Die Aufgaben müssen einzeln bearbeitet und abgegeben werden. Die Abgabe muss bis **spätestens Donnerstag 31.01.2012 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 **12** Punkte.

Vorbereitung

Für die Bearbeitung der zweiten Aufgabe von Hausaufgabe 9 benötigen Sie eine Implementierung des Klassendiagramms von Castle, welches in der Übung vorgestellt wurde. Sie können diese vom Blog zur Veranstaltung herunterladen (`http://seblog.cs.uni-kassel.de/category/currentterm/pmws1213/`) und als Projekt in Eclipse importieren.

Darüber hinaus benötigen Sie das *eDOBS*-Plugin für Eclipse welches im Fujaba4Eclipse Plugin enthalten ist.

Aufgabe 1 - Zetteltest (8P)

Die Klasse `Pawn` soll eine Methode `move():void` besitzen, welche die Spielfigur von ihrer aktuellen Position um die Anzahl gewürfelter Augen weitersetzt. Eine Beispielimplementierung ist in Listing 1 abgedruckt.

```
1  public void move ()
2  {
3      Field currentField = this.getField ();
4      Game game = this.getGame();
5      Dice dice = game.getDice();
6      Field nextField = current.getNext();
7
8      for(int pips = dice.getPips(); (pips > 0); pips--)
9      {
10         currentField.removeFromPawns(this);
11         nextField.addToPawns(this);
12
13         currentField = nextField;
14         nextField = nextField.getNext();
15     }
16 }
```

Listing 1: Beispielimplementierung der Methode Pawn::move():void

Falls Sie die im Blog zur Verfügung gestellte Implementierung verwenden, ist diese Methode bereits implementiert.

Fertigen Sie einen Zetteltest zur Ausführung der Methode `move():void` an. Nutzen Sie hierfür die Tabelle auf der nächsten Seite. Die Objektstruktur nach erstmaliger Ausführung bis inklusive Zeile 6 ist bereits vorgegeben. Der Würfel zeigt vor der Ausführung 3 (DREI) Augen. Zeichnen Sie für jeden Schritt die vollständige Objektstruktur **nach** einem Durchlauf der Schleife. Falls Sie die Implementierung in Listing 1 verwenden, ist die Schleife in den Zeilen 8-15 gemeint. Die finale Objektstruktur (also nachdem Schleife verlassen wurde) soll in die letzte Zeile der Tabelle eingetragen werden.

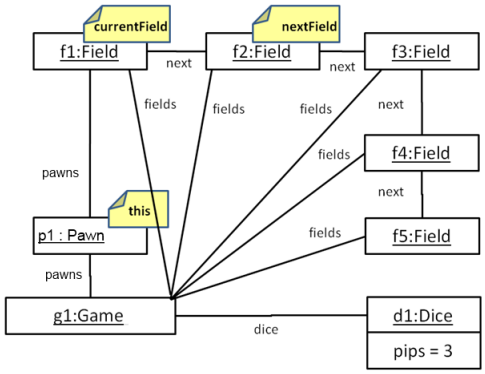
Schritt	Objektstruktur
1	
2	

Tabelle 1: Tabelle für Zetteltest

Schritt	Objektstruktur
3	
4	

Tabelle 2: Tabelle für Zetteltest Fortsetzung

Aufgabe 2 - eDOBS (4P)

Setzen Sie einen Breakpoint an das Ende der Methode `Castle::init():void`. Starten Sie die Methode `InitCastle::main(..)` im Debug-Modus und fertigen Sie einen Screenshot der eDOBS Objektstruktur am Breakpoint an.