Voorblad ***toets***

In te vullen door student:

Studentnummer:

_____

Naam:

_____

Klas:

_____

2412mm132A

In te vullen door docent:

Naam toets	Java Advanced
Toetscode	1915IN248Z
Datum	24-06-2020
Tijd	12:00 uur
Lokaal	Online proctored
Duur (in minuten)	120
Studiejaar	2019/2020
Toetsperiode	2.4
Kans	1
Examinator(en)	Wim Wiltenburg
Domein	Techniek, Ontwerpen & Informatica
Cluster	ICT
Opleiding + Variant	Informatica Voltijd
Locatie	Haarlem
Aantal pagina's (incl. voorblad)	4
Aantal opgaven	6
Cesuur	5.5 Max: 90, cijfer: (10 + behaalde punten) / 10
Toegestane hulpmiddelen	PC of laptop, internet, pen en papier
Antwoordblad	
Bijzonderheden	Geen digitale communicatiemiddelen toegestaan

(* doorhalen wat niet van toepassing is)

In te vullen door de toetsorganisatie:

Surveillanten:

Bijzondere voorzieningen:

Voorziening:	**Aantal:**	**Opmerking:**
Dyslexie		
A3		
Overig		

Tentamen Java Advanced

Woensdag 24-06-2019, 12:00 - 14:00

Inleiding

Dit tentamen bestaat uit 6 vragen, allen gerelateerd aan de cursus Java Advanced. ***Als er een bestand wordt genoemd in een vraag, dan is dat bestand te vinden op* Moodle.**

Eisen aan de code

1. Maak waar mogelijk gebruik Java Advanced features, zoals Stream API, lambda expressies, etc.
2. De code moet zijn gemaakt in Java 11
3. Gebruik de nieuwste versie van Spring Boot
4. Gebruik als groupId “nl.inholland.tentamen” en als artifactId je naam als een woord, bijvoorbeeld JaneDoe
5. Gebruik alleen de volgende methodes om je project te starten[1]:
    a. Maven archetype quickstart en refactor naar behoefte
    b. https://start.spring.io (AANBEVOLEN!). Eerst alleen Web toevoegen
6. Tenzij specifiek vernoemd voeg alleen spring-boot-starter dependencies toe
7. Hanteer de volgende (minimale) codeerstandaarden
    a. Java Naming Conventions
    b. Elk statement heft een eigen regel
    c. Niet meer dan 30 regels per methode
    d. Gebruik commentaar om te documenteren waarom je iets doet, niet wat.
    e. Formatteer je code correct.
    f. Gebruik packages. Gebruik van het default package levert puntenverlies op
8. Schrijf je eigen code
9. De applicatie moet kunnen opstarten. Als refactoring nodig is om de applicatie op te starten levert dat puntenverlies op

Inleverinstructie

Maak een zip-bestand van de root van het project, zodat dit bestand alleen de volgende bestanden/directories bevat:

/src
pom.xml
drivers.csv
cars.csv
f1.p12

De naam van je bestand is hetzelfde als het artifactId van je applicatie met als bestandsextensie ".zip", bijvoorbeeld JaneDoe.zip. Upload je zip-bestand op Moodle. Als je een identieke naam hebt als een andere student, voeg dan ook je studentnummer toe aan je bestandsnaam.

Puntenverdeling

De volgende (maximale) punten worden toegekend:

Question	Points
Question 1	5
Question 2	10
Question 3	10
Question 4	15
Question 5	25
Question 6	25
Total points	90

[1] Andere methodes zijn alleen betrouwbaar als je weet wat je doet!

Vraag 1: Kickstart een nieuwe Spring Boot Applicatie

- ☐ ~~Maak een nieuw Spring Boot project.~~
- ☐ ~~Voeg Lombok dependency toe~~
- ☐ ~~Verzeker je ervan dat de base application klasse Spring Boot runt~~
- ☐ ~~Als de applicatie is opgestart log "--- Exam Question 1 completed ---" op INFO niveau~~

Vraag 2: Formule 1, maak een Driver klasse

- ☐ ~~Maak een klasse genaamd Driver met de volgende eigenschappen:~~
  - ~~long id (id begint met 1_000_001)~~
  - ~~String firstName~~
  - ~~String lastName~~
  - ~~Ranking ranking -- Ranking is een Enum met de volgende waarden: NOVICE, INTERMEDIATE, ADVANCED, SENNA~~
  - ~~int age~~
  - ~~boolean wonBefore~~
- ☐ ~~Lees het bestand *drivers.csv* in en creëer hiervan nieuwe objecten en schrijf deze naar de database. Je kunt een van een String een Enum maken middels de Ranking.valueOf() methode.~~
- ☐ ~~Schrijf deze drivers naar het console en log "--- Exam Question 2 completed ---"op INFO niveau~~

Vraag 3: Formule 1, de Car klasse

- ☐ ~~Creëer een klasse genaamd Car met de volgende eigenschappen:~~
  - ~~long id (id begint met 9_000_001)~~
  - ~~String brand~~
  - ~~int topSpeed – Dit is een willekeurig getal tussen de 250 en 300 (exclusief)~~
  - ~~Driver driver~~
- ☐ Lees het bestand *cars.csv* in en maak hiervan nieuwe Car objecten. Het tweede veld van de csv-regel is het id van de driver uit vraag 2. Schrijf deze cars weg naar de database.
- ☐ Schrijf deze cars naar het console en log "--- Exam Question 3 completed ---"op INFO niveau

Vraag 4: Beveilig de applicatie met HTTPS

- ☐ Beveilig de applicatie met een PKCS12-keystore genaamd f1.p12 welke de volgende eigenschappen heeft:
  - Password: formulaOne
  - Alias: f1
  - Algorithm: RSA
  - Valid: 365 days
  - First and last name: <jouw-naam>
  - Organizational Unit: The Stable
  - Name of Organization: FormulaOne
  - City: Zandvoort
  - State or province: Noord-Holland
  - Country Code: NL
- ☐ Run de applicatie op port 8443
- ☐ Log "--- Exam Question 4 completed ---"op INFO niveau vanuit de base application klasse.

Vraag 5: Endpoints

- ☐ Creëer een endpoint om een specifieke driver op te halen aan de hand van het id.
- ☐ Log "--- Exam Question 5a completed ---"op INFO niveau
- ☐ Creëer een endpoint om een lijst met cars op te halen, waarbij je kunt kiezen of deze drivers bevat die wel iets hebben gewonnen of deze drivers bevat die niets hebben gewonnen.
- ☐ Log "--- Exam Question 5b completed ---"op INFO niveau

Vraag 6: Meer endpoints

- ☐ Creëer een endpoint waarmee je een nieuwe Car (merk: Mercedes, Driver is Max Vierstappen). Creëer hiervoor ook een klasse CarDTO met de volgende eigenschappen:
  - String brand
  - long driverId

  Gebruik een CarDTO in je request body en geef de nieuwe Car terug in de response body
- ☐ Log "--- Exam Question 6a completed ---"op INFO niveau
- ☐ Creëer een endpoint welke de som van de maximumsnelheid van alle cars in de database teruggeeft in een JSON-object met als enige eigenschap: total. **Gebruik geen SQL**
- ☐ Log "--- Exam Question 6a completed ---"op INFO niveau