

1. Angekündigter Test im Fach Informatik
am 27.11.2020

Klassen und deren Implementierung

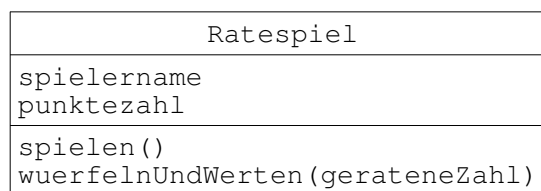
Szenario: In einem Computerspiel sollen die Spieler das Leben von Einhörnern in Einhornwelten steuern. Für jedes Einhorn muss gespeichert werden, wie glücklich es ist und wie fit es ist. Damit ein Einhorn glücklicher wird, kann man es spielen lassen (was gespielt wird und wie lange das dauert, wird zufällig bestimmt). Allerdings ist das Spielen für Einhörner anstrengend, danach sind sie erschöpft. Deshalb muss ein Einhorn für eine bestimmte Zeit schlafen gelegt werden, damit es wieder fit wird – je länger man es schlafen lässt, desto fitter wird es.

Aufgabe 1: Zeichne ein Klassendiagramm für die Klasse `Einhorn`.

6 BE

Szenario: Eines der Spiele, die man für sein Einhorn spielen kann, ist ein primitives Ratespiel für einen Spieler. Das Spiel umfasst 10 Runden. Am Anfang der ersten Runde hat der Spieler zehn Punkte. In jeder Runde ermittelt der Computer zufällig eine Zahl: entweder eine 0 oder eine 1. Der Spieler rät zu Beginn jeder Runde, ob die 0 oder die 1 „gewürfelt“ wird. Hat er richtig geraten, wird seine Punktzahl verdoppelt, hat er falsch geraten, bekommt er einen Punkt abgezogen.

Das folgende Klassendiagramm wurde für das Spiel entworfen:



Der Name eines Spielers soll bei der Erzeugung eines Objekts der Klasse `Ratespiel` – als Parameter des Konstruktors – angegeben werden. Die Punktezahll hingegen muss bei der Erzeugung eines Objekts auf 10 gesetzt werden – sie darf also kein Parameter des Konstruktors sein.

Die `spielen`-Methode führt das gesamte Spiel mit allen zehn Runden durch. Sie lässt in jeder Runde den Spieler die Zahl, die er raten möchte, eingeben und ruft mit dieser Zahl als Parameter dann die `wuerfelnUndWerten`-Methode auf. Letztere Methode ermittelt die Zufallszahl und aktualisiert die Punktezahll des Spielers entsprechend. Am Schluss jeder Runde wird eine Zwischenstandsmeldung ausgegeben (das gehört nicht mehr zur Aufgabe der `wuerfelnUndWerten`-Methode). Nach der zehnten Runde wird der Spielstand zum Schluss angezeigt.

Hinweis zur Implementierung in Kotlin: Zum Ermitteln einer Zufallszahl gibt es in der Programmiersprache Kotlin (wie wir wissen) das Objekt mit dem Bezeichner `Random`. Es steht immer zur Verfügung, wenn man es mit einer entsprechenden Importanweisung bereitstellt – wir setzen für die Aufgaben unten voraus, dass das bereits passiert ist und wir das Objekt also einfach verwenden können. Will man dieses Objekt zufällig eine 0 oder 1 ermitteln lassen, ruft man dessen `nextInt`-Methode mit dem Parameter 2 auf.

Aufgabe 2: Zeichne ein Aktivitätsdiagramm, das darstellt, wie die `spielen`-Methode arbeiten muss. Dabei brauchst du die einzelnen Arbeitsschritte der `wuerfelnUndWerten`-Methode (die ja von der `spielen`-Methode aufgerufen wird) nicht im Detail anzugeben – das folgt dann in der nächsten Aufgabe.

7,5 BE

Aufgabe 3: Gib eine Implementierung der Klasse `Ratespiel` in der Programmiersprache Kotlin an. Lasse dabei aber die `spielen`-Methode weg. Berücksichtige alle Anforderungen aus dem Szenario, insbesondere hinsichtlich der Deklaration der Attribute.

8,5 BE

Viel Erfolg!