

# Software Engineering I

## Organisation

### Termine

MGH-TINF23	18.09.2024	09:00 - 12:15 13:00 - 16:15	01 Analyse-/Entwurfsmodelle 02 Entwurfsprinzipien
	02.10.2024	09:00 - 12:15 13:00 - 16:15	03 Data Analytics - Lambda/Streams 04 Build-/Testmanagement
	09.10.2024	09:00 - 12:15	Präsentation / Q&A
MOS-TINF23A MOS-TINF23B	19.09.2024	09:00 - 12:15 13:00 - 16:15	01 Analyse-/Entwurfsmodelle 02 Entwurfsprinzipien
	30.09.2024	09:00 - 12:15 13:15 - 16:30	03 Data Analytics - Lambda/Streams 04 Build-/Testmanagement
	07.10.2024	13:00 - 16:15	Präsentation / Q&A
alle Kurse  MGH-TINF23 MOS-TINF23A MOS-TINF23B	11.10.2024	13:00 - 16:15 (O <sup>1</sup> )	- Spezifikation zu Prüfungsleistung u. Projekt - Anforderungsanalyse (→ Projekt) - Vorgehensmodelle (→ Projekt)
	17.10.2024	09:00 - 12:15 (O <sup>2</sup> ) 13:15 - 16:15	Selbststudium   Q&A Data Analytics, Build-/Test Management, Projekt
	18.10.2024	13:00 - 16:15 (O <sup>3</sup> )	Selbststudium   Q&A Projekt
	25.10.2024	13:00 - 16:15 (O <sup>4</sup> )	Selbststudium   Q&A Finalisierung und Abgabe: Data Analytics, Build-/Test Management und Projekt
	31.10.2024	09:00 - 12:15 (O <sup>5</sup> ) 13:15 - 16:15	Präsentation   Q&A Data Analytics, Build-/Test Management und Projekte

### Fristen

01 Analyse-/Entwurfsmodelle	Alle Kurse	27.09.2024
02 Entwurfsprinzipien		
03 Data Analytics - Lambda/Streams		
04 Build-/Testmanagement		
05 Projekt		25.10.2024

1 Online: <https://meet.goto.com/237688229>

2 Online: <https://meet.goto.com/237688229>

3 Online: <https://meet.goto.com/709025317>

4 Online: <https://meet.goto.com/270763197>

5 Online: <https://meet.goto.com/270507733>

- Modellierung grundsätzlich in **Englisch (BE)**.
- Modellierung in **Visual Paradigm 17.2** und **Nutzung** des bereitgestellten **Templates**.
- Modelle müssen **syntaktisch korrekt** sein und die **UML-Spezifikationen genau einhalten**.
- Modell-**Elemente** sollten **gemäß** ihrer **Definitionen / Regeln der UML** verwendet werden.
- Alle **relevanten Aspekte** des **Systems** sollten in dem UML-Modell **abgebildet** sein.
- Modellierung auf der **richtigen Abstraktionsebene**.
- Modelle sollten **hohe Kohäsion** (innerer Zusammenhalt) und **niedrige Kopplung** (geringe Abhängigkeit) der Elemente sicherstellen.
- Modelle sollten eine **klare Rückverfolgbarkeit** zu den **Anforderungen** bieten.
- Modelle sollen **eindeutig** sein, **ohne mehrdeutige Interpretationen**.
- **Diagrammtypen** sollten **konsistent zueinander** sein, insbesondere bei „Project“.
- **Namenskonventionen, Datentypen, und Beziehungen** sollten **durch alle Diagramme** hinweg **einheitlich** verwendet werden.
- Diagramme sollten **klar und einfach zu verstehen** sein, auch für Personen, die nicht direkt an der Entwicklung beteiligt sind.
- Diagramme sollten ein **geordnetes Gesamtbild (Look & Feel)** aufweisen.

- Implementierung grundsätzlich in **Englisch (BE)**.
- **Verbindliche Namenskonventionen zu OOP** sind zu **beachten**.
- Implementierung muss die **definierten Anforderungen korrekt umsetzen**.
- Implementierung sollte auf der **richtigen Abstraktionsebene** erfolgen, indem unwichtige Details verborgen und nur wesentliche Aspekte dargestellt werden.
- **Alle Methoden, Klassen und Schnittstellen sollten Aufgaben gemäß der Verantwortung ohne Fehler erfüllen.** ► Testen mit JUnit / Mockito / BDD.
- **Warnungen** der „Code Inspection“ von IntelliJ sind **weitestgehend zu bereinigen**.
- Im Rahmen der finalen Abgabe ist „**Reformat Code**“ in IntelliJ durchzuführen.

