

# Programmiermethodik

## Übung 4

Sommersemester 2011  
Fachgebiet Software Engineering

Andreas Koch  
andreas.koch@uni-kassel.de

# Agenda

- **eDOBS**
- **Rückblick: Implementierung eines Klassendiagramms von Hand**
- **Fujaba 4 Eclipse I**
  - Installation und Einrichtung
  - Klassendiagramme erstellen und bearbeiten
- **Praktische Übung I**
  - Risiko Klassendiagramm in Fujaba erstellen und Code generieren
- **Fujaba 4 Eclipse II**
  - StoryBoards erstellen und bearbeiten
- **Praktische Übung II**
  - Storyboard für Wizard::init(...) erstellen

# Fragen zu eDOBS?

# Rückblick: Implementierung eines KD von Hand

- **Nachteile**
  - Dauert lang
  - Fehleranfällig (z.B. bezüglich der Referenziellen Integrität)
  - „Man schreibt ständig das gleiche!“

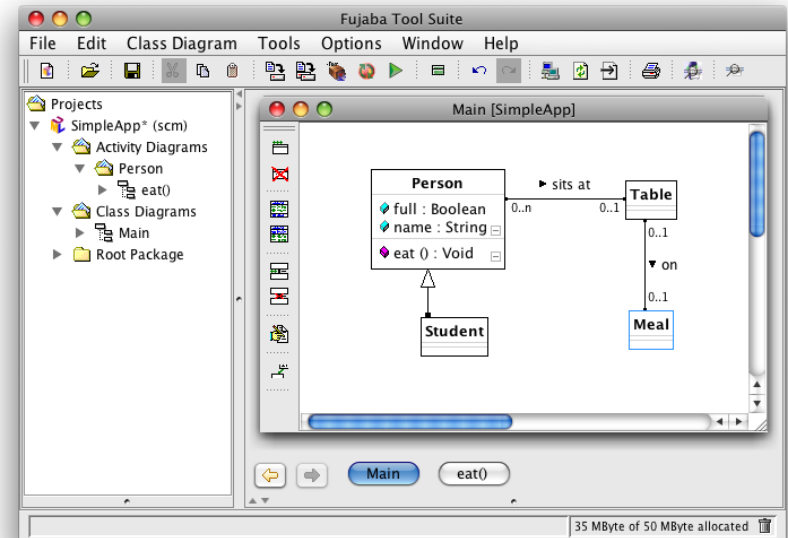
**Das lässt sich automatisieren: Codegenerierung!**

# Fujaba

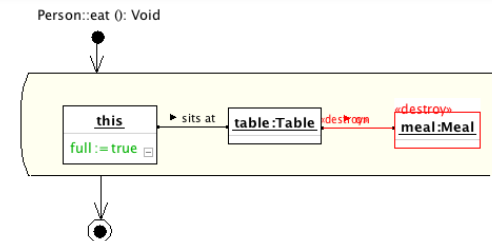
- **Fujaba: From UML to Java And Back Again**
  - Intention: Codegenerierung und Reverse Engineering

- **Funktionalitäten:**

- Erstellung von UML Klassendiagrammen  
(Modellierung von Struktur)



- Erstellung von Story- bzw. Aktivitätsdiagrammen  
(Modellierung von Verhalten)
- Erstellung von Storyboards (Modellierung von Unit-Tests)



