

PXJavaAdvanced2017

Week 2: Opdrachten Java Generics

Github repository: <https://github.com/PXJavaAdvanced2017/week2-generics>

Opdracht 1:

Maak een abstract klasse **Player** met een member variabele *name*. Maak hiervan 3 afgeleide klassen: **BaseballPlayer**, **VolleybalPlayer** en **SoccerPlayer**.

Maak nu een klasse **Team**. Voorzie de volgende eigenschappen: *name*, *played* (=aantal wedstrijden gespeeld), *won* (= aantal wedstrijden gewonnen), *lost* (= aantal wedstrijden gewonnen), *tied* (=aantal wedstrijden gelijkgespeeld) en *members* (lijst van spelers, gebruik hiervoor een **ArrayList**). Voorzie getters.

In de **constructor** geef je enkel een naam mee.

Voorzie de methode **addPlayer** om een speler aan het team toe voegen en een methode **numberOfPlayers** om te vragen hoeveel spelers er in het team zitten.

Kan je spelers met een verschillend type in een team toevoegen? Test uit! Zorg ervoor dat dit niet (meer) mogelijk is.

Voorzie de methode **matchResult**(**Team** opponent, **int** ourScore, **int** theirScore). Deze methode zorgt ervoor dat voor het Team waarvoor de methode wordt aangeroepen en de opponent het aantal gespeelde, gewonnen, verloren en gelijkspel wedstrijden wordt opgehoogd afhankelijk van de waarden van ourScore en theirScore.

Kan je bovenstaande methode aanroepen met een team van volleybalspelers tegen een team van baseballspelers? Los dit eventueel op, zodat dat niet langer mogelijk is.

Voeg tenslotte een methode **ranking()** toe. Deze geeft een geheel getal terug waarbij een team 3 punten krijgt voor elke gewonnen wedstrijd en 1 punt voor geen wedstrijd met gelijkspel.

Maak nu een **League** klasse aan. Aan een competitie nemen verschillende teams van hetzelfde type deel. Voorzie de methode **addTeam()**.

Maak een methode **printTeams()** die alle teams met hun ranking afdrukt.

Opdracht 2:

Een goochelaar heeft een magische hoed. In de hoed kan hij een dier verstoppen: een duif of een konijn. De goochelaar kan in de hoed kijken: de eerste keer komt er niets uit. De volgende keer komt het konijn of de duif uit de hoed tevoorschijn. Let wel op: de goochelaar houdt er niet van om het dier te “casten”.

Opdracht 3:

Bekijk de code voor opdracht 3 in de github repository.

Pas de code aan op de volgende mogelijk manieren en test je aanpassingen telkens uit in de main-methode in de klasse TestWorkingPlace.

1. De WorkingPlace kan alle Motorized objecten herstellen (dus geen Bike)
2. De WorkingPlace kan alle Vehicle objecten herstellen (dus geen CircularSaw).
3. De WorkingPlace kan enkel Bike objecten herstellen.
4. De WorkingPlace kan enkel Motorized Vehicle objecten herstellen.

Maak de klasse WorkingPlace terug generiek zonder beperkingen. We passen nu de grenzen van de parameter in de methode getScore in de WorkingPlaceUtility klasse aan. Test dit telkens uit!

5. Zorg ervoor dat je de methode getScore uit WorkingPlaceUtility enkel kan oproepen voor een WorkingPlace die Bike objecten herstelt.
6. Zorg ervoor de je de methode getScore uit WorkingPlaceUtility enkel kan oproepen voor WorkingPlace die Vehicle objecten herstelt.
7. Zorg ervoor de je de methode getScore uit WorkingPlaceUtility enkel kan oproepen voor WorkingPlace objecten die Motorized objecten herstelt .
8. Zorg ervoor de je de methode getScore uit WorkingPlaceUtility enkel kan oproepen voor WorkingPlace objecten die Motorized Vehicle objecten herstelt.

Opdracht 4:

Maak een static methode **maakIntegerLijst** die een ArrayList van type Integer vult met 10 identieke getallen (dit getal geef je mee als parameter in de methode).

Voorzie de static methode **printIntegerLijst** die de waarden in de ArrayList kan afdrukken.

Maak een main-methode waarin je de methode maakIntegerLijst en printIntegerLijst aanroept.

Maak nu een nieuwe static methode **maakLijst** die hetzelfde kan als maakIntegerLijst, maar dan voor eender welk type.

Voorzie de static methode **printLijst** die de waarden uit eender welk type lijst kan afdrukken.

Voor de uitbreiding voeg je extra elementen toe in je lijst.

Uitbreiding 1: Voeg ook een generische methode toe om de waarden in de lijst om te keren.

Een voorbeeld: de lijst met volgend resultaat van printLijst

abc abc beer abc olifant abc kanarie abc selfie abc abc abc abc abc

geeft na omkeren het volgende resultaat:

abc abc abc abc abc abc kanarie abc selfie olifant abc beer abc abc

Uitbreiding 2: Implementeer een generische methode om de waarden in de lijst te sorteren door gebruik te maken van selection sort. https://en.wikipedia.org/wiki/Selection_sort