

Uitwerking eerste deeltentamen Imperatief Programmeren

donderdag 23 februari 2006

Opgave 1

- Een `int` wordt naar een `double` geconverteerd als argument van `+` als het andere argument al een `double` is. Een `int` wordt naar een `String` geconverteerd als argument van `+` als het andere argument al een `String` is.
- Als `i` type `int` heeft, heeft `"" + i` of `String.valueOf(i)` type `String`.
- Een `Scanner`-object kan zinnen in woorden splitsen. De methode `next` levert het eerstvolgende woord op van de tekst die als parameter bij de constructor van het object werd aangeboden.
- De woorden `implements ActionListener` vormen een belofte dat in de klasse een methode met header

```
public void adjustmentValueChanged(AdjustmentEvent e)
```

voorkomt. Die methode wordt aangeroepen als een `AdjustmentEvent` optreedt, bijvoorbeeld als de gebruiker de stand van een schuifregelaar verandert.
- Een klasse is een groep bij elkaar behorende methoden. Een klasse is ook het type van objectverwijzingen. De methoden in een klasse hebben een object van dat type onderhanden.
- Methode `paint` heeft een parameter van type `Graphics`, methode `repaint` heeft geen parameter. Methode `paint` wordt door de browser automatisch aangeroepen, methode `repaint` wordt expliciet aangeroepen door de code van de applet.

Opgave 2

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

public class Rood extends Applet implements ActionListener
{
    private Color kleur;

    public void init()
    {
        kleur = Color.BLUE;
        Button rood = new Button("Rood");
        add(rood);
        rood.addActionListener(this);
    }
}
```

```

public void paint(Graphics g)
{
    g.setColor(kleur);
    g.fillRect(0,0,300,300);
}

public void actionPerformed(ActionEvent e)
{
    kleur = Color.RED;
    repaint();
}
}

```

Opgave 3

```

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

public class Food extends Applet
{
    public void paint(Graphics g)
    {
        String carbo = getParameter("carbohydrates");
        String fats = getParameter("fats");
        String prot = getParameter("proteins");

        int icarbo = Integer.parseInt(carbo);
        int ifats = Integer.parseInt(fats);
        int iprot = Integer.parseInt(prot);

        g.setColor(Color.RED);
        g.fillRect(20, 150-icarbo, 40, icarbo);
        g.setColor(Color.BLUE);
        g.fillRect(60, 150-ifats, 40, ifats);
        g.setColor(Color.ORANGE);
        g.fillRect(100, 150-iprot, 40, iprot);

        g.setColor(Color.BLACK);
        g.drawString("carbo", 20, 170);
        g.drawString("fats", 60, 170);
        g.drawString("prot", 100, 170);
    }
}

```

Opgave 4

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

public class Chase extends Applet implements MouseMotionListener
{
    int oldx, oldy; // vorige plaats van muis
    int newx, newy; // huidige plaats van muis
    int vecx, vecy; // coördinaten van bewegingsvector

    public void init()
    {
        addMouseMotionListener(this);
    }

    public void paint(Graphics g)
    {
        g.setColor(Color.RED);
        int cx, cy;
        cx = newx + vecx;
        cy = newy + vecy;
        g.fillOval(cx-10, cy-10, 20, 20);
    }

    public void mouseMoved(MouseEvent e)
    {
        oldx = newx; oldy = newy;
        newx = e.getX(); newy = e.getY();
        vecx = newx - oldx; vecy = newy - oldy; // bewegingsvector
        double length = 1 + Math.sqrt(vecx*vecx + vecy*vecy); // niet 0
        vecx = (int)(20*vecx/length);
        vecy = (int)(20*vecy/length); // genormaliseerd op lengte 20
        repaint();
    }

    public void mouseDragged(MouseEvent e)
    {
        mouseMoved(e);
    }
}
```