# Opdrachten 4-weeks voorprogramma

## Weekindeling

### Week 1

Onderwerpen: Leren kennen van de Java omgeving, leren omgaan met de tools. Simpele in-en uitvoer, compileren en uitvoeren via de command line.  
Simpele applicatie ontwikkelen. Variabelen en elementaire datatypen. Simpel object.

Vertrouwd raken met het werken met objecten.  
Alle belangrijkste java besturingselementen: while loop, if statement, enkele belangrijke standaardbibliotheken.

Hoe schrijf ik een use case

Intro GIT

### Week 2

Intro IntelliJ

Import van libraries, opzoeken van veel gebruikte Java libraries.

Object orientatie.

Lijsten van objecten maken en gebruiken.

JUnit om te testen. Try Catch

Hoe foutscenario’s toe te voegen aan de use cases

Hoe ontwerp en schrijf ik een test

### Week 3

Simpel front end (command line front end)  
Simpel back end (bestanden lezen en schrijven)

Case huishoudboekje uitwerken

Javadoc inzien om verder gebruik te maken van libraries (IntelliJ)  
Javadoc genereren

Hoe schrijf ik een integratietest

### Week 4

Webservice controller

Front-end met JSP of Angular (Kado gegeven)

Backend java hiervan wijzigen.   
Simpelst denkbare voorbeeld van Spring Boot applicatie met JPA. Studenten hoeven niet alle code te begrijpen maar moeten wel de samenhang zodanig begrijpen dat ze het back-end kunnen wijzigen zodat alles “naar wens” werkt.

Case over huishoudboekje nu uitwerken en gebruiken als webapplicatie.

|  |  |
| --- | --- |
| Cases week 1 | Demo’s en opdrachten  (Bedenk bij ieder vakje hieronder: Demo + opdracht. Dan hoef ik dat niet telkens te herhalen ☺) |
| Dag 1  Hello World  Simpele datatypen java  Simpele klasse en object  Naam invoer | Zo simpel mogelijk programma om compileren en runnen te tonen  Aftrappen van maven. Maven pom even laten zien.  int, String  Wat is een klasse wat is een object  Simpele invoer via console |
| Dag 2  Java besturingselementen  Filteren  Use case schrijven | While, itereren  If statement + simple demo programmaatjes  Probleemstelling, stappen, uitwerking in scenario’s |
| Dag 3  Object orientatie  Punt notatie voor attributen  Punt notatie voor methoden  Scope/zichtbaarheid | Wat is een object, methoden, attributen. Objecten maken  Gebruik attribuut van een ander object.  Roep methode aan van ander object  Het begrip scope. Demo van public en private |
| Dag 4  Java packages  Import statement  Java-bibliotheken gebruiken  Iets meer over Maven, projectstructuur en build path  Java conventies naamgeving | Demo van hoe een package structuur aan te leggen  Hoe kan ik een object benaderen in een ander package.  Laat zien wat nu met de pom moet gebeuren. Laat de structuur van het project zien. Jar aftrappen met een command line. Build path. Zelf laten doen met tekstverwerker en command line. Geef aan wat de conventies zijn, vertel dat dit een eis is voor ingeleverde programma’s |
| Dag 5  Intro GIT  Intro IntelliJ  Methode parameters  Call parameters zijn call by value | GIT repository aanleggen/connecten/push/pull  Laat het interface van IntelliJ zien. Neem een bestaand project en importeer dat. Leg uit wat voor- en nadelen van IntelliJ zijn (mbt tot OCA: je moet echt zelf de syntax weten)  Laat globaal zien wat IntelliJ zoal voor je kan doen  Hoe methode parameters te declareren  Geen zij-effecten want call by value, demo  We kunnen dat voor elkaar krijgen via velden van het object (niet veilig) en via een hulpklasse als return value. Toon beide aan.  Door een objecttype te maken (aparte klasse) kunnen er toch meerdere resultaten teruggegeven worden door een methode. |

