

# sheets Programmeren 1 — Java

college 2, Walter Kusters

De sheets zijn gebaseerd op de hoofdstukken  
2 tot en met 6 van:

D. Bell en M. Parr, Java voor studenten,  
Prentice Hall, 2002

<http://www.liacs.nl/home/kusters/java/>

## Java — hello world

Edit een file `Hello.java`, met:

```
// een hello world programma
public class Hello {
    public static void main (String[ ] args) {
        System.out.println ("Hallo wereld!");
    } // main
} // Hello
```

Compileren met `javac Hello.java`

Runnen met `java Hello`

Het kan ook met een grafische omgeving.

Het programma is een voorbeeld van een stand-alone tekstgerichte applicatie. Daarnaast bestaan ook nog applets, die in een webpagina (zeg `leuk.html`) draaien, en die je kunt bekijken met behulp van `appletviewer leuk.html`

## Java — **applet**

De file leuk.html bevat:

```
<html>
<head><title>Een WWW-bladzijde</title></head>
<body>
<applet code="Lijn.class" width=100 height=100>
</applet>
</body>
</html>
```

De file Lijn.java bevat:

```
import java.awt.*;
import java.applet.Applet;
public class Lijn extends Applet {
    public void paint (Graphics g) {
        g.drawLine(0,0,50,70); // teken lijn
    } // paint
} // Lijn
```

## Java — variabelen

Informatie zit in variabelen. Elke variabele is van een bepaald type (net zoals wij allen mensen zijn).

We hebben standaardtypes:

- `int` (integer, geheel getal),
- `float` of `double` (“reëel getal”),
- `boolean` (`true` of `false`).

En ook nog: `String` (zoals `"Hallo wereld!"`, zie later) en arrays (je gebruikt bijvoorbeeld `args[0]` en `args[2]` van het array `args`, zie ook later).

## Java — operatoren

Met behulp van operatoren als + en \* kun je op de gebruikelijke manier (waarden van) variabelen met elkaar combineren.

Tip: gebruik overvloedig haakjes!

Enkele voorbeelden, met integers `i` en `j`, dus met `int i;` en `int j;`:

- `i = 3 * ( 17 + 8 );` (`i` wordt 75)
- `i = i + 13;` (`i` wordt met 13 opgehoogd)
- `j = i % 7;` (`j` wordt rest bij deling van `i` door 7)
- `j++;` (hoog `j` met 1 op)
- `boolean oud = ( i > 65 );` (ja of nee)

## Java — operatoren, vervolg

Nog meer voorbeelden, met `int i`; en `double d`;

- `d = 1.2345 / 6.789;` (d wordt 0.18183...)
- `d = 5 / 2;` (d wordt 2!!!)
- `d = 5 / 2.0;` (d wordt 2.5)
- `d = (double)5 / 2;` (casting; d wordt 2.5)

Let op: `g.drawString ("iets" + 1 + 2,50,50);` levert `iets12` op het scherm op (de strings worden “geconcateneerd”), en niet `iets3`; gebruik eventueel haakjes of variabelen om “gewone” optelling af te dwingen.

## Java — **args**

Wat doet die parameter `args` uit het “hello world” programma? Daar kun je “command line parameters” mee opvangen:

```
public class Plus2 {  
    public static void main (String[ ] args) {  
        int i = Integer.parseInt (args[1]);  
        // i is een locale variabele  
        System.out.println (i+2);  
    } // main  
} // Plus2
```

Bij aanroep `java Plus2 7777 100` krijg je 102 (en niet 1002 overigens). Let ook op het omzetten van een string naar een integer.

Zo kun je “invoer” van de gebruiker vragen. Dat kan ook anders: of via “gebufferde input”, of via windows met knoppen en schuiven.

## Java — events, compleet voorbeeld

```
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
public class Iets extends Applet
    implements AdjustmentListener {
    private Scrollbar trekker;
    private int waarde = 0;
    public void init ( ) {
        trekker = new Scrollbar (
            Scrollbar.HORIZONTAL,0,1,0,100);
        add (trekker);
        trekker.addAdjustmentListener (this);
    } // init
    public void paint (Graphics g) {
        g.drawString ("Nu is het " + waarde,100,100);
    } // paint
    public void adjustmentValueChanged (
        AdjustmentEvent e) {
        waarde = trekker.getValue ( );
        repaint ( );
    } // adjustmentValueChanged
} // Iets
```



## Java — events, details

De eerste methode

```
public void init ( ) {
    trekker = new Scrollbar (
        Scrollbar.HORIZONTAL,0,1,0,100);
    add (trekker);
    trekker.addAdjustmentListener (this);
} // init
```

maakt een nieuwe “scrollbar”, voegt die aan het window toe, en vraagt “ons” (this) erop te attenderen als er wat mee gebeurt.

