

# Tentamen Programmeren 1 — Java

## Dinsdag 17 december 2002, 10.00–13.00 uur

### Universiteit Leiden — Informatica

Geef overal duidelijke uitleg. Veel succes!

#### 1. (2 punten)

- a. Waar staat het 100-ste array-element van het array `iets`?
- b. Wat is het verschil tussen Java-applets en stand alone Java applicaties?
- c. Wat wordt er afgedrukt door het volgende stukje Java:

```
int x = 1000; int y = 10;
System.out.println ( x + y );
System.out.println ("Som: " + x + y + ".");
System.out.println ("Som: " + ( x + y ) + ".");
```

Leg uit!

- d. Hoe kun je in een Java-applet zonder gebruik te maken van de methode `paint` (via `repaint`) toch iets afdrukken in het huidige grafische window, en waarin verschilt dit van `paint`?

#### 2. (2 punten)

We hebben een array `A` met `A.length` verschillende positieve gehele getallen.

- a. Geef een Java-functie `public int som (int[ ] A)` die de *som* van de array-elementen van het array `A` retourneert.
- b. Geef een Java-functie `gem (A)` die het gemiddelde van de array-elementen van het array `A` retourneert. Gebruik de functie van **a**.
- c. Leg in *woorden* uit hoe *binair zoeken* verloopt.
- d. Leg uit wat in Java het verschil is tussen het doorgeven van een `int` aan een functie en het doorgeven van een array aan een functie.

### 3. (2 punten)

a. Bij een functie kun je te maken hebben met *locale* en *globale* variabelen. Verder heb je ook nog variabelen (parameters) die tussen haakjes worden meegegeven. Leg deze begrippen aan de hand van het volgende voorbeeld duidelijk uit:

```
private int globaal = 10;
public void doewat (int x) {
    int temp = x + 2; x = 3; globaal = 5; }
```

b. Een zeker Java-programma bevat de volgende functies:

```
public int telop1 ( ) { int temp = a - b; return temp; }
public int telop2 (int a, int b) { int temp = a - b; return temp; }
public int telop3 ( ) { a = a - b; b = 0; return a; }
public int telop4 (int a, int b) { a = a - b; b = 0; return a; }

public void alert (int a, int b, int x) {
    System.out.println (a + " " + b + " " + x); }
```

Wat gebeurt er bij het volgende stukje Java? NB a, b en x zijn “globale” variabelen.

```
a = 15; b = 4; x = telop1 ( ); alert (a,b,x);
a = 15; b = 4; x = telop2 (a,b); alert (a,b,x);
a = 15; b = 4; x = telop2 (b,a); alert (a,b,x);
a = 15; b = 4; x = telop3 ( ); alert (a,b,x);
a = 15; b = 4; x = telop4 (a,b); alert (a,b,x);
a = 15; b = 4; x = telop4 (b,a); alert (a,b,x);
```

Wat wordt er via de alert’s afgedrukt? Geef hierbij uiteraard uitleg.

### 4. (2 punten)

Deze opgave gaat over een m bij n array A met gehele getallen tussen 0 en 99. Een voorbeeld, met m = 3 rijen en n = 5 kolommen:

```
11 30 0 4 5
0 0 18 3 6
3 0 1 99 1
```

a. Schrijf een Java-functie `drukaf (A)` die het array A precies als boven afdrukt, dus n getallen per regel, en de kolommen rechts uitgelijnd.

Gebruik `System.out.print(ln)` voor het schrijven naar het beeldscherm.

b. Schrijf een stukje Java dat een m bij n array A maakt (gebruik `new`), en de array-elementen als volgt vult. In de i-de rij, j-de kolom komt de som van i en j te staan, en 0 als deze som groter dan 99 is.

c. Schrijf een Java-functie `hoeveel (A)` die bepaalt hoeveel array-elementen er zijn waarvan *alle buren* ongelijk aan 0 zijn. Een buur van een array-element is een array-element dat er horizontaal, verticaal of diagonaal direct aan grenst. In het voorbeeld zijn het er 4, namelijk 99, 5, 6 en 1.

## 5. (2 punten)

We bekijken de volgende Java-applet die een scrollbar, **trekker** geheten, op het scherm zet:

```
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
public class Iets extends Applet implements AdjustmentListener {
    private Scrollbar trekker;
    private int waarde = 0;
    public void init ( ) {
        trekker = new Scrollbar (Scrollbar.HORIZONTAL,0,1,0,100);
        add (trekker);
        trekker.addAdjustmentListener (this);
    } // init
    public void paint (Graphics g) {
        g.drawString ("Nu is het " + waarde,100,100);
    } // paint
    public void adjustmentValueChanged (AdjustmentEvent e) {
        waarde = trekker.getValue ( );
        repaint ( );
    } // adjustmentValueChanged
} // Iets
```

- a. Leg de werking van dit programma uit. Met name: waarvoor is de functie `init`? En wat doet de functie `adjustmentValueChanged`?
- b. Voeg een functie `kwadraat (i)` toe (die het kwadraat van `i` berekent).
- c. Voeg een tweede scrollbar toe, met waarde `waarde2`. Zorg ervoor dat in `paint` het kwadraat van de som van de twee in de scrollbars ingevoerde waarden wordt afgedrukt.
- d. Stel dat we in de functie `adjustmentValueChanged` bij een “event” onderscheid willen maken tussen de twee scrollbars. Hoe doen we dat? En waarom is dit in het algemeen sowieso netjes?