

# Uitwerking 2e deeltent. Imperatief Programmeren

14 oktober 2009

## Opgave 1

```
public String replace(char x, char y)
{
    String res;
    int n;
    char c;
    res = "";
    for (n=0; n<this.length(); n++)
    {
        c = this.charAt(n);
        if (c==x)
            c = y;
        res += c;
    }
    return res;
}

public boolean endsWith(String s)
{
    int n, sl, tl;
    tl = this.length();
    sl = s.length();
    if (sl>tl)
        return false;
    for (n=0; n<sl; n++)
    {
        if (s.charAt(n)!=this.charAt(tl-sl+n))
            return false;
    }
    return true;
}
```

Als s en t verschillende String-objecten zijn, die echter dezelfde characters bevatten, dan zal s.equals(t) als resultaat true opleveren, maar s==t heeft de waarde false.

## Opgave 2

- a. x1: impl MouseListener  
x2: prim int  
x3: super TextComponent  
x4: sub Container  
x5: sub Component  
x6: impl Runnable
- b. this bewaart een verwijzing naar x1, zodat als de gebruiker iets met de muis doet, de bijbehorende methode uit MouseListener kan worden aangeroepen met x1 onderhanden
- c. er wordt waarschijnlijk t.start() aangeroepen. Dan wordt van x6 de methode run aangeroepen. Dat gebeurt in een aparte thread, het programma gaat dus ook meteen door met de opdracht na t.start()

### Opgave 3

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

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

    public void paint(Graphics g)
    {
        int kl = 0, gr = 0;
        for (int t=0; t<n; t++)
        {
            g.setColor(new Color(h[t], h[t], h[t]) );
            g.fillRect(x[t]-10, y[t]-10, 20, 20 );
            g.setColor(Color.BLACK);
            g.drawRect(x[t]-10, y[t]-10, 20, 20 );
            if (h[t]<h[kl]) kl=t;
            if (h[t]>h[gr]) gr=t;
        }
        if (met && n>0)
        {
            g.setColor(Color.BLACK);
            g.drawLine(x[kl], y[kl], x[gr], y[gr]);
        }
    }
}
```

```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource()==leeg)
        n = 0;
    else met = !met;
    this.repaint();
}
public void mousePressed(MouseEvent e)
{
    if (n<100)
    {
        x[n] = e.getX();
        y[n] = e.getY();
        h[n] = 255-helderheid.getValue();
        n++;
    }
}
this.repaint();
}
```