## 01 | Analyse-/Entwurfsmodelle

### Anwendungsfalldiagramm

#### UC01 Fahrkartenautomat

Ein intelligenter Fahrkartenautomat bietet für Kunden die Leistungen [i] Fahrplanauskunft, [ii] Ticketverkauf, [iii] Sitzplatzreservierung und [iv] Abholung vorbestellter Fahrscheine an. Für die Authentifizierung stehen die Verfahren [i] Gesichtserkennung, [ii] Fingerabdruck und [iii] SmartPhone mit TAN zur Verfügung. IT-Spezialisten sind für die Software- und Datenpflege und Techniker für die Fehlerdiagnose verantwortlich.

#### UC02 Getränkeautomat

Ein intelligenter Getränkeautomat für alkoholfreien Gin sowie Tonic bietet für registrierte Kunden die Leistungen [i] Auskunft, [ii] Verkauf, [iii] Pfandrückgabe, [iv] Vorbestellung und [v] Abholung an. Für die Authentifizierung bietet der Getränkeautomat die Verfahren [i] Gesichtserkennung, [ii] Fingerabdruck, [iii] SmartCard mit Pin und [iv] SmartPhone mit TAN an. Servicemitarbeiter sind für [i] Befüllung, [ii] Leerung der Pfandflaschen und [iii] Wartung verantwortlich. Mitarbeiter des Ordnungsamtes können den Automaten zwecks Prüfung öffnen, ob ausschließlich alkoholfreier Gin verkauft wird.

#### UC03 Automat in der Bibliothek

Ein intelligenter Automat in einer Bibliothek bietet für Studierende an der Hochschule die Leistungen [i] Ausleihe, [ii] Rückgabe, [iii] Abfrage Kontostand und [iv] Reservierung an. Für die Authentifizierung bietet der Automat die Verfahren [i] Gesichtserkennung und [ii] Smart-Card mit Pin an. Mitarbeiter der Bibliothek sind für die Datenpflege verantwortlich und können bei Bedarf den Automaten sperren und entsperren.

#### UC04 Automat im Parkhaus

Ein intelligenter Automat in einem Parkhaus bietet für Kunden die Leistungen [i] Ticket bezahlen, [ii] Parkzeit verlängern, [iii] Zeitkarte erwerben, [iv] Zeitkarte verlängern und [v] Zeitkarte vorzeitig zurückgeben an. Für die Bezahlung von Ticket und/oder Zeitkarte stehen die Verfahren [i] Gesichtserkennung, [ii] Fingerabdruck und [iii] SmartPhone mit TAN zur Verfügung. IT-Spezialisten sind für die Software- und Datenpflege und Techniker für die Fehlerdiagnose verantwortlich.

#### UC05 Automat für Impfungen

Ein intelligenter Automat für Impfungen bietet für registrierte Kunden die Leistungen [i] Impfstatus prüfen, [ii] Impfung durchführen, [iii] Anzeige Impfhistorie und [iv] Reservierung an. Für die Authentifizierung bietet die Automat die Verfahren [i] Gesichtserkennung, [ii] Fingerprint, [iii] SmartCard mit Pin und [iv] SmartPhone mit Pin an. Mitarbeiter des Gesundheitsamtes sind für die Befüllung mit Impfdosen verantwortlich. IT-Spezialisten sind für die Software- und Datenpflege und Techniker für die Fehlerdiagnose verantwortlich.

#### ► UC01-UC05 : Modellierung „Use-Case Diagram“

## Aktivitätsdiagramm

### AC01 Automat I

Der Kunde hält seine beiden Hände zwecks Desinfektion in eine Öffnung. Nachdem die Hände desinfiziert wurden, führt der Automat die Gesichtserkennung durch. Nach erfolgreicher Authentifizierung, wird der Sprachassistent und das Lesegerät gleichzeitig aktiviert. Über den Sprachassistenten wird bei dem Kunden die gewünschte Dienstleistung erfragt. Ist die Dienstleistung unbekannt, wird die Frage wiederholt. Nennt der Kunde dreimal hintereinander eine ungültige Dienstleistung, erfolgt vom Automaten eine Sprachmeldung, der Sprachassistent und das Lesegerät werden gleichzeitig deaktiviert und anschließend die Öffnung für die Desinfektion der Hände gereinigt. Die Aktivität terminiert. Wurde die Dienstleistung „Bezahlen“ erkannt, fordert der Sprachassistent den Kunden auf, sein Smartphone an das Lesegerät zu halten. Nachdem der Kunde sein Smartphone an das Lesegerät gehalten hat, wird gleichzeitig eine SMS versandt und die Transaktion protokolliert. Danach werden der Sprachassistent und das Lesegerät gleichzeitig deaktiviert und anschließend die Öffnung für die Desinfektion der Hände gereinigt. Die Aktivität terminiert.

