

Uitwerking¹ Imperatief Programmeren, eerste deeltentamen (INFOIMP) 29 september 2006

Opgave 1

- a) Alice gebruikt een compiler die Java-broncode omzet in een tussentaal die bekend staat als byte-code. Bob gebruikt een browser waarin een bytecode-interpretter is ingebouwd.
- b) Een object heeft een klasse als type. Een klasse kan beschouwd worden als een groepje methoden dat bij elkaar hoort. Een object kan onder handen worden genomen door het aanroepen van een methode uit de klasse die het type van dat object is.
- c) Een void method is een methode die geen resultaatwaarde retourneert. De aanroep ervan is dan ook geen expressie (want die heeft een waarde), maar een opdracht.
- d) Een static methode heeft geen object onder handen. Bij de aanroep staat er voor de punt dan ook niet een object, maar de naam van de klasse waarin de static methode zich bevindt.
- e)

```
n = (int) d;  
n = Integer.parseInt(s);  
d = n;  
d = Double.parseDouble(s);  
s = ""+n;  
s = ""+d;
```

in plaats van de laatste twee is ook mogelijk:

```
s = Integer.toString(n);  
s = Double.toString(d);
```

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

Opgave 2

Voor de duidelijk zijn hieronder de programmafragmenten opnieuw opgeschreven. In je antwoord hoefde je alleen het bijbehorende getal aan te geven.

| | | | |
|---|----------|---|----|
| <pre>x=0; while (x<5) { x=x+1; this.iets(); } </pre> | 5 | <pre>for (x=0; x<5; x++) { this.iets(); x=x+1; } </pre> | 3 |
| <pre>x=0; while (x<5) { this.iets(); x+1; } </pre> | ∞ | <pre>for (x=0; x<5; x++) this.iets(); x=x+1; </pre> | 5 |
| <pre>x=0; while (x=5) { this.iets(); x=x+1; } </pre> | F | <pre>for (x=0; x<x;) this.iets(); </pre> | 0 |
| <pre>x=0; while (x<5); { this.iets(); x=x+1; } </pre> | 0 | <pre>for (x=0; x<5) { this.iets(); x=x+1; } </pre> | F |
| <pre>x=0; {while (x<5) x=x+1; this.iets(); } </pre> | 1 | <pre>for (x=0; x<5; x++) for (y=0; y<5; y++) this.iets(); </pre> | 25 |
| <pre>x=0; {while (x<5) {; this.iets(); x=x+1; }; }; </pre> | 5 | <pre>for (x=0; x<5; x++) { for (y=0; y<5; y++) this.iets(); this.iets(); } </pre> | 30 |
| <pre>x=0; while (x<5) this.iets(); x=x+1; </pre> | ∞ | <pre>for (x=0; x<5; x++) this.iets() for (y=0; y<5; y++) this.iets(); </pre> | 10 |
| <pre>x=0; while (x<5); this.iets(); x=x+1; </pre> | 1 | <pre>for (x=0; x<5; x++) for (y=0; y<x; y++) this.iets(); </pre> | 10 |

Hieronder volgt een toelichting (hoefde je ook niet te geven).

While-opdrachten:

1. Het is gebruikelijker om $x=x+1$ aan het eind te schrijven, maar voor het aantal maakt dat niet uit.
2. $x+1$ verhoogt de teller niet: het rekent de nieuwe waarde wel uit, maar slaat deze niet op. Daardoor wordt x nooit 5, en blijft het oneindig doorgaan. (De programmeur had natuurlijk $x=x+1$; of $x+=1$; of $x++$; moeten schrijven).
3. $x=5$ is een toekenning, en niet een boolean vergelijking $x==5$.
4. Er staat een puntkomma achter de header. Deze lege opdracht wordt dus oneindig keer herhaald, en aan 'iets' komt hij helemaal niet toe.
5. Er staat een zinloos paar accolades, maar dit is niet fout. De body van de while bestaat alleen uit $x=x+1$. Na afloop wordt 'iets' 1 keer uitgevoerd.

6. De rare puntkomma's zijn een extra lege opdracht vlak voor elke 'iets'-aanroep, en nog eentje na afloop van de while-opdracht, maar dit is niet fout.
7. Er ontbreken accolades, dus x blijft hangen op 0
8. De voorwaarde is meteen al niet waar, dus de (lege!) body wordt nul keer uitgevoerd. De volgende opdracht 'iets' wordt gewoon eenmaal uitgevoerd.

For-opdrachten:

1. Zowel in de header als in de body wordt x opgehoogd. Het gaat dus met stappen van 2 tegelijk: 0, 2 en 4, dus driemaal.
2. Hoewel de layout anders suggereert, maakt $x=x+1$ nu geen deel uit van de body, dus 'iets' wordt gewoon 5 keer gedaan.
3. De drie onderdelen in de header zijn optioneel, dus dit is niet fout. De voorwaarde is nooit waar, dus de body wordt 0 keer uitgevoerd.
4. Er moeten altijd twee puntkomma's in de header staan, dus dit is fout.
5. De body van de buitenste for-opdracht is zelf een for-opdracht, in totaal dus 5 maal 5 herhalingen.
6. Voor elke x wordt 'iets' $5+1$ maal herhaald, in totaal dus 30 keer.
7. Eerst 5 maal met x als teller, dan 5 maal met y als teller, totaal 10 keer.
8. Voor $x=0$: 0 keer; voor $x=1$: 1 keer, voor $x=2$: 2 keer, voor $x=3$: 3 keer, en voor $x=4$: 4 keer; totaal $0+1+2+3+4=10$ keer.

Opgave 3

Dit is het kortste:

```
private double coshyp(double x)
{
    double res, a, b; int s, t;
    a=1;b=1;res=0;
    for (t=0; t<40; t+=2)
    {
        res += a/b;
        a *= x*x;
        b *= (t+1)*(t+2);
    }
    return res;
}
```

Maar je kunt natuurlijk ook eerst een methode `macht` en een methode `fac` schrijven:

```
private double macht(double x, int n)
{
    double res; int t;
    res = 1 ;
    for (t=0; t<n; t++)
        res *= x;
    return res;
}

private int fac(int n)
{
    int res, t;
    res = 1;
    for (t=2; t<=n; t++)
        res *= t;
    return res;
}
```

en dan:

```
private double coshyp(double x)
{   double res; int t;
    res = 0;
    for (t=0; t<40; t+=2)
        res += this.macht(x,t)/this.fac(t);
    return res;
}
```

Opgave 4

De cursieve gedeeltes mocht je weglaten.

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

public class Squarez extends Applet
    implements MouseListener, MouseMotionListener
{
    int n, mx, my;

    public void init()
    {   this.addMouseListener(this);
        this.addMouseMotionListener(this);
        n = 1;
        mx = 0;
        my = 0;
    }
    public void paint(Graphics g)
    {   int i, k;
        for (i=n; i>=1; i--)
        {   k = 32*(i-1)%256;
            g.setColor(new Color(k, k, k));
            g.fillRect(mx-10*i, my-10*i, 20*i, 20*i);
        }
    }
    public void mouseMoved(MouseEvent e)
    {   mx = e.getX();
        my = e.getY();
        this.repaint();
    }
    public void mouseClicked(MouseEvent e)
    {   n++;
        this.repaint();
    }

    public void mouseDragged (MouseEvent e) {}
    public void mousePressed (MouseEvent e) {}
    public void mouseReleased (MouseEvent e) {}
    public void mouseEntered (MouseEvent e) {}
    public void mouseExited (MouseEvent e) {}
}
```