## Rennschnecke

Erstellen Sie eine Klasse "Rennschnecke", die folgende Eigenschaften besitzt:

* Name
* Rasse
* Maximalgeschwindigkeit (Veränderung der Strecke pro Schritt)
* Zurückgelegte Strecke

Implementieren Sie einen Konstruktor für die Klasse "Rennschnecke", der den Instanzvariablen beim Erstellen einer neuen Instanz (Objekt zu dieser Klasse) Werte zuweist.

Erstellen Sie in der Klasse "Rennschnecke" die Methode "krieche( )", welche die Schnecke abhängig von ihrer Maximalgeschwindigkeit eine zufällige Strecke weiter bewegt. Dabei soll die Schnecke eine zufällige Strecke größer null und kleiner ihrer Maximalgeschwindigkeit zurücklegen. Verwenden Sie als Zeiteinheit 1.

Tipp: Schauen Sie sich die Methode Math.random0 aus der Java API an.

Implementieren Sie in der Klasse "Rennschnecke" die Methode "public String toString0", die die Daten der Schnecke als String zurückgibt.

Testen Sie Ihre Klasse, indem Sie probehalber ein Rennschneckenobjekt erzeugen und seine Daten auf der Konsole ausgeben. Verwenden Sie zum Ausgeben der Daten die toString0 Methode der Rennschnecke.

## Rennen

Erstellen Sie eine Klasse "Rennen", die folgende Eigenschaften besitzt:

* Namen
* Anzahl der teilnehmenden Schnecken

Die teilnehmenden Schnecken selbst, entweder als Array für eine einfachere Umsetzung mit konstanter Teilnehmerzahl oder als ArrayList für eine schöner Umsetzung mit variabler Teilnehmerzahl

Verwenden Sie als Datentyp für die teilnehmenden Schnecken ArrayList< Rennschnecke>, Beim Erstellen verwenden Sie new ArrayList «Rennschnecke> 0-

Legen Sie in der Klasse "Rennen" die Methode "void addRennschnecke(Rennschnecke neueSchnecke)" an, um dem Rennen eine Schnecke hinzuzufügen.

Legen Sie in der Klasse "Rennen" die Methode "void removeRennschnecke(String name)" an, um eine Schnecke aus dem Rennen zu entfernen (optional).

Implementieren Sie in der Klasse "Rennen" die Methode "public String toString0", die die Daten des Rennens mit return als String zurückgibt. Um die Daten der beteiligten Schnecken zurückzugeben, verwenden Sie deren toString® Funktion.

Testen Sie Ihre Klasse vom Hauptprogramm aus!

Legen Sie in der Klasse "Rennen" die Methode "Rennschnecke ermittleGewinnerO" an, die null zurückliefert, wenn noch keine der teilnehmenden Schnecken das Ziel erreicht hat und anderenfalls die Gewinnerschnecke zurückgibt.

Legen Sie in der Klasse "Rennen" die Methode "void lasseSchneckenKriechen0" an, die alle teilnehmenden Schnecken einmal kriechen lässt.

Legen Sie in der Klasse "Rennen" die Methode "void durchfuehrenO" an, die so lange lasseSchneckenKriechen0 aufruft, bis eine der Schnecken das Ziel erreicht hat. Ob eine Schnecke im Ziel angekommen ist, können Sie mit der Methode ermittleGewinner0 feststellen.

Wettbüro

Erstellen Sie eine Klasse Wettbuero mit folgenden Eigenschaften:

Das Wettbüro weiß, für welches Rennen es seine Wetten entgegennimmt.

Es verfügt über eine Liste (z.B. eine ArrayList) von angenommenen Wetten.

• Es hat einen festen Faktor, mit welchem Wetteinsätze bei einem Gewinn multipliziert werden.

Legen Sie in der Klasse "Wettbuero" die Methode "wetteAnnehmen (String schneckenName, int wettEinsatz, String spieler)" an, die eine Wette entgegennimmt.

Die Wette ist bezogen auf eine Schnecke für das Rennen, das von dem Büro bearbeitet wird. Um die einzelnen Wetten speichern zu können, sollten ihre Daten in eigenen Objekten der Klasse "Wette" gespeichert werden. Erstellen Sie diese Klasse selbst.

Implementieren Sie in der Klasse "Wettbuero" die Methode "rennenDurchfuehrenO", die das betreute Rennen durchführt.

Legen Sie in der Klasse "Wettbuero" die Methode "toString0" an, die die Daten des Wettbüros, die Daten des Rennens sowie sämtliche abgeschlossenen Wetten als String zurückgibt

Testen Sie Ihr Programm!