

1)

Wie kann diese Kommunikationsverbindung nun dennoch mit Hilfe einer zusätzlichen „Fabrik“-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. So sollte z.B. auch das Datum der Erzeugung eines Translator-Objekts gesetzt werden.

- In unserer Aufgabe, die Factory Klasse “TranslatorFactory” erzeugt/instantiiert eine GermanTranslator Objekt. Es ist dann sinnvoll dass es in dem gleichen Package wie GermanTranslator liegt (unter Control).

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?

- Factory Method Pattern. Damit kann Objekten zentral und konsistent erzeugt werden.

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

- Das Interface muss ein public Interface sein: so kann es von allen Klassen aufgerufen werden, selbst in anderen Package.

3)

Was ist der Vorteil einer separaten Test-Klasse?

- So ist die Test-Klasse getrennt von der Hauptcode. Beim Kompilieren ist die Test-Klasse nicht ausgeführt, kann aber trotzdem die Klassen von Hauptcode zugreifen.

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

- Mit Äquivalenzklassen kann man alle Bereichen von gültigen und ungültigen Eingaben feststellen; durch Repräsentanten jeder Äquivalenzklasse kann man die minimalen Anzahl von Tests machen und trotzdem sind alle möglichen Reaktionen bedeckt.

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

- Mit einem Blackbox-Test auf der Klasse Client weiß man nicht was die erwartete Ergebnisse sind da man kein Zugriff zu den inneren Codes hat (und damit hat man keine Information ob hier ein GermanTranslator, EnglishTranslator usw. benutzt wird)