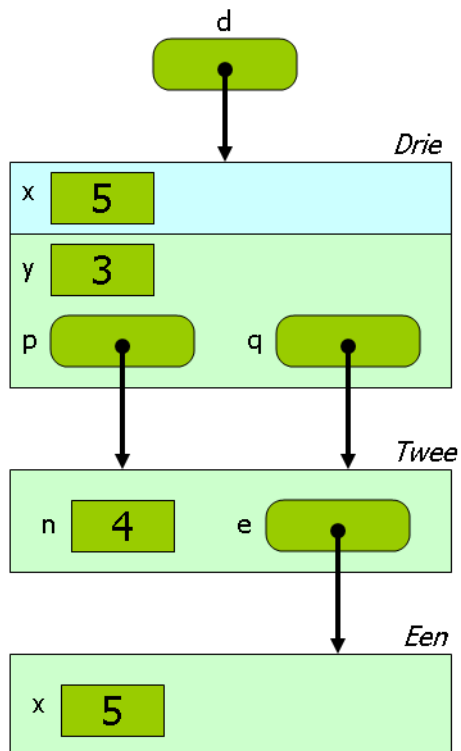


# Uitwerking 2e deeltentamen Imperatief Programmeren

14 oktober 2005

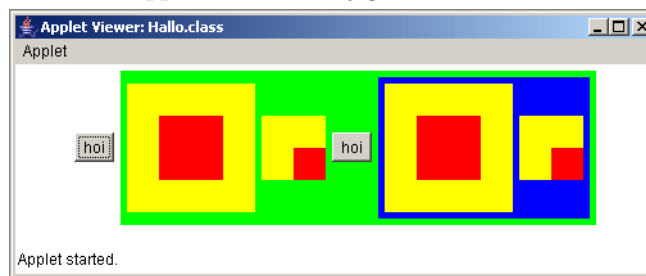
```
1. int compareTo(String s)
{
    int i, m, n;
    char c, d;
    m = this.length();
    n = s.length();
    for (i=0; i<Math.min(m,n); i++)
    {
        c = this.charAt(i);
        d = s.charAt(i);
        if (c!=d) return c-d;
    }
    return m-n;
}
```

2. Dit is de geheugenopbouw na `d=new Demo();`



3. (a)
- In klasse Test wordt this meegegeven als derde parameter aan de constructor van Gebied. die verwacht daar een ActionListener, dus moet Test implements ActionListener. In Test is er dus een methode actionPerformed.
  - In klasse Test wordt Gebied g1 meegegeven aan de constructor van Thread. Die verwacht een Runnable, dus moet Gebied implements Runnable. In Test is er dus een methode run.
  - In klasse Ruimte wordt this meegegeven aan addMouseListener. Die verwacht uiteraard een MouseMotionListener, dus moet Ruimte implements MouseMotionListener. In Ruimte zijn er dus een methoden mouseMoved en mouseDragged.

(b) Zo ziet de applet er uit als hij gerund wordt:



- (c)
- this heeft het type Test
  - Test extends Applet, die op zijn beurt extends Panel extends Container. Een Container wordt als tweede parameter verwacht, dus dat is OK.
  - Test implements ActionListener, en dat wordt als derde parameter verwacht, dus dat is ook OK.
- (d) Bij een animatie zal er in methode run typisch een while-opdracht staan die hem aan de gang houdt. Door de voortgang daarvan van een boolean te laten hangen, kun je de animatie stoppen door die boolean false te maken. Of schrijf 'while (t!=null)', en maak t gelijk aan null.
4. Zie listing.

```

5 public class LichtDonker extends Applet implements MouseListener , ActionListener
{   Scrollbar helderheid;
    Button leeg, lijn;
    int [] x, y, h;
    int n;   boolean met;

10
    public void init()
    {   helderheid = new Scrollbar(Scrollbar.HORIZONTAL, 128, 1, 0, 255);
        leeg = new Button("Leeg");
        lijn = new Button("Lijn");
15
        this.add(leeg);
        this.add(helderheid);
        this.add(lijn);
        leeg.addActionListener(this);
        lijn.addActionListener(this);
20
        this.addMouseListener(this);
        x = new int[100];
        y = new int[100];
        h = new int[100];
        n = 0;
25
        met = false;
    }

    public void paint(Graphics g)
    {   int kl = 0, gr = 0;

30
        for (int i=0; i<n; i++)
        {   g.setColor(new Color(h[i], h[i], h[i]) );
            g.fillRect(x[i]-10, y[i]-10, 20, 20 );
            g.setColor(Color.BLACK);
35
            g.drawRect(x[i]-10, y[i]-10, 20, 20 );

            if (h[i]<h[kl]) kl=i;
            if (h[i]>h[gr]) gr=i;
        }
40
        if (met && n>0)
        {   g.setColor(Color.BLACK);
            g.drawLine(x[kl], y[kl], x[gr], y[gr]);
        }
    }

45
    public void actionPerformed(ActionEvent e)
    {   if (e.getSource()==leeg)
        n = 0;
        else met = !met;
50
        this.repaint();
    }

    public void mousePressed(MouseEvent e)
    {   if (n<100)
55
        {   x[n] = e.getX();
            y[n] = e.getY();
            h[n] = 255-helderheid.getValue();
            n++;
        }
60
        this.repaint();
    }
}

```