De tweede methode

```
public void adjustmentValueChanged (
    AdjustmentEvent e) {
    waarde = trekker.getValue ( );
    repaint ( );
} // adjustmentValueChanged
```

vangt het desbetreffende event op, en handelt het af.

## Java — labels

En nu met twee scrollbars — met labels:

```
private Scrollbar trekker1, trekker2;
private int waarde1 = 0, waarde2 = 0;
public void init ( ) {
    Label titel1, titel2;
    titel1 = new Label ("Een: ");
    add (titel1);
    trekker1 = new Scrollbar (
        Scrollbar.HORIZONTAL,0,1,0,100);
    add (trekker1);
    trekker1.addAdjustmentListener (this);
    ... en nu voor titel2 en trekker2 ...
} // init
public void adjustmentValueChanged (
    AdjustmentEvent e) {
    waarde1 = trekker1.getValue ( );
    waarde2 = trekker2.getValue ( );
    repaint ( );
} // adjustmentValueChanged
```

## Java — **if**

We voegen aan de `paint`-methode van onze scrollbar een `if`-statement toe:

```
public void paint (Graphics g) {
    if ( waarde >= 21 )
        g.drawString ("Volwassen!",50,50);
    else {
        g.drawString ("Nog jong!",50,50);
        g.drawString ("Wanhoop niet ...",50,70);
    } // else
    g.drawString ("Dat was het",50,90);
} // paint
```

Een `else`-gedeelte is niet verplicht. Gebruik accolades (een `{` en een `}`) om statements bij elkaar te laten horen. Let op de layout.

## Java — **if, details**

Er zijn legio operatoren beschikbaar, zodat je op allerlei manieren condities kunt combineren. Enkele voorbeelden:

- `if ( 21 <= leeftijd && leeftijd < 42 ) ...`  
(met de EN-operator `&&`)
- `if ( lengte == 100 || geslaagd ) ...` (met de OF-operator `||`; hier is `geslaagd` een boolean; let ook op de test op gelijkheid: `==`, een dubbel is-gelijk-teken; ongelijk is `!=`)

Let ook op de “hangende else”:

```
if ( leeftijd > 6 )
    if ( leeftijd < 16)
        g.drawString ("Kinderkorting!",50,50);
    else // hoort bij "laatste" if!!!
        g.drawString ("Volle prijs!",50,50);
```

## Java — **switch**

Stel je hebt een langgerekte “if else if else if”:

```
if ( dag == 1 )
    g.drawString ("zondag",5,5);
else if (dag == 2 )
    g.drawString ("maandag",5,5);
...
else
    g.drawString ("foutje",5,5); break;
```

Soms is dan een switch-statement duidelijker:

```
switch (dag) {
    case 1: g.drawString ("zondag",5,5); break;
    case 2: g.drawString ("maandag",5,5); break;
    ...
    case 7: g.drawString ("zaterdag",5,5); break;
    default: g.drawString ("foutje",5,5); break;
} // case
```

## Java — knoppen

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class Knopteller extends Applet
    implements ActionListener {
    private Button knop;
    private int teller = 0;
    public void init ( ) {
        knop = new Button ("Druk hier");
        add (knop);
        knop.addActionListener (this);
    } // init
    public void paint (Graphics g) {
        g.drawString (teller + " keer",100,100);
    } // paint
    public void actionPerformed (
        ActionEvent event) {
        // soms: if ( event.getSource ( ) == knop )
        teller++;
        repaint ( );
    } // actionPerformed
} // KnopTeller
```

## Java — random getallen

De functieaanroep `Math.random ( )` levert een pseudo-random (schijnbaar willekeurig) getal tussen 0.0 en 1.0 op (kleiner dan 1.0).

Hoe kun je hiermee een zeskantige dobbelsteen nadoen?

```
int worp;  
worp = (int) ( 6 * Math.random ( ) ) + 1;
```

## Java — tekstvelden

Een tekstveld lijkt op een knop:

```
private TextField leeftijdVeld;
private int leeftijd = 0;
public void init ( ) {
    leeftijdVeld = new TextField (10);
    add (leeftijdVeld);
    leeftijdVeld.addActionListener (this);
} // init
public void paint (Graphics g) {
    g.drawString (leeftijd + " jaar",30,70);
} // paint
public void actionPerformed (
    ActionEvent event) {
    leeftijd = Integer.parseInt (
        leeftijdVeld.getText ( ));
    repaint ( );
} // actionPerformed
```