

## Uitwerking<sup>1</sup> tweede deeltentamen Imperatief Programmeren (INFOIMP) 12 oktober 2007

### 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;
}
```

### Opgave 2

- `this` heeft als type de klasse waarin `init` is gedefinieerd, wat een subklasse is van `Applet`. `Applet` is op zijn beurt een subklasse van `Panel`, en `Panel` van `Container`. De methode `add` is gedefinieerd in `Container`, alle subklassen erven hem daarvan.
- De methode `add` verwacht een `Component`; zowel `Button` als `TextArea` zijn subklassen van `Component` en dus ook acceptabel als parameter.
- `class Zelfgemaakt extends Component implements ActionListener`

```
{
    public Zelfgemaakt(int x)
    public void actionPerformed(ActionEvent e)
}
```

---

<sup>1</sup>Deze uitwerkingen zijn met de grootste zorg gemaakt. In geval van fouten kan de  $\mathcal{I}\mathcal{B}\mathcal{C}$  niet verantwoordelijk worden gesteld, maar wordt zij wel graag op de hoogte gesteld: [tbc@a-eskwadraat.nl](mailto:tbc@a-eskwadraat.nl)

### Opgave 3

```
public static char vaakste(String s)
{
    int [] a;
    int res, n;
    a = new int[256];
    for (n=0; n<s.length(); n++)
        a[ s.charAt(n) ]++;
    res = 0;
    for (n=0; n<256; n++)
        if (a[n]>a[res])
            res = n;
    return (char)res;
}
```

### Opgave 4

```
public class Tent200710 extends Applet
implements ActionListener, MouseListener, Runnable
{
    Scrollbar diam;
    Button color, clear;
    Thread t;
    int [] xs, ys, ds;
    int aantal, k;
    public void init()
    {
        xs = new int[100];
        ys = new int[100];
        ds = new int[100];
        aantal = 0;
        color = new Button("color");
        clear = new Button("clear");
        diam = new Scrollbar(Scrollbar.HORIZONTAL
        , 50, 1, 0, 100);
        this.add(color);
        this.add(clear);
        this.add(diam);
        color.addActionListener(this);
        clear.addActionListener(this);
        this.addMouseListener (this);
    }
    public void run()
    {
        while (t!=null)
        {
            k = (k+1) % 256;
            this.repaint();
            try
            { Thread.sleep(20);
            }
            catch (Exception e) {}
        }
    }
    public void paint(Graphics gr)
```

```

{
    Color col1, col2;
    col1 = new Color(k,k,255-k);
    col2 = new Color(255-k,255-k,k);
    for (int i=0; i<aantal; i++)
    {
        gr.setColor(col1);
        gr.fillRect(xs[i]-ds[i]/2, ys[i]-ds[i]/2
            , ds[i], ds[i]);
        gr.setColor(col2);
        gr.fillOval(xs[i]-ds[i]/2, ys[i]-ds[i]/2
            , ds[i], ds[i]);
    }
}
public void actionPerformed(ActionEvent e)
{
    if (e.getSource()==clear)
        aantal = 0;
    if (e.getSource()==color)
    { if (t==null)
        { t = new Thread(this);
          t.start();
        }
      else t = null;
    }
    this.repaint();
}
public void mousePressed(MouseEvent e)
{
    if (aantal<100)
    { xs[aantal] = e.getX();
      ys[aantal] = e.getY();
      ds[aantal] = diam.getValue();
      aantal++;
      repaint();
    }
}
}
}

```