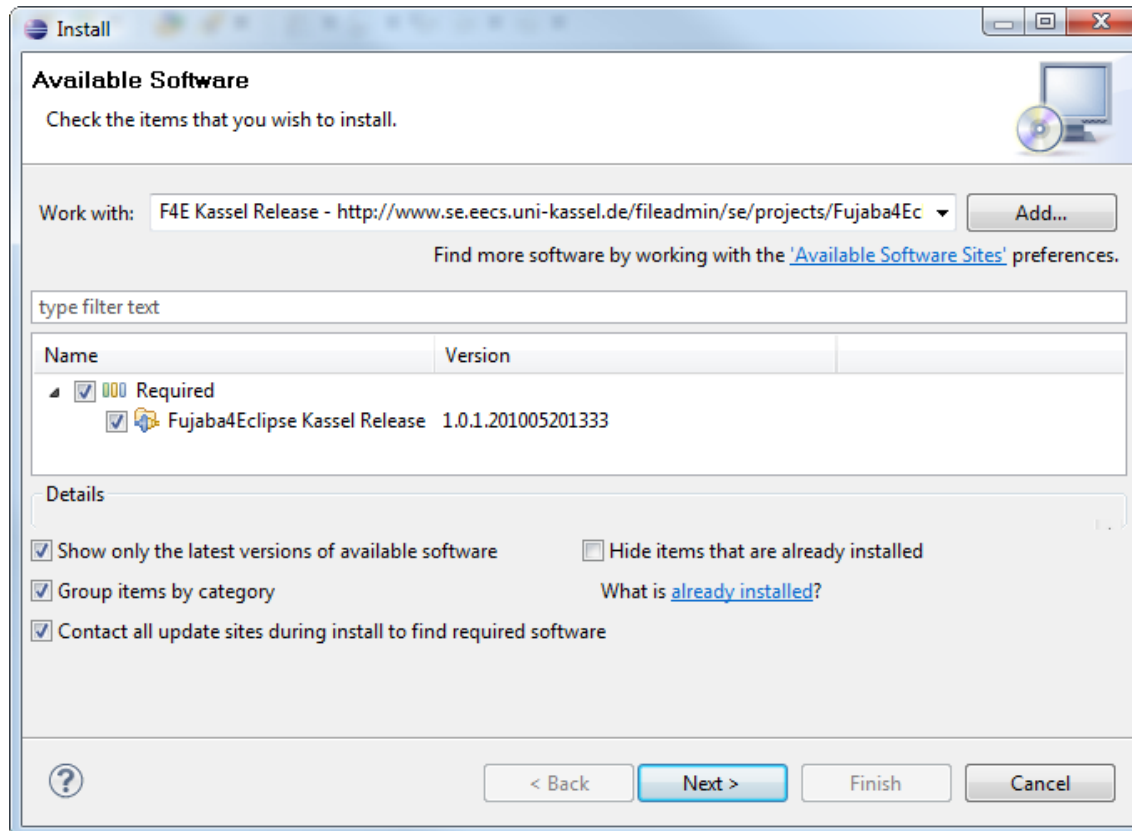
# Fujaba4Eclipse

- **Fujaba4Eclipse → Fujaba mit Eclipse Integration**
  - Ein Fujaba Projekt/Modell ist Teil eines Eclipse Projekts.
  - Ein Eclipse-Projekt hat (u.U.) viele Fujaba-Modelle
- **Vorteile:**
  - Zeitersparnis
  - Keine (Flüchtigkeits-)Fehler mehr in der Implementierung
  - Man kann sich auf das wesentliche (das Modell) konzentrieren
  - Besserer Überblick durch Diagramme statt Code
  - ...

# F4E: Installieren

- Über Eclipse Update Mechanismus. Update-Site URL:

<http://www.se.eecs.uni-kassel.de/fileadmin/se/update/>



# F4E: Projekt anlegen

## 1. Normales Java Projekt anlegen

## 2. (Neuer Unterordner „model“)

## 3. Fujaba Model erstellen

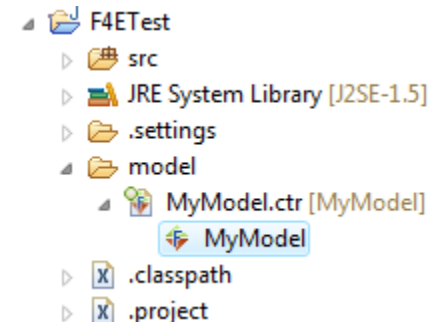
- Rechtsklick auf Java Projekt -> New -> Other -> Fujaba Model
- .ctr-Datei im Projekt- oder „model“ Ordner speichern
- (Erstellt auch gleich ein Klassendiagramm, man kann mehrere anlegen...)

