

Mustergültige zusammenarbeit

S06 Design Patterns

Patrick Kocsis & Stefan Polydor 4AHIT

11.12.2014

Table of Contents

<u>ANGABE.....</u>	<u>3</u>
ELEARNING	3
FOLIE	3
<u>DESIGNÜBERLEGUNG</u>	<u>4</u>
ANGEGEBENES UML-DIAGRAMM	4
IDENTIFIKATION UND BESCHREIBUNG DER PATTERNS	5
ABSTRACT FACTORY	5
DECORATOR	5
OBSERVER	5
STRATEGY	5
<u>ARBEITSAUFTEILUNG</u>	<u>6</u>
AUFWANDSABSCHÄTZUNG.....	6
ENDZEITAUFTeilUNG.....	6
<u>ARBEITSDURCHFÜHRUNG.....</u>	<u>7</u>
RESULTATE	7
<u>QUELLEN.....</u>	<u>7</u>

Angabe

eLearning

Geben Sie den entsprechenden Beispiel-Code (Java -> jar) und die gewünschte Dokumentation der Patterns (PDF) ab.

Es sind keine Test-Cases verlangt.

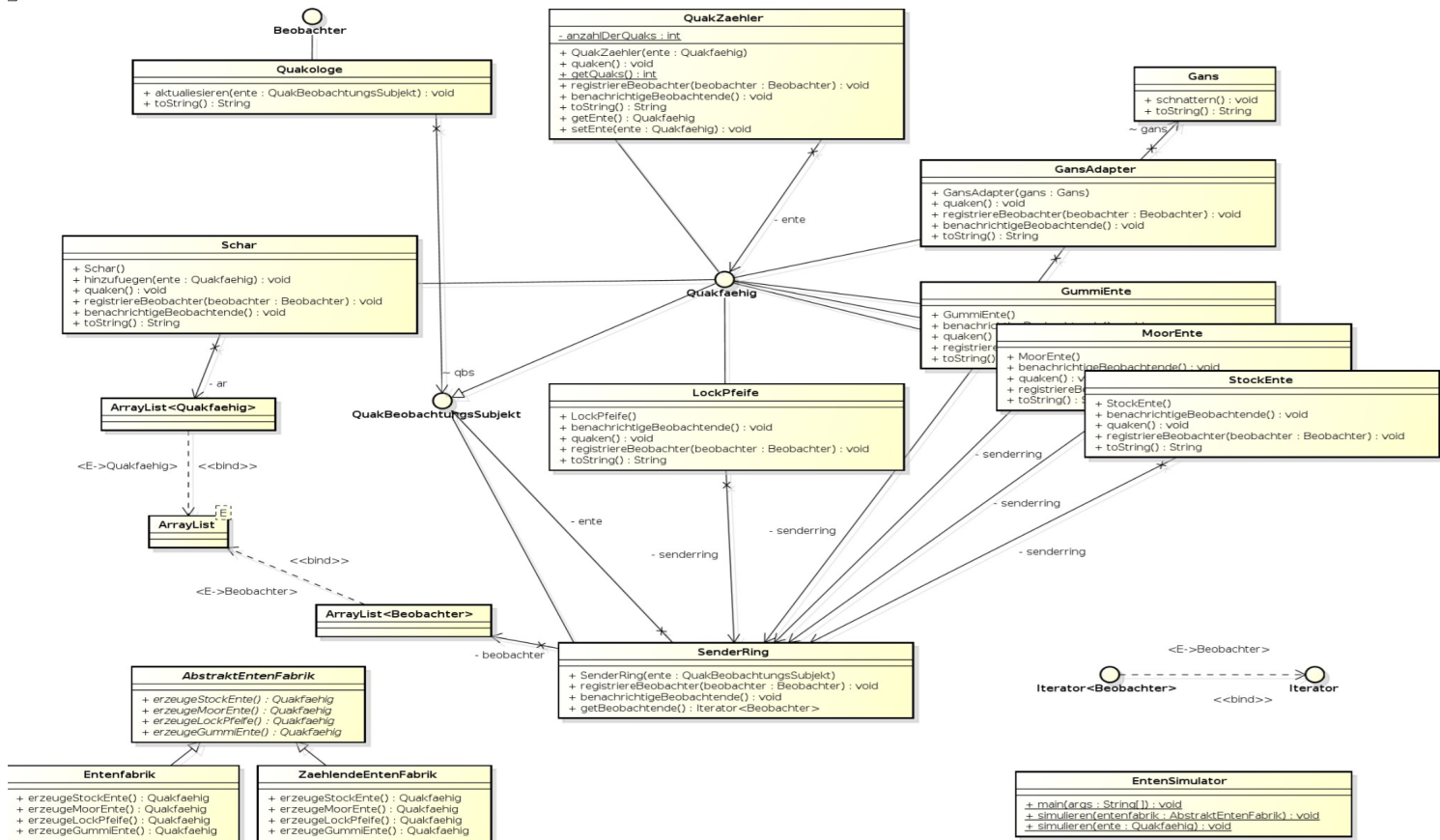
Folie

Eckpunkte

- Implementieren Sie die Quakologie!
- Erkennen Sie die verwendeten Muster!
- Wann und wo wurden die Muster eingesetzt?
- Erkläre die verwendeten Muster (kleiner Tipp, es sind deren sechs!)

Designüberlegung

Angegebenes UML-Diagramm



Identifikation und Beschreibung der Patterns

- Abstract Factory
- Decorator
- Observer
- Strategy

Abstract Factory

Bei dieser Aufgabe ist die Abstract Factory die AbstraktEntenFabrik. Je nachdem, welche Fabrik verwendet wird, kommt ein anderes Ergebnis zustande.

Decorator

Eine Klasse kann mit anderen erweitert werden. Z.B.: Einer Pizza wird mit einer Tomatensauce erweitert.

Hierbei werden Enten mit der Klasse QuackZaehler erweitert.

Observer

Für die Klassen, welche Enten beobachten möchten, wird das Interface Beobachter bereitgestellt.

Wenn eine Ente der Schar hinzugefügt wird, beobachtet sie die Schar. Dabei wird die Ente benachrichtigt, wenn die Schar z.B.: quakt und die Ente wird auch quaken.

Strategy

Bei dieser Aufgabe wurde das Quackverhalten in ein Interface geschrieben. Jede Entenklasse soll quaken() können. Daher wird sie mit dem Interface Quackfähig mitgeliefert.

Arbeitsaufteilung

Aufwandsabschätzung

	Stefan Polydor	Patrick Kocsis
UML-Begutachtung	1h	1h
Java Code	1h	1h
Protokoll	40min	40min

Endzeitaufteilung

	Stefan Polydor	Patrick Kocsis
UML-Begutachtung	1h	1h
Java Code	45min	45min
Protokoll	1h	1h

Arbeitsdurchführung

Resultate

- Gefundener Source Code auf GitHub
- Quellenangabe zum Code hinzugefügt
- Erkennung der Design Patterns -> Auswertung vom UML Diagramm
- Testcases

Quellen

- Java Code (Github-Repo); Online: https://github.com/tlins/Quakologie/tree/master/quack_java; abgerufen am 13.12.2014