### AC02 Automat II

Der Automat scannt das Gesicht des Kunden. Trägt der Kunde keinen Mund-Nasen-Schutz (MNS), erfolgt vom Automaten die Ablehnung und die Aktivität terminiert. Trägt der Kunde den MNS nicht korrekt, erfolgt vom Automaten eine Aufforderung. Trägt der Kunde nach der dritten Aufforderung den MNS noch nicht korrekt, erfolgt vom Automaten die Ablehnung und die Aktivität terminiert. Trägt der Kunde korrekt den MNS wird ein erstes Fach geöffnet und der Kunde desinfiziert seine Hände. Nach der Desinfektion wird das Lesegerät und der Bildschirm gleichzeitig aktiviert und danach das erste Fach geschlossen. Das zweite Fach wird geöffnet und der Kunde stellt seinen Mehrwegbecher in dieses Fach und hält seine Smart Card an das Lesegerät. Ist die Smart Card abgelaufen oder gesperrt, erfolgt vom Automaten eine Meldung und die Aktivität terminiert. Ist die Smart Card gültig, wählt der Kunde am Bildschirm den gewünschten Kaffee aus und bestätigt die Auswahl mit OK. Das Mahlen der Kaffeebohnen und das Erhitzen des Wassers startet gleichzeitig. Anschließend wird der Kaffee gebrüht und fließt in den Mehrwegbecher. Der Kunde entnimmt seinen Becher mit dem Kaffee. Das zweite Fach wird geschlossen und anschließend desinfiziert. Die Aktivität terminiert.

### AC03 Automat II

An einem Zugang zu einer U-Bahn steht ein Automat für die Zugangskontrolle. Dem Kunden stehen die Möglichkeiten [i] Barzahlung, [ii] Ticket mit Barcode und [iii] Smart Card zur Verfügung. [Barzahlung] Der Kunde wirft Münzen in den Automaten ein. Ist der zu zahlende Betrag entrichtet, werden die beiden Türen geöffnet. Nachdem der Kunde die Zugangskontrolle passiert hat, werden die beiden Türen geschlossen. Die Aktivität terminiert. [Barcode oder Smart Card] Alternativ kann der Kunde das Ticket mit dem Barcode oder seine Smart Card über ein Lesegerät halten. Für die Validierung wird eine Datenbankabfrage an das Rechenzentrum gestellt. Ist das Ticket abgelaufen oder bereits entwertet, wird der Kunde abgewiesen und die Aktivität terminiert. Ist die Smart Card gesperrt oder kein Mindestguthaben von 5 Euro vorhanden, wird der Kunde abgewiesen und die Aktivität terminiert. Wurde das Ticket oder die Smart Card angenommen, werden die beiden Türen geöffnet. Nachdem der Kunden die Zugangskontrolle passiert hat, werden die beiden Türen geschlossen und die Transaktion protokolliert. Die Protokollierung startet zeitgleich lokal in dem Logfile des Automaten sowie in der Datenbank im Rechenzentrum. Die Aktivität terminiert.

#### ► AC01 - AC03 : Modellierung „Activity Diagram“

## Klassendiagramm

### CL01 Automat I

Ein Automat verfügt über einen Bildschirm, ein Lesegerät und eine Kamera. Das Lesegerät kann aktiviert oder deaktiviert werden. Die Kamera kann an- oder ausgeschaltet werden. In je einem Attribut in Lesegerät und Kamera wird der aktuelle Status (an/aus, aktiv/inaktiv) gespeichert. Die Kamera verfügt über drei LED, eine Linse und ein Mainboard. Das Mainboard hat acht Prozessoren (vier Prozessen vom Typ CPU und vier vom Typ GPU). Das Mainboard und die Prozessoren bilden eine physische Inklusion. Die Prozessoren vom Typ GPU sind spezielle CPU mit einer höheren Taktrate. Der Prozessor bietet einen Dienst an, der eine Zeichenkette als Eingabe erwartet und die verarbeitete Zeichenkette zurückgibt.

