



## Stichting NIOC en de NIOC kennisbank

Stichting NIOC ([www.nioc.nl](http://www.nioc.nl)) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website [www.nioc.nl](http://www.nioc.nl) ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op [www.nioc2025.nl](http://www.nioc2025.nl) voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

[www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief](http://www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief)

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga [kennisbank@nioc.nl](mailto:kennisbank@nioc.nl).

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Kom  
verder



# Geflipt Leren Programmeren op Saxion



Ruud Greven  
[r.greven@saxion.nl](mailto:r.greven@saxion.nl)

Wilco Bonestroo  
[w.j.bonestroo@saxion.nl](mailto:w.j.bonestroo@saxion.nl)

Kom  
verder



# Inhoud

## Aanleiding

- Hoe uitgevoerd?
- Tips

Kom verder



The screenshot shows the Eclipse IDE interface. The main editor displays the following Java code:

```
1 import java.io.File;
2
3
4
5 public class ScannerVoorbeeld {
6
7     public static void main(String[] args) throws Exception {
8         try {
9             a();
10        } finally {
11            System.out.println("finally in main");
12        }
13    }
14
15    public static void a() throws Exception {
16        try {
17            System.out.println("Voor aanroep b");
18            b();
19            System.out.println("Na aanroep b");
20        } finally {
21            System.out.println("finally in a");
22        }
23    }
24 }
```

The bottom console window shows the output of the program:

```
<terminated> ScannerVoorbeeld [Java Application] C:\Program Files\Java\jdk1.8.0_20\bin\javaw.exe (12 jan. 2015 15:12:23)
Voor aanroep b
voor exception in b
Exception in thread "main" finally in a
```



- Aanleiding
- ➔ Hoe uitgevoerd?
- Tips

Kom  
verder

SAXION

Voor april 2013



Kom  
verder



april 2013

## 4 & 5 april 2013: NIOC 2013

### 5 april 2013

PS2. Nioc was wel leuk. Het lijkt me gaaf om volgende week Programmeren in een **Flip-the-classroom** principe te gieten. heb een praatje gehoord van iemand van de HAN die dat ook deed.

Hoi Ruud,

Ik helemaal goed. Kan jij morgen bij je laptop dan?

Het **flip-the-classroom** verhaal had ik ook van Tristan gehoord. Klinkt wel als een geschikte manier voor introductie programmeren. Ik voel een pilot aankomen ;) Ik denk dat het goed is om het bij de "studiedag", of wat daar van overblijft, te bespreken.

Goed weekend,

**Dr. ir. W.J. (Wilco) Bonestroo**

Lecturer and Researcher Information Technology

**12 april 2013: Plangesprek**

**15 april 2013: Studiedag**

**22 april 2013: Voorstel naar LPC**



# 7 mei 2013

Dag Willem, Paul,

We hebben inmiddels met z'n drieën overlegt en we zijn eruit. IT GIET OAN! WE GAAN HET DOEN!!!! ; -)

We hebben qua roostering en aantal uren wel een aantal wensen:

- Project Contact van Tristan naar VAC, Tristan krijgt dan een klasje IP extra. Daardoor heb je dat niet de vacature gelijk in ons **flipped** gebeuren springt en dat Tristan een aandachtsgebied minder heeft. het komt er dan op neer dat RGR 2 klassen heeft, TPO 3 en WBO ook 3.
- 1 blok van 4 uur "practicum" per docent (bij voorkeur een vast moment, als in elke week op hetzelfde moment, en bij voorkeur op di of wo). Dus niet de 2x2 die het nu is.
- Elke week 3 uur "extra practicum" voor klas ABCD (in 4 linuxlokalen) en 3 uur "extra practicum" voor klas EFGH (in 4 linuxlokalen). We verwachten hier gewoon alle studenten, alleen de begeleiding is minder intensief. De docenten roeleren hierover, dus bijvoorbeeld in week 1 doen TPO en RGR, week 2 RGR en WBO, week 3 WBO en TPO. We gaan er hierbij vanuit dat we 4 linuxlokalen hebben

Verder zouden we graag voor die 3 lesuur practica studentassistenten willen inzetten. Omdat er per 4 klassen maar 1 docent beschikbaar is. wil je het liefst

