# Cheat Sheet für Bewertung der Bewerbungsaufgabe IataExchangeRates

## Was wollen wir aus der Aufgabe herauslesen

Uns interessiert:

* Kann der Bewerber programmieren?
* Kann er eine Spezifikation lesen und verstehen?
* Wie sorgfältig arbeitet er?
* Denkt er kundenorientiert (Der Kunde sind zur Abwechslung mal wir 😊)?

Wie sehr wir die einzelnen Punkt gewichten hängt mit Sicherheit vom Profil des Bewerbers ab, aber mithilfe der folgenden Kategorien können wir die Bewertung grob strukturieren:

## Abgabeformat + Smoketest

* Wie einfach können wir die Lösung in unsere IDE importieren, war der Bewerber so aufmerksam uns mit einer Art readme abzuholen?
* Kompiliert der code?
* Gibt es unit tests und laufen sie durch?
* Startet die Anwendung und findet sie die \*.csv Datei?

## Funktionaler test (make it run)

* Tut die Anzeige eines Kurses (Funktion 1 in der Konsolenanwendung)?
  + Sind Mengen und Wertnotierung korrekt interpretiert (manche Bewerber zeigen den Kehrwert)?
  + Wird bei der Anzeige sowohl der Kurs als auch die Gültigkeit ausgegeben.
* Tut die Anzeige aller Kurse (Funktion 3)?
* Tut das Anlegen eines Kurses ohne Konflikt mit einem bestehenden Kurs (Funktion 2)?
* Tut das Anlegen eines Kurses mit Überschneidung zum Anfang des Gültigkeitszeitraumes eines bestehenden Kurses (Funktion 2)?
* … Überschneidung zum Ende?
* … Überschneidung in der Mitte des bestehenden Gültigkeitszeitraums (bestehender Kurs wird zerhackt)?
* … Komplette Überschneidung des bestehenden Gültigkeitszeitraums (bestehender Kurs wird gelöscht)?

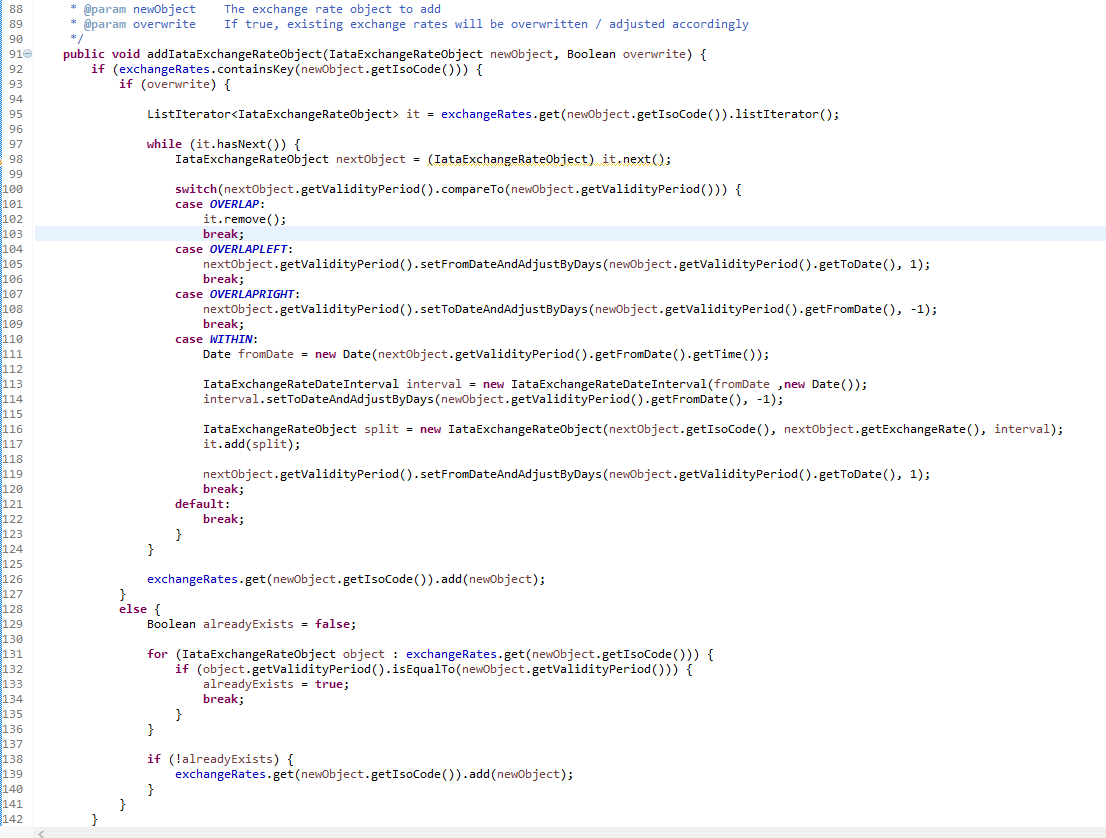
## Codequalität (make it right)

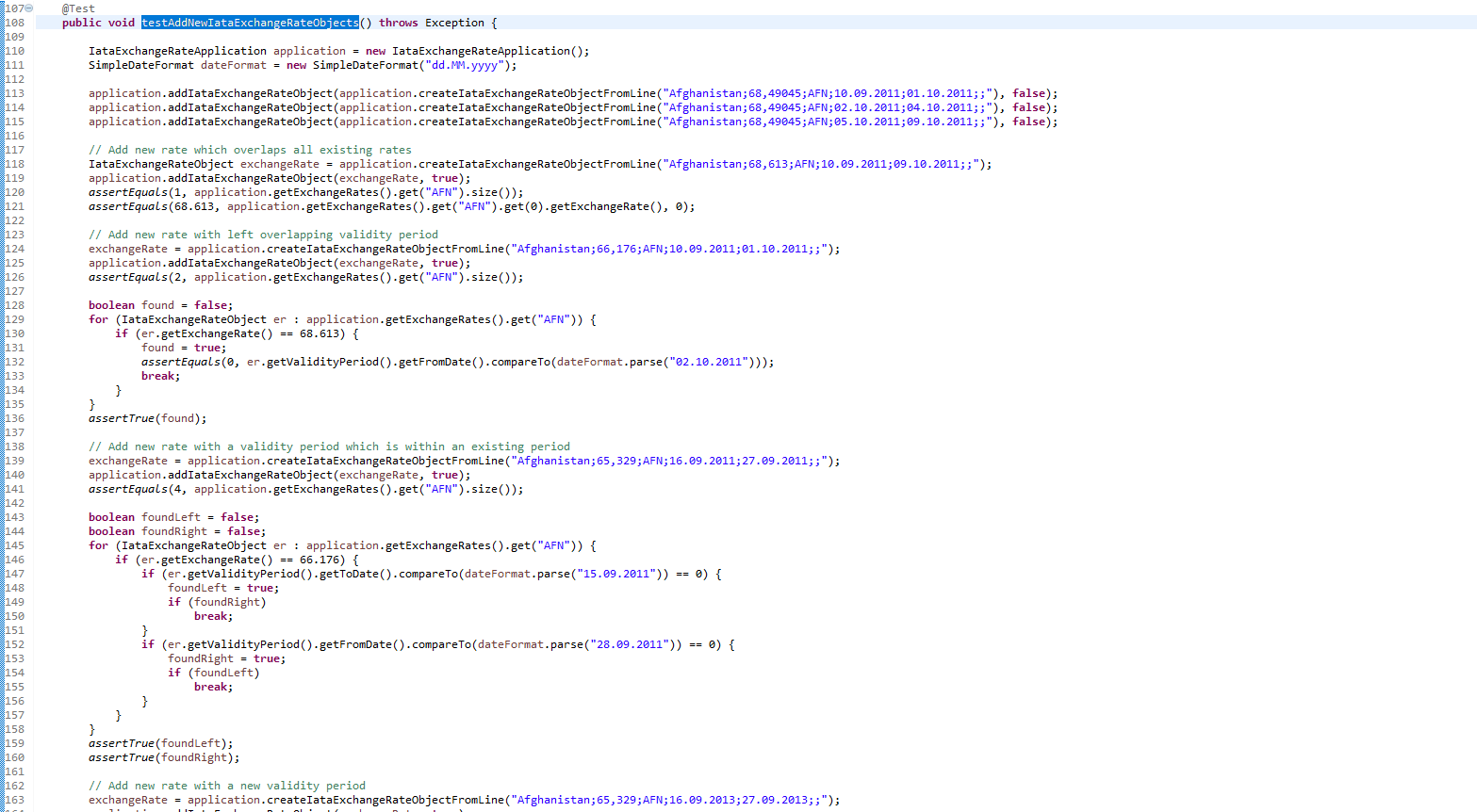
*Guter einstiegspunkt für den selbst geschriebenen Teil ist IataExchangeRateApplication#processUserInputAndCheckForExitRequest*

* Wie sind Klassen strukturiert?
* Wie sind Klassen, Methoden und variablen benannt?
* Wie ist die Testabdeckung?
* Sind die möglichen Fälle einer Kollision zwischen bestehendem und neuem Kurs übersichtlich dargestellt.

## Beispiel

Die angefügten Snippets sollen uns dazu ermahnen realistisch zu sein. Wir erwarten keine fertigen Superprogrammierer, die vom Fleck weg unsere hohen Anforderungen erfüllen. Daher hilft es den hier dargestellten Code zu überfliegen, dieser käme so wahrscheinlich nicht durch ein peer review, hat uns aber damals aufgezeigt, dass der Bewerber sein Handwerk beherrscht.





*Der code ist übrigends mit ausdrücklicher Erlaubnis des Urhebers (NSC) hier angefügt 😊*