|  |
| --- |
| Huiswerkopdrachten ISIOO |
| http://4.bp.blogspot.com/_N3QF9LXli28/S8IqYFUvZNI/AAAAAAAABE8/d0sYWertYAc/s1600/Fokke+en+Sukke+nieuwe+computer.gifInleiding Objectoriëntatie |
|  |

|  |  |  |
| --- | --- | --- |
| Bram Janssens | |  |
| School of Information Sciences | |
|  | Januari 2011 |

Inhoudsopgave

[Inleiding 3](#_Toc282776013)

[Week 1 4](#_Toc282776014)

[a. Klassen en objecten 4](#_Toc282776015)

[b. Associaties 5](#_Toc282776016)

[Week 2 6](#_Toc282776017)

[a. Werken met collections 6](#_Toc282776018)

[b. Programmeren in Java 7](#_Toc282776019)

[Week 3 9](#_Toc282776020)

[a. Arrays, Overerving(1) 9](#_Toc282776021)

[b. Toetspracticum 1 10](#_Toc282776022)

[Week 4 11](#_Toc282776023)

[a. Overerving(2) 11](#_Toc282776024)

[b. GUI (1) 13](#_Toc282776025)

[Week 5 14](#_Toc282776026)

[a. GUI (2) 14](#_Toc282776027)

[b. Toetspracticum 2 14](#_Toc282776028)

[Week 6 15](#_Toc282776029)

[a. Graphics 15](#_Toc282776030)

[b. Exceptions 18](#_Toc282776031)

[Week 7 19](#_Toc282776032)

[a. Bestanden 19](#_Toc282776033)

[b. Toetspracticum 3 19](#_Toc282776034)

## Inleiding

Deze reader is je wegwijzer door het zelfstudiegedeelte van het vak ISIOO. Per week vind je de huiswerkopdrachten. De opdrachten dienen iedere week te worden afgetekend binnen de werkplaats*.*

## Week 1

### Klassen en objecten

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H 1
* H 2

#### Opgave 1

In het college heb je de code voor de klasse Laptop gemaakt. Objecten hiervan kun je met BlueJ aanmaken. We gaan in deze opgave Laptop-objecten aanmaken m.b.v. Java-code.

**Uitleg:**

We maken een Laptop-object aan m.b.v. een stukje zelfgeschreven code in een nieuwe klasse Laptopfabriek:



In de klasse Laptopfabriek zit één methode: printLaptop().In de code van deze methode maak je één Laptop-object aan en druk je het merk en de prijs af m.b.v. System.out.println(………) (let op: voor het opvragen van de prijs van een laptop moet je nog een "getter" maken in de klasse Laptop!)

*Opdracht*

1. maak de code van de klasse Laptopfabriek
2. maak in BlueJ een object aan van de klasse Laptopfabriek en roep de methode printLaptop() dan daarop aan.

#### Opgave 2

Maak de volgende opgaven uit het boek:

* §1.10: 1 + 2
* §2.11: 1 t/m 3

### Associaties

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H 3

#### Opgave 1

Maak de volgende opgaven uit het boek:

* §3.11: 1 t/m 4

#### Opgave 2

* Maak de casus "Student" (zie blackboard)
* Maak in BlueJ de code van de ontworpen klassen
* Maak objecten aan van de klassen en test daarmee je code!

## Week 2

### Werken met collections

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H 4

#### Opgave 1

Maak de volgende opgaven uit het boek:

* §4.11: 5

#### Opgave 2

Programmeer het volledige webwinkelsysteem zoals in het college behandeld is. Zie het klassendiagram hieronder:

Maak de code voor elke klasse. Test het systeem in BlueJ, door:

* Enkele Artikel-objecten aan te maken
* Catalogus:
  + enkele van deze aangemaakt artikelen op te nemen in een catalogus en de catalogus te printen
  + laat zien dat je zoekfunctie werkt
* Bestelling
  + enkele van deze aangemaakt artikelen op te nemen in een nieuwe bestelling
  + de factuur van de bestelling te printen met daarop het totaalbedrag

### Programmeren in Java

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H5
* H6: §1 t/m 8

#### Opgave 1

Maak de volgende opgaven uit het boek:

* §5.15: 7

#### Opgave 2

**NetBeans starten**

Zorg dat je NetBeans hebt geïnstalleerd, en start vervolgens het programma op.

