

Java Essentials: Project 2

Een vriendengroep speelt iedere vrijdagavond gezelschapsspelen en/of online games. Ze willen nu een programma om hun beoordelingen over de spelen te digitaliseren.

Zorg ervoor dat je de voorgestelde methoden en klassen aanmaakt volgens de richtlijnen die hieronder worden beschreven. Je voegt GEEN extra eigenschappen en methoden toe.

1. De klasse Uitgever

Uitgever		
f	naam	String
f	jaarOprichting	int
f	hoofdkantoor	String
f	website	String
m	Uitgever(String, String, int)	
m	getNaam()	String
m	setNaam(String)	void
m	getJaarOprichting()	int
m	setJaarOprichting(int)	void
m	getHoofdkantoor()	String
m	setHoofdkantoor(String)	void
m	getWebsite()	String
m	setWebsite(String)	void
m	toString()	String

Deze klasse representeert een uitgever van spelletjes. We voorzien een constructor waarbij de parameters naam, hoofdkantoor (dit is de locatie waar het hoofdkantoor gevestigd is) en jaarOprichting worden meegegeven.

Let op dat de waarde van jaarOprichting niet hoger mag zijn dan het huidige jaar. Je kan hiervoor gebruik maken van `LocalDate.now().getYear()` (zie project 1). Indien een hogere waarde wordt gegeven, geef je jaarOprichting de waarde 0.

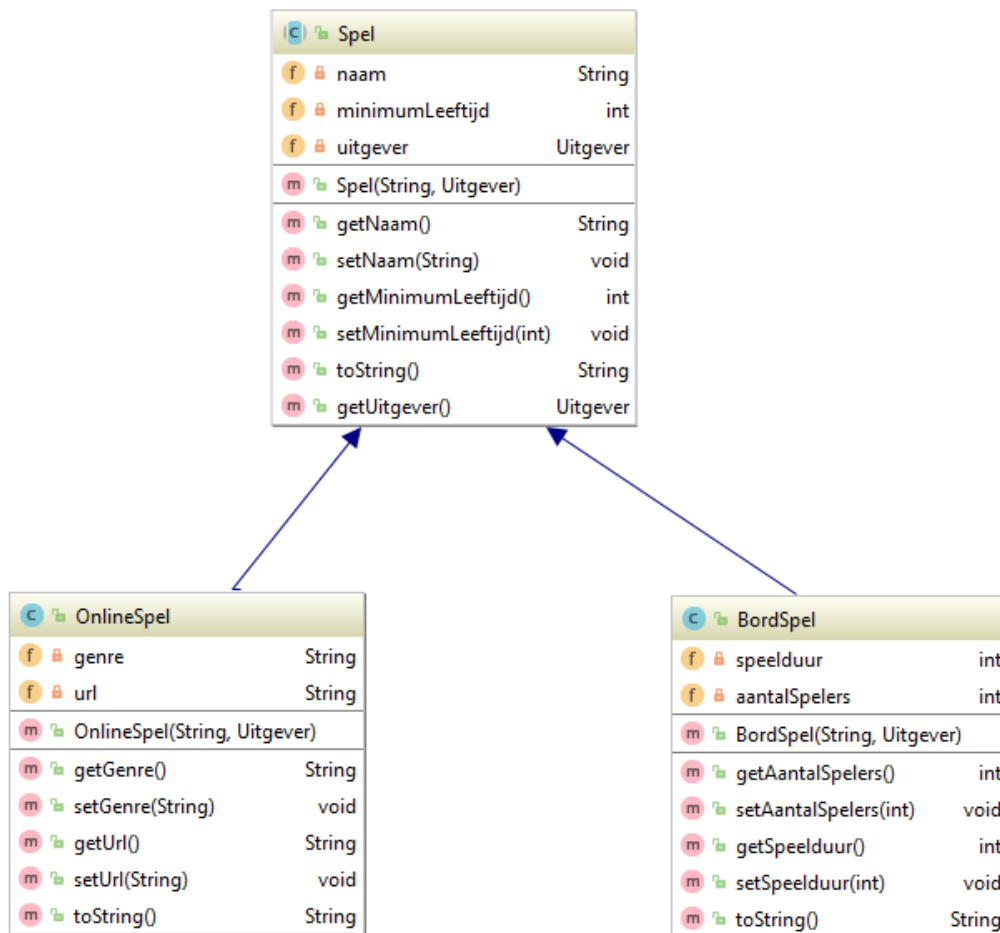
De `toString()` methode genereert een String met het volgende formaat:

```
Uitgever{naam='<<naam>>', jaarOprichting=<<jaarOprichting>>,
hoofdkantoor='<<hoofdkantoor>>', website='<<website>>'}
```

Bijvoorbeeld:

```
Uitgever{naam='Riot Games', jaarOprichting=2006, hoofdkantoor='LA',
website='www.riotgames.com'}
```

2. De abstracte klasse Spiel



De klasse `Spiel` is een abstracte klasse die we later gebruiken als basis voor `BordSpiel` en `OnlineSpiel`. We voorzien een constructor met 2 parameters: naam en `Uitgever`.

