

## uitwerking tentamen Imperatief Programmeren 8 maart 2001

```
1a. public void paint(Graphics gr)
{
    for (int y=0; y<10; y++)
        for (int x=0; x<=y; x++)
            gr.drawRect(x*20, y*20, 10, 10);
}
```

1b. Static methoden hebben geen object onder handen, niet-static methoden hebben dat wel. Daarom hoeft bij aanroep van een static methode geen object voor de punt te worden geschreven, maar kan worden volstaan met de naam van de klasse. Daarom kan een niet-static methode wel een static methode aanroepen (hij 'vergeet' dan gewoon het object waar hij mee bezig is), maar kan het niet andersom.

1c. De test s==t vergelijkt of de objectverwijzingen s en t naar hetzelfde object verwijzen, dat wil zeggen of de pijlen gelijk zijn. De test s.equals(t) vergelijkt de aparte letters van de twee strings. Als de pijlen naar hetzelfde object wijzen, is de inhoud natuurlijk ook hetzelfde, dus de eerste uitspraak is 'altijd'. Twee strings met gelijke inhoud hoeven echter niet dezelfde kopie te zijn, dus de tweede uitspraak is 'soms'.

1d. Zowel een array als een Vector kunnen worden gebruikt om meerdere gegevens op te slaan. Voordelen van array zijn bijvoorbeeld efficiënte opslag, en het feit dat ook primitive types als elementen kunnen optreden. Voordelen van Vector zijn bijvoorbeeld dat de lengte kan worden vergroot, en dat je gemakkelijk elementen kunt tussenvoegen.

```
1e. g.setColor(new Color(255,255,0));
```

2. zie aparte listing

3a. Het is een application, de gebruiker kan dus op de commandoregel filenamen intikken, die in het programma via de parameter van main beschikbaar zijn.

3b. De eerste new maakt een array van object-verwijzingen, maar nog niet de aparte objecten. Die worden daarom in de for-opdracht ieder apart gecreëerd.

3c. De foutmelding is 'array index out of bounds' die ontstaat als in methode rapportage info[m] wordt uitgerekend, terwijl info een array is met lengte 0. Dit is een **runtime** foutmelding.

```
3d.
BufferedReader br;
String regel;
br = new BufferedReader(new FileReader(naam));
while ((regel = br.readLine())!=null)
    verwerk(regel);
```

```
3e.
if (c>='A' && c<='Z')
    aantal[c-'A']++;
else if (c>='a' && c<='z')
    aantal[c-'A']++;
```

```
3f. m = 0;
for (int i=1; i<info.length; i++)
    if (info[i].freq(c) > info[m].freq(c))
        m = i;
```

```
3g.
class ExtraData extends Data
{
    // nieuwe variabelen
    int regels, totaal;
    // herdefinitie methode
    public void verwerk(String regel)
    { super.verwerk(regel);
      regels++; totaal+=regel.length();
    }
    public double gemRegLen()
    { return totaal/(double)regels;
    }
}
```

```

import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;

public class Cirkels extends Applet implements ActionListener,
        AdjustmentListener, MouseListener
{
    int []xs; int []ys; int aantal; Color kleur;
    Button clear, black, green;
    Scrollbar sizer;

    public void init()
    {
        xs = new int[100];
        ys = new int[100];
        aantal = 0;
        kleur = Color.black;

        clear = new Button("clear");
        black = new Button("black");
        green = new Button("green");
        sizer = new Scrollbar(Scrollbar.HORIZONTAL, 8, 1, 2, 20);

        add(clear); add(sizer); add(black); add(green);
        clear.addActionListener(this);
        black.addActionListener(this);
        green.addActionListener(this);
        sizer.addAdjustmentListener(this);
        this.addMouseListener(this);
    }

    public void paint(Graphics gr)
    {
        int i, r;
        r = sizer.getValue();
        gr.setColor(kleur);
        for (i=0; i<aantal; i++)
            gr.fillOval(xs[i]-r, ys[i]-r, 2*r, 2*r);
    }

    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource()==clear)
            aantal = 0;
        else if (e.getSource()==black)
            kleur = Color.black;
        else if (e.getSource()==green)
            kleur = Color.green;
        repaint();
    }

    public void adjustmentValueChanged(AdjustmentEvent e)
    {
        repaint();
    }

    public void mousePressed(MouseEvent e)
    {
        if (aantal<100)
        {
            xs[aantal] = e.getX();
            ys[aantal] = e.getY();
            aantal++;
            repaint();
        }
    }

    public void mouseReleased(MouseEvent e){}
    public void mouseClicked(MouseEvent e){}
    public void mouseEntered(MouseEvent e){}
    public void mouseExited(MouseEvent e){}
}

```