

Java Fundamentals - Labo 4

Richtlijnen

- **Vanaf dit labo werken we in de Netbeans IDE.** Het idee is dat je gebruik leert maken van de mogelijkheden van de omgeving : javadoc raadplegen + debugging!
- Volg systematisch de stijlregels : gebruik grote of kleine letters wanneer gepast, indentatie, blanco lijnen, zinvolle naamgeving, ...
- Voorzie je code van *zinvolle* commentaar.
- Schrijf java gebruikersdocumentatie bovenaan elke klasse: een korte beschrijving van de opgave, inclusief vermelding van de auteur (@author) en de versie (@version).

Netbeans IDE

1. **First Cup of Java in Netbeans** Om je eerste programma in Netbeans aan te maken, compileren en runnen volg je volgende tutorial : Hello World for the Netbeans IDE : <https://docs.oracle.com/javase/tutorial/getStarted/cupojava/netbeans.html>
2. **Debuggen in Netbeans** Netbeans heeft enkele handige tools om de fouten in je java programma op te sporen. Zo kan je de uitvoering van je programma onderbreken (even op pauze zetten) en de toestand van je programma op dat punt bekijken. Je kan de huidige waarde van al je variabelen bekijken en het programma stap voor stap laten uitvoeren. Hiertoe plaats je een breakpoint in je code door in netbeans te klikken op een lijnnummer. Via de groene driehoek (**continue**) kan je van breakpoint naar breakpoint gaan. Via de **step over** pijl kan je je code lijn per lijn uitvoeren en via de step into pijl kan je de code in die lijn zelf volgen.

Vertrek van het bijgeleverde DebugModel.java programma (cfr klasse Model van labo 3). Plaats breakpoints op die lijnen waarin uitspraak1 telkens een nieuwe waarde krijgt. Vervolgens ga je via de step over en continue doorheen de evaluatie van de code en hierbij noteer je achtereenvolgens de waarden van de variabelen uitspraak1 t.e.m. uitspraak3 (zonder naar je output window te kijken !)

3. **Stijl in Netbeans** Merk op dat de IDE automatisch de java indentatie stijl kan goed zetten door onder source de **Format** optie te gebruiken. Test dit uit !

Gebruik van de Java API

1. **KleinsteGetal** Schrijf een programma dat het kleinste van 2 in te lezen getallen bepaalt. Maak gebruik van de methode `min` van de `Math` klasse.
2. **WillekeurigGetal** Schrijf een programma dat iedere keer een ander willekeurig geheel getal genereert in het interval `[50 150]`.
3. **BMI** Schrijf een programma dat de BMI van een persoon kan berekenen door vooraf zijn lengte (in m) en gewicht (in kg) te vragen. De formule voor de BMI berekening is als volgt : $BMI = \frac{gewicht}{lengte^2}$. Gebruik de `Math` klasse.
4. **NaamMetHoofdletter** Schrijf een programma dat vraagt naar je voornaam, de eerste letter omzet naar een hoofdletter en vervolgens de voornaam afdruckt. Betrouw erop dat de gebruiker kleine letters ingeeft.
5. **ButFirst** Schrijf een programma dat een zin in één keer inleest als een `String`, en deze zin zonder het eerste woord terug schrijft naar het scherm.

```
Geef je zin :
The Great Gonzo
-> Great Gonzo
```

6. **ButSecond** Schrijf een programma dat een zin in één keer inleest als een `String`, en deze zin zonder het tweede woord terug schrijft naar het scherm.

```
Geef je zin :
The Great Gonzo
-> The Gonzo
```

7. **VraagZin** Schrijf een programma dat een zin inleest van minimum 2 woorden en er vraagzin van maakt. Dit doe je door de eerste 2 woorden van plaats te wisselen en achteraan een vraagteken toe te voegen. Denk ook aan het gebruik van de hoofdletter! Print de vraagzin uit naar het scherm.

```
Geef je zin :
Ik zou beter moeten weten
-> Zou ik beter moeten weten ?
```

8. **Nummerplaat** Schrijf een programma dat de nieuwe Belgische nummerplaten van 7 karakters in de vorm

cijfer1 – letter1letter2letter3 – cijfer2cijfer3cijfer4

kan inlezen. Ga na of de combinatie *cijfer1cijfer2cijfer3cijfer4* deelbaar is door 4. Gebruik de ternaire operator.

```
Geef je nummerplaat :
1-abc-003
-> Het getal 1003 is niet deelbaar door 4
```

9. **Rekeningnummer2** Schrijf een programma dat je rekeningnummer opvraagt en meedeelt of het geldig is of niet. Je programma leest je rekeningnummer in één keer in als een String. Het rekeningnummer van een financiële instelling bevat steeds de IBAN code gevolgd door 12 cijfers : IBAN BEXX XXXX XXXX XXXX . In een vorige oefening moest je de IBAN code scheiden van het eigenlijke rekeningnummer. Nu ga je na of het eigenlijke rekeningnummer (12 cijfers) goed opgebouwd is. Door een eenvoudige berekening kan gecontroleerd worden of er geen fouten gemaakt zijn bij het ingeven van het rekeningnummer. Het getal, gevormd door de eerste tien cijfers wordt gedeeld door 97. De rest van deze deling moet gelijk zijn aan het controlegetal gevormd door de laatste 2 cijfers. Gebruik de ternaire operator.

10. **KlantKey** Schrijf een programma dat op basis van een ingelezen achternaam en voornaam van een klant een klantcode genereert op volgende manier :

- De key begint steeds met de prefix "ODISEE"
- De key bestaat alleen uit hoofdletters
- Het volgende karakter bepaal je als volgt : neem de decimale waarde van de derde letter van de voornaam (in hoofdletters) tel hier 5 bij op en zet terug om naar een letter (let op: zorg ervoor dat je nog steeds een hoofdletter bekomt, dus als je karakter > 'Z' moet je er 26 aftrekken)
- Voor het volgende karakter neem je de laatste letter van de familienaam (als hoofdletter)

- Tenslotte eindig je de code door het volgende : neem de eerste letters van de voor en achternaam (als hoofdletter). Vermenigvuldig hun numerieke waarden, dit getal zet je om naar een string en plak je aan de code.

Dit geeft het volgende voor de naam : **Jan Janssens** (Merk op de voornaam bestaat minstens uit 3 karakters, de achternaam kan samengesteld zijn bvb **Van de winkel** beschouw dit als 1 woord, met spaties.)

- Het 7de karakter van de code is gebaseerd op de derde letter van de voornaam hier dus n . Hiervan nemen we $N + 5$ ofwel $78 + 5 = 83$ wat volgens de ASCII tabel de letter 'S' geeft.
- Het volgende karakter is eveneens 'S'
- Het produkt van 'J' met zichzelf hier geeft : $74 * 74 = 5476$
- De code voor Jan Janssens is dan als volgt : *ODISEESS5476*