Kom  
verder



# En verder...

**20 mei 2013: Lesopzet bedacht**

**21 mei 2013: Boeken besteld**

**26 juni 2013: Software licenties voor video editing geregeld**

**September 2013: Eerste  
uitvoering**

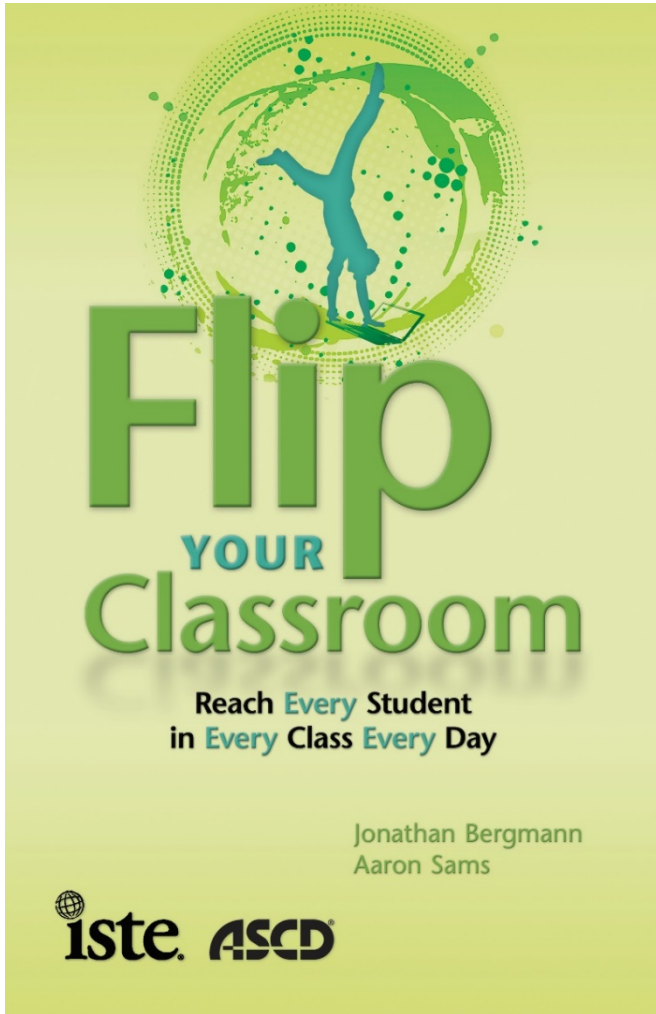
**September 2014: Tweede  
uitvoering**



Kom  
verder

SAXION

# Ideeën opdoen

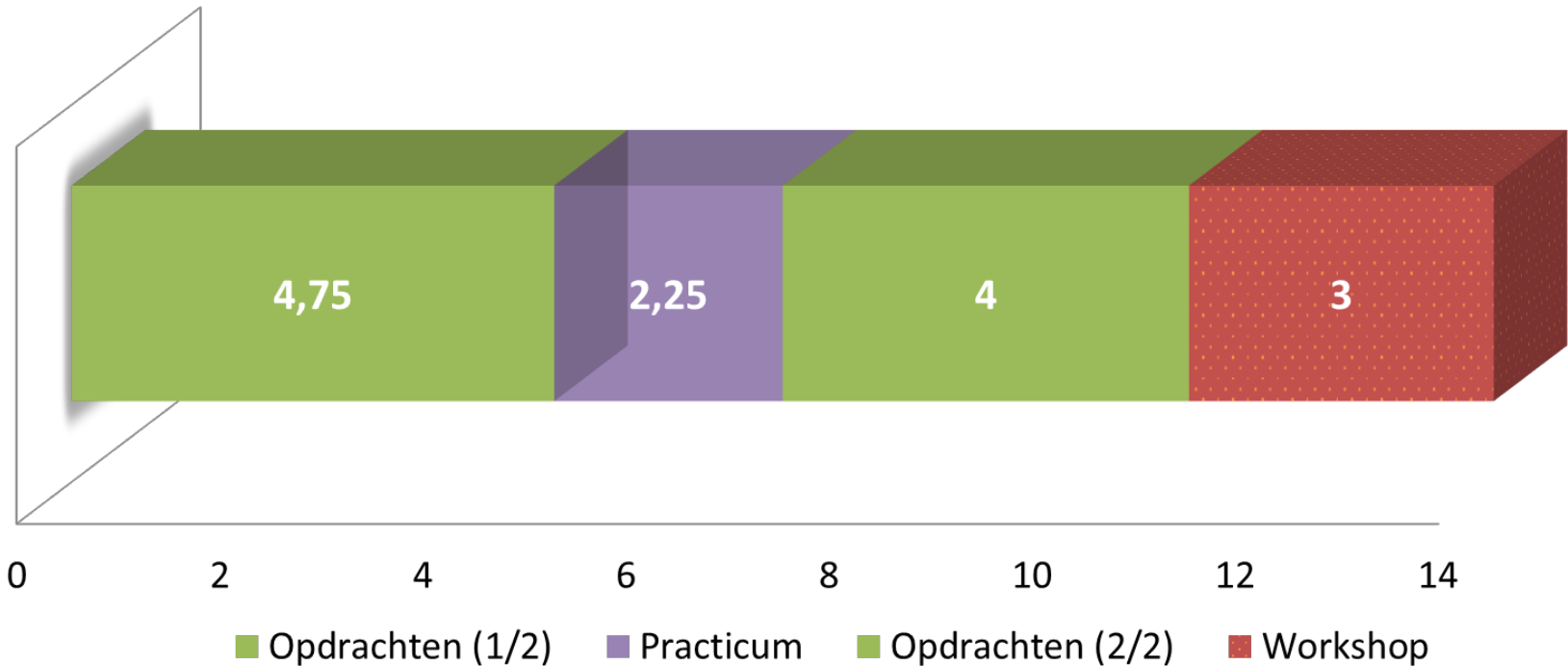


- Inlezen
- Zelf methode bedacht
  - Workshops
  - Practica

Kom verder



# Tijdsplanning per Week



## 3 Constructors, Problem Solving & Static

[Build Content](#)[Assessments](#)[Tools](#)[Partner Content](#)[Studentinstructie Constructors, Problem Solving & Static](#)

### Het gebruik van constructors in Java

Het gebruik van constructors in Java

```
public class ConstructorTest {  
    public static void main(String[] args) {  
        CashRegister register = new CashRegister();  
        //System.out.println(register.getItonCount());  
        //System.out.println(register.getItonCount());  
        BankAccount account = new BankAccount();  
    }  
}
```



### Static vs non-static variabelen en methoden

Static vs non-static variabelen en methoden

```
public class BankAccount {  
    public static void main(String[] args) {  
        BankAccount account = new BankAccount();  
    }  
