

Praktikum Programmierertechnik (Technische Informatik)

SS 2016, Hochschule für Angewandte Wissenschaften (HAW), Hamburg
Prof. Dr. Philipp Jenke, Prof. Dr. Axel Schmolitzky, Norbert Kasperczyk-Borgmann



Aufgabenblatt 4: Objektorientierung

Aufgabe 4.1: Rennauto

Schwerpunkte: Klasse mit Zustand

Aufgabe: Schreiben Sie eine Klasse Rennauto. Jedes Rennauto hat folgende Eigenschaften:

- Name der Fahrerin
- Fahrzeugtyp
- Maximalgeschwindigkeit (Veränderung der gefahrenen Strecke pro Zeitschritt)
- bisher gefahrene Strecke ab dem Start

Schreiben Sie einen Konstruktor, der die Objektvariablen initialisiert (Name, Typ und Maximalgeschwindigkeit sind Parameter). Das Rennauto hat außerdem eine Methode `fahren()`. Beim Aufruf der Methode bewegt sich das Rennauto zunächst einen Schritt (Zeitschritt = 1) mit Maximalgeschwindigkeit weiter. Eine Fahrerin schafft es aber nicht immer, die Maximalgeschwindigkeit auszureizen. Daher wird die tatsächlich gefahrene Distanz in einem Zeitschritt mit einem Zufallsfaktor skaliert (`Math.random()` liefert eine Zufallszahl aus $[0,1]$). Die Klasse benötigt außerdem eine Methode `ausgeben()`, die den aktuellen Zustand der Klasse auf der Konsole ausgibt. Testen Sie Ihre Implementierung, z.B. indem Sie eine Instanz von Rennauto erstellen, einige Schritte fahren und jeweils den Zustand des Objektes ausgeben.

Aufgabe 4.2: Rennen

Schwerpunkte: Klasse mit Referenzen auf Objekte anderer Klassen

Aufgabe: Schreiben Sie eine Klasse Rennen. Ein Rennen hat folgende Eigenschaften:

- Anzahl der Rennautos im Rennen
- List der Rennautos in einem Array
- Streckenlänge

Implementieren Sie einen Konstruktor. Initialisieren Sie darin das Array mit der Länge 1. Die Klasse soll außerdem eine Methode `addRennauto(Rennauto)` haben, die dem Rennen ein zusätzliches Rennauto hinzufügt. Falls im Array kein Platz mehr ist, soll die Arraylänge verdoppelt werden. Die bereits vorhandenen Rennautos dürfen dabei natürlich nicht verloren gehen. Ein Rennen hat außerdem drei Methoden, um ein Rennen zu steuern:

- `Rennauto ermittleSieger()`: Liefert das Rennauto, das bereits die gesamte Streckenlänge zurückgelegt hat. Falls dies für kein Rennauto zutrifft, liefert sie `null`. Falls mehrere Rennautos bereits im Ziel sind, liefert sie ein beliebiges davon.
- `void schritt()`: Alle Teilnehmerinnen des Rennens bewegen sich einen Schritt weiter.
- `void durchfueren`: Es wird so lange `schritt()` aufgerufen, bis ein Sieger des Rennens gefunden wurde.

Aufgabe 4.3: Wettbüro

Schwerpunkte: Algorithmenentwurf, Klasse ausschließlich zur Datenhaltung

Aufgabe: Schreiben Sie eine Klasse Wettbuero. Ein Wettbuero hat folgende Eigenschaften:

- Referenz auf ein Rennen
- Liste (Array) von Wetten
- (fester) Faktor mit dem Wetteinsätze bei Gewinn multipliziert.

Wetten werden in einer eigenen Klasse `Wette` implementiert. Mit der Methode `wetteAnnehmen(String fahrerin, int wetteEinsatz, String spieler)` nimmt das Wettbüro eine Wette an und speichert sie intern als `Wette`-Objekt im Array. Das Wettbüro hat außerdem eine Methode `auswerten()`, die überprüft, ob einer (oder mehrere) der Spieler gewonnen hat. In dem Fall berechnet sie den Gewinn und gibt den Spieler und den Gewinn aus.

Zum Überblick finden Sie hier ein Klassendiagramm des Systems:

