

# Uitwerking eerste deeltentamen Imperatief Programmeren

donderdag 19 februari 2004

---

## Opgave 1

- Declaraties van variabelen worden afgesloten met een puntkomma, die van parameters niet. In declaraties van variabelen kunnen een typenaam verschillende variabelennamen volgen, bij parameters kan dat niet. Bij parameters kunnen we ervan uitgaan dat ze al een waarde hebben, gekregen door de aanroep; variabelen moeten een waarde krijgen door expliciete toekenning.
- De compiler kan door declaraties controleren of methode-aanroepen zinvol zijn en detecteren of er in variabelennamen tikfouten voorkomen.
- Een `int` wordt naar een `double` geconverteerd als argument van `+` als het andere argument al een `double` is. Een `int` wordt naar een `String` geconverteerd als argument van `+` als het andere argument al een `String` is.
- Als `i` type `int` heeft, heeft `" " + i` of `String.valueOf(i)` type `String`.
- Als `d` type `double` heeft, heeft `(int) d` type `int`.
- Als `s` type `String` heeft en een `int`-waarde voorstelt, heeft `Integer.parseInt(s)` type `int`.

## Opgave 2

- ```
public String metStreepjes(int aanElkaar)
{
    int jaar, maand, dag;
    jaar = aanElkaar / 10000;
    maand = aanElkaar % 10000 / 100;
    dag = aanElkaar % 100;
    return dag + "-" + maand + "-" + jaar;
}
```
- ```
public int aanElkaar(String metStreepjes)
{
    StringTokenizer st;
    int dag, maand, jaar;
    st = new StringTokenizer(metStreepjes, "-");
    dag = Integer.parseInt(st.nextToken());
    maand = Integer.parseInt(st.nextToken());
    jaar = Integer.parseInt(st.nextToken());
    return 10000 * jaar + 100 * maand + dag;
}
```

Opmerkingen:

- De twee methoden gebruiken geen objectvariabelen van `this`, ze kunnen dus beter als `static` worden gedefinieerd.
- Voor gebruik van `StringTokenizer` is expliciete import uit `java.util` nodig.

### Opgave 3

- a. Een methode in een klasse kan precies die objecten bewerken waarvan die klasse het type is.
- b. Na afloop heeft `t` de waarde "voorbeeld".
- c. Een `StringTokenizer` dient om een `String` in delen (zoals woorden) te splitsen. Een `StringTokenizer` wordt gemaakt door de expressie `new StringTokenizer(s, t)`, waarin `s` de te splitsen `String` is en `t` de scheidingstekens bevat. De delen van `s` worden nu geleverd door achtereenvolgende aanroepen van de methode `nextString()`.

### Opgave 4

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;

public class Vlaggen extends Applet
{
    private void baan(Graphics g, int y)
    {
        Color kleur;
        int b1, b2, b3;
        b1 = (int)(255 * Math.random());
        b2 = (int)(255 * Math.random());
        b3 = (int)(255 * Math.random());
        kleur = new Color(b1, b2, b3);
        g.setColor(kleur);
        g.fillRect(15,y,500,100);
    }

    public void paint(Graphics g)
    {
        g.setColor(Color.orange.darker());
        g.fillRect(5,5,5,500);
        baan(g, 5);
        baan(g, 105);
        baan(g, 205);
    }
}
```