

## GROEIPROJECT – ALGEMENE INFO

De bedoeling van dit “groeiproject” is om een uitgebreid project in IntelliJ te maken dat elke week “meegroeit”. Het komt er dus op neer dat je de leerstof van elke week toepast in een nieuw toegevoegde module. Je krijgt hiervoor wekelijks specifieke instructies.

Op het einde lever je de gezippte code in; goed voor **10%** van de evaluatie voor dit vak. En uiteraard heb je het bijkomend voordeel dat je dan de leerstof onder de knie hebt en goed bent voorbereid op het examen.

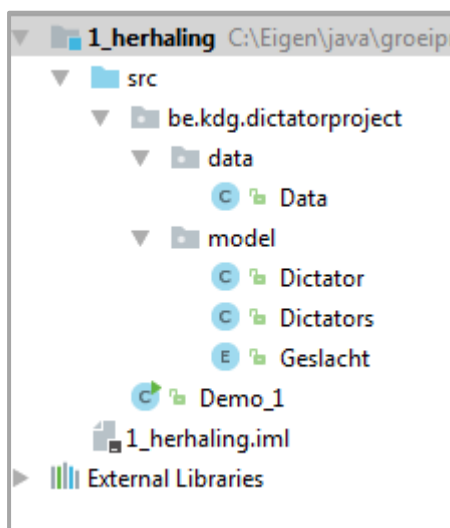
Veel succes!



## MODULE 1: HERHALING

### 1. Voorbereiding

- 1.1. Kies een concreet onderwerp dat je interesseert of dat je goed kent. Bijvoorbeeld: wielrenners, kamerplanten, trekvogels, racewagens, smartphones, voetballers, rockbands, scoutsgroepen, planeten, vlinders... De voorwaarde is dat je daarover wat interessante data hebt en dat er **niemand anders in jouw klasgroep** hetzelfde onderwerp heeft gekozen.  
→ Als voorbeeld waarnaar we voortdurend zullen verwijzen, kiezen we: **dictators** (biografische data over beruchte en historische dictators zoals bijvoorbeeld Jozef Stalin en Adolf Hitler)  
→ Je moet eerst via het forum op BB goedkeuring krijgen voor jouw gekozen onderwerp. Lees ook de details onder 2.1
- 1.2. Maak een nieuw (empty) project in IntelliJ en voeg daarin de eerste module toe: `1_herhaling`.
- 1.3. Maak een aparte package: `be.kdg.<projectnaam>`. Voorzie daaronder een aparte subpackage: `model` met daarin 2 klassen; een **basisklasse** en een **multiklasse**. De woordkeuze is belangrijk: de multiklasse is het meervoud van de naam van de basisklasse.  
→ In ons voorbeeld heet de basisklasse: `Dictator` en bevat de beschrijving van één dictator.  
→ De multiklasse heet: `Dictators` en bevat een groep `Dictator`-objecten.
- 1.4. De uiteindelijke package structuur zal er zo uitzien:



## 2. De basisklasse (in ons voorbeeld: Dictator )

2.1. In de basisklasse voorzie je minstens 6 attributen. Het is erg belangrijk voor het verdere groeiproject dat daar verschillende datatypes tussen zitten. Zorg voor minstens één attribuut van volgende types:

- `String`
- `double`
- `int`
- een enum-type
- een `LocalDate`

→ In ons voorbeeld heeft de klasse `Dictator` volgende attributen: `naam`, `geslacht` (enum), `geboorte` (`LocalDate`), `land`, `regime`, `duur` (aantal jaren aan de macht), `wreedheid` (percentage) en `slachtoffers` (aantal in mln)

2.2. Voorzie in de basisklasse constructors, getters en setters.

- In de **setters** doe je controle op de nieuwe waarden die via de parameters binnenkomen. Indien verkeerd, doe je een throw van een `IllegalArgumentException` (met gepaste foutmelding).  
→ In ons voorbeeld kan de naam van een `Dictator` niet leeg zijn, de geboortedatum moet in het verleden liggen, het percentage wreedheid moet in de range 0..1 liggen, enz...
- In de **constructor** komen alle attribuutwaarden via parameters binnen. Roep hier de setters aan voor de controle.
- Voorzie ook een **default-constructor**. Daarin roep je de andere constructor op met dummy-waarden (bijvoorbeeld "*Anoniem*" of "*Ongekend*" voor Strings, nul voor getallen, ...)

2.3. Laat het uniek-zijn van een object van de basisklasse afhangen van één of meer attribuut(en); werk dus een gepaste `equals` en `hashCode` methode uit.

→ In ons voorbeeld is elke `Dictator` uniek door zijn naam.

