

W4 Patterns - QuartoFactory opgave



Pas het **Static Factory-method pattern** toe om volgende opdracht uit te werken:

1. Bekijk de klasse **QuartoBlock**. Deze klasse vertegenwoordigt één blokje van het populaire denkspel Quarto. De klasse is immutable gemaakt door het gebruik van final. De 4 attributen bepalen het uitzicht van het blokje en zijn allemaal van een enumtype. Merk ook op dat het uniek zijn van een blokje afhangt van al deze 4 eigenschappen (zie **equals** en **hashCode**).
Het project bevat een test, die momenteel faalt.
2. Zorg ervoor dat de test slaagt. Je werkt hiervoor exclusief in de klasse **QuartoFactory**. Deze klasse is verantwoordelijk voor het aanmaken van Quartoblokjes voor een online game. Om onze game efficient te maken, maken we niet elke keer een nieuw blokje aan als er eentje nodig is. Als een bepaald type blokje al eens is aangemaakt, hergebruiken we dit.
 - Voorzie 2 static attributen:
 - een **HashSet** om de types blokjes te verzamelen. (Waarom een HashSet?)
 - een teller die het aantal blokjes bijhoudt dat we geleverd hebben (ook als een bestaand blokje werd teruggegeven gaat de teller omhoog).
 - Zorg ervoor dat de klasse **QuartoFactory** niet kan geïnstantieerd worden. (Hoe doe je dat ook weer?)
 - Implementeer de methode **createBlock**:
Hiermee wordt een blokje aangemaakt. Controleer eerst of er al een blokje met dezelfde 4 attributen in de set zit. Indien dat het geval is dan maak je geen nieuw aan, maar je retourneert een referentie naar het bestaande blokje.
Indien het type blokje nog niet werd aangemaakt, dan creëer je eerst een nieuw en bewaar je het in de HashSet.
 - Implementeer de methode **countBlocks** die het aantal blokjes dat de factory leverde teruggeeft.
3. Voer de test uit en pas QuartoFactory aan tot de test slaagt.

Uitbreiding: verwijderen Factory klasse

4. De factory zorgt ervoor dat een bepaald blokje slechts één keer aangemaakt wordt, maar je kan nog altijd dubbele blokjes aanmaken via de QuartoBlock constructor. Maak nu QuartoBlock zélf de factory: kopieer de static methoden en variabelen naar QuartoBlock en zorg dat de constructor niet meer van buitenaf aangeroepen kan worden.
5. Pas de factory aanroepen aan in de test, en zorg dat de test terug slaagt.