



Klausurprüfung - Haupttermin Mai 2023
PROGRAMMIEREN UND SOFTWARE ENGINEERING
Höhere Lehranstalt für Informatik

Reife- und Diplomprüfung 2022/23

Haupttermin Mai 2023

Aufgabenstellung für die Klausurprüfung (Fachtheorie)

| | |
|--------------------------|---|
| Jahrgang | 5AHIF, 5BHIF, 5SCHIF, 5EHIF |
| Prüfungsgebiet | Programmieren und Software Engineering |
| Prüfungstag | 8. Mai 2023 |
| Arbeitszeit | 6 Std. (gem. §6 VO BGBl. II Nr. 8/2022) |
| Kandidaten/Kandidatinnen | |
| Prüfer/Prüferin | DI Joachim Grüneis ; DI (FH) Martin Schrutek ; DI Thomas Tschernko ; Klaus Unger , MSc |
| Aufgabenblätter | 10 Seiten inkl. Umschlagbogen |

Inhaltsübersicht der Einzelaufgaben im Umschlagbogen
(Unterschrift des Prüfers/der Prüferin auf den jeweiligen Aufgabenblättern)

Das versiegelte Kuvert mit der der Aufgabenstellung wurde geöffnet von:

Name: Unterschrift:
Datum: Uhrzeit:

Zwei Zeugen

Name: Unterschrift:
Datum: Uhrzeit:

Geprüft: Wien am

.....
.....DI Robert **Jelinek**.....
.....Abteilungsvorstand.....

RS.

.....
..... Dr. Gerhard **Hager**
.....Direktor

Genehmigt:

Wien, am
.....

RS.

.....
.....MinR Mag. Gabriele **Winkler-Rigler**



Klausurprüfung (Fachtheorie)
aus Programmieren und Software Engineering
im Haupttermin 2022/23
für Höhere Lehranstalt für Informatik

Liebe Prüfungskandidatin,
liebe Prüfungskandidat!

Bitte füllen Sie zuerst die nachfolgenden Felder in Blockschrift aus, bevor Sie mit der Arbeit beginnen.

Maturaaccount (im Startmenü sichtbar):

Vorname

Zuname

Klasse

Teilaufgabe 2 a) RESTful API

☐

Teilaufgabe 2 b) Web Anwendung

☐



Generelle Hinweise

Die Klausurarbeit besteht aus zwei Teilaufgaben und zwei Theoriefragen.

Die Arbeitszeit für die Bearbeitung der gestellten Aufgaben und Fragen beträgt **6 Stunden** (360 Minuten).

Die beiden Teilaufgaben sind unabhängig voneinander zu bearbeiten, Sie können sich die Zeit frei einteilen.

Sie können die beiden Aufgabenstellungen entweder mit dem Technologie- und Toolstack:

- ``C#`, `.NET 6`, `Visual Studio`/`Jetbrains Rider`` oder
- ``Java 19`, `Spring Boot 3`, `IntelliJ``

lösen.

Sie finden auf den Arbeitsplatzrechnern die erforderlichen Tools vollständig installiert und konfiguriert vor. Ebenso existieren die erforderlichen Projekte inklusive aller Abhängigkeiten - also entweder eine ``.NET Solution`` oder ein ``Maven` & `Spring Boot`` basiertes Projekt.

Ihre Aufgabe ist es nun die existierenden Projekte mit den geforderten Lösungen zu füllen.

Sollte es notwendig sein, dass die Annahmen (Abweichungen von der Angabe) treffen, um in Ihrer Implementierung weiter zu kommen dann kommunizieren sie diese bitte mittels Kommentaren im erstellten Quellcode.

Übersicht Beurteilung

| Teilaufgabe / Theoriefrage | Punkte |
|--------------------------------|------------|
| Domain Modelling & Persistence | 40 |
| Presentationlayer | 55 |
| Dependency Injection | 5 |
| Model View Controller | 5 |
| Summe | 100 |



Teilaufgabe 1 - Domain Modelling & Persistence

Überblick

Wir haben an der Schule das Innolab. In diesem findet einerseits Unterricht statt und zusätzlich können SchülerInnen die Räume, die einzelne Arbeitsplätze und das Equipment (wie z.B. 3D Drucker) auch für schulische Projekte nutzen.

