

## Entwicklung komplexer Software Systeme

### Praktikumsblatt 3

#### Gruppe C

### - Hausaufgaben -

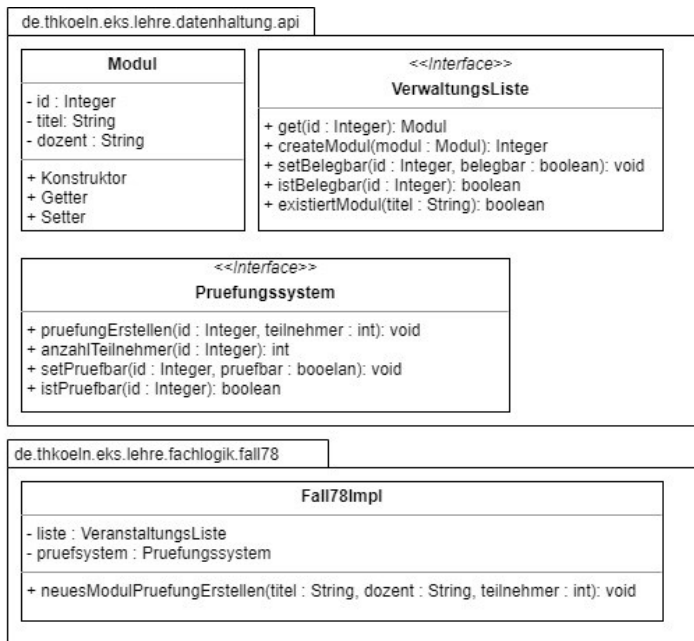
**Ziel:** Erstellung von Mockito-Testfällen

**Abgabe der Lösungen:** Bis zum 24.01., 08:00 Uhr morgens, im Master-Branch des Gitlab-Repositories P3EKS<IhreTeamnummer>. Abzugeben ist das vollständige IntelliJ-Projekt.

**Hinweis:** Ihr Gitlab-Repository zu diesem Praktikumsversuch enthält bereits ein IntelliJ-Projekt, welches Sie als Grundlage für Ihre Implementierung verwenden sollen – Sie sollen also den Ordner P3EKS<IhreTeamnummer> als IntelliJ-Projekt öffnen.

Gegeben sind die

- folgenden 3 Klassen der Schicht Datenhaltung:
  - Entitätsklasse Modul,
  - Interface-Klasse VeranstaltungsListe,
  - Interface-Klasse Pruefungssystem,
- und die Klasse Fall78Impl in der Schicht Fachlogik, welche den (fiktiven) Anwendungsfall 78 implementiert. Dieser Anwendungsfall (bzw. seine Implementierung) ist nur für diese Aufgabe konstruiert und macht sonst wenig Sinn.



**Modul1:** repräsentiert ein Modul an einer Hochschule

- **titel** ist der Titel des Moduls
- **dozent** ist der Name des Dozenten des Moduls

- `id` ist die eindeutige Nummer des Moduls

VeranstaltungsListe: ist die Liste aller an der Hochschule vorhandenen Module

- `get(id)` liefert das Modul mit der Nummer `id`
- `createModul(modul)` erzeugt das Modul `modul` in der VeranstaltungsListe und liefert als Ergebnis die eindeutige Nummer des neuen Moduls. (Hinweis: Die `id` des als Parameter übergebenen Moduls muss `null` sein)
- `setBelegbar(id, belegbar)` setzt die Belegbarkeit des Moduls mit der Nummer `id` auf `belegbar` (also `true` oder `false`).
- `istBelegbar(id)` liefert als Ergebnis die Belegbarkeit des Moduls mit der Nummer `id` (also `true` oder `false`).
- `existiertmodul(titel)` prüft, ob in der VeranstaltungsListe ein Modul mit dem Titel `titel` bereits erzeugt wurde.

