



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

Reife- und Diplomprüfung

Nebentermin September 2024

Aufgabenstellung für die Klausurprüfung (Fachtheorie)

Jahrgang	5AHIF, 5BHIF, 5SCHIF, 5EHIF
Prüfungsgebiet	Programmieren und Software Engineering
Prüfungstag	September 2024
Arbeitszeit	5 Std. (gem. §6 VO BGBl. II Nr. 8/2022)
Kandidaten/Kandidatinnen	
Prüfer/Prüferin	DI Joachim Grüneis ; Michael Schletz BEd; 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 Nebentermin September 2024
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



Generelle Hinweise

Die Klausurarbeit besteht aus drei Teilaufgaben und zwei Theoriefragen.

Die Arbeitszeit für die Bearbeitung der gestellten Aufgaben und Fragen beträgt **5 Stunden** (300 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 8`, `Visual Studio`/`Jetbrains Rider`` oder
- ``Java 21`, `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	25
Businesslogic & Algorithms	30
Presentationlayer	30
Theoriefrage 1	5
Theoriefrage 2	5
Theoriefrage 3	5
Summe	100



Teilaufgabe 1 - Domain Modelling & Persistence

Overview

You must implement model and data access classes which allow interaction between a business application and a relational database.

Your model is about organizing the final exams for Spengergasse. It is about which **exams** every **student** must take; which is the **subject** of the exam; who are the **examiners**; **when** will the exam take place and more.

The required **model classes** should be taken from the provided UML diagram.

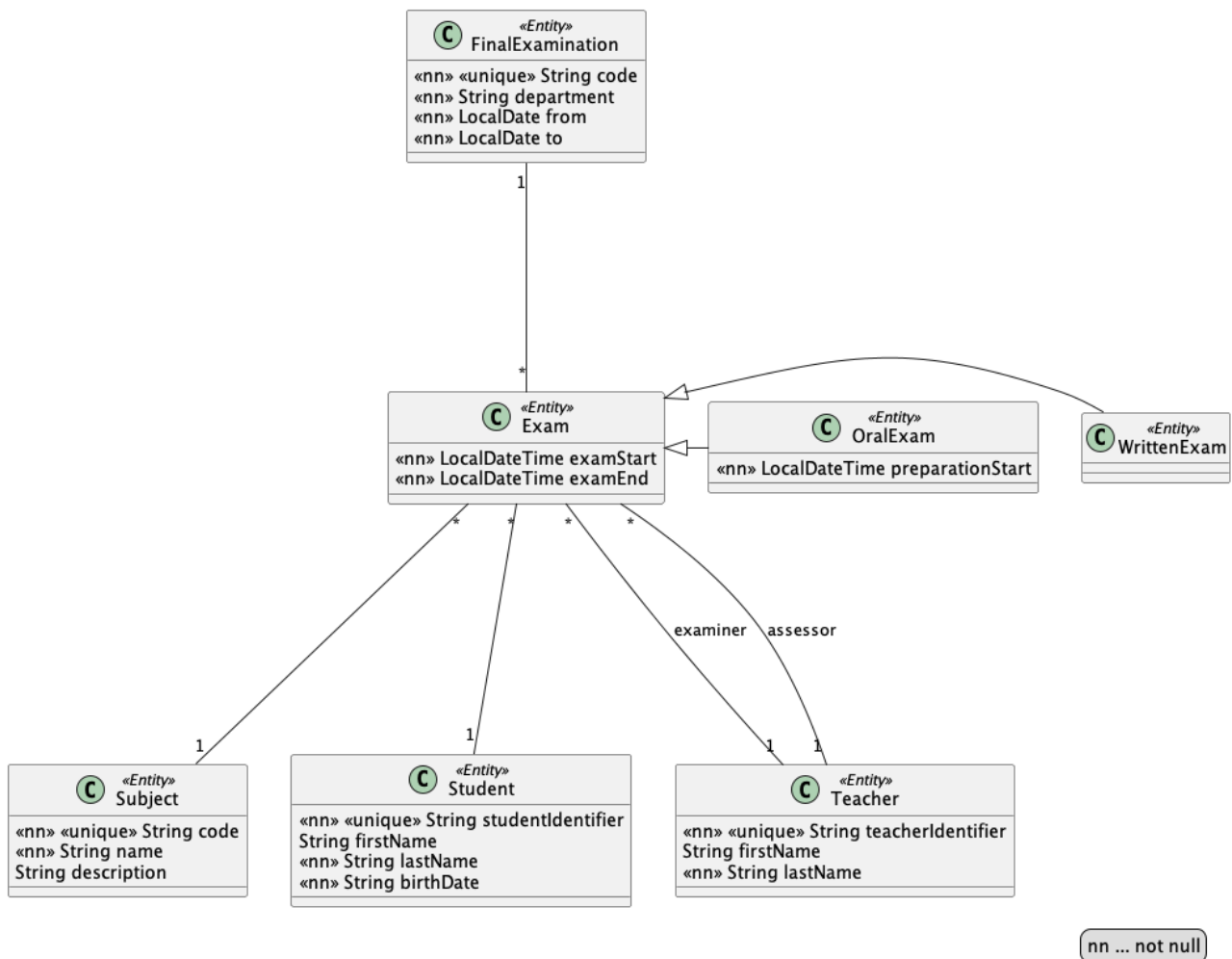
In addition, you must provide **data access classes** and showcase some typical sets of data.