### CL02 Automat II

Ein Automat hat fünf Fächer, ein Lesegerät und einen Ausgabeschacht. Zu einem Automaten werden die Seriennummer (z.B. 99292833), der Hersteller (AB oder IQ) und das Datum (z.B. 15.09.2024) der letzten Befüllung gespeichert. Je Fach werden die Getränke [i] Wasser, [ii] Apfelsaftschorle, [iii] alkoholfreies Bier, [iv] alkoholfreier Gin und [v] Tonic angeboten. Jedes Fach hat eine Kapazität für maximal 10 Getränke. Ein Fach kann ein Getränk ausgeben. Der Automat hat ein Lesegerät. Das Lesegerät ist mit einem Prozessor und einem Kühler ausgestattet. Der Prozessor hat zwei Kerne. Prozessor und Kerne bilden eine physische Inklusion. Das Lesegerät bietet die Dienste [i] Empfangen einer Zeichenkette, [ii] Senden einer Zeichenkette und [iii] Freigabe an. Freigabe liefert als Rückgabewert – für den Ausgabeschacht – das selektierte Getränk aus dem Fach.

### CL03 Automat III

Ein Geldautomat verfügt über ein Touchpad, ein Lesegerät, eine Kamera, ein Drucker und zwei Fächer für Einzahlungen. Zu einem Automaten werden die Seriennummer (z.B. 838831), die Sicherheitsstufe (S1, S2 oder S3), der aktuelle Status (entsperrt, gesperrt) und das Datum (z.B. 15.09.2024) der letzten Befüllung gespeichert. Das Lesegerät existiert in zwei Varianten. Das Lesegerät der Variante A akzeptiert eine Smart Card. Das Lesegerät der Variante B erweitert die Variante A um das Einlesen eines Fingerabdrucks. In dem Geldautomat ist die Variante B verbaut. Die KI-Kamera nutzt zwei Prozessoreinheiten. Eine Prozessoreinheit verfügt über vier Prozessoren. Jeder Prozessor hat vier Kerne. Kerne und Prozessor bilden eine physische Inklusion. Jeder Prozessor ist mit einem Kühler und einem Temperatursensor ausgestattet. Jede Prozessoreinheit bietet einen Dienst an, der ein Bild als Eingabe erwartet und den klassifizierten Typ Motorrad, Auto, LKW oder NA zurückgibt.

► CL01 - CL03 : Modellierung „Class Diagram“ und Implementierung<sup>6</sup>.

---

6 SOLID-Design Prinzipien sind weitestgehend zu berücksichtigen.

## Sequenzdiagramm

### SQ01 Automat I

Für die Bezahlung eines Tickets sendet der Automat eine Nachricht mit Informationen zu Typ und Preis an das Smartphone des Kunden. Der Kunde bestätigt auf seinem Smartphone diese Transaktion durch die Eingabe einer Pin. Der Automat sendet eine Anfrage an die Bank, ob das mit dem Smartphone assoziierte Bankkonto eine ausreichende Deckung aufweist. Bei ungültiger Pin und/oder unzureichender Deckung erhält der Automat eine Nachricht von der Bank, dass die Transaktion abgelehnt wurde. Der Automat sendet eine Nachricht an den Kunden und die Transaktion wird abgebrochen. Bei gültiger und ausreichender Deckung erhält der Automat eine Nachricht von der Bank, dass die Transaktion genehmigt wurde. Der Automat sendet das Ticket an das Smartphone des Kunden.

### SQ02 Automat II

Der Kunde hält seine Smart Card an den Automaten der Autovermietung. Der Automat fordert den Kunden auf, seine PIN einzugeben. Gibt der Kunde dreimal hintereinander die inkorrekte PIN ein, wird die Smart Card gesperrt. Ist die PIN korrekt, wird der Bildschirm aktiviert. Basierend auf einer Abfrage der Datenbank im Rechenzentrum werden die verfügbaren Fahrzeuge ermittelt. Der Kunde markiert einen Eintrag auf dem Bildschirm und bestätigt die Auswahl mit OK. Im nächsten Schritt fragt der Automat den Benutzer nach der Mietdauer in Tagen. Der Benutzer gibt die Mietdauer ein. Der Automat erstellt einen Beleg und der Schlüssel für das gemietete Fahrzeug wird in einem Fach bereitgestellt.

### SQ03 Automat III

Am Tankautomat hält der Benutzer seine Kreditkarte an das Lesegerät. Der Automat fordert den Kunden auf, seine PIN einzugeben. Gibt der Kunde dreimal hintereinander die inkorrekte PIN ein, wird die Kreditkarte für die Benutzung an diesem Tankautomaten gesperrt. Ist die PIN korrekt, wird der Kunde aufgefordert, die Nummer der Zapfsäule einzugeben. Es stehen die Zapfsäulen 1-4 zur Verfügung. Gibt der Kunde dreimal hintereinander eine Nummer einer nicht existierenden Zapfsäule ein, wird der Vorgang abgebrochen. Wurde die Nummer einer existierenden Zapfsäule eingegeben, wird die korrespondierende Zapfsäule aktiviert. Nach Abschluss des Tankvorgangs übermittelt die Zapfsäule den Benzintyp (Super oder Diesel) und die Menge in Liter an den Tankautomaten. Der Tankautomat erstellt einen Beleg und übermittelt den Gesamtbetrag zwecks Abbuchung an die Bank.

