

SEW 2 - Übung: Fahrrad (Teil II - IV)

Ziel

Implementieren von *uni-* und *bidirektionalen Assoziationen* zwischen zwei Klassen.

Lernziele

- Uni- und bidirektionale Assoziationen.
- Arbeiten mit *Arrays* von Objekten.
- Testgetriebene Entwicklung.

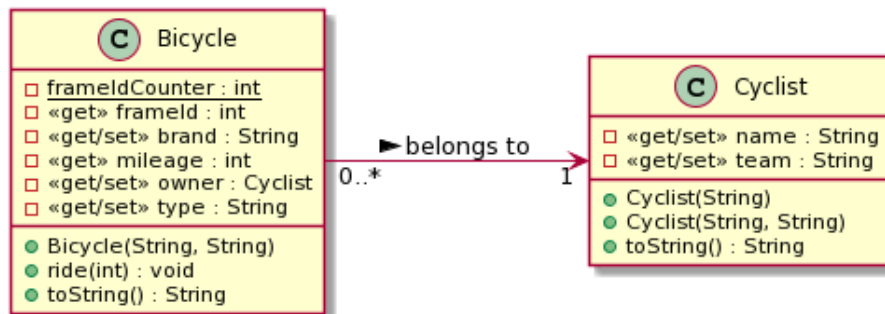
Abgabelinien

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

Aufgabe

Nachdem Sie sich im ersten Teil dieser Übung mit dem Erstellen von Klassen für Fahrräder und RadfahrerInnen befasst haben, wird es nun Zeit diese zu verbinden. Arbeiten Sie die folgenden Teile Schritt für Schritt durch: Für jeden Teil gibt es einen eigenen *Testtreiber* - stellen Sie nach jedem Teil sicher, dass auch die vorherigen Tests noch funktionieren!

Teil II: Von Rad Zu BesitzerIn

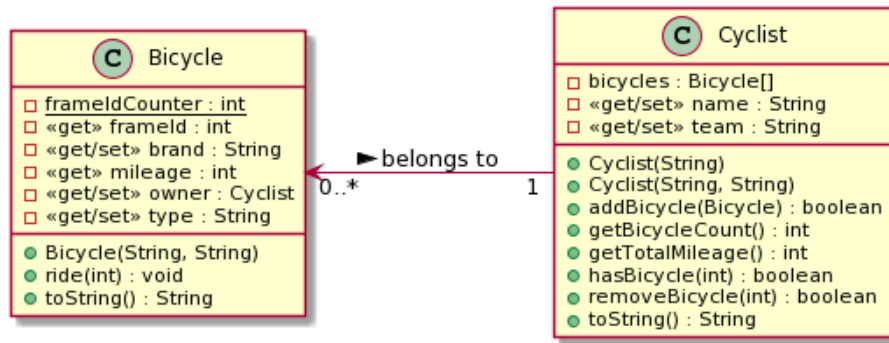


Fügen Sie den Testtreiber `BicycleAssociationTest` Ihrem Projekt hinzu.

Bilden Sie nun die oben gezeigte Beziehung zwischen Fahrrädern und FahrerIn - *0 oder mehr Räder gehören einer/einem FahrerIn* - ab, indem Sie das Fahrrad durch eine private Eigenschaft für den/die BesitzerIn erweitern, die durch öffentliche *Getter/Setter* abgerufen bzw. gesetzt werden kann. Beachten Sie, dass bei einem Fahrerwechsel der Kilometerstand zurückgesetzt wird. Weiters ändert sich die

Standardausgabe (i.e. `toString`) in Abhängigkeit davon, ob ein `BesitzerIn` vorhanden ist oder nicht (i.e. Überprüfung aus `null`).

Teil III: Von *BesitzerIn* Zu *Rädern*



Fügen Sie den Testtreiber `CyclistAssociationTest` Ihrem Projekt hinzu.

Jetzt ist die Beziehung zwischen `FahrerIn` und mehreren Rädern zu implementieren. Dafür wird ein `FahrerIn` durch ein privates *Array* von Rädern erweitert. Für diese Übung können Sie eine fixe Länge von 3 verwenden.

