

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?

Antwort

Es sollte eine extra Klasse angelegt werden welche im view package hinzugefügt wird. Diese Klasse importiert die Klasse GermanTranslator und Intanziert diese in einer Funktion auf welche der Client zugreifen kann. Um zu verhindern das die Klasse Client etwas erzeugt mit "new" muss die Methode von TranslatorFactory statisch sein.

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? Gratistipp: Hinweise für das korrekte Pattern finden sie bei unten angegeben Video-Tutorien ;-)

Antwort

Man koennte die Factory Method verwenden, um dem Client den Zugriff auf die Methode der Klasse GermanTranslator zu ermoeöglichen.

Frage

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

Antwort

Man koennte eingetter fuer die Variable "version" im GermanTranslator erstellen um Zugriff auf die versions Nummer zu gewaehrleisten.

Frage

Was ist der Vorteil einer separaten Test-Klasse?

Antwort

Der Vorteil ist das es Uebersichtlicher ist fuer andere Entwickler sowie auch die Test-Klassen direkt zuordnen kann beim raussuchen von Fehler.

Frage

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

Antwort

Der Sinn von Äquivalenzklassen ist, die Einschraenkung von Werten welche die Äquivalenzklasse vorgibt erlaubt genutzt zu werden.

Frage

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

Antwort

Die Klasse Client besitzt nur eine Methode welche ein void Wert zurueck gibt. Da wir keine Methode haben welche ein Wert zum Vergleichen zurueck gibt, koennen wir auch keine Blackbox-Tests mit dieser Klasse durchfuehren.