**Een nieuw project aanmaken**

Op <http://www.netbeans.org/kb/60/java/quickstart.html> staat beschreven hoe je een nieuw Hello World programma in Java aanmaakt. Volg deze aanwijzingen en geef je project en zelfbedachte naam.

In plaats van in de main-methode, zetten we de System.out.println(..) code in de constructor van de aangemaakte klasse Main.

Zet vervolgens de volgende code in de main-methode:

public static void main(String[] args) {

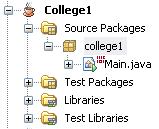
Main m = new Main();

}

Dit zorgt ervoor dat een Main-object wordt aangemaakt, waardoor de constructor van Main wordt aangeroepen en de tekst wordt geprint. Test of dit inderdaad is gelukt.

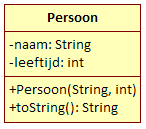
**De klasse Persoon**

In het menu van je nieuwe project staat onder andere ‘college1’ (als dat je gekozen naam was). Dit is een **package**, wat een verzameling is van een aantal Java-klassen. Packages zijn vergelijkbaar directory's (mappen) op je harde schijf. We gaan hierin een nieuwe klasse maken om personen in op te kunnen slaan.



Klik met de rechtermuisknop op college1 en ga naar New ►Java Class... Vul als naam voor de klasse *Persoon* in en klik op *Finish*. Er is nu een bestand aangemaakt waar alvast de basis van de klasse in is gezet.

Programmeer vervolgens de rest van de klasse volgens onderstaand klassendiagram:



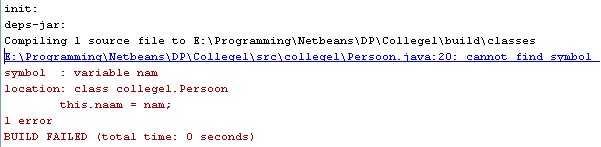
Ga vervolgens terug naar Main.java en zet in de constructor Main() code om 4 Persoon-objecten te maken en te printen.

🛈 **Fouten opsporen:**

Fouten in de code worden door NetBeans aangegeven, zoals:



Als je vervolgens probeert om de code te compileren en runnen, krijg je de volgende melding:



Als je op de blauwe link klikt, wordt er direct naar de regel gesprongen waar Java een fout ziet. Met ‘cannot find symbol: variable nam’ wordt bedoeld dat Java geen variabele kent met de naam ‘nam’. Dat klopt, dit is een typfoutje waar eigenlijk ‘naam’ moest staan!

## Week 3

### Arrays, Overerving(1)

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H6: §12
* H7

#### Opgave 1

Schrijf een programma dat 20000 gehele willekeurige getallen in een array zet. De getallen bevinden zich in het bereik van 0 tot en met 100.

Bereken en toon:

* het grootste getal
* het kleinste getal
* de som van de getallen
* het gemiddelde van de getallen

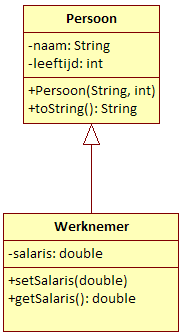
#### Opgave 2

Maak de oefeningen uit de gele sheets uit het college en laat dit aftekenen.

#### Opgave 3

**De klasse Werknemer**

Een werknemer is natuurlijk ook een persoon, maar heeft nog meer specifieke eigenschappen. Maak een nieuwe klasse in NetBeans. Maak de klasse af volgens onderstaand klassendiagram:



Bedenk hoe de constructor er uit komt te zien!