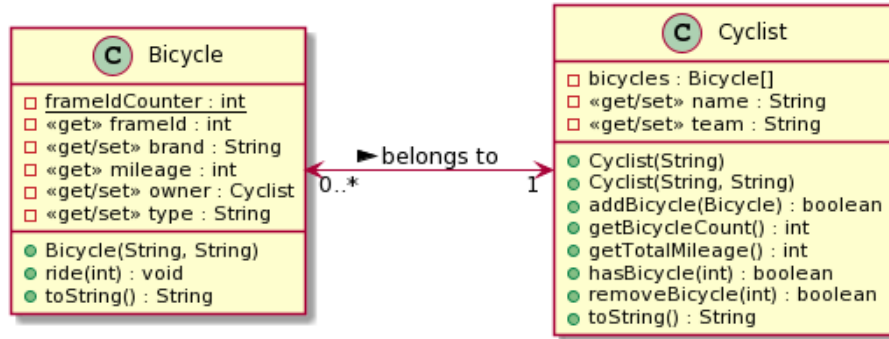
Nachdem weder *Getter* noch *Setter* für die Räder zur Verfügung stehen, sind folgende Methoden für die Verwaltung (in folgender empfohlener Reihenfolge) zu implementieren:

- `hasBicycle` geht das *Array* durch und überprüft ob ein Rad mit der übergebenen Rahmennummer vorkommt. Vergessen Sie nicht auf `null` zu überprüfen, da es sonst beim Aufruf der `getFrameId` zu einem Fehler kommt!
- `addBicycle` fügt das übergebene Rad dem *Array* hinzu, sofern es noch nicht enthalten ist. Gehen Sie dafür alle Positionen durch. Sobald eine Position `null` ist (i.e. ein freier Platz verfügbar ist), können Sie das Rad dort einfügen. Der Erfolg des Hinzufügens ist als boolescher Wert zurückzugeben.
- `removeBicycle` entfernt ein Fahrrad mit einer bestimmten Rahmennummer. Hierbei ist die Implementierung analog zur `hasBicycle`-Methode, wobei die gefundene Position im *Array* auf `null` gesetzt wird. Auch hier soll der Erfolg der Operation als boolescher Wert zurückgegeben werden.

Desweiteren sind folgende kleine Methoden zu implementieren:

- `getBicycleCount` gibt die Anzahl der verwalteten Fahrräder zurück.
- `getTotalMileage` gibt die Gesamtanzahl der mit allen im Besitz stehenden Rädern gefahrenen Kilometer zurück.
- Die `toString`-Methode ist um die Anzahl der Räder zu erweitern - sofern mindestens eines besessen wird.

Teil IV: Beide Richtungen



Fügen Sie den Testtreiber `BidirectionalAssociationTest` Ihrem Projekt hinzu. Nachdem die letzte Aufgabe nicht ganz unkompliziert ist, macht es Sinn den funktionierenden Zwischenstand zwischenspeichern.

Jetzt fügen wir die Teile zusammen, so dass man zwischen beiden Klassen navigieren kann:

- Der *Setter* des Fahrrads soll sich darum kümmern, dass das Fahrrad auch in *Array* des FahrerIn-Objekts landet (bzw. daraus entfernt wird).
 - Verwenden Sie eine Kombination aus `add-`, `has-` und `removeBicycle`.
 - Achten Sie auf `null`-Werte - sowohl die/den VorbesitzerIn als auch die/den NachfolgerIn könnte es nicht geben.
 - Wir müssen nur dann handeln, wenn Vor- und NachbesitzerIn unterschiedliche Objekte sind!
 - Ein möglicher Weg wäre:
 1. Aufhebung der alten Beziehung.
 2. Erstellung der neuen Zuordnung von Rad zu FahrerIn.
 3. Erstellung der neuen Zuordnung von FahrerIn zu Rad. Beachten Sie hier das Fahrrad-Limit!
- Die Methoden zum Hinzufügen bzw. Entfernen von Fahrrädern sollen den *Setter* im `Bicycle`-Objekt aufrufen, um den Fahrerwechsel auch in den Rädern abzubilden.
 - Wenn ein neues Rad hinzugefügt wird, muss auch dessen *Setter* aufgerufen werden - allerdings nur, wenn beim Rad einE andereR BesitzerIn eingetragen ist!
 - Beim Entfernen von Rädern kann deren BesitzerIn einfach auf `null` gesetzt werden.