

SEW 2 - Übung: Fahrrad (Teil 1)

Ziel

Erstellen zweier Klassen für Fahrräder und Radfahrer.

Lernziele

- Anfertigen von *UML*-Diagrammen.
- Definieren von Klassen.
- Konstruktoren(-überladung), *Getter* und *Setter*.
- Einfache Methoden und `toString`-Überschreibung.
- Statische Variablen.
- Einstieg in die *testgetriebene Entwicklung*.

Abgaberichtlinien

- Ihr *UML*-Diagramm.
- Ihre implementierte Lösung als `.java`-Dateien. **Vergessen Sie dabei nicht auf Kommentare und Kommentarkopf!**

Aufgabe

Zu implementieren sind zwei Klassen zur *objekt-orientierten* Abbildung von Fahrrädern (`Bicycle`) und Radfahrern (`Cyclist`).

Bevor Sie mit der Implementierung beginnen, sind *UML*-Diagramme für beide Klassen zu erstellen. Sämtliche dafür benötigten Informationen finden Sie in den Beschreibungen unterhalb. **Besprechen Sie das Diagramm mit dem/der LehrerIn bevor Sie mit der Implementierung beginnen!**

- Klasse `Cyclist`:
 - Ein Radfahrer besteht aus einem Namen (`name`) und einem Team (`team`).
 - Auf beide Eigenschaften kann über *Getter* und *Setter* zugegriffen werden.
 - * Wird versucht das Team auf einen leeren Text oder `null` zu setzen, so soll der Standard-Text `None` verwendet werden.
 - Die Klasse soll zwei Konstruktoren zur Verfügung stellen: Einen in dem nur der Name und einen in dem beides gesetzt werden kann.
 - * Verwenden Sie Konstruktoren-Überladung um Code-Verdopplung zu vermeiden!
 - * Auch hier gelten die gleichen Regeln für leere Teams (s.o.)!
 - Für eine schöne Ausgabe soll natürlich auch die `toString`-Methode überschrieben werden.
- Klasse `Bicycle`

- Jedes Fahrrad hat eine eindeutige Rahmennummer (**frameId**), die im Konstruktor **automatisch** (kein Parameter!) gesetzt wird und nachträglich **nur gelesen** werden kann.
- Weiters hat ein Fahrrad eine Marke und einen Typ, die über *Getter* und *Setter* gelesen und gesetzt werden können.
 - * Gültige Typen sind **City**, **Mountain**, **Road** und **Trekking**. Bei anderen Eingaben ist standardmäßig **City** zu verwenden.
 - * Denken Sie daran für das Vergleichen von Texten **equals** zu verwenden!
- Zusätzlich besitzt ein Fahrrad einen Kilometerstand (**mileage**), der zu Beginn 0 ist und nur gelesen werden kann. Der Kilometerstand kann einzig mit der Methode **ride** um einen positiven Wert erhöht werden.
- Daraus ergibt sich auch der einzige Konstruktor, der lediglich Marke und Typ (beachten Sie die o.g. Regeln!) akzeptiert.
- Auch das Fahrrad überschreibt die **toString**-Methode.

Für die Implementierung werden Ihnen - wie mittlerweile gewohnt - zwei Testtreiber zur Verfügung gestellt.