

- 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.
 - Beim Erstellen eines Client-Objektes wird die static-Methode der Factory-Klasse aufgerufen. In dieser wird ein Translator-Objekt deklariert und ein neues GermanTranslator-Objekt diesem zugewiesen. So kann der GermanTranslator nur auf die Methoden des Interfaces Translator zugreifen.
 - In der static-Methode wird zudem das aktuelle Datum dem neuen GermanTranslator übergeben.
 - Die Factory-Klasse sollte in einem eigenen Package liegen
- 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-Methode: Dient als Schnittstelle für die Erstellung des Objektes (GermanTranslator)
- Wie muss man den Source Code des Interface ggf. anpassen, um mögliche auftretende Kompilierfehler zu beseitigen?
 - Das Interface muss auf „public“ gesetzt werden, damit die Factory-Klasse das Interface „finden kann“
- Was ist der Vorteil einer separaten Test-Klasse?
 - Übersichtlichkeit: Tests für die Klassen werden vom Code getrennt
 - Unabhängigkeit: Tests in separaten Klassen beeinflussen sich nicht gegenseitig
 - Testklassen können beim Compilieren vom Code getrennt werden. So sind diese nicht in der eigentlichen Software vorhanden
- Was ist bei einem Blackbox-Test der Sinn von Äquivalenzklassen?
 - Da man den internen Code beim Blackbox-Test nicht kennt, helfen Äquivalenzklassen die Anzahl der Testfälle zu reduzieren. Dabei kann man die Werte mit ähnlichem/gleichen Verhalten in die jeweiligen Klassen einteilen. So kann man aus jeder Äquivalenzklasse Repräsentanten für seine Tests nutzen und die Ergebnisse der Tests abgleichen.

- Warum ist ein Blackbox-Test mit JUnit auf der Klasse Client nicht unmittelbar durchführbar?
 - Da die Klasse Client nur eine display-Methode hat, die mit „void“ deklariert ist, kann man hier keinen direkten „Rückgabewert“ der Methode testen.
 - Zudem ist die Methode nicht auf „public“ gesetzt