Maak nu ook een aantal werknemer-objecten aan en print ze. Wat valt hierbij op? Los dit probleem op!!

*Nieuwe methodes*

Breid nu de klasse Persoon uit met de volgende methode:

public void jarig() {

..

}

Deze methode zorgt ervoor dat de leeftijd van een persoon met 1 jaar omhoog gaat. Test vervolgens de methode op personen én op werknemers. Merk op dat de klasse Werknemer deze methode automatisch ‘erft’ van Persoon!

Voeg vervolgens aan de klasse Werknemer de volgende methode toe:

public void verhoogSalaris(double bedrag) {

..

}

In deze methode wordt het salaris verhoogd met het meegegeven bedrag. Test deze methode op werknemers én op personen. Wat valt er op? Hoe kan je dit verklaren?

### Toetspracticum 1

Bereid je voor op het toetspracticum door alle huiswerkopdrachten, collegesheets en het boek van betreffende weken (zie weekplanning) door te nemen en te begrijpen! Succes.

## Week 4

### Overerving(2)

#### Opgave 0 (voorbereiding)

Lees nogmaals uit het boek door:

* H7

#### Opgave 1

**Casus TV-gids**

In de tv-gids worden een aantal tv-programma’s opgenomen. Tv-programma’s hebben een titel, een tijdsduur in minuten en worden op een bepaalde zender uitgezonden. Talkshows hebben daarnaast een presentator en een onderwerp. Tv-series worden geregisseerd door een regisseur en worden geïdentificeerd door een afleveringsnummer. De tv-gids kan worden geprint, waarbij een net overzicht van alle programma’s wordt gegeven.

1. Maak het klassendiagram voor deze casus. Voeg toString() methodes toe en maak juist gebruik van overerving!
2. Programmeer de klassen in Netbeans. Test de klassen uitgebreid door er (in code) objecten van aan te maken en de methodes aan te roepen

#### Opgave 2

Op blackboard staat (als bijlage) een powerpoint over Interfaces en Abstracte klassen. Bestudeer deze en maak hiervan de gele sheets.

#### Opgave 3

Maak de superklasse 'TVProgramma" uit opgave 1 nu abstract (want: een TVProgramma op zichzelf is niets, dus daar mag je geen objecten van kunnen aanmaken, maar alleen van de klassen Talkshow en Serie).

Voer de nodige aanpassingen door en zorg ervoor dat de code blijft werken.

#### Opgave 4

1. Maak in NetBeans een GUI (scherm) waarmee je objecten van onderstaande klasse **Persoon** kunt aanmaken. Let op: Persoon heeft géén getters/setters voor zijn *private* attributen. Maak en gebruik ook de **interface** **Benoembaar**.
2. Onthoud de **Persoon**-objecten in een **ArrayList** in de klasse **Scherm**. De klasse scherm moet je uitbreiden naar een **JFrame**.
3. Toon deze lijst van Persoon-objecten in een **TextArea**, waarbij je van elke persoon in de ArrayList de methode geefVolledigeNaam moet aanroepen om



### GUI (1)

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H8

#### Opgave 1 – Cd-Winkel

**1 Voorbereiding**

Maak een nieuw project in NetBeans. Voeg hier de gegeven klassen Cd en Winkel aan toe (zie blackboard).

