***DEJAN+LOUIS maven***

***ALLARD+DUNCAN exceptions  
JASON+ROBIN threading***

***allard duncan unit testing  
jason louis dejan***

|  |  |
| --- | --- |
| Titel: | Maven |
| Beschrijving: | *As a student, I need to learn Maven so I can build, test and run applications with a script outside an IDE to prepare for working in a continuous integration environment.* |
| Acceptatiecriteria: | * Kan externe libraries toevoegen als Maven dependency. * Kan versies van dependencies aanpassen. * Kan Maven command-line gebruiken voor het runnen en unit-testen van een applicatie. * Kent de lifecycle van Maven met minimaal de goals *clean*, *compile*, *test* en *package*. * Hoeft geen rekening te houden met multi-modules |

|  |  |
| --- | --- |
| Titel: | Exceptions |
| Beschrijving: | *As a student, I need to learn checked and unchecked exceptions so I can handle runtime errors properly.* |
| Acceptatiecriteria: | * Kan gebruik maken van *try* en *catch* om een exception af te vangen * Kent de verschillen tussen checked en unchecked exceptions * Kan zelf nieuwe exceptions definieren als subclasses van checked of unchecked exceptions in Java. * Kan zelfstandig nieuwe exceptions instantieren en "gooien" met de keywords *throw* en *throws*. * Kan in unit tests checken of de juiste exception op het juiste moment komt. * Hoeft exceptions niet te loggen met een Logging library, System.out.println is voldoende. |

|  |  |
| --- | --- |
| Titel: | Unit testing |
| Beschrijving: | *As a student, I need to learn unit testing so I can prove that my software matches the requirements.* |
| Acceptatiecriteria: | * Kent de TDD red-green-refactor-cycle. * Kan op de TDD wijze testcases definieren die zorgen voor 100% testcoverage van een of twee classes bestaande uit enkele methoden. Hierbij worden niet alleen "happy day"-scenario's getest maar ook randgevallen en uitzonderingen. * Kan JUnit als framework gebruiken om testcases te schrijven en uit te voeren * Kan unit tests uitvoeren in IntelliJ waarbij coverage rapportage gegenereerd wordt. * Kan unit tests uitvoeren met Maven op de command-line. * Hoeft geen unit tests te maken voor samenwerkende classes waarbij mocking of stubbing noodzakelijk is. |

|  |  |
| --- | --- |
| Titel: | Threading |
| Beschrijving: | *As a student, I need to learn the Java Thread API so I can make applications run smoothly.* |
| Acceptatiecriteria: | * Kan met de API-onderdelen *Thread* en *Runnable* een class multi-threaded maken * Kan een thread correct starten en stoppen en kent de beperkingen van de API bij het stoppen. * Kan het keyword *synchronized* correct inzetten om te voorkomen dat threads tegelijk een gedeelde resource benaderen. * Weet het verschil tussen multi-threading en multi-processing en welke rol de JVM, een besturingssysteem en de processor met zijn cores hierin spelen. * Hoeft geef rekening te houden met threads die op elkaar moeten wachten (concurrency) waarbij methoden als *wait*, *notify* en *notifyAll* nodig zijn. |

|  |  |
| --- | --- |
| Titel: | RMI |
| Beschrijving: | *As a student, I need to learn the RMI-protocol so I can build distributed applications and know the concepts of distributed computing.* |
| Acceptatiecriteria: | * Kan een interface definieren voor een remote class gebruik makend van de API-onderdelen Remote en RemoteException * Kan een class remote benaderbaar maken gebruik makend van de API-onderdelen UnicastRemoteObject en Java interfaces. * Kan een server voor een remote class schrijven die via het RMI-protocol en de Registry aanroepen van een client kan ontvangen. * Kan een client schrijven die via het RMI-protocol en de Registry methoden op een remote class aanroept. * Weet het verschil tussen pass-by-value en pass-by-reference en kan het API-onderdeel Serializable inzetten voor pass-by-value. * Kent het concept van client-registry-server. * Hoeft geen rekening te houden met security (SecurityManager), codebases en classpaths. |