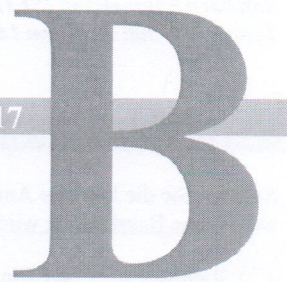


1 (7) 1509

SOFTWARE ENGINEERING 1 KLAUSUR SoSe 2017



Name, Vorname [REDACTED]

Matrikelnummer [REDACTED]

Pseudonym [REDACTED]

freiwillig, wenn Sie möchten, dass Ihr Klausurergebnis im Internet veröffentlicht wird

Platznummer 4.2
 wird vom Betreuer (evtl.) zu Beginn der Klausur vergeben

Erster (X) Zweiter () Letzter () Versuch

Note für Punkte	0 bis 55, ab 60, ab 70, ab 75, ab 80, ab 85, ab 90, ab 95, ab 100, ab 105, ab 110
<i>1,0 für Mo ↓</i>	5,0 4,0 3,7 3,3 3,0 2,7 2,3 2,0 1,7 1,3 <u>1,0</u>

Lesen Sie zunächst alle Aufgaben sorgfältig durch. Sollten Sie Fragen haben, **können Sie diese in den ersten zehn Minuten laut stellen**. Spätere Fragen sind nicht mehr zulässig, denn laute Fragen stören, und leise Fragen widersprechen dem Gleichbehandlungsprinzip. Es sind keine Hilfsmittel zugelassen. Schreiben Sie Ihre Lösungen auf dieses Aufgabenblatt (beachten Sie auch die Rückseite!) sowie **auf höchstens zwei einseitig beschriebene Zusatzblätter** mit Ihrem Namen; kennzeichnen Sie die Aufgabennummer eindeutig. Schreiben Sie am besten mit Kugelschreiber (Bleistift ist nicht zulässig!). Für falsche oder unverständliche Lösungen bekommen Sie grundsätzlich keine Punkte. Wenn aber aus Ihren Notizen oder Bemerkungen ersichtlich ist, dass Ihr Gedankengang korrekt war, können Sie Teilpunkte erreichen. Sie verlieren diese Möglichkeit jedoch, wenn abschreiben oder Kommunikation während der Klausur nachgewiesen werden kann. Der Kern der Fragen wurde *kursiv* gesetzt. **Die Aufgaben sind ungefähr gleich aufwändig und jeweils 40 Punkte wert.**

Bearbeiten Sie bitte **unbedingt** die Aufgaben 2 und 4 (siehe Rückseite) und nur **eine** der beiden Aufgaben 1 und 3. Kennzeichnen Sie **deutlich**, welche Aufgabe Sie ausgewählt haben. Geben Sie zum Aufgabenblatt höchstens **ZWEI EINSEITIG BESCHRIEBENE** Zusatzblätter ab. Weitere Blätter werden nicht bewertet.

1. AUFGABE

Viele Informationen des Pflichtenhefts aus Ihrem ersten Meilenstein werden in den Geschäftsprozessbeschreibungen (GP-Überblicksdiagramm, GP-Schablonen, Aktivitätsdiagramme) des zweiten Meilensteins *erneut verwendet*. Erläutern Sie diese **Informationsbezüge** (beispielsweise was? woher? wohin?) **tabellarisch in Stichpunkten (kein Fließtext!)**.

Lösung auf dem Extrablatt Nr. Punkte (0 5 10 15 20 25 30 35 40)

2. AUFGABE UNBEDINGT BEARBEITEN!

Modellieren Sie ein Fachklassendiagramm für folgenden Sachverhalt: Eine Hochschule bietet Lehrveranstaltungen für *Erstsemester* und für *Höhere Semester* an. Diese werden *jedes Semester wiederholt*. Die Veranstaltungen besitzen eine *Beschreibung ihres Themeninhalts*. Eine *Erstsemesterveranstaltung* kennt zusätzlich ihre *maximal zulässige Teilnehmerzahl*. Ist diese Zahl erreicht, werden weitere Teilnehmerbelegungen *auf einer Warteliste vermerkt*. Im Diagramm bietet sich die Verwendung einer *{or}-Restriktion* und mindestens einer Vererbung an.

Lösung auf dem Extrablatt Nr. 1 Punkte (0 5 10 15 20 25 30 35 40)

3. AUFGABE

Das Konzept der *Polymorphie* eröffnet wichtige Möglichkeiten innerhalb der objektorientierten Softwareentwicklung. Bearbeiten Sie **tabellarisch in Stichpunkten (kein Fließtext!)**:

- Beschreiben Sie die drei *Unterarten* der Polymorphie. Welche Bedeutung haben in diesem Zusammenhang die Begriffe *Überschreiben* und *Überladen*? Und was ist eigentlich *Überdecken*?
- Was ist der Unterschied zwischen *Early Binding* und *Lazy Binding*, und anhand welcher Merkmale bestimmt die Java Virtual Machine (JVM) zur Laufzeit, welche Methode sie aufruft?
- Was ist der Unterschied zwischen *Operation* und *Methode*?

