

**Hausaufgabe 3**

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/ws2122/programming-and-modelling/> zu berücksichtigen.

**Abgabefrist ist der 25.11.2021 - 23:59 Uhr**

## Abgabe

Wir benutzen für die Abgabe der Hausaufgaben Git. Jedes Repository ist nur für den Studierenden selbst sowie für die Betreuer und Korrektoren sichtbar.

Für die Hausaufgabe benötigt ihr ein **neues Repository**, welches über folgenden Link angelegt werden kann, falls nicht bereits geschehen:

<https://classroom.github.com/a/6NJVftz4>

**Nicht oder zu spät gepushte (Teil-)Abgaben werden mit 0 Punkten bewertet.**

## Vorbereitung

Zur Bearbeitung der Hausaufgabe sollte eine Entwicklungsumgebung verwendet werden. Wir empfehlen aufgrund der Nachvollziehbarkeit die Verwendung von VSCode (siehe Anhang). Die Abgabe muss als **lauffähiges** Projekt abgegeben werden.

**Abgaben, die nicht lauffähig sind, werden mit 0 Punkten bewertet!**

## Java

Wir verwenden für die Veranstaltung Java 17. Dieses könnt ihr unter folgendem Link herunterladen und installieren:

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

Wählt dort die für euer Betriebssystem passende Datei aus und installiert diese. Bei der Verwendung einer anderen Java-Version wird diese Hausaufgabe mit 0 Punkten bewertet.

## Aufgabe 1 - Klassendiagramm zu Java (14P)

In dieser Aufgabe implementieren wir ein vereinfachtes Klassendiagramm des Spiels "Nine men's morris". Hierzu ist das in Abbildung 1 dargestellte Klassendiagramm gegeben.

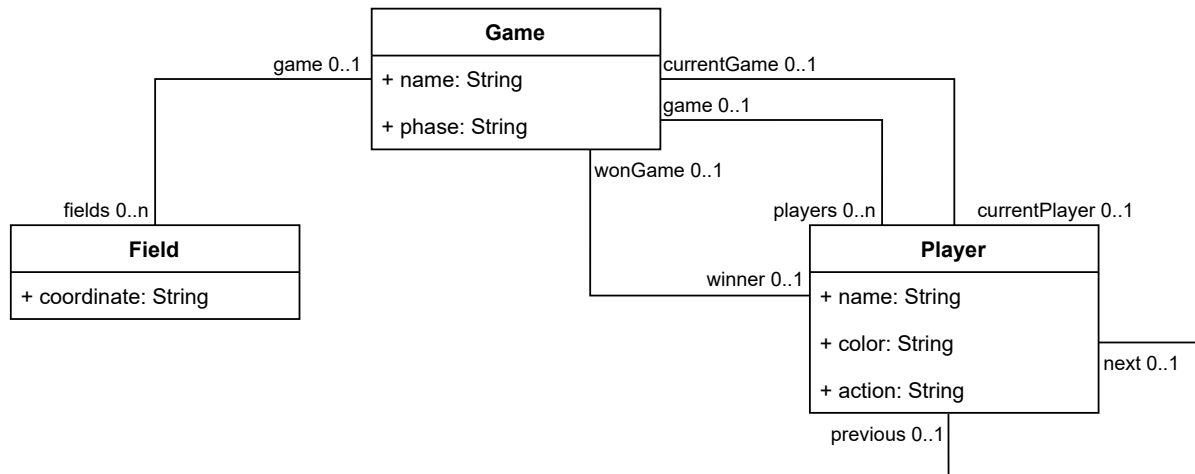


Abbildung 1: Nine men's morris-Klassendiagramm (vereinfacht)

Folgende Vorgehensweise wird vorgeschlagen:

1. Erstelle für jede Klasse im Diagramm eine Klasse in einer separaten .java-Datei unter dem Modul `src/main/java` im zu erstellenden Package `de.uniks.ws2122.ninemen.model`
2. Füge den Klassen **keine** Konstruktoren hinzu.
3. Füge den Klassen die entsprechenden Attribute hinzu. Achte dabei auf die Zugriffskapselung der Attribute mit den Zugriffsmethoden Set und Get aus der Vorlesung!
  - `<Class> set<Field>(<Type> <Variable>)` und `<Type> get<Field>()`
4. Implementiere die Assoziationen und stelle die referenzielle Integrität sicher. Dies verlangt folgende Punkte:
  - Füge den Klassen für 0..1-Assoziationen die Zugriffsmethoden `<Type> get<Field>()` und `<Class> set<Field>(<Class> <Variable>)` hinzu
  - Füge den Klassen für 0..n-Assoziationen die Zugriffsmethoden `List< <Class> > get<Field>()` sowie `<Class> with<Field>(<Class> <Variable>)` und `<Class> without<Field>(<Class> <Variable>)` hinzu
  - Wähle für die 0..n-Assoziationen eine geeignete Containerklasse (z.B. ArrayList)

Du kannst deine Implementierung mit dem Test aus Aufgabe 2 ausprobieren. Die Datei `.gitkeep` kannst du nach dem Implementieren der Klassen löschen.

**Bei der Bewertung wird vor allem auf die Projektstruktur, Zugriffskapselung sowie die korrekte Umsetzung der referenziellen Integrität geachtet.**

**Achtet darauf, das Repository der aktuellen Hausaufgabe zu verwenden.**

## Aufgabe 2 - Testing (8P)

In dieser Aufgabe soll der fehlende Test `testAssociationGameToField` unter `src/test/java` im Package `de.uniks.ws2122.nineman.model` implementiert werden.

Als Orientierung kannst du die bereits vorhandenen Tests verwenden, beachte, dass die Tests mit sinnvollen Asserts durchgeführt werden müssen. Folgende Methoden müssen aus der Game-Klasse

1. `withFields`
2. `withoutFields`
3. `getFields`

und aus der Klasse `Field`

1. `setCoordinate`
2. `getCoordinate`
3. `setGame`
4. `getGame`

getestet werden.

Committe und pushe die Änderungen an deinem Gradle-Projekt abschließend auf den `main`-Branch.

**Bei der Bewertung wird vor allem auf die korrekte Nutzung von JUnit sowie die korrekte Umsetzung der Objektdiagramme geachtet.**

**Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.**

## Anhang

Es folgt eine Auflistung hilfreicher Webseiten und weiterer Erklärungen zu den Themen dieser Hausaufgabe. Die Links sind als Startpunkt zur selbstständigen Recherche angedacht. Das Durcharbeiten der folgenden Quellen ist kein bewerteter Anteil der Hausaufgaben.

### VS Code

- Download: <https://code.visualstudio.com/>
- Erweiterungen: *Extension Pack for Java* und *Gradle Extension Pack*