## 4. Die Fujaba4Eclipse Perspektive wählen

- Window -> Open Perspective -> Other... -> Fujaba4Eclipse

## 5. Nun kann die .ctr Datei im *Project Explorer* (!)

„aufgeklappt“ werden

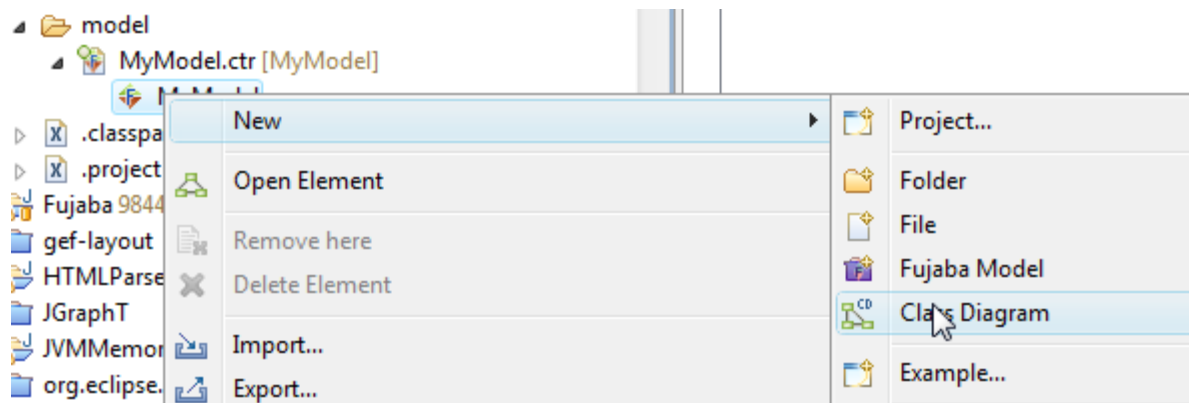




# F4E: Klassendiagramm erstellen

- **Klassendiagramm erstellen**

- Rechtsklick auf Fujaba Projekt -> New -> Class Diagram



- Name eingeben

- **„Alternate Editing Mode“ einschalten**

- Vereinfacht das Anlegen von Klassen, Attributen, Methoden und Assoziationen



# F4E: Alternate Editing Mode

- **Maus:**

- Klasse durch Aufziehen eines Kastens erstellen
- Eine Assoziation erstellen, indem die eine Klasse auf die andere gezogen wird



- **Tastatur**

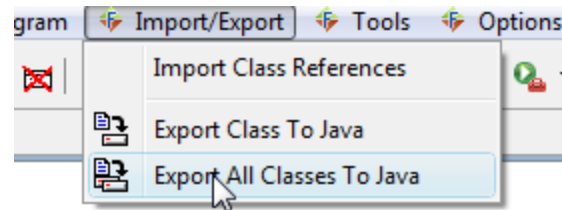
- Umbenennen von Elementen (Klassen, Attribute, Assoziationen) durch Anklicken und losstippen
- Erstellen von Attributen und Methoden durch Selektieren einer Klasse und Eingabe der Signatur:
  - myAttribute : Integer
  - aMethod():Void