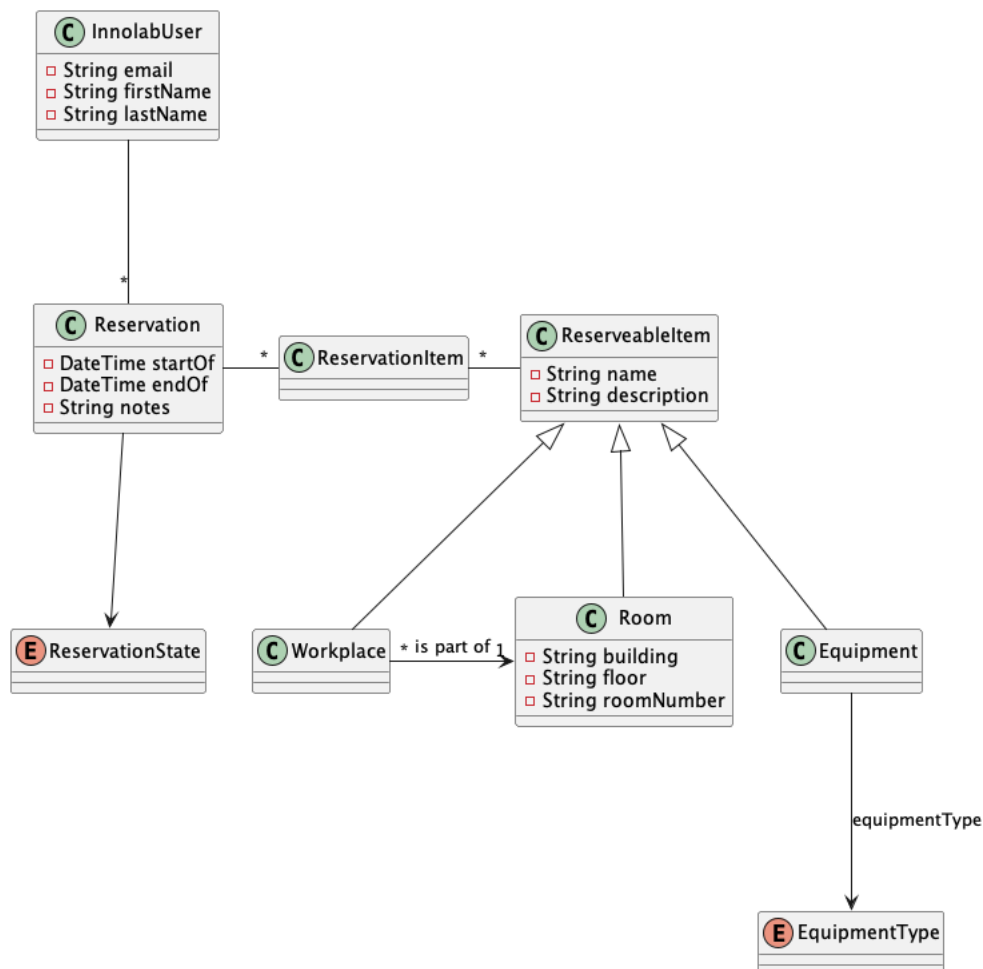
Es wird nun eine Möglichkeit (Anwendung) benötigt, um diese Nutzungen zu synchronisieren. SchülerInnen - welche sich zuvor registriert haben - sollen Räume, Arbeitsplätze (in Räumen) und Equipment reservieren können.

Sowohl den betreuenden LehrerInnen als auch für SchülerInnen soll es damit einfacher werden folgende Fragen zu beantworten:

- Wie hoch ist die Auslastung des Innolab?
- Ist zur gewünschten Zeit ein Raum, ein Arbeitsplatz bzw spezifisches Equipment verfügbar
- Von wem wurden zu einem bestimmten Zeitpunkt Plätze, Räume, ... benutzt

ClassModel

Das folgende Modell ist gedacht als erster Schritt zu einer vollständigen Lösung für das Innolab. Ihre Aufgabe ist die Umsetzung dieses Modells!