► SQ01 - SQ03 : Modellierung „Sequence Diagram“ und Implementierung<sup>7</sup>.

---

<sup>7</sup> SOLID-Design Prinzipien sind weitestgehend zu berücksichtigen.  
Interaktion ist analog dem Beispiel „Jet Engine“ im Code nachvollziehbar zu kommentieren.

## Zustandsdiagramm

### SD01 Automat I

Initial befindet sich der Automat im Status wartend. Der Kunde hält sein Smartphone an das Lesegerät und der Automat wechselt in den Status „Warten auf Eingabe der PIN“. Nach Eingabe der PIN wechselt der Automat in den Status „Validierung PIN“. Ist die Validierung nicht erfolgreich, wechselt der Automat in den Status „Warten auf Eingabe der PIN“. Wurde dreimal hintereinander die inkorrekte PIN eingegeben, wechselt der Automat in den Status „Abbruch“. Ist die Validierung der PIN erfolgreich, wechselt der Automat in den Status „Warten auf Auswahl“. Wählt der Kunde „Information“, wechselt der Automat in den Status „Anzeige“. Wählt der Kunde „Einlösen“, wechselt der Automat in den Status „Bestellung“. Wählt der Kunde „Abbruch“ wechselt der Automat in den Status „Abbruch“.

### SD02 Automat II

Initial befindet sich der Parkautomat im Status wartend. Durch das Drücken der Taste wird das Touchpad aktiviert. Wählt der Kunde den Menüpunkt Ticket aus, wechselt der Automat in den Status „Warten auf Auswahl Ticket“. Wählt der Kunde den Menüpunkt Bezahlung aus, wechselt der Automat in den Status „Warten auf Ticket“. Der Kunde führt das Ticket in das Lesegerät ein und der Automat wechselt in den Status „Validierung Ticket“. Ist die Validierung des Ticket nicht erfolgreich, wechselt der Automat in den Status „Abbruch“. Ist die Validierung des Ticket erfolgreich, wechselt der Automat in den Status „Bezahlung“.

### SD03 Automat III

Initial befindet sich der Tankautomat im Status wartend. Der Kunde hält seine Kreditkarte an das Lesegerät und der Automat wechselt in den Status „Warten auf Eingabe der PIN“. Nach Eingabe der PIN wechselt der Automat in den Status „Validierung PIN“. Ist die Validierung nicht erfolgreich, wechselt der Automat in den Status „Warten auf Eingabe PIN“. Wurde dreimal hintereinander die inkorrekte PIN eingegeben, wechselt der Automat in den Status „Abbruch“. Ist die Validierung der PIN erfolgreich, wechselt der Automat in den Status „Warten auf Eingabe Nummer Zapfsäule“. Gibt der Kunde dreimal hintereinander eine Nummer einer nicht existierenden Zapfsäule ein, wechselt der Automat in den Status „Abbruch“. Wurde die Nummer einer existierenden Zapfsäule eingegeben, wechselt der Automat in den Status „Warten auf Beendigung Tankvorgang“. Nach Abschluss des Tankvorgangs wechselt der Automat in den Status „Belegdruck“. Nach Entnahme des Beleges wechselt der Automat in den Status „Buchung“. Nach Abschluss der Buchung wechselt der Automat wieder in den Status wartend.

### ► SD01 - SD03 : Modellierung „State Machine Diagram“



X01   UC02   AC03   CL03   SQ02   SD02	X17   UC05   AC01   CL03   SQ02   SD03
X02   UC03   AC01   CL02   SQ03   SD02	X18   UC02   AC02   CL03   SQ02   SD03
X03   UC04   AC03   CL02   SQ03   SD03	X19   UC04   AC01   CL01   SQ03   SD03
X04   UC03   AC02   CL02   SQ02   SD01	X20   UC02   AC01   CL01   SQ01   SD01
X05   UC01   AC03   CL02   SQ02   SD02	X21   UC03   AC03   CL02   SQ03   SD02
X06   UC05   AC01   CL02   SQ01   SD03	X22   UC05   AC01   CL03   SQ03   SD03
X07   UC01   AC01   CL02   SQ01   SD01	X23   UC01   AC02   CL01   SQ02   SD01
X08   UC01   AC02   CL03   SQ02   SD01	X24   UC02   AC02   CL02   SQ01   SD01
X09   UC05   AC02   CL01   SQ01   SD01	X25   UC03   AC03   CL01   SQ03   SD01
X10   UC02   AC02   CL01   SQ02   SD02	X26   UC02   AC03   CL01   SQ01   SD01
X11   UC05   AC02   CL02   SQ01   SD02	X27   UC03   AC02   CL01   SQ01   SD03
X12   UC05   AC01   CL01   SQ02   SD02	X28   UC01   AC01   CL01   SQ02   SD01
X13   UC03   AC03   CL03   SQ03   SD02	X29   UC01   AC02   CL03   SQ01   SD01
X14   UC04   AC01   CL03   SQ01   SD02	X30   UC04   AC03   CL03   SQ02   SD03
X15   UC04   AC01   CL02   SQ03   SD01	X31   UC01   AC02   CL03   SQ02   SD03
X16   UC02   AC03   CL02   SQ03   SD02	X32   UC04   AC03   CL02   SQ03   SD02
X33   UC03   AC02   CL02   SQ02   SD01	

