>javac Problem2.java
C:\tomas\java\gavle\Lab7>java Problem2
Student is: number 2 Name is: pelle Adress is: stigen points are: 25
C:\tomas\java\gavle\Lab7>_

//1 c
```

Problem 3 – ArrayTools

emacs@LAPTOP-T7EPIS9M

```
/*
 * Uppgift 3
 * ArrayTools.
 * Tomas Berggren, tombe691@gmail.com
 * 2019-05-08
 */
import java.util.Arrays;
//import java.util.*;

public class ArrayTools {
    // Driver method
    /* public static void main(String args[])
    {
        int arr[] = createIntArray(3);

        arr[0] = 45;
        arr[1] = 1;
        arr[2] = 67;

        printIntArray(arr);
        int min = getMinInt(arr);
        System.out.print(min);
        int max = getMaxInt(arr);
        System.out.print(max);
        int find = containsIntElement(arr, 1);
        System.out.print(find);
    } */

    public static int[] createIntArray(int l)
    {
        // returning array
        int[] arr = new int[l];
        return arr;
    }

    public static void printIntArray(int[] arr)
    {
        for (int i = 0; i < arr.length; i++)
            System.out.print(arr[i]+" ");
    }

    public static int getMinInt(int[] arr)
    {
        int arrcopy[] = arr.clone();
        Arrays.sort(arrcopy);
        return arrcopy[0];
    }

    public static int getMaxInt(int[] arr)
    {
        int arrcopy[] = arr.clone();
        Arrays.sort(arrcopy);
        return arrcopy[arrcopy.length-1];
    }

    public static int containsIntElement(int[] arr, int val)
    {
        int pos = -1;
        for (int i = 0; i < arr.length; i++) {
            if (val == arr[i]){
                pos = i;
            }
        }
        return pos;
    }
}
```

Kommandotolken

```
C:\tomas\java\gavle\Lab7>javac Problem3Test.java ArrayTools.java
C:\tomas\java\gavle\Lab7>java Problem3Test
Test ArrayTools.createIntArray - skapar en array med 10 element
Resultat: Array med 10 element
OK
Tilldelning: A[2] <= 3, A[6] <= 8.
Test ArrayTools.printIntArray
0 0 3 0 0 0 8 0 0 0 Test ArrayTools.getMinInt
Resultat: 0
OK
Test ArrayTools.getMaxInt
Resultat: 8
OK
Test ArrayTools.containsIntElement - element som efterlyses: 3
Resultat: 2
OK
Test ArrayTools.containsIntElement - element som efterlyses: 0
Resultat: 9
OK
Test ArrayTools.containsIntElement - element som efterlyses: 9
Resultat: -1
OK
C:\tomas\java\gavle\Lab7>
```

Problem 4 – Point

Metoden för att komma till origo anropar den första metoden och använder 0, 0, 0 som en av punkterna.

```

/*
Uppgift4
Point.
Tomas Berggren, tombe691@gmail.com
2019-05-08
*/

public class Point {
    // Instance Variables
    private double x;
    private double y;
    private double z;

    public Point(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    // method 1
    public String getInfo(Point point) {
        return ("Point coordinates are: x: "+point.x+" y: "+point.y+" z: "+point.z);
    }

    public double[] getValues(Point point)
    {
        // returning array
        double[] arr = new double[3];
        arr[0] = this.x;
        arr[1] = this.y;
        arr[2] = this.z;
        return arr;
    }

    public double calcDistance(Point pointA, Point pointB)
    {
        double dist;
        dist = Math.sqrt(Math.pow(pointA.x - pointB.x, 2) +
                        Math.pow(pointA.y - pointB.y, 2) +
                        Math.pow(pointA.z - pointB.z, 2));

        return dist;
    }

    public double calcDistanceToOrigo(Point pointA)
    {
        double dist;
        Point pointB = new Point(0, 0, 0);
        dist = Math.sqrt(Math.pow(pointA.x - pointB.x, 2) +
                        Math.pow(pointA.y - pointB.y, 2) +
                        Math.pow(pointA.z - pointB.z, 2));

        return dist;
    }

    public void test(){
        Point pointA = new Point(0, 0, 0);
        Point pointB = new Point(2, 2, 0);
        double dist = calcDistance(pointA, pointB);
        System.out.println(dist);
    }

    public static void main(String[] args) {
        Point point = new Point(1, 2, 3);
        System.out.println(point.getInfo(point));
        point.test();
    }
}

```

Kommandotolken

```

C:\tomas\java\gavle\Lab7>javac Point.java
C:\tomas\java\gavle\Lab7>java Point
Point coordinates are: x: 1.0 y: 2.0 z: 3.0
2.8284271247461903
C:\tomas\java\gavle\Lab7>

```