

Test Driven Development (TDD)

Befolgen Sie die folgenden Schritte für diese Aufgabe sehr genau:

1. Für diese Übung ist eine 2-er Gruppe zwingend erforderlich. Deshalb suchen Sie sich bitte einen Kommilitonen/in und bilden eine 2 er oder 3 er Gruppe! Auf jeden Fall
2. Beschäftigen Sie sich mit Test-Driven-Development und lesen die Kernideen nach: z.B. hier
 - a. <http://agiledata.org/essays/tdd.html>
 - b. <https://martinfowler.com/bliki/TestDrivenDevelopment.html> oder auch
 - c. <https://technologyconversations.com/2013/12/20/test-driven-development-tdd-example-walkthrough/>
3. Die genaue Gruppen-Aufteilung und Zuordnung erfolgt im Praktikum.
4. Jede Gruppe implementiert
 - a. die Funktionalität für eine Aufgabe X, testet Aufgabe X aber nicht,
 - b. Testcases für Aufgabe Y, implementiert aber nicht die Funktionalität für Aufgabe Y.
5. Erstellen Sie dann **Schritt für Schritt** mindestens **8 JUnit-Testcases** für die zu testende Methode und erweitern Sie ihren Code nach jedem Schritt so, dass alle bisher geschriebenen Testcases laufen.
 - a. Ihre Testfälle sollten die in den Anforderungen (unten) beschriebene Funktionalität abdecken.
 - b. Stellen Sie - wenn Sie möchten - am Ende mit zusätzlichen Testcases sicher, dass Ihre Methode die Spezifikation erfüllt.
 - c. Schicken Sie den Code der ungetesteten Methode einer anderen Gruppe, die Testcases dafür geschrieben hat. Sie selbst bekommen von dieser Gruppe Code zu der Methode, die Sie getestet haben.
 - d. Geben Sie in Ihrer Lösung an, mit wem Sie den Code ausgetauscht haben. Testen Sie den Code der anderen Gruppe mit Ihren Testcases und dokumentieren Sie die Ergebnisse, also schreiben Sie auf wieviele und welche Testcases fehl-schlagen.

Hinweis zur Vorgehensweise:

- **Achten Sie darauf, dass Sie wirklich gemäß TDD schrittweise vorgehen, d.h. Sie implementieren nicht alle 8 Testcases auf einmal und dann die gesamte Funktionalität, sondern immer ein Testcase, dann die Funktionalität für diesen Testcase usw.**

Aufgabe A:

Zu einem beliebigen Text soll ein Vorschautext angezeigt werden, der maximal 140 Zeichen lang ist (wie z.B. bei Blogbeiträgen oder News-Beiträgen). Um zu symbolisieren, dass es sich nur um eine Vorschau handelt, soll der Vorschautext mit '...' enden, allerdings nur, wenn wirklich gekürzt wurde. Wenn möglich sollen beim Kürzen keine Wörter in der Mitte getrennt werden. Stattdessen soll möglichst vor einem Leerzeichen abgeschnitten werden, auch wenn der Text dadurch insgesamt kürzer als 140 Zeichen wird.

Implementieren Sie in der Klasse `Utilities` die statische Methode

```
public static String shortenText (String s),
```

die einen beliebigen Text entgegennimmt und einen entsprechend gekürzten Text zurückliefern soll.

Aufgabe B:

In einer Zeichenkette sollen einige Sonderzeichen entfernt werden, um sie als Teil einer URL verwenden zu können. Implementieren Sie in der Klasse `Utilities` die statische Methode

```
public static String prepareStringForUrl (String s)
```

Die Methode soll den übergebenen String so ändern, dass

- Fragezeichen '?', und '&' und Hash-Zeichen '#' jeweils in die Strings 'qm', 'amp' und 'hash' umgewandelt werden,
- er keine Leerzeichen oder sonstigen Sonderzeichen (außer Minuszeichen) enthält, sondern nur Buchstaben von 'a' bis 'z' und Ziffern von '0' bis '9'. Alle Leer- und Sonderzeichen sollen durch ein Minuszeichen ('-') ersetzt werden,
- das Ergebnis keine aufeinanderfolgenden Minuszeichen enthält,
- das Ergebnis keine Großbuchstaben enthält. Großbuchstaben sind also in Kleinbuchstaben umzuwandeln,
- das Ergebnis nicht mit einem Minuszeichen beginnt oder endet,
- das Ergebnis nicht die leere Zeichenkette ist.
- Wenn die eingegebene Zeichenkette es nicht zulässt, ein entsprechendes Ergebnis zu erzeugen, soll `null` zurückgegeben werden.

Hinweise:

- **Es ist sicher hilfreich, sich in der Java API-Dokumentation nochmals die Klassen `String` und `Character` anzuschauen. Bitte verwenden Sie keine weiteren Bibliotheken, da Sie sonst der anderen Gruppe das Testen erschweren. Es sollte generell nur die Klasse `Utilities` ausgetauscht werden.**
- **Die Tests sind mit `JUnit` zu implementieren**
- **Beachten Sie, dass Sie Strings in ein Array von `Characters` umwandeln können und umgekehrt.**
- **Sie können die Klasse `Utilities` um weitere Hilfsmethoden erweitern, wenn das nötig ist.**