► Abgabe bis spätestens 27.09.2024.

## 02 Entwurfsprinzipien

<b>Refactoring:</b>	Single Responsibility Principle	(Self-Service Terminal)
	Open-Closed Principle	(Self-Service Terminal)
	Liskov Substitution Principle	(Order)
	Interface Segregation Principle	(Self-Service Terminal)
	Dependency Inversion Principle	(Order)

► Modellierung „Class Diagram“ und Implementierung je Design-Prinzip.

► Abgabe bis spätestens 27.09.2024.

---

### Refactoring einer Biometrischen Boarding-Pass-Kontrolle

An einem internationalen Flughafen wurde ein System zur biometrischen Boarding-Pass-Kontrolle implementiert. Das aktuelle System besteht aus einem monolithischen Code mit minimaler Modularisierung, harter Kopplung der Komponenten, und mangelnder Flexibilität für zukünftige Erweiterungen. Die SOLID-Design-Prinzipien (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) wurden bei der Implementierung komplett ignoriert, was zu zahlreichen Problemen wie schlechter Wartbarkeit, Testbarkeit und Erweiterbarkeit geführt hat.

► Modellierung „Class Diagram“ oder Implementierung.

► Abgabe bis spätestens 27.09.2024.

<b>Erwartungen : des Dozenten</b>	Themen aus dem Teil 01 sind souverän zu beherrschen und anzuwenden. Qualitativ sehr gute Lösungen als Kompendium für die Prüfungsleistung.
---------------------------------------	---

### 1. Fahrkartenautomat

Entwickeln Sie ein vereinfachtes System, mit dem Fahrkarten für öffentliche Verkehrsmittel gekauft werden können. Das System sollte die Auswahl von Zielorten, die Berechnung des Preises, die Bezahlung mit Bargeld oder Karte sowie den Druck der Fahrkarte unterstützen.

**Kernfunktionen:** Auswahl der Strecken, Ticketpreisberechnung, Zahlungsintegration.

### 2. Bibliothekssystem

Entwickeln Sie ein vereinfachtes System für eine Bibliothek, das die Verwaltung von Büchern, Ausleihen und Rückgaben unterstützt. Das System sollte Benachrichtigungen für fällige Rückgaben senden können. **Kernfunktionen:** Buchverwaltung, Ausleihe und Rückgabe, Benachrichtigungssystem.

### 3. Kursregistrierungssystem

Entwickeln Sie ein vereinfachtes System, das es Studierenden ermöglicht, sich für Kurse an einer Hochschule zu registrieren, wobei die Kurskapazitäten berücksichtigt werden.

**Kernfunktionen:** Kursauswahl, Kapazitätsmanagement, Benachrichtigungen bei Überbelegung.

### 4. Bankautomat

Entwickeln Sie ein vereinfachtes System für einen Bankautomaten, der Bargeldabhebungen, Kontostandabfragen und Überweisungen ermöglicht. **Kernfunktionen:** Kontoabfrage, Geldabhebung, Überweisungen.

### 5. Warenkorb

Entwickeln Sie einen Warenkorb für einen Online-Shop, der Artikel speichert, deren Gesamtpreis berechnet und den Bestellvorgang abschließt. **Kernfunktionen:** Warenkorbverwaltung, Preisberechnung, Bestellabschluss.

### 6. Kalendersystem

Entwickeln Sie ein vereinfachtes Kalendersystem, das Benutzer dazu befähigt, Meetings zu planen, einzuladen und Benachrichtigungen zu erhalten. **Kernfunktionen:** Meetingplanung, Einladungen, Benachrichtigungen.

### 7. To-Do-Liste

Erstellen Sie eine vereinfachte To-Do-Liste, die es Benutzern ermöglicht, Aufgaben hinzuzufügen, Prioritäten festzulegen und Aufgaben nach Dringlichkeit zu sortieren.