Klausurprüfung - Haupttermin Mai 2023
PROGRAMMIEREN UND SOFTWARE ENGINEERING
Höhere Lehranstalt für Informatik

| | |
|-------------------------|---|
| InnolabUser | A registered user of Innolab. The `email` is mandatory and should be unique. |
| ReserveableItem | An item which can be reserved using this app. Every item has to have a unique `name`. |
| Room | Rooms are of `ReserveableItem` and have three mandatory attributes: `building`, `floor` and `roomNumber`. |
| Workplace | Workplaces are of `ReserveableItem` and are part of a `Room`. |
| Equipment | Equipment items are of `ReserveableItem` and have an `EquipmentType`. |
| EquipmentType | Is an enumerated with the values: `SOLDERING_STATION` and `OSCILLOSCOPE`. |
| Reservation | An `InnolabUser` creates a `Reservation` for a certain time slot and for one or many `ReserveableItem`s. The time slot is described with the attributes `startOf` and `endOf` which are both mandatory. |
| ReservationItem | Links one `ReserveableItem` with a certain `Reservation`. |
| ReservationState | A `Reservation` is first `REQUESTED` and later `ACCEPTED`. |

Aufgabenstellung

Umsetzung des geforderten **class models** als entsprechende `C#` oder `Java` Klassen inklusive:

- Mapping für eine **Relationale Datenbank** (z.B. `H2` oder `Microsoft.EntityFrameworkCore.InMemory`)
- Klassen für **`CRUD` Operationen** (create, read, update und delete)
- **Unittests** welche die Korrektheit der Implementierung überprüfen

Folgende Anwendungsfälle sollten mindestens **als Test** umgesetzt werden:

Anlegen von `ReserveableItem`

Alle drei Arten von `ReserveableItem` sollen angelegt werden.

Anlegen von `InnolabUser`

Es sollen `InnolabUser` angelegt werden.

Anlegen einer `Reservation`

Es soll eine vollständige `Reservation` angelegt werden - also mit einem `InnolabUser` und zumindest einem zugeordnetem `ReserveableItem`.

Die Implementierung von weiteren - sinnvollen - Testfällen bzw welchen Detailgrad das implementierte Modell aufweist, wird in der Note berücksichtigt. Die minimalen Anforderungen führen zu einer positiven Note.



Klausurprüfung - Haupttermin Mai 2023
PROGRAMMIEREN UND SOFTWARE ENGINEERING
Höhere Lehranstalt für Informatik

Beurteilung

Punkte erhalten Sie wenn ihre Implementierung baut (compile), es Unittests mit Überprüfungen (Assertions) gibt und diese Unittests auch erfolgreich durchlaufen!

Sie erhalten:

| | |
|-----------------------------|--|
| 4 Punkte | Wenn ihre Implementierung eine korrekte Umsetzung des <code>`ReserveableItem`</code> , inklusive Zugriffsoperationen und den entsprechenden Unittests aufweist. |
| 4 Punkte | Wenn ihre Implementierung eine korrekte Umsetzung des <code>`InnolabUser`</code> , inklusive Zugriffsoperationen und den entsprechenden Unittests aufweist. |
| 4 Punkte | Wenn ihre Implementierung eine korrekte Umsetzung der <code>`Reservation`</code> , inklusive Zugriffsoperationen und den entsprechenden Unittests aufweist. |
| 4 Punkte | Wenn ihre Implementierung eine korrekte Umsetzung der <code>`ReservationItem`</code> , inklusive Zugriffsoperationen und den entsprechenden Unittests aufweist. |
| 10 Punkte | Wenn ihre Implementierung eine korrekte Umsetzung von <code>`Workplace`</code> , <code>`Room`</code> und <code>`Equipment`</code> , inklusive Zugriffsoperationen und den entsprechenden Unittests aufweist. |
| 14 Punkte | Wenn weitere - sinnvolle und erfolgreiche - Tests vorhanden sind sowie ihre Implementierung Details wie Validierung, Fehlerbehandlung, ... (korrekt und getestet) aufweist. |
| Summe: 40 Punkte | |