Verder voorzien we nog een eigenschap `minimumLeeftijd` die niet negatief mag zijn. Indien een negatieve waarde wordt gegeven, gebruik je de waarde 12.

3. De klasse `BordSpiel`

De klasse `BordSpiel` is afgeleid van de klasse `Spiel`. Volgende eigenschappen worden toegevoegd: `aantalSpelers` en `speelduur`. Beide eigenschappen mogen geen negatieve waarde hebben. Indien toch een negatieve waarde wordt gegeven, gebruik je respectievelijk de waarden 1 (`aantalSpelers`) en 30 (`speelduur`).

4. De klasse OnlineSpel

De klasse OnlineSpel is eveneens afgeleid van de klasse Spel. Voeg de eigenschappen genre en url toe.

Hieronder vind je nog het resultaat van toString() van een OnlineSpel en een BordSpel terug:

```
OnlineSpel{naam='League of Legends', minimumLeeftijd=0, uitgever=Uitgever{naam='Riot Games', jaarOprichting=2006, hoofdkantoor='LA', website='www.riotgames.com'}, genre=Multiplayer online battle arena, url=prod.eu.lol.riotgames.com}
```

```
BordSpel{naam='Bonanza', minimumLeeftijd=12, uitgever=Uitgever{naam='999 games', jaarOprichting=1990, hoofdkantoor='Almere', website='www.999games.nl'}, aantalSpelers=4, speelduur=45}
```

5. De klasse Speler

Speler		
f	naam	String
f	geboortejaar	int
m	Speler(String, int)	
m	getNaam()	String
m	setNaam(String)	void
m	getGeboortejaar()	int
m	setGeboortejaar(int)	void
m	getLeeftijd()	int

Een Speler-object heeft 2 eigenschappen: een naam en een geboortejaar. Een geboortejaar mag nooit in de toekomst liggen en ook niet kleiner zijn dan 1900. Wanneer een ongeldig geboortejaar wordt gegeven, ken je de waarde 0 toe aan geboortedatum.

Je berekent de leeftijd met behulp van het geboortejaar. Indien het geboortejaar 0 is, geef je -1 als waarde.

6. De klasse Score

Score		
f	MAX_SCORE	int
f	waarde	int
m	Score(int)	
m	Score(Score[])	
m	setWaarde(int)	void
m	getStars()	String
m	equals(Object)	boolean

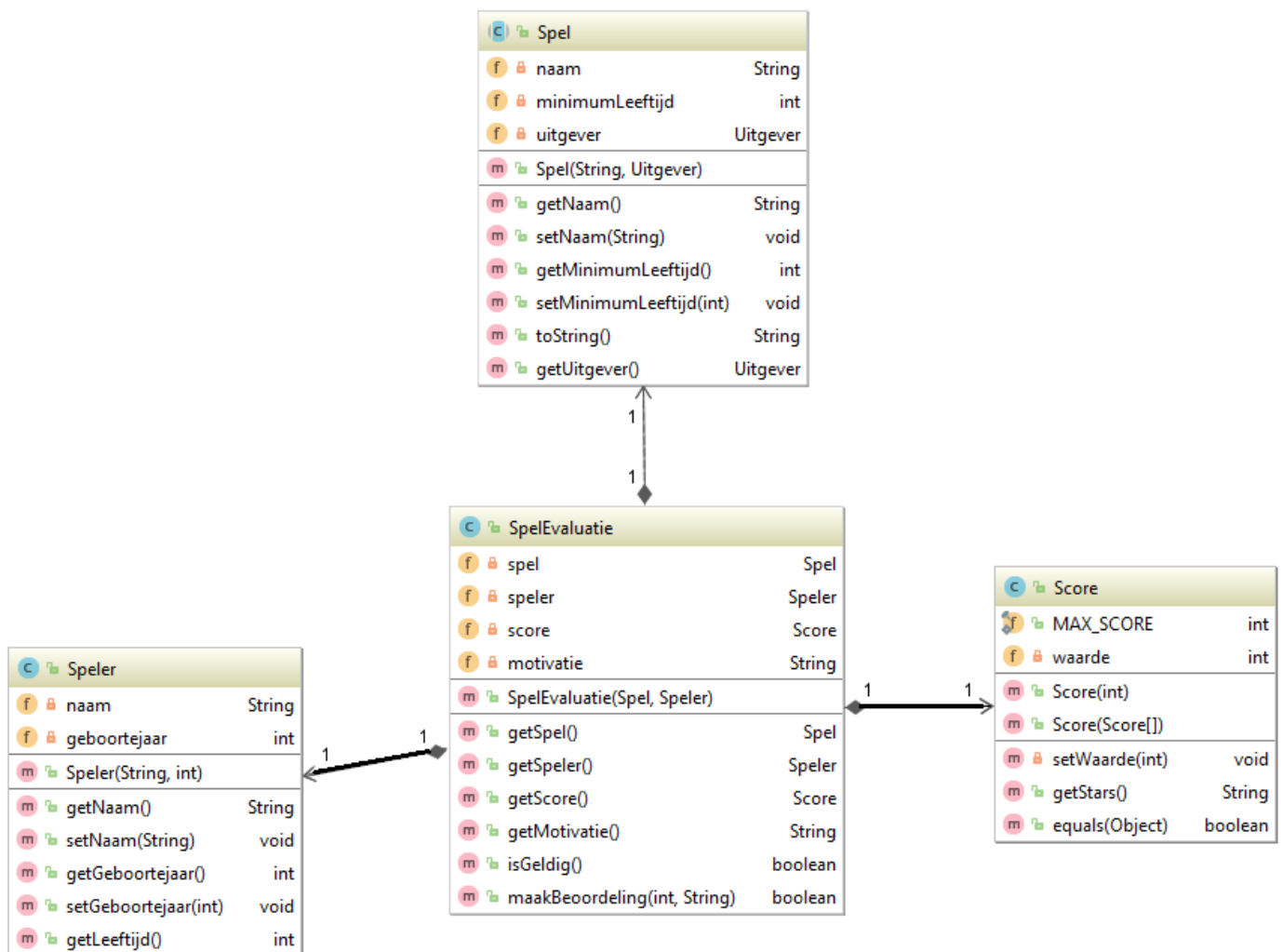
Een Score-object gebruiken we om de score van een spel bij te houden. Geef constante MAX_SCORE de waarde 5. Indien een score wordt gegeven die minder is dan 0 of meer dan MAX_SCORE, worden respectievelijk 0 of MAX_SCORE als waarde gebruikt.