**2 User interface**

Maak een klasse GUI:

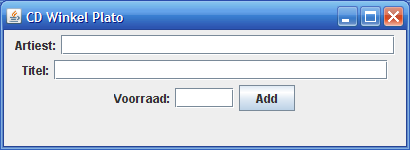
public class GUI extends JFrame

{

…

}

Hierin wordt de user interface gemaakt, volgens onderstaand voorbeeld:



Maak gebruik van een 'flowlayout', dit doe je door middel van onderstaande code:

setLayout(new FlowLayout());

Voeg een klassevariabele toe aan de GUI klasse:

private Winkel wkl = new Winkel();

**3 Events afhandelen**

Breid de klasse zo uit dat je kan reageren op het klikken op de knoppen. Zorg dat er daadwerkelijk cd’s aan de winkel kunnen worden toegevoegd, bijvoorbeeld m.b.v. wkl.addCd(…);

## Week 5

### GUI (2)

#### Opgave 1

**Uitbreiden applicatie CD-Winkel**

Maak een nieuwe knop Clear, waarmee alle velden in de GUI worden gewist.

#### Opgave 2

Maak een menu ‘Opties’ met het item ‘Overzicht’. Maak code voor menu-item ‘Overzicht’, waarmee een modal dialoog wordt getoond. In deze dialoog wordt een overzicht getoond van alle ingevoerde cd's. Maak hiervoor een JTextArea. Zoek in de Java API-documentatie hoe je zo'n tekstgebied aanmaakt.

#### Opgave 3

Bedenk een manier om bij het invoeren van een nieuwe cd ook één of meerdere tracks aan de cd te kunnen toevoegen m.b.v een dialoog en programmeer dit!

*HINTS:*

* Een Track bestaat uit een titel en een tijdsduur.
* Een Cd krijgt een Arraylist van tracks en methode(s) om track(s) toe te voegen aan de cd

### Toetspracticum 2

Bereid je voor op het toetspracticum door alle huiswerkopdrachten, collegesheets en het boek van betreffende weken (zie weekplanning) door te nemen en te begrijpen! Succes.

## Week 6

### Graphics

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H9

#### Opgave 1

**De lift**

Download van Blackboard de uitgangssituatie.

*Stap 1: de Lift-klasse*

Maak een klasse Lift. Objecten van Lift zijn eigenlijk gewoon maar tellertjes: de lift heeft als eigenschappen het aantal etages, en de etage waar de lift zich bevindt. Maak hiervoor attributen in de klasse aan. Daarnaast weet een lift hoe hij er grafisch uitziet: hij kan zichzelf tekenen.

We maken de lift van het T-gebouw na, dus het aantal etages krijgt de waarde 5. Let op dat we ons op 6 verschillende verdiepingen kunnen bevinden, namelijk 0 (de begane grond) t/m 5.

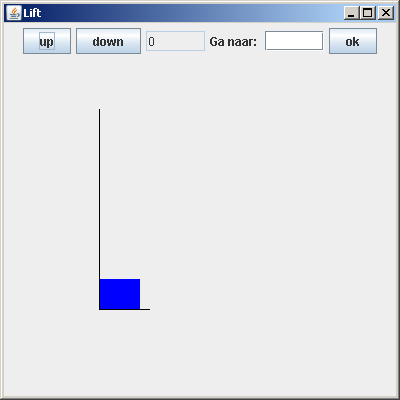
Maak de volgende methodes aan in de klasse:

* public void up() Hiermee verplaatst de lift zich één etage omhoog.
* Een methode public void down() om één etage omlaag te gaan.
* Een methode die teruggeeft op welke etage de lift zich bevindt.

#### Opgave 2

*Stap 2: de GUI-klassen*

De user interface ziet er als volgt uit:



Een groot deel van de GUI is al gegevens, maar de grafische lift moet je nog maken. Maak daarvoor een klasse LiftPanel, subklasse van JPanel. Deze klasse is het gebied waarop de lift daadwerkelijk wordt getekend. Maak daarom in de klasse LiftPanel een klassevariabele van het type Lift aan. Laat in de constructor van deze klasse een lift-object meegeven, als volgt:

public LiftPanel(Lift lift) {

this.lift = lift;

}

De klasse LiftFrame roept deze constructor aan. Deze klasse maakt dus ook een lift-object aan, zodat hij deze kan meegeven bij het aanroepen van bovenstaande constructor.

