

Uitwerking 2e deeltentamen Imperatief Programmeren

13 oktober 2004

Opgave 1 (30%)

```
public boolean equalsIgnoreCase1(String s)
{
    int t, n;
    char c, d;
    n = this.length();
    if (n!=s.length()) return false;

    for (t=0; t<n; t++)
    {
        c = this.charAt(t);
        d = s.charAt(t);
        if (c>='a' && c<='z')
            c = (char)(c-'a'+'A');
        if (d>='a' && d<='z')
            d = (char)(d-'a'+'A');
        if (c!=d)
            return false;
    }
    return true;
}

public boolean equalsIgnoreCase2(String s)
{
    return this.toUpperCase().equals(s.toUpperCase());
}
```

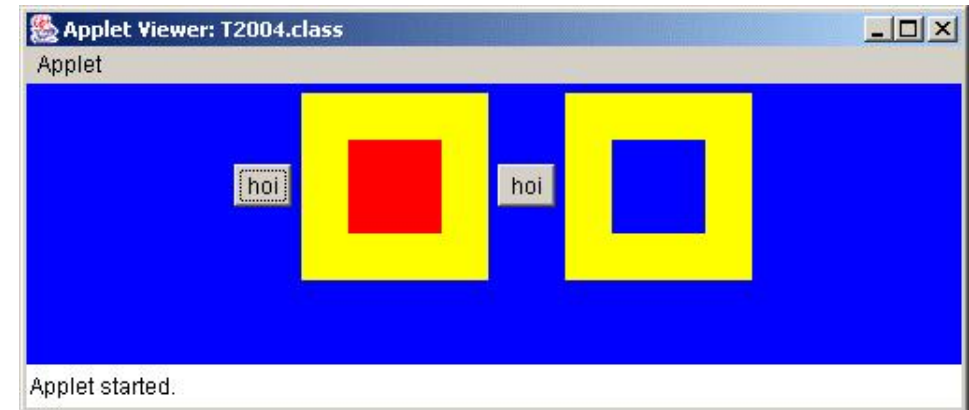
Opgave 2 (30%)

- a. Klasse Test implements `MouseMotionListener`. Dit is nodig omdat in de klasse Test `this` wordt gebruikt als parameter van `addMouseMotionListener`, en die heeft een `MouseMotionListener` nodig. Klasse Ruimte implements `ActionListener`, `Runnable`. Het eerste is nodig omdat in de klasse Ruimte `this` wordt gebruikt als parameter van `addActionListener`, en die heeft een `ActionListener` nodig. Het tweede

is nodig omdat in klasse Test het object `r2`, wat een Ruimte-object is, als parameter wordt meegegeven aan de constructor van `Thread`, en die heeft een `Runnable` nodig.

- b. In Test moeten de methoden `mouseMoved` en `mouseDragged` worden gedefinieerd, want dat wordt beloofd door `MouseListener`. In Ruimte moeten de methoden `actionPerformed` en `run` worden gedefinieerd, want dat wordt beloofd door resp. `ActionListener` en `Runnable`.

c.



De button wordt weliswaar in de constructor van Ruimte gemaakt, maar hij wordt toegevoegd aan de Container die daaraan als parameter wordt meegegeven, en dat is de hele applet! De button komt dus niet in het gele gedeelte te staan, maar ernaast. En omdat de constructor van Ruimte tweemaal wordt aangeroepen, worden er dus ook twee buttons gemaakt.

- d. Omdat het in die klasse staat, is `this` hier een object van type Test. Dat is een subklasse van Applet, en dat is weer een subklasse van Panel, en dat is weer een subklasse van Container. Daarom is `this` acceptabel als eerste parameter van de Ruimte-constructor, die immers een Container verwacht. Omdat de klasse Test de interface `MouseMotionListener` implementeert, is `this` ook acceptabel als parameter van `addMouseMotionListener`.
- e. De aanroep `t.start()` roept de methode `run` aan van het object dat bij de constructie van `t` als parameter is meegegeven, dus van `r2`. Het bijzondere is dat de aanroep van `start` meteen terugkeert, ook als `run` veel tijd kost. Vanaf dat moment gebeuren er dus twee dingen tegelijk.

Opgave 3 (40%)

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

public class BB extends Applet implements
    MouseListener, ActionListener, AdjustmentListener
{
    int [] xs; int [] ys;
    int n, r, m, minx, miny, maxx, maxy;
    Button b, c;
    Scrollbar s;
    boolean box;

    public void init()
    {
        r = 15;
        m = 150;
        box = false;

        b = new Button("Box");
        this.add(b); b.addActionListener(this);
        c = new Button("Clear");
        this.add(c); c.addActionListener(this);
        s = new Scrollbar(Scrollbar.HORIZONTAL, 1, 1, 1, 20);
        this.add(s); s.addAdjustmentListener(this);
        xs = new int[100];
        ys = new int[100];
        this.schoon();
        this.addMouseListener(this);
    }

    public void schoon()
    {
        xs[0] = m;
        ys[0] = m;
        n = 1;
        minx = m;
        miny = m;
        maxx = m;
        maxy = m;
    }
}
```

```
public void paint(Graphics g)
{
    int t;
    for (t=0; t<n; t++)
    {
        g.fillOval(xs[t]-r, ys[t]-r, 2*r, 2*r);
    }
    if (box)
    {
        for (t=0; t<s.getValue(); t++)
            g.drawRect(minx-r-t, miny-r-t,
                maxx-minx+2*(r+t), maxy-miny+2*(r+t) );
    }
}

public void actionPerformed(ActionEvent e)
{
    if (e.getSource()==c)
        this.schoon();
    else box = !box;
    this.repaint();
}

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

public void mouseClicked(MouseEvent e)
{
    int x, y;
    x = e.getX();
    y = e.getY();
    if (n<100)
    {
        xs[n] = x;
        ys[n] = y;
        n++;
        if (x<minx) minx = x;
        if (x>maxx) maxx = x;
        if (y<miny) miny = y;
        if (y>maxy) maxy = y;
        this.repaint();
    }
}

public void mousePressed(MouseEvent e){}
public void mouseReleased(MouseEvent e){}
public void mouseEntered(MouseEvent e){}
public void mouseExited(MouseEvent e){}
}
```