getStars() geeft een string-representatie van de score weer met sterren. Bijv. een score met waarde 3 geeft "***".

Je kan ook een nieuwe score maken door een array van bestaande Score-objecten te geven. Dit nieuwe Score-object is het gemiddelde (naar beneden afgerond) van de gegeven Score-objecten.

Voorzie de methode equals(): twee Score-objecten zijn gelijk, indien de waarde van de score gelijk is.

7. De klasse SpelEvaluatie



Een **SpelEvaluatie**-object wordt gebruikt om een **Speler** een **Spel** te laten evalueren. In de methode `maakBeoordeling()` geeft de **Speler** een score (int), die wordt bijgehouden in een **Score**-object, en een motivatie. Wanneer de methode `maakBeoordeling()` een tweede keer wordt opgeroepen, heeft dit geen effect en blijven de oorspronkelijke waarden behouden. Je kan namelijk maar 1 keer een beoordeling doen.

De methode `isGeldig()` geeft `true` indien `spel`, `speler` en `score` een geldige (niet null) waarde hebben. En de leeftijd van de speler groter of gelijk is aan de minimum leeftijd van het spel.

8. Het hoofdprogramma

Maak nu met behulp van bovenstaande klassen een programma om gezelschapsspelen te evalueren.

Maak eerst een **Spel**-object aan, je mag kiezen of je een **BordSpel** of een **OnlineSpel** gaat beoordelen, en druk alle gegevens af.

Vraag vervolgens hoeveel spelers het spel gaan evalueren. Indien je een BordSpel evalueert, mag dit aantal niet meer zijn dan het aantal spelers dat het spel toelaat. Dit controleer je in het programma.

Van iedere speler vraag je de naam en het geboortjaar.

Vervolgens geven de spelers hun score en de motivatie bij de score.

Druk daarna alle evaluaties af. Geef een aanduiding of de evaluatie geldig is of niet (+ voor een geldige evaluatie en - voor een niet-geldige evaluaties). Tenslotte toon je nog de eindscore voor het spel. Deze eindscore is het gemiddelde (naar beneden afgerond) van alle geldige evaluaties.

Hieronder volgt een voorbeeld van een uitvoer van het programma:

```
BordSpel{naam='Bonanza', minimumLeeftijd=12,  
uitgever=Uitgever{naam='999 games', jaarOprichting=1990,  
hoofdkantoor='Almere', website='www.999games.nl'},  
aantalSpelers=4, speelduur=45}
```

Hoeveel spelers evalueren het spel?

4

Geef speler naam:

Mickey

Geef geboortjaar:

1999

Geef uw score (0-5):

4

Geef uw motivatie:

Heel leuk spel, we hebben uren gespeeld.

Geef speler naam:

Minnie

Geef geboortjaar:

2001

Geef uw score (0-5):

3

Geef uw motivatie:

Moeilijk spel.

Geef speler naam:

Kwik

Geef geboortejaar:

2011

Geef uw score (0-5):

2

Geef uw motivatie:

Ik snap het spel niet.

Geef speler naam:

Kwek

Geef geboortejaar:

2006

Geef uw score (0-5):

4

Geef uw motivatie:

Heel leuk spel!

Evaluatie Bonanza 999 games

Mickey ***** Heel leuk spel, we hebben uren gespeeld. +

Minnie ***** Moeilijk spel. +

Kwik ***** Ik snap het spel niet. -

Kwek ***** Heel leuk spel! -

Totaalscore: ***