

Oefeningen – Exception handling

Oefening 1

Maak de **klasse Persoon**. Deze klasse heeft als velden:

naam (datatype String)

geboorteDatum (datatype LocalDate)

Voorzie een constructor die een waarde voor naam, geboortedag, geboortemaand en geboortjaar heeft.

```
Persoon p = new Persoon("Sam",3,4,2001);
```

Zorg ervoor dat twee personen als gelijk beschouwd worden als ze dezelfde naam hebben.

Maak de methode toString die de gegevens afdruckt in onderstaand formaat:

```
"Bert: 198 dagen tot verjaardag"
```

Via de methode aantalDagenTotVerjaardag wordt het aantal dagen tot de volgende verjaardag van de persoon teruggegeven.

Voorzie de getter voor naam.

Test de klasse met de voorziene unit tests. Pas aan waar nodig.

Maak de **klasse Vriendengroep**. Deze klasse heeft als velden

vrienden (een array met daarin waarden van datatype Persoon)

MAXAANTALVRIENDEN een constante met waarde 10

In de constructor wordt de rij vrienden aangemaakt. De inhoud van vrienden is nu een rij met daarin 10 keer de null-waarde.

Via de methode `voegVriendToe` wordt een `Persoon`-object toegevoegd aan de array vrienden. Let wel, er wordt gecontroleerd of de naam van de persoon nog niet voorkomt in de array, indien dit wel zo is dan wordt een exception opgeworpen met de boodschap "Deze naam bestaat al", gevolgd door de naam van de vriend. Verder wordt ook gekeken of er nog een vrije positie in de rij is, indien dit niet zo is dan wordt een exception opgeworpen met de boodschap "De rij vrienden is al vol". Voor beide exceptions maak je zelf een custom exception aan.

Maak de methode `verwijderVriend`. Deze methode neemt een `String` als argument, deze `String` bevat de naam van de te verwijderen vriend. De rij vrienden wordt doorlopen en op de index waar een `Persoon`-object met de opgegeven naam zich bevindt, wordt de waarde null geplaatst. Indien de naam niet voorkomt in de rij vrienden, wordt er een exception opgegooid met de boodschap "Deze persoon is geen vriend en kan niet verwijderd worden uit de vriendenlijst".

Voorzie de methode `toString`. Via deze methode worden alle `Persoon`-objecten in de array teruggegeven, ook wordt de naam van de persoon wiens verjaardag het dichtst bij de huidige datum ligt afgedrukt. De uitvoer van de methode `toString` is van de vorm:

```
Bert 3 4 2001
Ernie 1 1 1995
Tommie 2 2 1999
De volgende jarige is Ernie.
```

In de main-method van de klasse `VriendenApp` wordt telkens een waarde voor naam, geboortedag, geboortemaand en geboortjaar gevraagd tot voor naam de waarde stop ingegeven wordt. Er wordt telkens een `Persoon`-object aangemaakt en dit object wordt toegevoegd aan een vriendengroep. Ten slotte wordt heel de vriendengroep afgedrukt.

Mocht er een exception opgeworpen worden dan moet de main methode hier mee om kunnen. Er wordt in dat geval een foutboodschap geprint in de console, waarna de volgende persoon kan worden ingegeven.

Schrijf de nodige unit tests om na te gaan of de methoden voegVriendToe en verwijderVriend de gevraagde exceptions opwerpen bij foutieve ingave.

Test nadien je zelf geschreven unit tests en pas aan waar nodig.

UML