|  |  |
| --- | --- |
| Cases week 2 | Demo’s en opdrachten  (Bedenk bij ieder vakje hieronder: Demo + opdracht. Dan hoef ik dat niet telkens te herhalen ☺) |
| Dag 1  Introductie IntelliJ  Import van libraries  Introductie object oriëntatie | Toon importeren project. Toon dashboard functionaliteiten, integratie met Maven, toon openen/sluiten projecten, editor, hoe fout te vinden, compileren, debug plus breakpoint, starten/stoppen applicatie. Opdracht: importeer vorig project, hiermee een aantal opdrachten  Importeer library, toevoegen library aan POM, toon plugin POM plus effect (library kan ineens worden gebruikt incl java doc ingezien)  Klassen, objecten, inheritance voorbeelden  Een racefiets is een speciaal soort fiets met versnellingen. Die kun je alleen gebruiken als je object een racefiets is en geen fiets. |
| Dag 2  Call by value maar toch meerdere resultaten terugkrijgen?  Lijsten en arrays in Java  Gebruik van lijsten en arrays in Java | Hiervoor kunnen we een object gebruiken. Laat zien dat bij call bij value wel degelijk sprake is van een neveneffect als je een veld wijzigt van het object.  We tonen ArrayList en Map en wat we hiermee kunnen  Voorbeelden van java programma’s met arrays en lijsten bouwen. Use case voor beheren huishoudboekje – hiervoor arrays of lijsten gebruiken |
| Dag 3  JUnit  Try Catch  Logging  Foutscenario’s schrijven | Voorbeelden van unit tests. Schrijf en demonstreer je eigen unit tests  Maak een zachte landing als je programma een fout heeft die je al dan niet kunt voorzien  Breid use case uit met foutscenario’s (per definitie voor gebruiker zichtbaar) en verwerk deze foutscenario’s in unit tests |
| Dag 4  Logging | Simpel gebruik van logging framework (bijvoorbeeld logBack). Gebruik unit tests om ook de niet voor de gebruiker zichtbare fouten netjes worden opgevangen |
| Dag 5 | Herhaling van alles maar dan in het kort |

|  |  |
| --- | --- |
| Cases week 3 | Demo’s en opdrachten |
| Dag 1  Simpel front-end met alleen command line  Lezen en schrijven bestanden | Menu structuurtje, door een getal in te toetsen kies je, vervolgens krijg je een prompt en kun je in- en uitvoer doen. Hier verschillende cases van.  Streams (idealiter). |
| Dag 2  Case huishoudboekje uitwerken  intern lijstje (arrayList van objecten)  Backend is bestand, front end is command line met menu | Huishoudboekje, invoeren van uitgaven met label, omschrijving, bedrag  Opslaan van regels in bestand  Ophalen van regels uit bestand  Toon overzicht aan hand van label met totaalbedrag  Schrijf use case, programma stapsgewijs opbouwen  Schrijf test scenarios  Schrijf alternatieve scenario’s |
| Dag 3  Case verder uitwerken zowel ontwerp als bouw en test  JUnit gebruiken  Logging gebruiken  Itereren over collectie | Gebruiken van JUnit om de unit test te doen  Gebruik van logging om bepaalde situaties te loggen en voor auditing  Gebruik een iterator om over de lijst te itereren. |
| Dag 4  Use case diagram van het huishoudboekje  State transition diagram  Cases rond huishoudboekje  Switch statement in Java  Foutafhandeling: specifieke exceptie handling met betrekking tot file io. Try with resources. | Het menu heeft verschillende keuzen vertaalt zich in elementen op het use case diagram  Wat is een STD, voorbeelden van STD’s  Maak het std aan de hand van een deel van de applicatie Welke toestanden hebben we? Herbouw de control flow aan de hand van het state transition diagram en gebruik hiervoor een state variabele  Gebruik voor de control flow het switch statement met deze variabele Vertel waarom try with resources enorme voordelen heeft voor foutafhandeling |
| Dag 5  Herhaling van alles | Herhaling van alles |

|  |  |
| --- | --- |
| Cases week 4 |  |
| Dag 1 Front end, back end en web server  Front end en controller, bijvoorbeeld Angular voorbeeld (javascript)  Wat is een controller? Hoe zijn de taken verdeeld tussen deze drie? MVC controller patroon, wat is dat, hoe zit dat verwerkt in deze opzet?  Verdeling van taken tussen front end, back end en controller. Angular voorbeeld.  Wat is een web service? Wat is een web server?  Hoe gebruik je die vanuit een browser? | Leg HTML get en HTML post uit en laat zien hoe een controller deze verwerkt.  Onderscheid de functionaliteiten voor front end en back end. Welke functionaliteiten heeft front end en welke uitdrukkelijk niet? Welke voor back end? Wat is een controller?  Spring Boot voorbeelden (dwz ingebouwde webcontainer) van een zo simpel mogelijke web server die een html get request verwerkt  Starten, stoppen van de webservice via IntelliJ. Laat de uitvoer loggen naar console. |
| Dag 2  Schermen tonen/testen huishoudboekje via webservice en cadeau gegeven front end | Huishoudboekje omzetten naar werkende webapplicatie, bijvoorbeeld JSP pagina’s of Thymeleaf. Back end wijzigen, front end blijft min of meer gelijk |
| Dag 3  Hoe schrijf ik een integratietest | Vertel wat een integratietest is. Maak cases voor de integratietest. Wat is een teststub. Geef voorbeelden en demo’s over testen met stubs. Hoe bouw je je integratietest in JUnit. Hoe maak ik verschillende testomgevingen. Maak een integratietest met en zonder teststubs. |
| Dag 4  Database integratie | Simpelst van het simpelste Spring Boot applicatie met JPA. Zie spring.io voorbeelden. |
| Dag 5  Herhaling van stof van deze week met nieuwe voorbeelden |  |