Veranstaltungseinheit Javascript TDD Grundlagen

Test Driven Herleitung

Ziel der Übung

- Sie können eine Programmieraufgabe in kleine, testbare Teilaufgaben unterteilen
- Sie verstehen den "Test Driven Development" Prozess konzeptuell

Aufgabe(n)

In dieser Übung sollen sie ein Problem testgetrieben angehen. Anstatt ein Problem in Gedanken in eine Anzahl Aufgaben zu unterteilen ("Was soll der Code tun") muss man im TDD ein Problem in eine Anzahl testbare Kriterien unterteilen ("Welches Verhalten erwarte ich").

In dieser Übung sollen sie diese konzeptuelle Dekomposition anhand eines Beispiels üben. Ihre Aufgabe ist es, Code zur Kreditkarten-Nummer-Validierung zu schreiben. Eine Kreditkartennummer ist folgendermassen aufgebaut (Wikipedia.de):

```
Eindeutige Identifikationsnummer mit üblicherweise 16 Stellen

Die ersten sechs Ziffern bilden den BIN-Code:
 * Die ersten 4 Ziffern stehen für die Kreditkartengesellschaft.
 * Die 5. Ziffer steht für die Kreditkartenart (z. B. bei
   American Express: blau, grün, gold, platin).
 * Die 6. Ziffer dafür, ob es sich um eine Zweitkarte,
   Partnerkarte, Firmenkarte etc. handelt.

Die restlichen 10 Ziffern sind die Kontonummer mit der letzten
Ziffer als Prüfziffer (nach dem Luhn-Algorithmus).
```

Führen sie die folgende Auflistung weiter, indem sie die möglichen Implementierungsschritte aufzählen. Der erste Satz bezieht sich auf den traditionellen Entwicklungsprozess, der zweite Satz beinhaltet dieselbe Aufgabe aus der Sicht der Testgetriebenen Entwicklung. Achten Sie bei der Aufzählung darauf, die Schritte möglichst klein zu halten.

1. Dekomposition des Problemes in Tasks - Dekomposition des Problemes in Tests

2. Erstellen einer Methoden Signatur (Eingabeparameter String creditCardNo, Rueckgabewert boolen) - Eine Methode zur Validierung von Kreditkartennummern ist aufrufbar
3. Eingabeparameter auf null überprüfen und wenn dies zutrifft false zurückgeben - Wird die Methode mit null als Eingabeparameter aufgerufen gibt sie false zurück
4. Eingabeparameter auf leeren String überprüfen und wenn dies zutrifft false zurückgeben - ...
5. ...

Erwartete Resultate und Abgabe

- Die oben begonnene Tabelle ist sinnvoll fortgeführt.
- Sie können für diese Übung natürlich fast beliebig tief auf die fachliche Problematik eingehen. Im Rahmen der Übung wird als Resultat eine Tabelle mit ca. 10 einzelnen Entwicklungsschritten erwartet.
- Für diese Aufgabe wird kein Programmcode erwartet.
- Abgabe schriflich während dem Labor.

Jasmine benutzen

Ziel der Übung

Sie besitzen ein lauffähiges Jasmine Projekt.

Aufgabe(n)

- Laden Sie die Jasmine Library herunter und entpacken Sie diese. Benutzen Sie das sogenannte Standalone Release unter https://github.com/jasmine/jasmine
- Öffnen Sie die Datei `SpecRunner.html` sowohl in einem Browser als auch einem Editor. `SpecRunner.html` ist ein Beispiel für ein einfaches Javascript Projekt unter Test.
- Legen Sie analog den bereits in `SpecRunner.html` vorhandenen Tests eine eigene Javascript Testsuite an. Diese soll die Specs ihrer Applikation enthalten und soll `spec/ApplicationSpec.js` heissen.
- Legen Sie eine weitere Javascript Datei an die den zu testenden Code enthalten wird. Nennen sie diese Datei `src/Application.js`
- Binden Sie die neuen Dateien in `SpecRunner.html` ein.
- Schreiben sie eine einfache Funktion, welche immer den String “Hello, World” zurückgibt. Versuchen Sie die Funktion Test-Driven zu entwickeln, d.h. schreiben Sie zuerst Tests, dann die Funktionalität.

- Schreiben sie eine Funktion zur Berechnung der Fibonacci Folge bis zu einem parametrisierbaren Endwert. Versuchen Sie die Funktion Test-Driven zu entwickeln, d.h. schreiben Sie zuerst Tests, dann die Funktionalität.

Erwartete Resultate und Abgabe

- Komplettes Javascript Projekt mit den beschriebenen Funktionen und zugehörigen Tests.
- Abgabe während dem Labor.

Erste Zahlen der Fibonacci Folge:

```
F0      0
F1      1
F2      1
F3      2
F4      3
F5      5
F6      8
F7     13
F8     21
F9     34
F10    55
F11    89
F12   144
F13   233
F14   377
F15   610
F16   987
F17  1597
F18  2584
F19  4181
F20  6765
```