Teilaufgabe 2 – Presentationlayer

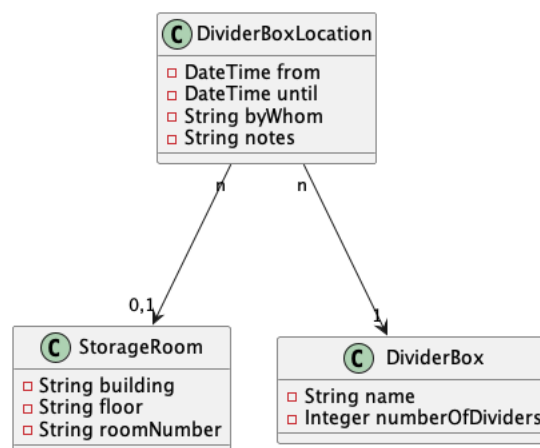
Thema 2 verlangt die Implementierung und das Testen eines RESTful API's **oder** einer Web Anwendung.

Fachlichkeit

In der Schule werden bei Schularbeiten Trennwände verwendet. Diese Trennwände sind in Boxen in verschiedenen Räumen verwahrt. Es soll nun möglich sein den Aufenthaltsort dieser Trennwände einzusehen und zu setzen.

Class Model

Das Klassenmodell inklusive der erforderlichen Zugriffsklassen sind bereits für sie implementiert. Diese sollen verwendet werden.



Umzusetzende Logik

Es sollen Instanzen der Objekte aus dem obigen Modell verwaltet werden können. Dies beinhaltet:

- Neuanlage (Create)
- Abrufen einer Gesamtliste sowie einzelner Datensätze
- Entfernen (Delete)

Im besonderen sind folgende Regeln zu implementieren.

- Ein `StorageRoom` muss alle drei Attribute gesetzt haben.
- Eine `DividerBox` muss alle beiden Attribute gesetzt haben.
- Bei der `DividerBoxLocation` sind mehr Regeln zu beachten. Es sind zumindest die Attribute `from`, `byWhom` und um welche `DividerBox` es sich handelt immer anzugeben. Gibt es für das neue Startdatum (`from`) bereits einen Eintrag ist in diesem das `until` Datum zu setzen. D.h. es darf zu einem Zeitpunkt auch immer nur einen Eintrag in der `DividerBoxLocation` zu jeder `DividerBox` geben.



RESTful API / Web Application

Es soll ein RESTful API bzw eine Web Application **entworfen und implementiert** werden um folgende Operationen durchführen zu können:

1. Verwaltung ``StorageRoom`` mit den Operationen:
Neuanlage, Gesamtliste, Einzelabfrage, Entfernen
2. Verwaltung ``DividerBox`` mit den Operationen:
Neuanlage, Gesamtliste, Einzelabfrage, Entfernen
3. Erstellen einer ``DividerBoxLocation`` für eine ``DividerBox``
 - a. wenn noch keine ``DividerBoxLocation`` existiert
 - b. wenn eine ``DividerBoxLocation`` existiert
 - c. mit einem ``StorageRoom``
 - d. ohne ``StorageRoom``
4. Abfrage aller ``DividerBoxLocation`` zu einem ``StorageRoom``
5. Abfrage aller ``DividerBoxLocation`` zu einer ``DividerBox``
6. Abfrage aller ``DividerBox`` die an keinem ``StorageRoom`` sind
7. Abfrage aller ``DividerBox`` die keine ``DividerBoxLocation`` haben
8. Sucher aller ``DividerBox`` s in einem Stockwerk (``floor``)
9. Sucher aller ``DividerBox`` s in einem Gebäude (``building``)

Implementieren sie entweder a) ein RESTful API oder b) eine Web Anwendung!
Nutzen sie die bereitgestellten Domain und Repository Klassen!
Sie müssen den Presentation Layer und Services implementieren!

Teilaufgabe 2a – RESTful API

Im Falle einer RESTful API Implementierung sind die **entsprechenden Endpunkte zu entwerfen**. Entsprechende **Controller** und **Service Klassen** zu implementieren. Und **Tests** welche die **geforderten Anwendungsfälle abdecken** (inklusive Überprüfungen - Assertions).