2.4. Maak de basisklasse `Comparable` op hetzelfde attribuut(en) als hierboven (2.3)

2.5. Schrijf in de basisklasse een `toString` methode die een geformatteerde string teruggeeft met de belangrijkste attribuutwaarden. Zorg voor een mooie uitlijning (bekijk daarvoor de voorbeeldafdruk op de volgende blz.)

→ Voorbeeld van de klasse `Dictator`:

Ayatollah Khomeini    (°1902) Iran    regime: Fundamentalisme    8,5 mln doden

## 3. De multiklasse (in ons voorbeeld Dictators)

3.1 In de multiklasse maak je gebruik van een ingekapselde **Treeset**. Het uniek-zijn en de volgorde binnen deze set heb je al vastgelegd in de vorige opdrachten (2.3 en 2.4) .

3.2 Voorzie de volgende methoden:

a. **toevoegen**:

→ `public boolean voegToe(Dictator dictator)`

- b. **verwijderen**: als parameter komt het attribuut binnen waarvan het uniek-zijn afhangt (zie 2.3; in ons voorbeeld: de naam van de Dictator) .  
→ `public boolean verwijder(String naam)`
- c. **zoeken**: als parameter komt het attribuut binnen waarvan het uniek-zijn afhangt (zie 2.3; in ons voorbeeld: de naam van de Dictator). Returnwaarde: de gevonden Dictator of null.  
→ `public Dictator zoek(String naam)`
- d. **sorteren**: voorzie minstens 3 methoden die een gesorteerde List teruggeven. Maak gebruik van bijkomende **Comparator** klassen (inner classes).  
→ `public List<Dictator> gesorteerdOpNaam()`  
→ `public List<Dictator> gesorteerdOpGeboorte()`  
→ `public List<Dictator> gesorteerdOpSlachtoffers()`
- e. `public int getAantal()`

#### 4. De Dataklasse en de main

- 4.1. Maak een klasse Data in een aparte data-package. Daarin is er een static methode die een gevulde datalist retourneert. Voorzie minstens 15 elementen met realistische en uiteenlopende data.  
→ In ons voorbeeld dus een List met 15 Dictator-objecten.
- 4.2. Maak een main-klasse Demo\_1 waarin je het volgende uittest:
  - a. Vraag een gevulde datalist op (zie 4.1)
  - b. Voeg alle objecten van deze list toe aan de multiklasse (zie 3.3)
  - c. Voeg ook eens een dubbel object toe (dat zou niet mogen lukken; zie 2.3)
  - d. Test de methoden zoek, verwijder en getAantal uit (zie 3.3)
  - e. Druk de 3 gesorteerde listen af (zie 3.3)
  - f. Test beide constructors uit en ook de IllegalArgumentException (zie 2.2)

#### 5. Mogelijke output van het dictator-project:

Dictators gesorteerd op naam:

|                    |                   |                             |                |
|--------------------|-------------------|-----------------------------|----------------|
| Adolf Hitler       | (°1889) Duitsland | regime: Nazisme             | 66,0 mln doden |
| Augusto Pinochet   | (°1915) Chili     | regime: Militaire dictatuur | 0,1 mln doden  |
| Ayatollah Khomeini | (°1902) Iran      | regime: Fundamentalisme     | 8,5 mln doden  |
| Benito Mussolini   | (°1883) Italië    | regime: Fascisme            | 2,0 mln doden  |
| enz...             |                   |                             |                |

Dictators gesorteerd op geboortedatum:

|                  |                   |                             |                |
|------------------|-------------------|-----------------------------|----------------|
| Jozef Stalin     | (°1878) Rusland   | regime: Stalinisme          | 45,0 mln doden |
| Benito Mussolini | (°1883) Italië    | regime: Fascisme            | 2,0 mln doden  |
| Adolf Hitler     | (°1889) Duitsland | regime: Nazisme             | 66,0 mln doden |
| Francisco Franco | (°1892) Spanje    | regime: Militaire dictatuur | 5,0 mln doden  |
| enz...           |                   |                             |                |

Dictators gesorteerd op slachtoffers:

|                    |                   |                         |                |
|--------------------|-------------------|-------------------------|----------------|
| Adolf Hitler       | (°1889) Duitsland | regime: Nazisme         | 66,0 mln doden |
| Jozef Stalin       | (°1878) Rusland   | regime: Stalinisme      | 45,0 mln doden |
| Ayatollah Khomeini | (°1902) Iran      | regime: Fundamentalisme | 8,5 mln doden  |
| Mao Tse-tung       | (°1893) China     | regime: Communisme      | 8,0 mln doden  |
| enz...             |                   |                         |                |