Diese Aufgabe ist zu bearbeiten wenn Sie über eine **gerade** Matrikelnummer verfügen.

Die Aufgaben müssen einzeln bearbeitet und abgegeben werden. Die Abgabe muss bis **spätestens Donnerstag 29.11.2012 um 23:59 Uhr** über unser Hausaufgabenabgabesystem **http://seblog.cs.uni-kassel.de/pmws1213/** erfolgen. Die Abgabe ist nur als einzelne *.zip oder *.jar-Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden. Diese Hausaufgabe gibt **18** Punkte.

Hinweise zur Abgabe:

- Die Hausaufgabe soll als exportiertes Eclipse Projekt (*.zip, **nicht** den gesamten Workspace) abgeben werden. Das kann mit Hilfe der Eclipse Export Funktion durchgeführt werden. Ist das Projekt nicht korrekt exportiert, kann es bei der Korrektur nicht berücksichtigt werden (es bietet sich also an, den Import des exportierten Projektes auszuprobieren).

  WICHTIG Benennen Sie ihre Projekte für diese und alle zukünftigen Abgaben nach folgendem Schema:

  `PMWS1112_HA<a>_<Matrikelnummer>,`

  wobei <a> für die aktuelle Hausaufgabe und <b> für die Aufgabennummer steht. Beispiel:

  `PMWS1112_HA4_12345678.`

Vorbereitung

Sie benötigen das Fujaba Eclipse Plugin, welches Sie über die Update Site: (`http://www.se.eecs.uni-kassel.de/fileadmin/se/update/`) in Ihrer Eclipse IDE installieren können. Legen Sie eine Fujaba Model Datei an und erstellen Sie ein leeres Klassendiagramm. Das Vorgehen kann mit Hilfe des Screencasts der Übung nachvollzogen werden.

Aufgabe 1 - Modellierung mit Fujaba (12P)

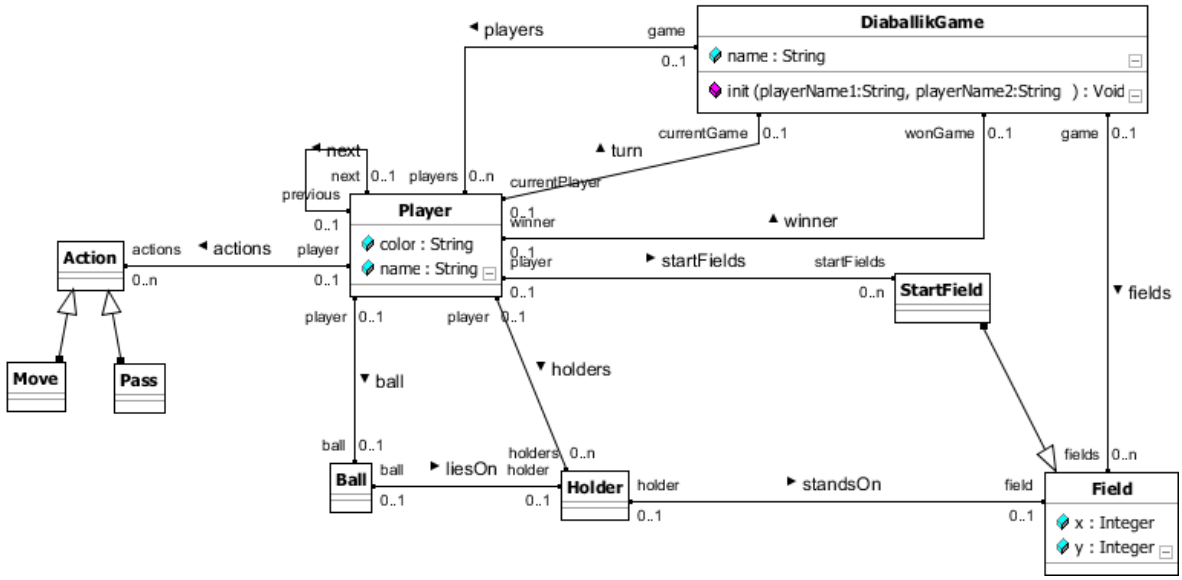

Abbildung 1: Diaballik Klassendiagram

Gegeben ist das in Abbildung 2 dargestellte Klassendiagramm. Modellieren Sie das Klassendiagramm in Fujaba und lassen Sie anschließend Quellcode daraus generieren. Folgende Vorgehensweise wird allgemein für die Modellierung vorgeschlagen:

1. Erstellen Sie für jede Klasse im Diagramm eine Klasse auf der Zeichenfläche
2. Fügen Sie den Klassen die entsprechenden Attribute hinzu
3. Fügen Sie den Klassen die ensprechenden Methoden hinzu
4. Modellieren Sie die Generalisierungen
5. Modellieren Sie die Assoziationen und passen Sie die Rollennamen und Kardinalitäten dem Klassendiagramm entsprechend an
6. Lassen Sie nun aus dem Diagramm Javaquellcode generieren

Aufgabe 2 - Implementierung (6P)

Implementieren Sie in der Klasse DiaballikGame den Rumpf der Methode

```
init(playerName1:String, playerName2:String):void
```

Sie soll das Spiel initialisieren und folgende Objekte erstellen:

- Für den ersten Paramenter einen Spieler mit der Farbe ”red” und dem entsprechenden Namen
- Für den zweiten Parameter einen Spieler mit der Farbe ”green” und dem entsprechenden Namen
- Alle 25 Felder:
  - 15 Felder
  - 5 Startfelder pro Spieler mit den zugehörigen Koordinaten
- 5 Holder pro Spieler auf den zugehörigen Startfeldern
- 1 Ball pro Spieler auf dem zugehörigen mittleren Holder

Das Spielfeld sieht folgendermaßen aus:

1,1	2,1	3,1	4,1	5,1
1,2	2,2	3,2	4,2	5,2
1,3	2,3	3,3	4,3	5,3
1,4	2,4	3,4	4,4	5,4
1,5	2,5	3,5	4,5	5,5

Abbildung 2: Diaballik Spielfeld mit Koordinaten

Die erste Koordinate entspricht Field.x, die zweite Koordinate Field.y. Die roten Startfelder befinden sich in der obersten, die grünen Startfelder in der untersten Reihe. Vergessen Sie nicht, alle Objekte korrekt zu verlinken. Anlegen von Actions und das Setzen des aktuellen Spielers ist nicht gefordert.