Vakinhoud Programming Advanced - Java

Onderwerp	Kennen	Kunnen
maven	ik weet wat maven is	ik begrijp de mappenstructuur van een maven project en vind hierin vlot mijn weg
	ik weet wat het bestand pom.xml	ik kan een maven project aanmaken
	ik begrijp de tags die gebruikt worden in pom.xml	ik kan dependencies door maven laten beheren
	ik begrijp de scope van dependencies	ik gebruik de correcte scope bij de dependencies
	ik ken de 3 verschillende build lifecycles	ik kan de nodige maven commando's gebruiken
	ik begrijp het concept van fasen en goals	
	ik begrijp het concept van een maven repository	

logging	Ik weet wat een logging framework is en welke voordelen het kan bieden	Ik kan dmv een appender van het log4j framework log statements in de console laten verschijnen onder een bepaald formaat (pattern)
	Ik ken het concept van log levels (ERROR, WARN, INFO, DEBUG)	Ik kan dmv een appender het log4j framework log statements in een bestand laten verschijnen onder een bepaald formaat (pattern)
	Ik begrijp het configuratie bestand log4j2.xml van het open source logging framework log4j	Ik kan de verschillende log levels (ERROR, WARN, INFO, DEBUG) gebruiken en configureren

unit testing	ik begrijp de annotaties @BeforeAll, @AfterAll, @BeforeEach, @AfterEach, @Test	ik kan unit testen in de correcte map in een maven project toevoegen
	ik begrijp wat een mock-object is	ik ken het maven commando om mijn unit testen uit te voeren
		ik kan relevante unit testen schrijven voor mijn methoden (grenzen testen)
		ik kan unit testen schrijven voor exception handling

		ik kan op een zinvolle manier gebruik maken van mockito

JDBC	ik weet wat JDBC is	ik kan een verbinding maken met een database dmv JDBC
	ik begrijp het concept van JDBC-drivers	ik kan SQL-commando's uitvoeren dmv JDBC
	ik begrijp het concept en de voordelen van Prepared Statements	ik kan Prepared Statements gebruiken
	ik begrijp wat een transactie is	ik kan gebruikmaken van transacties
	ik ken de verschillende transactie isolation levels en hun gedrag (dirty read, phantom read, non-repeatable read)	ik gebruik een persistence-laag met DAO-klassen

JPA	ik weet wat JPA is	ik kan JPA gebruiken in mijn java toepassingen
	ik ken 2 implementaties van JPA	ik kan JPA correct configureren
	ik ken de verschillende lagen in een 3-tier architectuur	ik kan entity-klassen met eigenschappen met verschillende datatypes implementeren
	ik weet wat ORM betekent	ik kan de relaties tussen entity-klassen correct implementeren
	ik weet de correcte locatie van persistence.xml	ik kan JPQL gebruiken om zoekopdrachten uit te voeren
	ik begrijp de inhoud van persistence.xml	ik kan named queries gebruiken
	ik weet wat een entity-klasse is	ik kan relevante unit testen schrijven voor mijn DAO-klassen
	ik ken de annotaties belangrijkste annotaties in een entity klasse	
	ik weet wat een persistence unit is	
	ik weet wat een entity manager is	
	ik weet wat een persistence context is	
	ik ken en begrijp de eigenschappen van een transactie	

	ik ken en begrijp de mogelijke relaties tussen entity-objecten	
	ik kan voor ieder type relatie tussen entity-objecten een voorbeeld geven	

Webcomponenten	ik weet wat HTTP is	ik kan een web-applicatie (war) schrijven in java en deployen in een webcontainer
	ik weet hoe een webclient communiceert met een webserver via het HTTP protocol (request, response, status-code)	ik maak gebruik van een 3-tier architectuur in mijn applicaties
	ik weet wat een webcontainer is en kan een voorbeeld van een webcontainer geven	ik kan relevante unit testen schrijven voor klassen in de business-laag

Servlets	ik ken de levenscyclus van een servlet	ik kan een afgeleide klasse van HTTPServlet implementeren en deployen
	ik weet waarvoor servlets gebruikt kunnen worden	ik kan sessie-status bijhouden in een servlet
		ik kan het ServletContext-object gebruiken

REST	ik kan uitleggen wat REST is (https://www.codecademy.com/articles/what-is-rest)	ik kan een REST api implementeren met JAX-RS
	ik kan een concreet voorbeeld geven van een (eenvoudige) REST api (requests, responses, models)	
	ik weet wat JAX-RS is	
	ik ken 2 implementaties va JAX-RS	

Spring Boot (optioneel)		ik kan Spring Data gebruiken
		ik kan een REST api implementeren met Spring MVC