

1 Grundlegendes

Im Rahmen dieser Live-Aufgabe soll mit Hilfe des Observer-Patterns ein sehr stark vereinfachtes Roulette-Spiel abgebildet und in Java umgesetzt werden:

2 Erstellen Sie folgende Klasse

2.1 Roulette

Entspricht dem Roulette-Spiel.

Diese Klasse enthält die Methode „int nextSpin()“ welche eine Spielrunde simuliert. Diese Methode liefert eine Zufallszahl von 0 bis 36 (jeweils inklusive) zurück, welche die Zahl repräsentiert, auf der die Kugel liegen blieb.

3 Observer-Pattern

Es gibt mehrere Parteien, die sich für die Ergebnisse der einzelnen Durchgänge interessieren. Diese sollen über das Resultat eines jeden Spins jeweils per Observer-Pattern informiert werden.

3.1 Spieler

Es gibt einen Spieler, der immer auf die Nummer 7 setzt und sich freut wenn diese Zahl getroffen wird. Dieser Freude verleiht er mit einer Ausgabe auf „System.out“ Ausdruck, sobald er darüber informiert wurde.

3.2 Das Haus

Das Haus möchte sicher stellen, dass die Roulette-Räder gleichverteilte Resultate erzielen und keine Zahlen gehäuft auftreten.

Aus diesem Grund wird die Häufigkeit des Auftretens der einzelnen Zahlen mit protokolliert.

3.3 Programmablauf

Instantiieren Sie (mit Hilfe einer main-Methode oder eines JUnit-Tests) ein Roulette, registrieren Sie die beiden Listener „Spieler“ und „Haus“ und führen Sie 100 Spiel-Runden durch.

Anschließend geben Sie das Ergebnis der Protokollierung des „Hauses“ aus. Dies könnte z.B. in einer einfachen Auflistung in folgender Form erfolgen.

0: {Häufigkeit}

1: {Häufigkeit}

usw.

Hinweise

Tipps bezüglich der Datenstruktur

Am einfachsten scheint die Speicherung der Häufigkeit in Form eines Arrays zu sein, bei der die erzielte Zahl dem Index des Arrays entspricht. Andere Datenstrukturen sind aber natürlich ebenfalls möglich.

Hinweis: Zufallszahlen

Am einfachsten lassen sich Zufallszahlen mit Hilfe der statischen Methode „random“ aus der Klasse „java.lang.Math“ generieren. Diese Methode liefert einen pseudozufälligen double-Wert zwischen 0.0 (inklusive) und 1.0 (exklusiv) zurück.

Unit-Tests

Erstellen Sie ausreichend Unit-Tests mit den entsprechenden Assertions, um die Korrektheit Ihrer Lösung beweisen zu können.

Observer-Pattern nutzen

Die Lösung ist nur dann korrekt, wenn sie mit Hilfe des Observer-Patterns umgesetzt wurde. Alternative Lösungen ohne das Observer-Pattern sind nicht gestattet.

Encoding

Bitte achten Sie darauf, dass Ihre Lösung im UTF-8-Encoding abgegeben wird.

Abgabe

Die Abgabe der Lösung erfolgt über das Relax. Bitte laden Sie Ihre Lösung (nur den „src“-Folder) in einem ZIP-File hoch.