Lösung auf dem Extrablatt Nr. 2 Punkte (0 5 10 15 20 25 30 35 40)

Achtung, beachten Sie auch die Rückseite!

4. AUFGABE (Richtig-/Falsch-Fragen)

UNBEDINGT BEARBEITEN!

Kreuzen Sie die korrekte Antwort an und geben Sie Ihre Begründung in Stichworten dazu. Jede Einzelfrage ist fünf Punkte wert. Ohne Begründung wird Ihre Antwort nicht bewertet.

- Richtig
 Falsch

4.1 Ein Pflichtenheft enthält nur die *Verpflichtungen des Auftragnehmers* gegenüber dem Auftraggeber.

Grund Es enthält auch Konkretionen die nicht zw.-grad verpflichtet sind. *Begründung in Ordnung* (5)

- Richtig
 Falsch

4.2 Das Objekt einer *assoziativen Klasse* existiert nur so lange wie die *Assoziation* zwischen den Objekten, mit der es verbunden ist.

Grund Ohne Assoziation kann kein Objekt bestehen. *genau!* (5)

- Richtig
 Falsch

4.3 Im *Analysemodell* wird die Objektverwaltung nicht modelliert, sondern davon ausgegangen, dass *jede Klasse ihre Objekte selbst verwaltet* (beispielsweise, um eine Suchfunktion zu realisieren).

Grund Das wäre zu fest versch. (5)

- Richtig
 Falsch

4.4 Das polymorphe Konzept der *Überdeckung* gilt nur für *Attribute*.

Grund Ein Attribut der Untv Klasse überdeckt ein Attribut der Oberklasse. *Methoden, Klassen, etc. sind nicht betroffen* (5)

- Richtig
 Falsch

4.5 Ein Geschäftsprozess muss stets einen *sichtbaren, quantifizierbaren oder qualifizierbaren Einfluss* auf die Systemumgebung haben.

Grund Es muss einen Sinn ergeben einen GP zu modellieren, ohne einen Einfluss wäre ein GP unnötig. (5)

- Richtig
 Falsch

4.6 Der *Rückgabety*p einer Operation gehört *nicht* zur Signatur.

Grund operations name (Übergabeparameter, Datentyp): Rückgabety ist + 7P *Begründung unzureichend* (10)

- Richtig
 Falsch

4.7 Die im Pflichtenheft dokumentierten *Abgrenzungskriterien* beschreiben *spezielle Leistungsmerkmale* der Softwareanwendung, die in vergleichbaren Softwareprodukten im allgemeinen nicht vorkommen.

Grund Sie beschreiben klare Merkmale die in der zu entwickelnden SW nicht vorhanden (5)

- Richtig
 Falsch

4.8 Im Fachklassendiagramm ist die *Aggregation* ein *Spezialfall der Assoziation*.

Grund die Aggregation ist eine teil-Ganzes Assoziation. (5)

Punkte (0 5 10 15 20 25 30 35 40)

④ Ad hoc Polymorphism

wenn 2 Klassen die in keine direkte Vererbungslinie stehen die gleiche Methode ~~implementieren~~ besitzen. (5)

Vererbte Polymorphism

ist wenn 2 Klassen, die in direkter Vererbungslinie stehen, die gleiche Methode deklarieren, aber unterschiedlich implementieren. (5)

Parametrische Polymorphism

ist wenn in 1 Klasse mehrere gleichnamige Methoden vorhanden sind, die sich ausschließlich durch unterschiedliche Übergabe Parameter unterscheiden. (5)

Überdecken

Überdecken ist wenn ein Attribut der Unterklasse das gleichnamige Attribut der Oberklasse überdeckt. Auf ein Attribut der Oberklasse, kann mittels type casting aus der Unterklasse zugegriffen werden. (5)

② Early binding

(5) } bereits beim compilieren wird beim Early Binding etwas getan. **aber was denn?**

beim Lazy Binding jedoch erst zur Laufzeit. **was?**

Die JVM kann das Early Binding benutzen wenn sie alle nötigen Informationen zur Verfügung hat **und welche sind das?**

Hier haben Sie ein bisschen "um den heißen Brei" herumzuphischen. :)

③ Der Unterschied zwischen Operation und Methode.

erst die Operation ist die Deklaration einer Methode. Sie muss nicht wissen wie sie funktioniert. (5)

eine Methode ist die Implementierung. Sie enthält den komplizierten Code :-)

(35) Prima!

