

OOP1 – oefenopdracht 2

Gooi Dobbelsteen deel 2

Inleiding

Dit is het vervolg op oefenopdracht 1: Gooi Dobbelsteen deel 1. Ja gaat dus verder werken in het project dat je daar hebt gemaakt.

Het UML Class Diagram van de class Dobbelsteen is voor deel 2 uitgebreid met *encapsulation*. In deel 1 had elke dobbelsteen zijn eigen `getalGenerator`. We kunnen echter voor elke dobbelsteen dezelfde `getalGenerator` gebruiken. Daarom is `getalGenerator` in deel 2 een class variable (Zie §9.7 van Liang).

Het UML Class Diagram ziet er nu als volgt uit:

Dobbelsteen
<code>-worp: int</code> <code>-getalGenerator: Random</code>
<code>+gooi(): void</code> <code>+print(): void</code> <code>+getWorp(): int</code>

Stappenplan

1. Open je Java project genaamd “GooiDobbelsteen” in IntelliJ.
2. Verbeter je class `Dobbelsteen` op drie manieren:
 - a. Door *encapsulation* toe te passen (zie §9.9 van Liang). Gebruik op de juiste plekken `public` en `private`.
 - b. Door van `getalGenerator` een class variable te maken.
 - c. Door in de methode `gooi()` het gebruik van `getalGenerator` te wijzigen naar `Dobbelsteen.getalGenerator`.
3. Test of je `main()` methode van class `GooiDobbelsteen` nog correct werkt.

Nu je een goed werkende class `Dobbelsteen` hebt, kun je hiermee ook andere functies implementeren. In stap 4 en 5 gebruik je de class om twee dobbelstenen te gooien en te printen, totdat met beide dobbelstenen dezelfde worp is gegooid.

4. Maak in de class `Main` een static methode met de volgende signatuur: `public static void gooiTotDubbel(Dobbelsteen dobbelsteen1, Dobbelsteen dobbelsteen2)`. Deze methode gooit en print twee dobbelstenen, totdat er dubbel is gegooid.
5. Maak in je `main()` een tweede dobbelsteen aan met de default constructor. Roep vervolgens de methode `gooiTotDubbel()` aan met je de twee dobbelstenen.

Richtlijnen bij coderen (zie ook HBO-ICT code conventions)

- Zorg dat je naam en het doel van het programma bovenin staan (ICC #1).
- Gebruik de juiste inspringing (*indentation*) bij de lay-out (ICC #2) .
- Let op juist gebruik hoofdletters en kleine letters (ICC #3).
- Gebruik goede namen (ICC #4).
- Voeg waar nodig commentaar toe die inzicht geven in je code (ICC#7).
- Gebruik javadoc tag `@author`.