**Bei LÖSCHEN mit Entf aufpassen!**

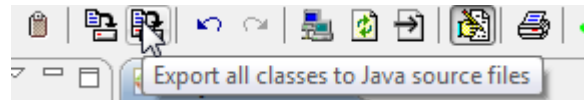
# F4E: Code generieren I

- **Zwei Möglichkeiten:**

- Fujaba Menü -> Import/Export -> Export all Classes to Java

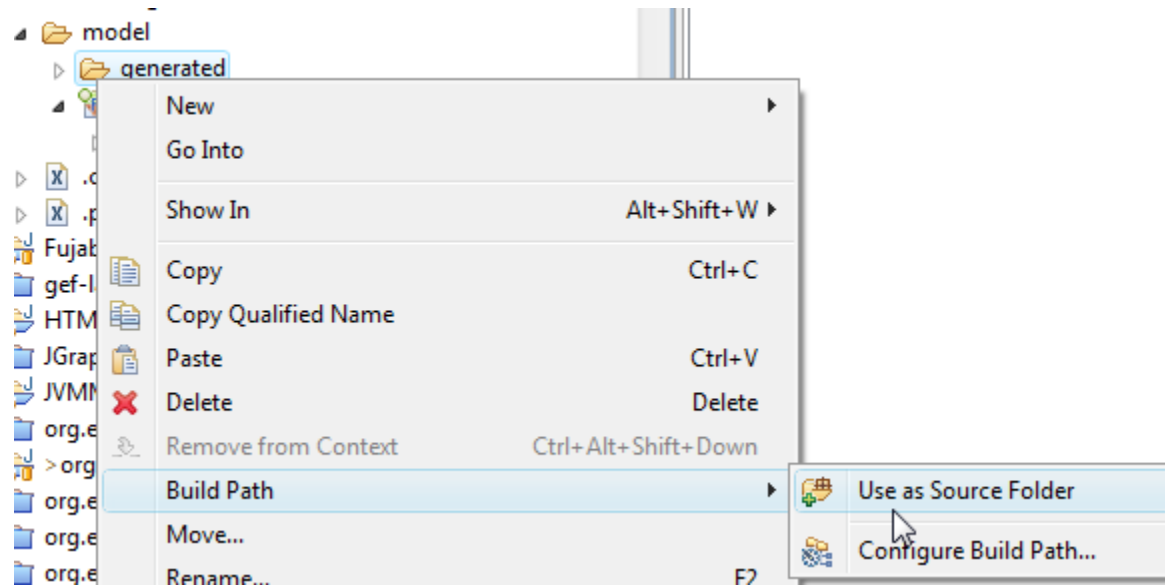


- „Export all Classes to Java source files“ Button

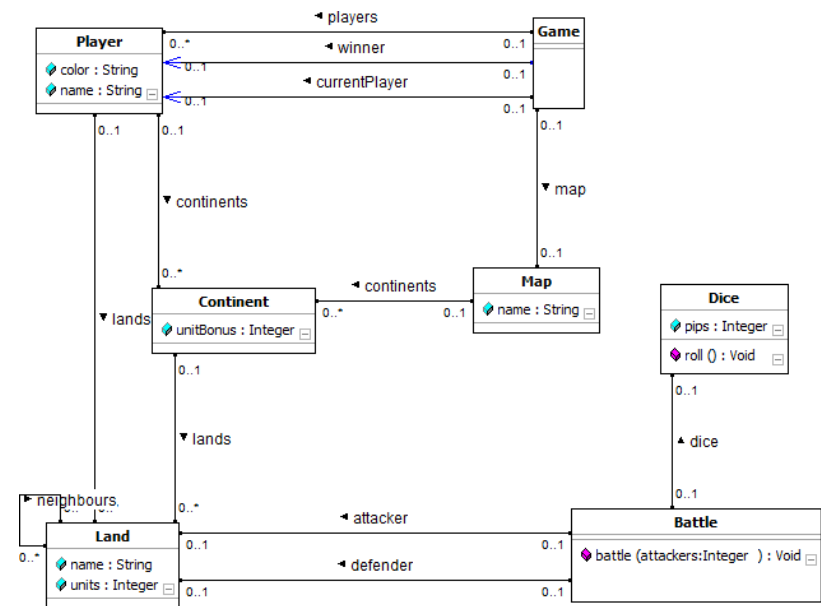


# F4E: Code generieren II

- **Generierte Klassen zum Build Path hinzufügen**
  - Rechtsklick auf „generated“ Verzeichnis -> Build Path -> Use as Source Folder