}
```

1 1.2: Programmeren-1  
(2014-2015)

H

[Mededelingen](#)[Informatie](#)[Voorbeeldtoetsen](#)[1 Array Lists](#)[2 Classes & Objecten](#)[3 Constructors, Problem Solving & Static](#)[4 Meerdere Classes & Debuggen](#)[5 Overerving & Interfaces](#)[6 I/O & Exceptions](#)[7 OO Ontwerp](#)[Android](#)[Bijspijkeropdrachten](#)[Inleveropdrachten](#)[Feedback](#)[Copyrighted Content](#)

COURSE MANAGEMENT

[Control Panel](#)[Files](#)

# Introductie Programmeren

## Week 5

---

### Leerdoelen

- (Geneste if-statements gebruiken)
- Het toepassen van herhalingen binnen je code d.m.v. *for*, *while* en *do-while* lussen
- Het kunnen beschrijven welke variabelen op welk niveau van toepassing zijn (variable scoping)
- Het schrijven van een eigen methode
- Het gebruik van argumenten bij het schrijven van methoden
- Het gebruik van returntypen bij eigen methoden.



### Instructiemateriaal

#### Boek Big Java

- Hoofdstuk 4 (zonder 4.9) en 5 t/m 5.8 (t/m pagina 228)

#### Videos en weblinks (zie blackboard)

- De while-lus
- De for-lus
- Uitwerking voorbeeld: het decimaal naar binair omzetten
- Een eigen methode schrijven (met return type en argumenten)
- Uitwerking voorbeeld: het decimaal naar binair omzetten aankleden met methoden
- Hoger / Lager op de commandline.

