

## PROG-2

Aufgabenstellung für **SPRINT 2**.

# ASSIGNMENT SCRUM-SPRINT 2

## ADRELI\_2\_Threads

Aufgabenstellung für **SPRINT 2**:

1. Die Teams entwickeln aus dem **ASSIGNMENT** die **SCRUM-USER-STORIES** und die **ACCEPTATION CRITERIA**.
2. Beide Artefakte (SCRUM-USER-STORIES und ACCEPTATION CRITERIA) werden mittels Haftnotiz in das **SCRUM-BOARD** eingetragen und dann mit dem **PRODUCT-OWNER** (Prof. Illik) diskutiert und
3. danach fixiert.
4. Danach werden alle anderen weißen **SCRUM-BOARD**-Flächen bearbeitet.
5. **Nach Abnahme** durch den **SCRUM-MASTER** wird das **SCRUM-BOARD** mit dem Handy **fotografiert und archiviert**.
6. **Siehe letzte Seite „SPRINT-Planning-Meeting“**

### Aufgabenbeschreibung

**Ihre Lösung aus dem SPRINT-1 ADRELI\_1\_CON läuft.**

Machen Sie davon eine Projekt-Kopie für die Implementierung von ADRELI\_2\_Threads.


Im Projekt 2 wird die bisherige Lösung ADRELI\_1\_CON, die aus einem einzigen Thread besteht, so umgebaut, dass eine Multi-Thread-Lösung entsteht (Weiter Details im Kolloquium).

### **Ziel:**

1. Ein Thread kümmert sich um die *Kommunikation mit dem User („View“-Thread = User Interface)*.
2. Ein zweiter Thread kümmert sich um die Lese- und Schreibvorgänge auf der *Datenbasis (= „Model“-Thread)*.
3. Die Datenbasis muss die Integration mit Microsoft-Excel oder OpenOffice.Org Calc / Apple Numbers unterstützen (*„Datenbasierte Integration“*): die Datenbasis ist

(spätestens) ab jetzt eine **CSV-Datei** (*Comma Separated Values* oder *Character Separatd Values* gemäß [RFC 4180](#)) mit einer **Zeile pro Datensatz**. Die Datenbasis heißt „**adrel.csv**“. (Testdatensätze können jetzt auch mit Excel / Calc / Numbes eingepflegt werden.)

**HINWEIS:** Nutzen Sie die **Hilfsklassen** aus dem STREAM-Umfeld. Sie dürfen auch **Objekte** schreiben/lesen. Zum Beispiel: **FileOutputStream / ObjectOutputStream** und **FileInputStream / ObjectInputStream**.

4. Der Datenaustausch und die Kommunikation der beiden Threads (Model-Thread, View-Thread) erfolgt mit Hilfe einer **Pipe**. (Siehe  im Skript Kapitel 7. Dort steht ein Empfehlung, was Sie über die Pipe streamen sollten.)
5. Der Hauptthread ist der „**Controler**“-Thread.
6. Als Puffer verwenden Sie nun einen anderen Datentyp/Container wie ADRELI\_1\_CON!!

## Weitere Aufgabedetails in der Live-Veranstaltung Vorlesung/Kolloquium.

**Hinweise:** beachten Sie das „**Prinzip der konstruktiven Voraussicht**“

Zu Ziel 1): Der „View“-Thread wird später erweitert zu einem GUI-basierten Client.

Zu Ziel 2) Der „Model“-Thread wird später zu einer eigenständigen Server-App transformiert.

Zu Ziel 3) CSV wird das Export-Format der Daten nach der Umstellung auf MySQL  
Sie Server-App wird im CSV-Format ein Logfile produzieren

Zu Ziel 4) Die Kommunikation via Pipe wird später durch eine Netzwerk-Kommunikation (via Sockets) ersetzt.

Das hier angesprochene Model-View-Controller-Modell ist (noch) nicht das im strenge Sinne aus der der Software-Architektur bekannte Modell (sondern eine „Näherung 0“).

Es gelten die gleichen Regeln wie für Projekt 1!

Aufbau der Doku wie bisher:

1. Deckblatt
2. Autorendatenblatt
3. Inhaltsverzeichnis
4. Screen-Shots der Benutzerschnittstelle
5. Print out der Datenbasis (10 Sätze)
6. UML-Diagramme
7. Kommentierter Quell-Code

Doku auch in die Dropbox kopieren.

Inverted Classroom – im Plenum: Methode Gruppen-Diskussion im Raum A3.11:

**Sprint-Planning-Meeting 25.10.17**

**Frage 1: „*Threading – Möglichkeiten und Potenziale? Oder: Sinn und Zweck des Multi-Threadings!*“**

**Frage 2: Was ist MVC? Und was hat MVC mit Multi-Threading zu tun?**