**Kernfunktionen:** Aufgabenverwaltung, Prioritäten.

### 8. Buchungssystem|Kurse

Entwickeln Sie ein vereinfachtes System, das es Mitgliedern eines Fitnessstudios ermöglicht, Kurse zu buchen, zu stornieren und Benachrichtigungen zu erhalten. **Kernfunktionen:** Kursbuchung, Stornierungen, Benachrichtigungen.

### 9. Restaurant-Bestellsystem

Entwickeln Sie ein vereinfachtes Self-Service-Bestellsystem für ein Restaurant, bei dem Kunden Gerichte auswählen, bestellen und den Gesamtpreis berechnen lassen können.

**Kernfunktionen:** Menüauswahl, Bestellverwaltung, Preisberechnung.

### 10. Filmreservierungssystem

Entwickeln Sie ein vereinfachtes System zur Buchung von Kinotickets. Das System sollte es ermöglichen, Filme, Sitzplätze und Vorführungszeiten auszuwählen sowie die Bezahlung durchzuführen. **Kernfunktionen:** Filmauswahl, Sitzplatzreservierung, Zahlungsabwicklung.

### 11. Musik-Streaming-Service

Entwickeln Sie ein vereinfachtes Musik-Streaming-System, das Playlists erstellt und abspielt sowie einfache Such- und Filterfunktionen für Songs unterstützt. **Kernfunktionen:** Playlist-Erstellung, Song-Suche und -Filter.

### 12. Wettervorhersagesystem

Erstellen Sie ein vereinfachtes System, das (zufallsbasierte) Wetterdaten von einem API abrufen und eine tägliche sowie stündliche Vorhersage anzeigen. **Kernfunktionen:** Wetterabfrage, Anzeige der Vorhersage für verschiedene Orte.

### 13. Hotelbuchungssystem

Entwickeln Sie ein vereinfachtes Buchungssystem für Hotels, das es Benutzern ermöglicht, Zimmer zu suchen, zu buchen und die Zahlung durchzuführen.

### 14. Fahrdienst

Erstellen Sie eine vereinfachte Anwendung für einen Fahrdienst, bei dem Benutzer eine Fahrt anfordern, den Fahrpreis berechnen und den Fahrer bewerten können.

**Kernfunktionen:** Fahrtanfrage, Preiskalkulation, Fahrerbewertung.

### 15. Buchungssystem|Konferenzräume

Entwickeln Sie ein vereinfachtes System, das es Unternehmen ermöglicht, Konferenzräume zu buchen, die Verfügbarkeit zu prüfen und Buchungen zu verwalten. **Kernfunktionen:** Raumverfügbarkeit prüfen, Buchungen erstellen, Stornierungen.

### 16. Fundbüro

Erstellen Sie ein vereinfachtes System, das es Benutzer ermöglicht, eine Verlustmeldung aufzugeben, nach gefundenen Gegenständen zu filtern/sortieren und Benachrichtigungen bei Fund zu erhalten. **Kernfunktionen:** Verlustmeldung aufgeben, Suche und Sortieren, Benachrichtigungen.

- Erstellen Sie eine **Hauptapplikation** mit einem aussagekräftigen Gesamtszenario, ggf. auf realitätsnahen Annahmen/Ergänzungen.

- **Anforderungsmanagement | S01**

Spezifikation der Anforderungen mit der SOPHIST-Satzschablone „FunktionsMaster“.

Erstellen Sie ein **geordnetes und übersichtliches Word-Dokument** mit

- Definition und Aufgaben des Funktionsmasters,
- Definitionen/je drei Beispielen zur rechtl. Verbindlichkeit auf Basis der Aufgabestellung,
- Funktionalen Anforderungen auf Basis der Aufgabenstellung,
- den präzise formulierten Anforderungen (Schritte 0-5) auf Basis der Aufgabenstellung.

- **Vorgehensmodell | Agile Vorgehensweise mit Scrum | S01**

Erstellen Sie ein **geordnetes und übersichtliches Word-Dokument** mit User Stories, Akzeptanzkriterien, Product Backlog (einschl. exemplarischer Priorisierung), Skizzierung der Vorgehensweise und des Zusammenspiels der Rollen auf Basis der Aufgabenstellung.