Voeg de methode public void paintComponent(Graphics g) {} toe. Schrijf in deze methode de code om de lift en de liftschacht te tekenen.

#### Opgave 3

*Stap 3: Events afhandelen*

In de methode actionPerformed(ActionEvent e) van de klasse LiftFrame moet, afhankelijk van de knop waarop is geklikt, de lift omhoog en omlaag bewegen. Vervolgens moet de lift worden getekend, door de methode repaint() van de klasse LiftPanel aan te roepen.

In het tekstveld moet de huidige etage worden weergegeven. In het begin staat de lift op de 0e etage.

#### Opgave 4

*Stap 4: Uitbreiding*

Breid de klasse Lift uit met de volgende methode:

* Een methode naarEtage(int etage) die de lift direct naar een bepaalde etage laat gaan. Controleer ook hier weer op een geldige etage!

Zorg ook dat de betreffende knop in het LiftFrame werkt.

### Exceptions

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H10

#### Opgave 1

We gaan verder met de Lift.

*Stap 5: afhandeling exceptions*

Probeer uit wat er gebeurt als je in het tekstveld ‘Ga naar’ de tekst ‘hoi’ invult. We willen deze exception gaan afvangen en netjes afhandelen. Zoek in paragraaf 10.3 van het boek op welke exception hier optreedt. Zorg dat met een try/catch blok het volgende gebeurt bij foute invoer:

1) laat een melding zien:



2) verplaats de lift naar etage 0.

#### Opgave 2

*Stap 6: zelf een exception maken.*

Zorg ervoor dat, als de gebruiker de lift hoger wil laten gaan dan de hoogste etage of de lager dan de laagste etage, er een (zelf te maken) OngeldigeVerdiepingException wordt opgegooid. Gooi de exception op in de methodes up() en down() van Lift. Vang deze exception op de juiste plek af en geef daar een passende foutmelding.

## Week 7

### Bestanden

#### Opgave 0 (voorbereiding)

Lees de volgende hoofdstukken uit het boek:

* H12: §1, 3 en 5

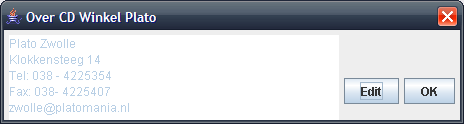
#### Opgave 1

Download van Blackboard de uitgangssituatie.

Breid de menubalk uit met het menu-item *Info* 🡪 *Over.*

Als er op dit item wordt geklikt, wordt een dialoog getoond met:

* Een tekstveld met informatie (de inhoud van bestand: “over.txt”). Deze tekst kan nog niet worden gewijzigd.
* Een knop ‘Edit’ die het tekstveld beschikbaar maakt, waardoor de tekst kan worden gewijzigd.
* Een knop ‘Ok’ waarmee de (gewijzigde) tekst kan worden opgeslagen en de dialoog wordt gesloten.



#### Opgave 2

Implementeer de menu-items *Open* en *Sluit.*

Het item ‘open’ leest cd-objecten uit het bestand ‘cd.dat’. Dit is de verantwoordelijkheid van de klasse Winkel, waar al een lege methode read(..) in te vinden is.

Het item ‘sluit’ schrijft cd-objecten naar het bestand ‘cd.dat’. Ook dit is de verantwoordelijkheid van klasse Winkel, implementeer de methode write(..).

#### Opgave 3

Wijzig nu de uitwerking van opdracht 2 zo, dat de gebruiker zelf het bestand kan kiezen waaruit gelezen en naar geschreven wordt. Doe dit door in de klasse GUI een JFileChooser dialoog te gebruiken.

### Toetspracticum 3

Bereid je voor op het toetspracticum door alle huiswerkopdrachten, collegesheets en het boek van betreffende weken (zie weekplanning) door te nemen en te begrijpen! Succes.