Z.B. werden für ``StorageRoom`` zumindest vier Endpunkte erforderlich sein.

Teilaufgabe 2b – Web Anwendung

Im Falle einer **Web Anwendung** ist es erforderlich die benötigten **Pages** zu entwerfen. Entsprechende **Controller** und **Service Klassen** zu implementieren. Und **Tests** welche die **geforderten Anwendungsfälle abdecken** (inklusive Überprüfungen - Assertions).

Z.B. werden für ``StorageRoom`` zumindest eine Listendarstellung, eine Einzeldarstellung und die jeweiligen Buttons um die Operationen auszulösen erforderlich sein.



Klausurprüfung - Haupttermin Mai 2023
PROGRAMMIEREN UND SOFTWARE ENGINEERING
Höhere Lehranstalt für Informatik

Beurteilung

Punkte erhalten Sie, wenn Ihre **Implementierung baut** (compile), es **Tests** mit Überprüfungen (Assertions) gibt und diese auch **erfolgreich** durchlaufen!

Sie erhalten:

| | |
|-----------------------------|--|
| 7 Punkte | Wenn die Verwaltung von <code>`StorageRoom`</code> vollständig ist |
| 7 Punkte | Wenn die Verwaltung von <code>`DividerBox`</code> vollständig ist |
| 10 Punkte | Wenn das Erstellen von <code>`DividerBoxLocation`</code> s vollständig ist |
| 10 Punkte | Wenn alle Abfragen implementiert sind - hier ist vor allem auch Logik in den zu erstellenden Service Klassen gefordert |
| 21 Punkte | Welche davon anhängen wie hoch die Testabdeckung der Lösung ist, wie die Qualität der Umsetzung beurteilt wird. Korrekte Behandlung von Fehlern entweder über HTTP Status Codes (RESTful API) oder entsprechende Fehlertexte (Web Anwendung). |
| Summe: 55 Punkte | |



Theoriefrage 1 – Dependency Injection

Sie haben im Unterricht mehrmals mit dem Design Pattern **Dependency Injection** gearbeitet.

Bitte erläutern Sie anhand eines **class models** und einer **Beschreibung** dieses Design Pattern.

Erläutern Sie wie dieses Design Pattern in dem von Ihnen gewählten Technologiestack umgesetzt wurde und Sie in Ihrer Implementierung damit arbeiten.

Mit der Antwort auf diese Frage sind **5 Punkte** erreichbar



Theoriefrage 2 – Model View Controller

Sie haben im Unterricht mehrmals mit dem Design Pattern **Model View Controller** gearbeitet.

Bitte erläutern Sie anhand eines **class models** und einer **Beschreibung** dieses Design Pattern.

Erläutern Sie wie dieses Design Pattern in dem von Ihnen gewählten Technologiestack umgesetzt wurde und Sie in Ihrer Implementierung damit arbeiten.

Mit der Antwort auf diese Frage sind **5 Punkte** erreichbar



Klausurprüfung - Haupttermin Mai 2023
PROGRAMMIEREN UND SOFTWARE ENGINEERING
Höhere Lehranstalt für Informatik

Bewertungsblatt (vom Prüfer auszufüllen)

In Summe sind 100 Punkte zu erreichen.

Für eine Berücksichtigung der Jahresnote müssen mindestens 30 % der Gesamtpunkte erreicht werden.

Für eine positive Beurteilung der Klausur müssen mindestens 50 % der Gesamtpunkte erreicht werden.

Beurteilungstufen:

88 - 100 Punkte – Sehr gut

76 - <88 Punkte – Gut

64 - <76 Punkte – Befriedigend

51 - <64 Punkte – Genügend

Erreichte Punkte je Aufgabe

| Teilaufgabe / Theoriefrage | Erreichte Punkte / von |
|--------------------------------|------------------------|
| Domain Modelling & Persistence | / 35 |
| Presentationlayer | / 55 |
| Dependency Injection | / 5 |
| Model View Controller | / 5 |
| Summe | / 100 |