

Fragen SE1 Übung 1

Frage:

Wie kann diese Kommunikationsverbindung nun dennoch mit Hilfe einer zusätzlichen Klasse, welche die dazu notwendige Objekt-Erzeugung übernimmt, aufgebaut werden? In welchem Package sollte diese zusätzliche Klasse liegen? Bitte beachten Sie dabei auch die Hinweise bzw. Anforderungen aus den Kommentaren der Klassen, Methoden und des Interfaces.

Antwort:

Hierzu wird eine Factoryklasse verwendet, welche im Controlpackage liegt, damit der Client keinen Zugriff auf diese hat. Diese Klasse liefert ein fertig generiertes Translatorobject, ohne dass der Client dieses großartig konfigurieren kann.

Frage:

Welches Entwurfsmuster (engl.: design pattern) könnte für die Problematik der Objekt-Erzeugung verwendet werden? Was ist der software-technische Nutzen bei der Verwendung des Entwurfsmusters?

Antwort:

Factory Design. Dies erlaubt es ein fertig konfiguriertes Objekt der gewünschten Klasse, in unserem Fall Translator, durch einen einzelnen Methodenaufruf erzeugen zu können. Dies erleichtert zum einen die Erzeugung, da nicht jedes Mal alles neu konfiguriert werden muss und zum anderen ist es auch eine Sicherheitsebene, da es zu sehr einfacher Replizierbarkeit von Fehlern führt.

Frage:

Wie muss man den Source Code des Interface ggf. anpassen, um mögliche auftretende Kompilierfehler zu beseitigen?

Antwort:

Das Interface muss als public überall sichtbar gemacht werden, damit der Client dieses im Sourcecode sehen und darauf zugreifen kann.

Frage:

Was ist der Vorteil einer separaten Test-Klasse?

Antwort:

Eine separate Testklasse ermöglicht es den eigentlichen Sourcecode von dem Testcode zu trennen und das Projekt übersichtlich zu halten. Ebenfalls kann der Anwendungscode separat vom Testcode ausgeliefert werden.

Frage:

Was ist bei einem Blackbox-Test der Sinn von Äquivalenzklassen?

Antwort:

Äquivalenzklassen ermöglichen es mehrere Testfälle parallel abzudecken, ohne diese alle einzeln implementieren zu müssen. Dies erleichtert die Testabdeckung.

Frage:

Warum ist ein Blackbox-Test mit JUnit auf der Klasse Client nicht unmittelbar durchführbar?

Antwort:

Die Methode display hat den returnwert void, dies macht es schwierig diese zu testen, da sie keinen überprüfbaren Wert zurückliefert. Ebenfalls hat sie keinen Sichtbarkeitsindikator, was bedeutet, dass diese nicht im testpackage gesehen werden kann, weil dafür der Indikator „public“ erforderlich wäre.

Darüber hinaus geht es bei Blackbox Testing darum die Software extern zu testen, ohne die Tests im Anwendungscode zu schreiben. Dies bedeutet, dass alle Testfälle ausgelagert werden sollten.

Deswegen würde es wenig Sinn ergeben die Testfälle direkt im Client zu implementieren.