

Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog (<https://seblog.cs.uni-kassel.de/ws1819/programming-methodologies/>) und aus den Übungen zu berücksichtigen.

Abgaben per Mail werden nicht akzeptiert.

Die Abgabefrist endet am Freitag der folgenden Woche nach erscheinen dieses Übungsblattes um 13 Uhr.

Aufgabe 1 - Implementierung des (unvollständigen) Klassendiagramms zu Live Risk (60P)

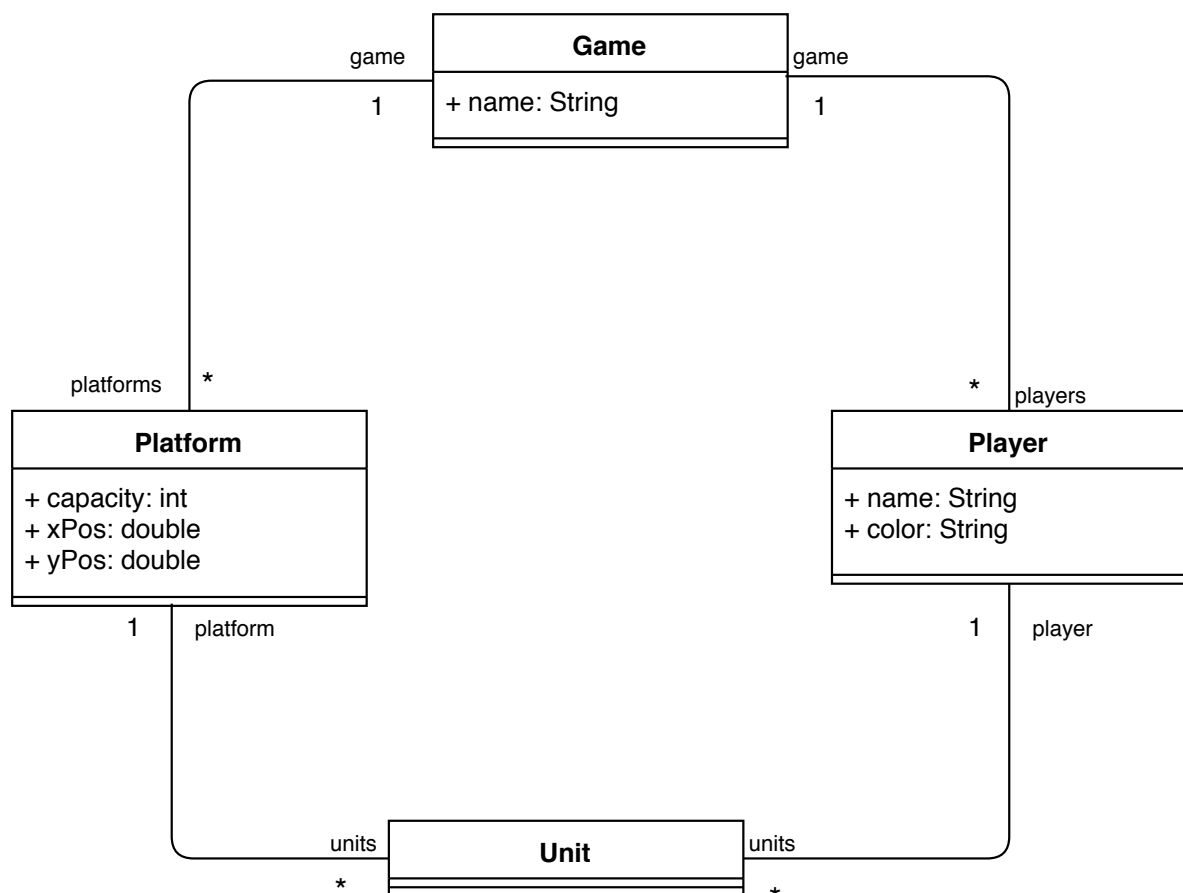


Abbildung 1: LiveRisk -Klassendiagramm

Gegeben ist das in Abbildung 1 dargestellte Klassendiagramm. Implementieren Sie das Klassendiagramm in Java.

Folgende Vorgehensweise wird vorgeschlagen:

1. Erstellen Sie für jede Klasse im Diagramm eine 'public class' in einer separaten .java-Datei.
2. Fügen Sie den Klassen die entsprechenden Attribute hinzu. Achten Sie auf Verkapselung der Attribute durch getter/setter!
3. Fügen Sie den Klassen die entsprechenden Methoden hinzu - eine Implementierung der Methodenrumpfe ist nicht gefordert.
4. Fügen Sie den Klassen keine weiteren Konstruktoren hinzu.
5. Implementieren Sie die Assoziationen und stellen Sie referenzielle Integrität sicher: Dazu fügen Sie den Klassen für zu-1-Assoziationen die Zugriffsmethoden set/get/with, für zu-n-Assoziationen die Zugriffsmethoden add<Role>/remove<Role>/get<Role>/with<Role> hinzu und wählen eine geeignete Containerklasse (z. B. `HashSet`). Sorgen Sie dafür, dass bei bidirektionalen Assoziationen die Rückrichtung automatisch mitgesetzt wird. Die implementierten Methoden besitzen nur die notwendigen Klassenparameter.

Hinweis: Bitte als Package `de.uniks.ws1819.liverisk.model` verwenden.

Aufgabe 2 - JUnit Tests (40P)

Schreiben Sie JUnit Tests, die die referenzielle Integrität der folgenden bidirektionalen Assoziationen testen:

- standsOn (Unit - Plattform * - 1)
- units (Player - Unit 1 - *)
- players (Game - Player 1 - *)

Es soll sowohl das Setzen/Hinzufügen als auch das Löschen von Elementen überprüft werden. Schreiben Sie Tests die **beide** Richtungen der Assoziationen testen. Die Tests sind in einer eigenen Klasse zu implementieren. **Hinweis:** Weil die Tests zum Löschen eines Elements voraussetzen, dass das Element vorher hinzugefügt wurde, können diese in einem einzigen Test zusammengefasst werden. Sie benötigen also $3 \text{ (Assoziationen)} * 2 \text{ (Rollen)} = 6$ Tests. Die Tests sollen in einer Testklasse `TestModel.java` erstellt werden. Bitte als Package `de.uniks.ws1819.liverisk.model` verwenden. Nutzen Sie zur Verifizierung ihrer Tests passende assert-Methoden. Testklassen gehören in den Test-Sourcefolder

Bsp:



- `Assert.assertTrue(object.trueExpected());` bzw.
- `Assert.assertFalse(object.falseExpected());`
- `Assert.assertEquals(3, object.getSize());`