

Einführung in Software Engineering



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Übung #10

Aufgabenblatt im Web:

<https://cage.st.informatik.tu-darmstadt.de/eise/public/exercise/ex10>

Dr.-Ing. Michael Eichberg
Dipl.-Inform. Ralf Mitschke



Software Design Patterns

- Erkennen und diskutieren von Design Patterns im JDK
- Design Patterns diskutieren
- Factory Method anwenden in der Flashcards-Software

Aufgabe 1 – Design Patterns im JDK

- a) **Analyse der Klasse** `java.rmi.server.RMIClientSocketFactory`
- Verwendung welcher Patterns
 - Angabe der Rollenverteilung im Rahmen der Implementierung
- b) **Analyse des Iterator Patterns in** `java.util.AbstractList.Itr`
- Ist der Iterator robust?
- c) **Analyse der Klasse** `java.lang.Runtime`
- Verwendung welcher Patterns
 - Angabe der Rollenverteilung im Rahmen der Implementierung

Aufgabe 2 - Design Patterns diskutieren

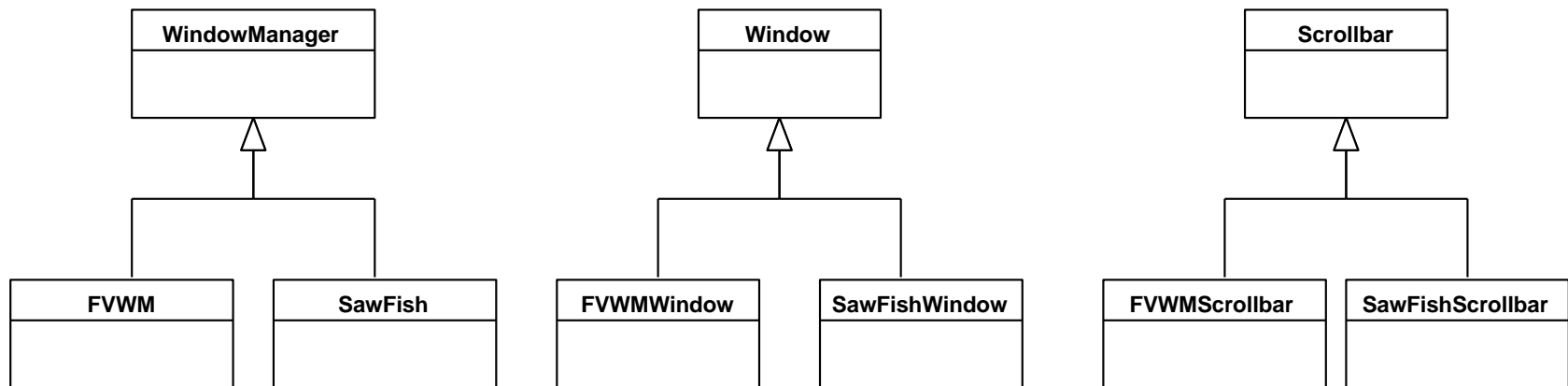
a) Singletons und Testen

Warum können Singletons die Testbarkeit einer Anwendung behindern?

Aufgabe 2 - Design Patterns diskutieren

b) Factory Method und Abstract Factory

Anforderung: Konsistente Nutzung von zugehörigen Klassen



Design mit Factory Method und/oder Abstract Factory

- Angabe der Rollenverteilung und der Verantwortlichkeiten Objekte zu erstellen
- Diskussion der Erweiterbarkeit (neue GUI Elemente, neue WindowManager)

Aufgabe 3 – Factory Method Anwenden



Verbesserung des Designs aus Übung 09.

Ist Zustand: Verschiedene Lernstrategien implementiert

a) Beschreibung des bisherigen Designs

Wer hat in Ihrem Design (aus Übung 09) die Verantwortlichkeit inne konkrete Lernstrategien zu erstellen?

Aufgabe 3 – Factory Method Anwenden



Verbesserung des Designs aus Übung 09.

Soll Zustand: Kapselung des Wissens um alle Lernstrategien im Domänen Modell

b) Factory Method Pattern anwenden

GUI sollte nur noch Factory Method und Strategy Interface benutzen
(keine Kopplung zu konkreten Strategien)

c) Dokumentation des Patterns als UML Klassendiagramm

- Relevante Klassen und Methoden
- Rollen Beschreibung