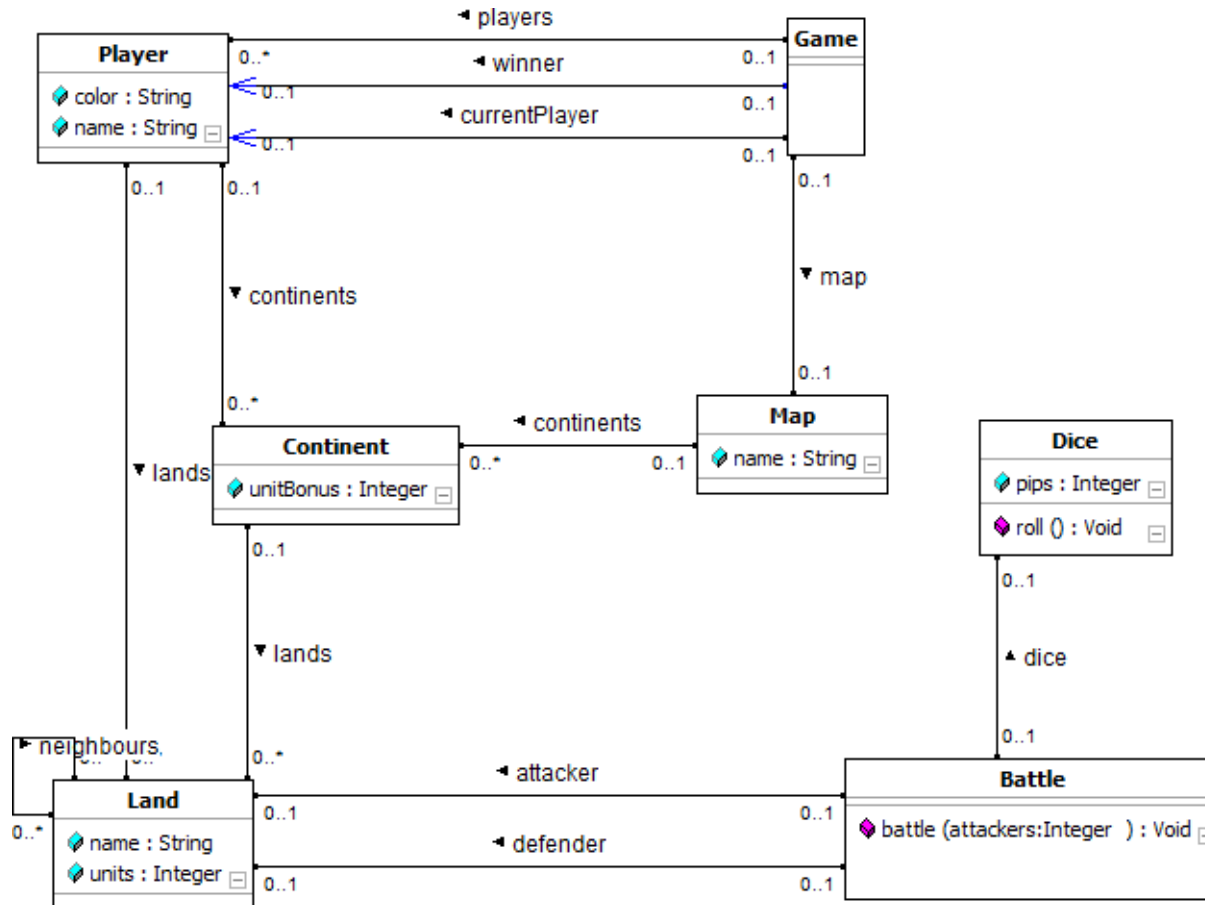


# Praktische Übung I: Risiko Klassendiagramm



# Praktische Übung I

- Erstellt das Klassendiagramm zu dem Brettspiel Risiko in Fujaba und generiert Code

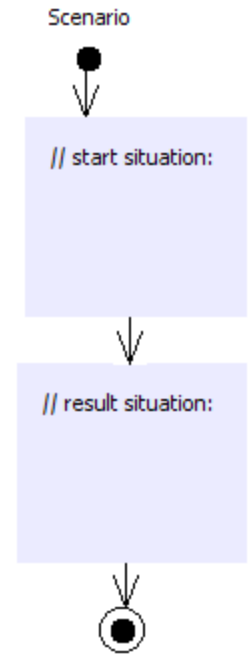


# Fujaba Storyboards I

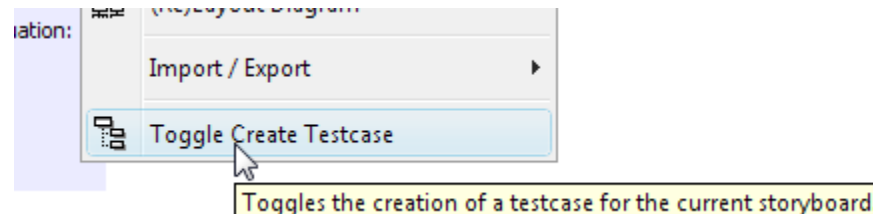
- Tests schreiben ist aufwendig -> Storyboards
- „Closed World Assumption“: Objekte die nicht gezeichnet wurden gibt es nicht
- Ein Storyboard ist – wie ein „normaler“ JUnit Test auch – nicht allgemein gehalten sondern testet IMMER einen konkreten Fall.

# Fujaba Storyboards II

- Fujaba Model öffnen -> Diagrams -> New Story Board
- Start situation:
  - Startsituation eines Szenarios herstellen
  - Methode aufrufen
- Result situation:
  - Endsituation prüfen



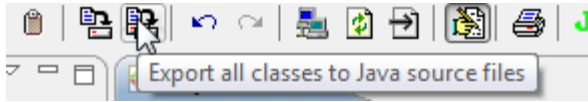
- Wichtig: Rechtsklick auf das Diagramm -> Toggle Create Testcase



