

Imperatief Programmeren, toets 2 (INFOIMP) 13 oktober 2004

Opgave 1

(3 punten)

In de klasse `String` zit een methode `equals`, waarmee je kunt vergelijken of twee strings dezelfde symbolen bevatten. De methode `equalsIgnoreCase` doet hetzelfde, maar maakt daarbij geen onderscheid tussen hoofdletters en kleine letters. Dus "aap", "Aap" , "aAP" en zijn volgens deze methode allemaal gelijk. Je mag er in deze opgave van uitgaan dat een string alleen symbolen met code kleiner dan 128 bevat. Je hoeft dus geen rekening te houden met letters met accenten of met exotische alfabetten.

Stel je voor dat de klasse `String` nog niet bestaat, en dat je die zelf aan het schrijven bent. Maak twee versies van de methode `equalsIgnoreCase`, waarbij

- je gebruik mag maken van de andere methoden die er in de klasse `String` zitten, maar niet van `toUpperCase` en `toLowerCase`
- je gebruik mag maken van de andere methoden die er in de klasse `String` zitten, maar niet van `substring` en `charAt`

Opgave 2

(3 punten)

Bekijk de listing hieronder, waarin de klassen `Test` en `Ruimte` worden gedefinieerd. Dit programma wordt door de compiler zonder foutmeldingen geaccepteerd. In de HTML-file staat: `<APPLET code="Test.class" width="500" height="150">`.

- Wat moet er achter implements worden geschreven in de klasse `Test`, en waarom is dat in dit programma nodig? En in de klasse `Ruimte`?
- Welke methode(n) moet(en) er nog worden gedefinieerd in de klasse `Test`? En welke in de klasse `Ruimte`?
- Maak een schets van hoe de applet in werking er uit komt te zien, en geef daarin aan wat er rood is, wat blauw, en wat geel.
- Bekijk de twee aanroepen `new Ruimte(this, Color.red)` en `r1.addMouseMotionListener(this)`. Wat is hier het type van `this`, en waarom is dat in deze twee gevallen een acceptabele parameter?
- Wat gebeurt er als gevolg van de aanroep `t.start()`? (Beschrijf dit in termen van de betrokken Java-objecten en methoden, dus niet alleen maar "de thread wordt gestart". Wat is er speciaal aan deze aanroep?

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

public class Test
    extends Applet
    implements ...
{
```

```

public void init()
{
    Ruimte r1, r2;
    Thread t;
    r1 = new Ruimte(this, Color.red);
    this.add(r1);
    r2 = new Ruimte(this, Color.blue);
    this.add(r2);
    r1.addMouseListener(this);
    t = new Thread(r2);
    t.start();
}

...
...
}
class Ruimte
    extends Canvas
    implements ...
{
    Color f;
    Button b;

    Ruimte(Container c, Color k)
    {
        f = k;
        this.setSize(100,100);
        this.setBackground(Color.yellow);
        c.setBackground(k);
        b = new Button("hoi");
        c.add(b);
        b.addActionListener(this);
    }

    public void paint(Graphics g)
    {
        g.setColor(f);
        g.fillRect(25,25,50,50);
    }

    ...
    ...
}

```

Opgave 3

(4 punten)

Om het schrijfwerk te beperken mag je in deze opgave:

- De HTML-file weglaten: je hoeft alleen de Java-file te schrijven
- De import-regels bovenaan het programma weglaten
- De muis-methoden die een lege body hebben weglaten

Het is niet erg als er in dit programma door afrondfouten afwijkingen van maximaal 1 beeldpunt ontstaan.

Schrijf een applet met de volgende eigenschappen:

1. De afmetingen van het applet zijn 300 bij 300 beeldpunten
2. De gebruiker ziet twee knoppen met het opschrift "Box" en "Clear", en een schuifregelaar.
3. In het midden van het applet is een cirkel zichtbaar met een doorsnede van 30 beeldpunten.
4. Iedere keer als de gebruiker ergens klikt ontstaat er, gecentreerd op dat punt, een extra cirkel.
5. Er zijn maximaal 100 cirkels zichtbaar. Als de gebruiker daarna toch meer punten aanklikt, gebeurt er niets (ook geen foutmelding!).
6. Na het indrukken van de "Box" knop verschijnt er een zo klein mogelijke rechthoek waar alle cirkels nog net binnen vallen.
7. Na het nogmaals indrukken van "Box" verdwijnt de rechthoek weer, enz.
8. Na het indrukken van "Clear" verdwijnen alle door kliks ontstane cirkels; alleen de cirkel in het midden blijft over (met, als dat nodig is, een rechthoek er direct omheen). De gebruiker kan dan weer met 99 nieuwe kliks beginnen.
9. Met de schuifregelaar kan de gebruiker de dikte van de lijn van de rechthoek instellen tussen 1 en 20 beeldpunten. Aan het begin is de lijndikte 1.
10. De rand van de rechthoek mag de cirkels niet overlappen; het kader moet dus naar buiten toe groeien!