All interactions need to be **tested**!



Class model

Class model to be implemented for the final examination application.





You should implement

- The required model classes (in Java or C#)
- All relations as defined in the UML model
- Mapping definitions for a **relational database** (use `H2` or `Microsoft.EntityFrameworkCore.InMemory`)
- Data access classes for **`CRUD` operations and queries**
- **Unit tests** to ensure the correctness of the implementation
- **Assertions** in your test to verify results

Pay attention on:

- OO design principles
- Correct and meaningful naming of namespace/packages, classes, attributes, methods
- On code style

The following uses cases should be implemented as tests:

- CRUD operations for **students**
- CRUD operations for **teachers**
- CRUD operations for **subjects**
- CRUD operations for **written exams** (all relations should also be filled)
- CRUD operations for **oral exams** (all relations should also be filled)
- a query to receive all **exams** for a **student** (exam is enhanced by teacher, student and subject attributes)
- a query to receive all **exams** for a **teacher** (exam is enhanced by teacher, student and subject attributes)
- a query to receive all **exams** for a **subject** (exam is enhanced by teacher, student and subject attributes)



Teilaufgabe 2 – Business Logic and Algorithms

Assignment of students to elective subjects.

Students are allowed to freely choose an **elective subject** when entering their fourth year.

There is a distinctive list of **elective subjects** with only a limited number of available seats for each. Because of this the student must provide a list of his priorities. The students are then processed in order of their average grades (best grade first).

Each student is placed into the subject of highest priority in case of an available free seat.

The following data is required:

- list of **elective subjects** with the number of seats
- the list of **students**
- for every student the **prioritized list of subjects**
- for every student the **grades**

You should implement

An application which takes the described input data and outputs a list of assigned **students per elective subject**.

- The required basic classes to hold data of elective subjects, students, grades, ...
- The required service class implementing the assignment algorithm
- Any helper classes
- Tests to ensure the correctness of the project

Minimal required test would be with a data set like:

- 3 elective subjects: IOT - 5 seats, BAP - 5 seats and SOS - 3 sets
- 13 students
- three prioritized subjects per student
- a list of three grades per student

You are advised to implement more tests – which will have a positive influence of your marks.



Teilaufgabe 3 – Presentationlayer

You should implement a RESTful API.

Your application is used to organize enrollment of students for optional topics. The business data exchanged should be:

- * `OptionalTopic` with attributes like `topicIdentifier` and `name`
- * `Student` with attributes like `studentIdentifier`, `firstName`, `lastName`, `birthDate`, ...
- * `Enrollment` - with attributes to protocol who enrolled to which topic at which date and time

Remember what RESTful means

- Resources as paths
- Use of HTTP verbs
- Use of HTTP status codes
- Data is exchanged in JSON format
- Input validation
- Correct handling of exceptions

The following **requests** should be implemented:

- Create a new **optional topic**
- Get a complete list of all **optional topics**
- Get all **enrolled students** for a specific **optional topic**
- Create a new **student**
- **Enroll** a **student** for an **optional topic**
- Get all **optional topics** a **student** is enrolled for

You should implement

You must design the required endpoints and data exchange formats to fulfill all six requests as mentioned above.

And you must implement and thoroughly test the required **controller** and **data transfer** classes (all endpoints and data mappings). The **service** classes need to exist and be called from the **controller** but are **not** required to be functional.



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



Theoriefrage 3 – Template Method

Sie haben im Unterricht mehrmals mit dem Design Pattern **Template Method** gearbeitet.

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

Wie kann man dieses Design Pattern in der von Ihnen gewählten Technologie umsetzen?

Wie kommen **classes, interfaces, abstract classes, functional interfaces**, ... zum Einsatz?

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



Klausurprüfung - Nebentermin September 2024
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	/ 25
Business Logic / Algorithms	/ 30
Presentationlayer	/ 30
Dependency Injection	/ 5
Model View Controller	/ 5
Template Method	/ 5
Summe	/ 100