- **Modellierung | S01 und S02**

**Die Hinweise unter „Modellierung | Qualitätssicherung | Checkliste“ sind bindend.**

Modellieren Sie in Visual Paradigm 17.2 und dem bereitgestellten Template:

- |   |             |
|---|-------------|
| - Anwendungsfalldiagramm  | <b>S01,</b> |
| - Aktivitätsdiagramm(e) zu einem nicht-trivialen Anwendungsfall nach Wahl | <b>S02,</b> |
| - Klassendiagramm   | <b>S02,</b> |
| - Sequenzdiagramm   | <b>S02,</b> |
| - Zustandsdiagramm  | <b>S02.</b> |

- **Implementierung | S03**

**Hinweise unter „Implementierung | Qualitätssicherung | Checkliste“ sind bindend.**

Implementieren Sie in **Java 21.0.4** (LTS) und dem bereitgestellten **IntelliJ-Projekt** eine technisch einwandfreie und vorzugsweise realitätsnahe Lösung. Die fundamentalen

**SOLID-Design Prinzipien** sind **weitestgehend** zu **berücksichtigen**. Die Zustände aus dem Zustandsdiagramm sind als Enum bzw. mit dem State-Design Pattern zu integrieren.

- **Testmanagement | S04**

Erstellen Sie ein **geordnetes und übersichtliches Word-Dokument** mit einer Strategie/Übersicht mit Kurzbeschreibung der Tests zu JUnit, Mockito und BDD.

Realisieren Sie im IntelliJ-Projekt die **aussagekräftigen/leistungsfähigen Tests zu**

- **JUnit** (einschl. @Display, @Nested, @Order, @ParameterizedTest und @TestFactory).
- **Mockito** (einschl. verify, times, never, any und inorder),
- **BDD** mit einem vollständigen Szenario zu Normalprozess und Fehlerprozess, Cucumber (einschl. Datenbeispiele in feature-Datei) und JGiven unter Einbeziehung der Aufgabenstellung, funkt. Anforderungen und Akzeptanzkriterien.

- **Lambdas/Streams | S05**

Realisieren Sie im IntelliJ-Projekt auf Basis der Aufgabenstellung nachfolgend aufgeführte aussagekräftige **Abfragen mit den Rahmenbedingungen** [01] 3 Filter (and, or, not), [02] Filter und groupBy ohne count [03] partitioningBy, [04] Sortierung mit Comparator nach zwei Attributen und [05] Filter und zwei ineinandergeschachtelte groupBy mit/ohne count.

Team	Zuordnung
Z01   1139344 9092563 7267944 4213003	8   Buchungssystem Kurse
Z02   8173419 2430672 4551906 1893630 5922305	16   Fundbüro
Z03   2436495 4458459 5239938 6120345	1   Fahrkartenautomat
Z04   7192477 2244032 1244358 2769545	5   Warenkorb
Z05   7087408 7050271 1799818 3141911	14   Fahrdienst
Z06   3260935 8466840 3498287 2576582	3   Kursregistrierungssystem
Z07   7335909 6464626 4749590 9830011	4   Bankautomat
Z08   9796557 2058677 9796557 4718461	6   Kalendersystem
Z09   1086439 4302297 1216113 9207280	7   To-Do-Liste
Z10   1634406 9833220 3181933 8102267	11   Musik-Streaming-Service
Z11   3844189 8595958 6812340 7976979	2   Bibliothekssystem
Z12   9497063 7162545 3244269 5102447 9346726	10   Filmreservierungssystem
Z13   6564425 4089333 4445357 8832277	12   Wettervorhersagesystem
Z14   6289380 7359548 1279358 8898914	13   Hotelbuchungssystem
Z15   1478661 1299169 7557437 6575277	15   Buchungssystem Konferenzräume
Z16   8330177 8765488 8855870 9822862 8531610	9   Restaurant-Bestellsystem

► **Abgabe einer vollständigen 7-Zip-Datei [team\_id].7z mit**

- Word-Datei „**requirements\_management.docx**“ zu „Anforderungsmanagement | S01“
- Word-Datei „**scrum.docx**“ zu „Vorgehensmodell | Agile Vorgehensweise mit Scrum | S01“
- Visual Paradigm-Datei „**uml.vpp**“ zu „Modellierung | S01 und S02“
- 7-Zip-Datei „**intellij.7z**“ mit der vollständigen und technisch einwandfreien Implementierung sowie den korrespondierenden Tests in JUnit, Mockito, BDD: Cucumber und JGiven zu „Implementierung | S03“ und „Testmanagement | S04“.
- Word-Datei „**test\_management.docx**“ zu „Testmanagement | S04“.
- Ggf. 7-Zip-Datei „**data\_analytics.7z**“ mit der vollständigen und technisch einwandfreien Implementierung sowie den korrespondierenden Tests in JUnit zu „Lambdas/Streams | S05“.

**mit qualitativ sehr guten Lösungen bis spätestens 25.10.2024.**