

Uitwerking 1e deeltent. Imperatief Programmeren

26 september 2008

Opgave 1

- Met `this` wordt het object aangeduid dat de methode onder handen heeft (het object dat bij aanroep van de methode voor de punt is vermeld). Het `this` object is niet beschikbaar in static methodes, want die hebben geen object onder handen.
- Primitieve types zijn bijvoorbeeld `int`, `double` en `boolean` (en `char`, `byte`, `short`, `long` en `float`). Object-types zijn bijvoorbeeld `Graphics`, `Color`, `String`, `Button`, en alle andere klassen. Waarden van primitieve types worden direct in de variabele opgeslagen, bij object-types is het een *verwijzing* naar het eigenlijke object.
- Een variabele die is gedeclareerd in de header van een methode krijgt zijn waarde bij aanroep van de methode. Een variabele die is gedeclareerd in de body van een methode is alleen voor intern gebruik binnen die methode. Deze is niet toegankelijk vanuit andere methoden. Een variabele die bovenin de klasse is gedeclareerd is vanuit elke (niet-statische) methode toegankelijk.
- Een imperatieve programmeertaal kent een vorm van *opdrachten* die het *geheugen veranderen* en die *één voor één* worden uitgevoerd.
- De programmeur gebruikt een *compiler* om de broncode te vertalen in bytecode; de gebruiker gebruikt een *interpreter* (vaak ingebouwd in een webbrowser) om de bytecode uit te voeren.

Opgave 2

Decl	Type	Expr
Expr	Opdr	Decl
Opdr	Type	Expr
Type	Decl	Opdr
Expr	Expr	Opdr

Opgave 3

```
public int laatste(int n)
{
    return n%10;
}
```

```
public boolean geldig(int x)
{
    int totaal, plaats;
    totaal = 0; plaats=1;
    while (x>0)
    {
        totaal += plaats * laatste(x);
        plaats++;
        x = x/10;
    }
    return totaal%11 == 0;
}
```

Opgave 4

```
public class Tent200809
    extends Applet implements AdjustmentListener
{
    Scrollbar sb;

    public void init()
    {
        sb = new Scrollbar(Scrollbar.HORIZONTAL, 5, 1, 2, 11);
        this.add(sb);
        sb.addAdjustmentListener(this);
    }

    public void adjustmentValueChanged(AdjustmentEvent e)
    {
        this.repaint();
    }

    public void paint(Graphics g)
    {
        int x, y, n, t, k;
        Scanner sc;

        sc = new Scanner( this.getParameter("midden") );
        x = Integer.parseInt(sc.next());
        y = Integer.parseInt(sc.next());
        n = sb.getValue();
        for (t=n; t>0; t--)
        {
            k = 255 - 255*(t-1)/(n-1);
            g.setColor(new Color(k,k,k));
            g.fillRect(x-10*t, y-10*t, 20*t, 20*t);
        }
    }
}
```