

Assignment 26 – A Game for Two Part 1

Doel

Het doel van deze opdracht is het oefenen met classes en objecten, waarbij de focus ligt op het gebruik van een class in een andere class.

Overzicht

Deze opdracht heeft 3 classes:

- Class Assignment26
- Class Game
- Class Player

Opmerking: Alleen de methode **printScoreBoard** heeft **System.out.println** methode(s) om informatie te printen naar console.

1. Voorbereiding

- Maak een package aan met de naam **assignment26** (let op packages beginnen altijd met een kleine letter)
- Maak in de package een nieuwe class met de naam **Assignment26** en maak de **main** methode aan. De verdere implementatie van de **main** methode wordt verder op in de opdracht besproken.

2. Class Player

We beginnen met het maken van de class **Player**, omdat deze gebruikt gaat worden door de class **Assignment26** en class **Game**.

- Maak in de package **assignment26** de nieuwe class met de naam **Player**.
- De class **Player** heeft twee eigenschappen (fields), resp. **name** en **score**. Bepaal zelf meest voor de hand liggend data type. De fields dienen beide **private** te zijn.
- Maak voor beide fields de bijbehorende getters en setters.
- De class heeft geen custom constructor(s), we maken gebruik van de default constructor, die Java zelf maakt bij het compileren.

3. Class Game

De class **Game** gebruikt de class **Player**.

- a. Maak in de package **assignment26** de nieuwe class met de naam **Game**.
- b. De class **Game** heeft drie eigenschappen (fields), resp. **title**, **player1** en **player2**. Bepaal zelf meest voor de hand liggend data type voor field **title**. De fields **player1** en **player2** dienen beide van het data type **Player** te zijn, welke we eerder hebben aangemaakt.
- c. Alle fields dienen beide **private** te zijn.
- d. Maak voor alle fields de bijbehorende getters en setters.
- e. De class heeft geen custom constructor(s), we maken gebruik van de default constructor, die Java zelf maakt bij het compileren.
- f. De class **Game** krijgt (naast de getters en setters) een methode **printScoreBoard**

Method **printScoreBoard**

De methode **printScoreBoard** print het scorebord naar console.

- a. De methode geeft niets terug en heeft geen parameters.
- b. De output van de methode is hieronder afgebeeld, de implementatie ervan moet jezelf bedenken. Let op maak gebruik van de fields van de class.

```
### Scoreboard of the game Dammen ###  
Player one Mark has a score of 21  
Player two Chantal has a score of 29
```

4. Class Assignment26

De **main** methode hebben we al aangemaakt in de Voorbereiding (1b). Nu gaan we de implementatie doen van de **main** methode.

- a. Instantieer van de class **Game**, een nieuw object met de naam checkers (dammen).
- b. Geef het object **checkers** de **titel** “Dammen”, gebruik hiervoor de bijbehorende setter.
- c. Instantieer van de class **Player**, een nieuw object met de naam player1.
- d. Geef het object player1 de naam “Mark”, gebruik hiervoor de bijbehorende setter.
- e. Zet de score van player1 op 21, gebruik hiervoor de bijbehorende setter.
- f. Instantieer van de class **Player**, een tweede nieuw object met de naam player2.
- g. Geef het object player2 de naam “Chantal”, gebruik hiervoor de bijbehorende setter.
- h. Zet de score van player2 op 29, gebruik hiervoor de bijbehorende setter.
- i. Zet de eerste speler voor het object **checkers**, gebruik hiervoor de bijbehorende setter.
- j. Idem voor de tweede speler.
- k. Print het score bord uit van het object **checkers**, of anders gezegd print het score bord uit van het spel checkers.

Extra:

- Instantieer een nieuw spel en geeft het spel een titel
- Instantieer twee nieuwe spelers en zet de namen en scores
- Print het score bord
- Merk op dat je dus meerdere spellen kunt maken met verschillende spelers, maar wel gebruik maakt van dezelfde classes, dat is 1 van de krachtige eigenschappen van OOP!

5. Opmerking

Na het maken van de opdracht zul je wellicht opmerken dat bepaalde getters niet gebruikt worden. Deze mogen verwijderd worden, maar kunnen handig blijken wanneer een bepaald field opgevraagd moet worden.

In deze implementatie van deze opdracht worden ze niet gebruikt, maar stel dat iemand anders gebruik wil maken van jou classes en een bepaald field wil opvragen.