Pruefungssystem: ist das Prüfungssystem der Hochschule und speichert zu einem Modul die Anzahl der zugelassenen Prüfungsteilnehmer

- `pruefungErstellen(id, anzahl)`: erstellt das Modul mit der Nummer `id` im Prüfungssystem, damit dieses Modul geprüft werden kann. Dabei wird die Teilnehmeranzahl auf den Wert `anzahl` begrenzt.
- `anzahlTeilnehmer(id)`: gibt die im Prüfungssystem derzeit gespeicherte Anzahl an zugelassenen Prüfungsteilnehmer für das Modul mit der Nummer `id` zurück.
- `setPruefbar(id, pruefbar)` setzt die Prüfbarkeit des Moduls mit der Nummer `id` auf `pruefbar` (also `true` oder `false`).
- `istPruefbar(id)` liefert als Ergebnis die Prüfbarkeit des Moduls mit der Nummer `id` (also `true` oder `false`).

Fall178Impl:

- `neuesModulPruefungErstellen(modultitel, dozent, teilnehmer)` erzeugt ein neues Modul basierend auf dem Titel `modultitel` und dem Namen des Dozenten `dozent` in der VeranstaltungsListe und meldet dieses Modul im Prüfungssystem mit `teilnehmer` als der Anzahl der zugelassenen Prüfungsteilnehmer an. Diese Methode ist eine korrekte Implementierung des Anwendungsfalls 78 und darf nicht verändert werden.

### Ihre Aufgaben:

Erstellen Sie im Modul `de.thkoeln.eks.lehre.fachlogik` im dortigen Verzeichnis `src/test/java` im Paket `de.thkoeln.eks.lehre.fachlogik.fall178` die Testklasse `TestFall178Impl` mit dem unten aufgeführten Testfall.

Erstellen Sie hierfür für die Interfaces VeranstaltungsListe und Pruefungssystem entsprechende Mocks. Verwenden Sie die Dependency Injection (DI) für die Injektion dieser Mocks.

Testfall: Prüft mit Mockito die Aufrufe der Methode

`neuesModulPruefungErstellen(modultitel, dozent, teilnehmer)` der Klasse `Fall178Impl`.

In diesem Testfall müssen die folgenden Bedingungen geprüft werden:

- Die Reihenfolge der Aufrufe muss exakt wie hier angegeben eingehalten werden!

- Aufruf `existiertModul()` mit `modultitel` (Hinweis: Rückgabe von `existiertModul()` muss natürlich `false` sein.)
- Aufruf `createModul()` muss mit Modul mit folgenden Attributen erfolgen (ArgumentCaptor hierfür verwenden!):
  - `titel` ist String "Modul"+`modultitel` (Hinweis: Hört sich falsch an, soll aber in dieser Aufgabe so gemacht werden ;-)
  - `dozent` ist der übergebene String `dozent`
  - `id` ist `null`
- Aufruf `setBelegbar()` mit `ID` und `false` (ID = Rückgabe von `createModul()`)
- Aufruf `setPruefbar()` mit `ID` und `false`
- Aufruf `get()` mit `ID` (Hinweis: Hier soll das zuvor erzeugte Modul als Ergebnis geliefert werden)
- Aufruf `existiertModul()` mit `titel` (Hinweis: Rückgabe sollte `true` sein)
- Aufruf `setBelegbar()` mit `ID` und `true`
- Aufruf `istBelegbar()` mit `ID` (Hinweis: Rückgabe sollte `true` sein)
- Aufruf `pruefungErstellen()` mit `ID` und `teilnehmer`
- Aufruf `anzahlTeilnehmer()` mit `ID` (Hinweis: Rückgabe sollte `teilnehmer` sein)
- Aufruf `setPruefbar()` mit `ID` und `true`
- Aufruf `istPruefbar()` mit `ID` (Hinweis: Rückgabe sollte `true` sein)
- `istBelegbar()` muss mindestens 1 Mal aufgerufen werden
- `istPruefbar()` muss mindestens 1 Mal aufgerufen werden
- `createModul()` muss genau 1 Mal aufgerufen werden
- `pruefungErstellen()` muss genau 1 Mal aufgerufen werden
- 

Für alle diese Aufrufe ist korrektes stubbing zu erstellen.