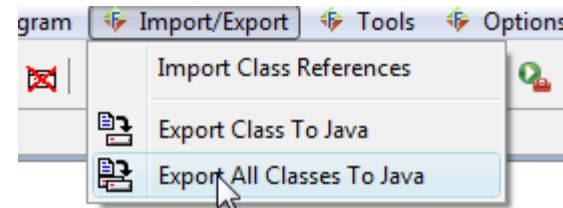


# Fujaba Storyboards III

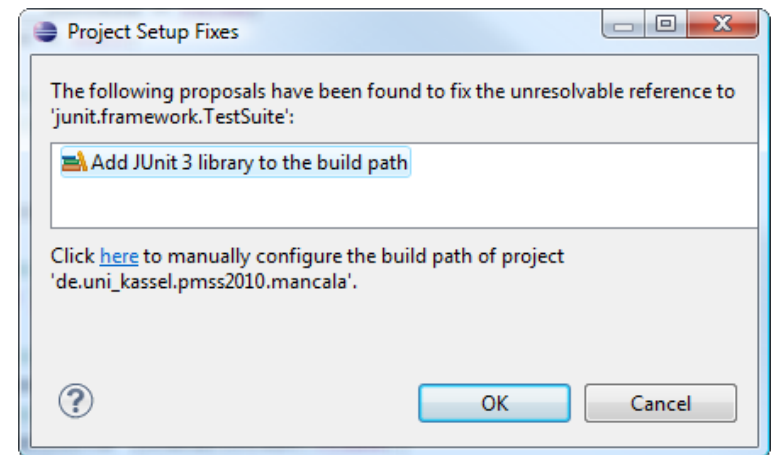
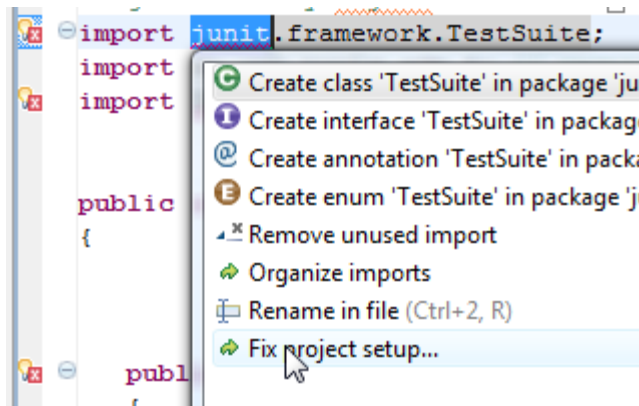
- Wie gewohnt Code generieren



oder

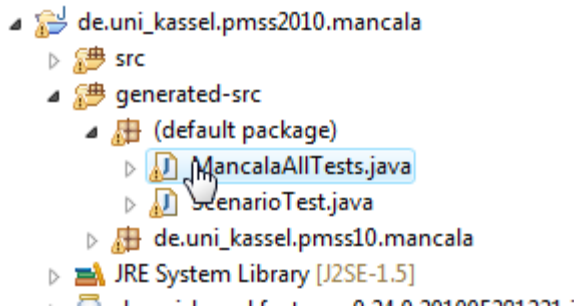


- Neben dem Modell werden JUnit3 Testklassen generiert
- Falls nötig, JUnit3 Dependency hinzufügen

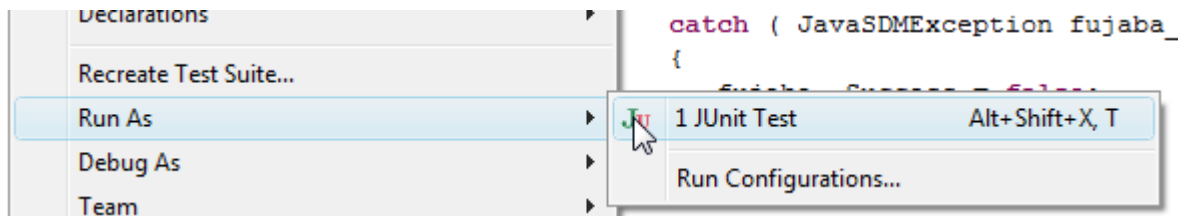


# Fujaba Storyboards IV

- **Generierte Tests ausführen:**
  - Rechtsklick



- Run As -> JUnit Test



# Praktische Übung II: Erstellen von Storyboards

- **Wizard Projekt vom PM Blog runterladen**
- **Storyboard zu Wizard::init(..) erstellen**
- **Wenn bereits vorhanden, testet eure Implementierung mit dem Storyboard**

**Ende**

**Schönes WE!**