### Opgaven

Werk voor de volgende workshop de volgende opdrachten van het boek uit in Java

- o P4.2 (blz. 188)
- o P4.8, P4.9 en P4.10 (blz. 189)
- o P4.12 (blz. 190)
- o P5.1 en P5.2 (blz. 238)
- o P5.5 en P 5.7 (blz. 239)
- o Maak ook je eigen *Hoger-lager* spel voor de commandline in Java.  
Bekijk voor de werking de video op Blackboard.
- o Als je tijd over hebt: P4.24 (blz 192)

Kom verder



# Camtasia

Camtasia Studio - Debugging.camproj

File Edit View Play Tools Help

Record the screen Import media Produce and share

Debugger Bug = fout in je software

Debugger Bug = fout in je software

Java - Test/src/Point.java - Eclipse

```
1 // Represents a two-dimensional point.
2
3
4 public class Point {
5     private int x, y;
6     private String name;
7
8
9     // Gets the x-value of this point.
10    // Returns the x-value.
11    public int getX() {
12        return x;
13    }
14
15
16    // Sets the x-value of this point.
17    // Replaces x with new x-value.
18    public void setX(int x) {
19        this.x = x;
20    }
21
22
23    // Gets the y-value of this point.
24    // Returns the y-value.
25    public int getY() {
26        return y;
27    }
28
29
30    // Sets the y-value of this point.
31    // Replaces y with the p-value.
32    public void setY(int y) {
33        y = this.y;
34    }
35 }
```

1:1 Maintain aspect ratio

Scale: 94 %

SmartFocus

Apply SmartFocus to clips added

Bin Library Callouts Zoom-in Audio Transitions More

0:00:32:22 / 0:09:52:08

0:00:00:00 0:00:10:00 0:00:20:00 0:00:32:22 0:00:40:00 0:00:50:00 0:01:00:00 0:01:10:00 0:01:20:00 0:01:30:00 0:01:40:00

Track 3 Animated Title 2

Track 2 Debuggen.trec (Screen)

Track 1 Debuggen.trec (System Audio)

Fill Erase Copy

Kom  
verder



# Inhoud

- Aanleiding
- Hoe uitgevoerd?



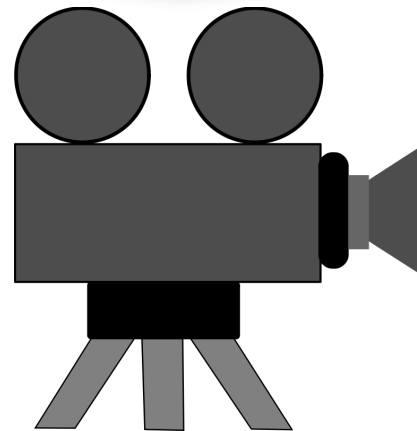
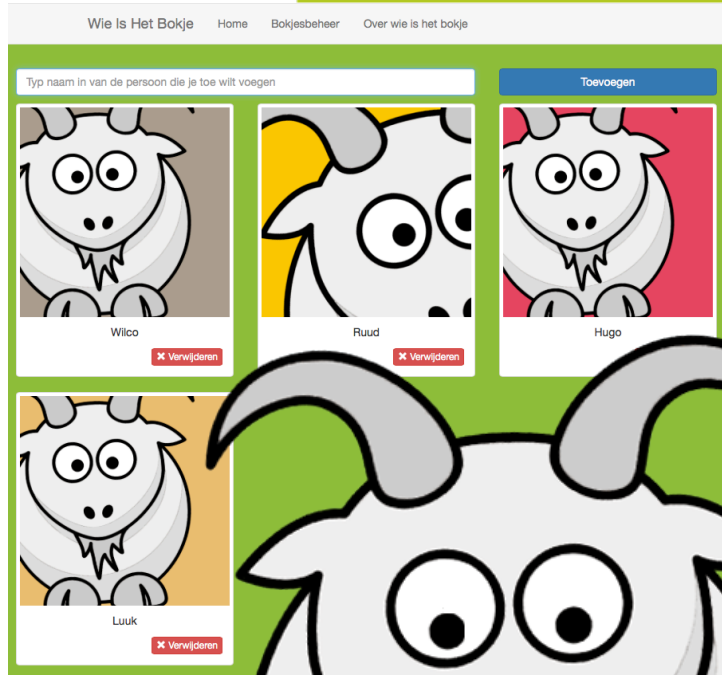
Tips



Kom verder



# Tips



- Bedenk of je het zelf wilt
  - Niet of het kan qua tijd
- Vind draagvlak bij collega's
  - Begin klein (uitvoerenden)
- Doe alsof het gewoon is
- Zorg voor goede faciliteiten
  - Microfoon
  - Geschikte ruimte
  - Goede editing software
  - Richtlijnen voor filmpjes