

Uitwerking tweede deeltentamen Imperatief Programmeren

maandag 22 maart 2004

Opgave 1

- a. Een interface bevat alleen methode-headings, een klasse kan ook objectvariabelen, constructoren en methode-bodies bevatten. Een klasse kan worden gebruikt om objecten aan te maken, een interface niet.
- b. De woorden `implements ActionListener` in de header van een klasse betekenen dat die klasse een methode `public void actionPerformed(ActionEvent e)` bevat.
- c. Een boolean expressie is een expressie die een waarheidswaarde (`true` of `false`) voorstelt.
- d. De klasse `Thread` dient om een afzonderlijk proces te starten dat parallel aan de gewone programma-uitvoering verloopt. Deze wordt gebruikt door een klasse de interface `Runnable` te laten implementeren door middel van een methode `void run()` en een object van die klasse mee te geven aan de constructor van `Thread`.
- e. De waarde `null` dient om de afwezigheid van een object te symboliseren.
- f. Zowel `this` als `super` verwijzen naar het object dat de code uitvoert, maar bij `super.m()` wordt de implementatie van `m()` in de superklasse van het object gekozen.

Opgave 2

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

public class Kakutani extends Applet implements ActionListener
{
    TextField t;

    public void init()
    {
        t = new TextField("1", 10);
        this.add(t);
        t.addActionListener(this);
    }

    public void paint(Graphics g)
    {
        int k, waarde;
        waarde = Integer.parseInt(t.getText());
    }
}
```

```

        for (k=0; k<30; k++)
        {
            g.drawString("" + k, 10, 50 + 15 * k);
            g.drawString("" + waarde, 40, 50 + 15 * k);
            waarde = opvolger(waarde);
        }
    }

    public void actionPerformed(ActionEvent e)
    {
        repaint();
    }

    private int opvolger(int w)
    {
        if (w % 2 == 0)
            return w / 2;
        return 3 * w + 1;
    }
}

```

Opgave 3

```

public static boolean isAddress(String s)
{
    int k, aantal;
    k = 0;
    aantal = 0;
    while (k < s.length() && aantal <= 2)
    {
        if (s.charAt(k) == '@') aantal++;
        k++;
    }
    return aantal == 1;
}

```

Opgave 4

```

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

public class Zon extends Applet implements AdjustmentListener
{
    Scrollbar s;
    int hoogte;

    public void init()
    {
        s = new Scrollbar(Scrollbar.HORIZONTAL, 10, 1, 0, 200);
        this.add(s);
    }
}

```

```

        s.addAdjustmentListener(this);
        hoogte = 0;
    }

    public void paint(Graphics g)
    {
        tekenAchtergrond(g);
        tekenZon(g);
        tekenVoorgrond(g);
    }

    public void adjustmentValueChanged(AdjustmentEvent e)
    {
        hoogte = s.getValue();
        repaint();
    }

    private void tekenAchtergrond(Graphics g)
    {
        g.setColor(Color.blue);
        g.fillRect(0, 0, 300, 200);
    }

    private void tekenZon(Graphics g)
    {
        int k;
        double hoek;
        g.setColor(Color.yellow);
        g.fillOval(130, 180-hoogte, 40, 40);
        for (k=0; k<20; k++)
        {
            hoek = k * Math.PI / 10;
            g.drawLine( 150,
                        200-hoogte,
                        150 + (int)(300 * Math.cos(hoek)),
                        200 - hoogte - (int)(300 * Math.sin(hoek))
                        );
        }
    }

    private void tekenVoorgrond(Graphics g)
    {
        g.setColor(Color.white);
        g.fillRect(0, 200, 300, 100);
    }
}

```