

## Voorblad **toets**

### In te vullen door student:

Studentnummer:

\_\_\_\_\_

Naam:

\_\_\_\_\_

Klas:

\_\_\_\_\_

\*2412mm132A\*

### In te vullen door docent:

|                                  |   |
|----------------------------------|---|
| Naam toets                       | Java Advanced   |
| Toetscode                        | 1915IN248Z  |
| Datum                            | 29-10-2020  |
| Tijd                             | 15:00 uur   |
| Lokaal                           | Online  |
| Duur (in minuten)                | 150   |
| Studiejaar                       | 2019/2020   |
| Toetsperiode                     | 2.4   |
| Kans                             | 2   |
| 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<br>Max: 90, cijfer: (10 + behaalde punten) / 10                                 |
| Toegestane hulpmiddelen          | PC of Laptop, Internet  |
| Antwoordblad                     | Lijntjespapier / ruitjespapier / op opgave schrijven / schrapkaart / grafiekpapier* |
| Bijzonderheden                   |   |

(\* delete if not applicable)

### In te vullen door de toetsorganisatie:

#### Surveillanten:

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |

#### Bijzondere voorzieningen:

| Voorziening: | Aantal: | Opmerking: |
|--------------|---------|------------|
| Dyslexie     |         |            |
| A3           |         |            |
| Overig       |         |            |

---

# Exam Java Advanced

Thursday 29-10-2020, 15:00 - 17:00

---

## 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<sup>1</sup>:
  - 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. Formateer 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`  
`venues.csv`  
`concerts.csv`

De naam van je bestand is hetzelfde als het artifactId van je applicatie aangevuld met je studentnummer gescheiden met streepjes met als bestandsextensie ".zip", bijvoorbeeld Jane-Doe-123456.zip. Upload je zip-bestand op Moodle.

## Puntenverdeling

De volgende (maximale) punten worden toegekend:

| Vraag   | Punten |
|---------|--------|
| Vraag 1 | 5      |
| Vraag 2 | 10     |
| Vraag 3 | 10     |
| Vraag 4 | 15     |
| Vraag 5 | 25     |
| Vraag 6 | 25     |
| Totaal  | 90     |

---

<sup>1</sup> 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. Alle logging in de applicatie gebeurt met Lombok
- ☐ Verzeker je ervan dat de main applicatieklasse Spring Boot runt
- ☐ Als de applicatie is opgestart log "--- Exam Question 1 completed ---" op INFO niveau

## Vraag 2: Concerts, maak een Venue klasse

- ☐ Maak een Venue klasse aan met de volgende eigenschappen:
  - o long id (id begint met 1\_000\_001)
  - o String name
  - o String city
  - o Country country -- Country is een Enum met de volgende waarden: BELGIUM, FRANCE, GERMANY, NETHERLANDS
  - o int capacity
- ☐ Lees het bestand [venues.csv](#) in en creëer hiermee nieuwe Venue objecten. Schrijf deze weg naar een database. Om van een String een Country te maken gebruik je de method `Country.valueOf()`
- ☐ Print deze venues naar console en log op INFO niveau "--- Exam Question 2 completed ---"

## Vraag 3: Concerts, maak een Concert klasse

- ☐ Maak een Concert klasse aan met de volgende eigenschappen:
  - o Long id (id begins at 5\_000\_001)
  - o String artist
  - o LocalDate date
  - o Venue venue
- ☐ Lees het bestand [concerts.csv](#) in en creëer hiermee nieuwe Concert objecten. Het eerste veld van elke regel is de naam van een Venue uit vraag 2. Schrijf deze Concerts weg naar een database.
- ☐ Het mag niet mogelijk zijn om een artiest twee keer op dezelfde datum op dezelfde Venue te boeken
- ☐ Print deze Concerts naar console en log op INFO niveau "--- Exam Question 2 completed ---"

## Vraag 4: Beveilig de applicatie met een API token

- ☐ Beveilig alle endpoints van de applicatie met een API token. Voeg de juiste dependency toe. De naam van de HTTP header voor het API token staat in een property genaamd "nl.inholland.tokenheader" en heet "token-auth". De waarde die in de header moet worden ingevuld is **A9C5BCAB4ADD226E**

## Vraag 5: Endpoints

- ☐ Maak een endpoint met een path variable genaamd "concertId" waarmee je een enkel Concert kunt ophalen aan de hand van het ID
- ☐ Log op INFO niveau "--- Exam Question 5a completed ---"
- ☐ Maak een endpoint welke een lijst van Concerts teruggeeft aan de hand van de naam van een land. **Gebruik hiervoor een database query.**
- ☐ Log op INFO niveau "--- Exam Question 5b completed ---"

## Vraag 5: Nog meer endpoints

- ☐ Maak een endpoint waarmee je een nieuw Concert kunt aanmaken (artist: KoolAid and the Bang, Venue: Rock Tempel in Assen, datum: 1 juni 2021. Maak hiervoor een klasse genaamd ConcertDTO met de volgende eigenschappen:
  - String artist
  - String venueName
  - String dateGebruik het ConcertDTO object in het request body, and retourneer het nieuw aangemaakte Concert in de response body.
- ☐ Log op INFO niveau "--- Exam Question 6a completed ---"
- ☐ Maak een endpoint genaamd "capacity" welke de totale capaciteit voor alle venues teruggeeft in een JSON object met een enkele property: totalCapacity. **Gebruik geen SQL om dit te berekenen.**
- ☐ Log op INFO niveau "--- Exam Question 6b completed ---"