

## Lösningar för tenta 3 DAT043, 2018-08-22.

### Uppgift 1

```
class Person{
    public String förnamn;
    public String efternamn;
    public int ålder;
    private double längd;
    private double vikt;

    public Person(String förnamn, String efternamn, int å,
                  double l, double v){
        this.förnamn = förnamn;
        this.etternamn = efternamn;
        ålder = å;
        längd = l;
        vikt = v;
    }

    public Person(String förnamn, String efternamn, int å){
        this.förnamn = förnamn;
        this.etternamn = efternamn;
        ålder = å;
        // Eftersom vi låter bli att initiera längd/vikt får dessa
        // defaultvärdet för double dvs 0.0
    }
}
```

### Uppgift 2

```
public static double[][] transponera(double[][] a) {
    int m = a.length;
    int n = a[0].length;
    double[][] b = new double[n][m];
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            b[j][i] = a[i][j];
    return b;
}
```

### Uppgift 3

Felet är att man glömt lägga till knappen till vår JFrame. Detta görs med metoden add, som även kan ta ett Layout-objekt som argument (här har vi valt BorderLayout.SOUTH eftersom vi vill ha knappen längst ner).

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Uppgift3 implements Runnable {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Uppgift3());
    }

    private class ButtonActionListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("Löst uppgiften!");
        }
    }

    public void run() {
        JFrame frame = new JFrame("Uppgift 3");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton button = new JButton("Skriv 'Löst uppgiften!'");
        button.addActionListener(new ButtonActionListener());

        frame.setPreferredSize(new Dimension(300, 250));

        // Denna rad saknades.
        frame.add(button, BorderLayout.SOUTH);

        frame.pack();
        frame.setVisible(true);
    }
}
```

### Uppgift 4

Vi påminner oss om att variabler av objekttyp i Java egentligen innehåller en referens till en minnesadress där objektets data finns lagrad. Om man bara ger ett objekt direkt som argument till metoden System.out.println, så kommer denna referens att skrivas ut, inte innehållet i objektet (detta beror på att den ärvda metoden toString från klassen Object anropas av System.out.println, och den skriver ut objektets referens).

Vill man skriva ut innehållet i en array är det lämpligt att först gör om den till en sträng som man sedan skriver ut. Detta görs exempelvis med följande biblioteksmetod som finns i lathunden:

```
import java.util.*;

public class Uppgift4{

    public static void main(String[] args) {
        int [] a = {1,2,3,4};
        System.out.println(Arrays.toString(a));
    }
}
```

### Uppgift 5

B, D och H är sanna.

### Uppgift 6

Klassvariablerna min och max måste deklarerars med nyckelordet static, vilket programmeraren har missat. Annars tror Java att min och max är instansvariabler, dvs man kommer ha en kopia av max och min för varje termometer-objekt, vilket inte var avsikten. Det ska bara finnas en global variabel min/max som registrerar högsta/lägsta temperatur som någon termometer har mätt upp. De ska inte registrera min/max för varje enskild termometer (i så fall skulle de mycket riktigt ha varit instansvariabler, men det är inte det vi är ute efter här).

```
class Termometer{

    public static double max;
    public static double min;
    ...
}
```

## Uppgift 7

Notera att för att få godkänt på denna uppgift krävs att man även demonstrerar att man förstår hur generics används genom att korrekt kunna skapa objekt som tillhör den generiska klassen, genom att instantiera typvariabeln (se main metoden nedan). Poängen är ju att man inte längre behöver göra någon type cast när man sedan plockar ut objekt ur boxen. Här hade tyvärr många bara kopierat koden i frågan rakt av, vilket inte gett rätt svar.

```
class GenericBox<T>{
    private T content;

    public void put(T thing){
        content = thing;
    }
    public T remove(){
        T thing = content;
        content = null;
        return thing;
    }

    public static void main(String[] args) {
        GenericBox<String> b1 = new GenericBox<String>();
        GenericBox<Integer> b2 = new GenericBox<Integer>();
        b1.put("hello");
        b2.put(0);
        b1.put("world");
        //b1.put(1); // Kompilerar ej med detta!
        String s = b1.remove();
        Integer i = b2.remove();
    }
}
```

## Uppgift 8

```
public static <E extends Comparable<E>> void selectionSort(E[] arr) {
    for (int i = 0; i < arr.length; i++) {
        E min = arr[i]; // Håller reda på minsta element den osorterade delen.
        int minidx = i; // Index för minsta element i osorterad del.
        // Inre loop går igenom resten av arr för att se om det finns något mindre element
        for (int j = i + 1; j < arr.length; j++) {
            int c = arr[j].compareTo(min);
            if (c < 0) { // Om vi hittar något mindre än nuvarande minimum...
                min = arr[j]; // ...updatera min och...
                minidx = j; // ... uppdatera index för minsta element.
            }
        }
        if (minidx > i) { // Om vi hittade ett mindre element!
            arr[minidx] = arr[i]; // Byt plats på minsta element och arr[i]
            arr[i] = min; // Byt plats, minsta element hamnar i sorterad del.
        }
    }
}
```

## Uppgift 9

```
public static List<Car> getSelectedCarList(String filename, Predicate<Car> p){
    Scanner in;
    ArrayList<Car> l = new ArrayList<>();
    try{
        in = new Scanner(new File(filename));
    }
    catch(FileNotFoundException e){
        System.out.println(e.getMessage());
        return l;
    }
    while(in.hasNext()){
        String s = in.nextLine();
        String[] values = s.split(",");
        if (values.length != 3){
            System.out.println("Faulty entry in input file: "+Arrays.toString(values));
            return l;
        }
        Car c = new Car(values[0],values[1], Integer.parseInt(values[2]));
        if(p.test(c)){
            l.add(c);
        }
    }
    return l;
}
```

```
List<Car> l = getSelectedCarList("Uppgift9.txt",
                                c -> c.brand.equals("Volvo") && c.year >= 2000);
```