



Probeklausur
zur Vorlesung **Software Engineering**
im Wintersemester 2023

Dauer: 90min

Textaufgabe 1:

Textaufgabe 2:

Multiple Choice:

Vorname(n):

Nachname(n):

Matrikelnummer:

Studiengang:

z.B. Bachelor

Studienfach:

z.B. Medizininformatik

Teamprojekt bereits belegt?

☐ Nein

☐ Ja, im Semester:

Diese Klausur besteht aus offenen Aufgaben und Multiple-Choice-Fragen. Jede der Multiple-Choice-Fragen hat genau eine richtige Antwort. Für jede Multiple-Choice-Frage können Sie keine Antwort, eine Antwort, oder mehrere Antworten geben, indem sie die entsprechenden Felder ankreuzen.

Wir benutzen die Konvention dass ☒ "Antwort ausgewählt" bedeutet, und sowohl ☐ als auch ☐ bedeuten "Antwort **nicht** gewählt". Ein Kreuz neben einem ausgemalten Kästchen (☒) bedeutet ebenfalls "Antwort ausgewählt".

Hinweise zur Ankreuzstrategie:

- Wenn Sie die Antwort sicher wissen, kreuzen Sie nur diese an.
- Wenn Sie sich nicht sicher sind aber einige Antworten ausschliessen können, kreuzen Sie die Antworten an, die korrekt sein könnten.
- Wenn Sie überhaupt nicht wissen, was die Antwort sein könnte und nichts ausschliessen können, kreuzen Sie nichts an.

Die Antworten werden so bewertet, dass zufälliges Raten null Punkte ergibt, also eine falsche Antwort Minuspunkte gibt.

Aufgabe 1 [20 Punkte insgesamt]:

Aufgabe 1.1 [3 Punkte]: Lesen Sie folgenden Code. Wenn Sie den Code von einer höheren Abstraktionsebene aus betrachten, was sind die wesentlichen 4 Berechnungsschritte? Folgen Sie dem Beispiel, markieren und beschriften Sie die restlichen Berechnungsschritte.

<pre> 1 def main(filename: String): Unit = { 2 var lines: List[String] = readLinesFromFile(filename) 3 var gradeCounter: mutable.HashMap[String, Int] = mutable.HashMap() 4 var grade: String = null 5 lines.foreach { line => 6 grade = line.split(",")(1) // grade is stored in second column 7 if (gradeCounter.contains(grade)) { 8 gradeCounter.update(grade, gradeCounter(grade) + 1) 9 } else { 10 gradeCounter.update(grade, 1) 11 } 12 } 13 val possibleGrades 14 = List("mit_auszeichnung", "bestanden", "nicht_bestanden") 15 var count: Int = 0 16 possibleGrades.foreach { grade => 17 if (gradeCounter.contains(grade)) { 18 count = gradeCounter(grade) 19 } 20 println(grade + ":" + " " .repeat(count)) 21 count = 0 22 } 23 }</pre>	<p>1. <i>Dateiinhalt lesen</i></p> <p>2.</p> <p>3.</p> <p>4.</p>
--	--

Beispiel-Eingabedatei:

```

Paul,bestanden
Emilia,mit auszeichnung
Karl,nicht bestanden
Marie,bestanden
```

Aufgabe 1.2 [3 Punkte]: Identifizieren Sie drei verschiedene Code-Smells im obigen Code und markieren Sie diese im Code-Beispiel mit Pfeilen, Unterstreichungen oder anderen Markierungen. Benennen Sie den Smell mit einem Begriff aus der Vorlesung und beschreiben Sie kurz, warum dieser Code-Smell problematisch sein kann. Achten Sie darauf, dass sich die Smells *nicht* auf Ein- oder Ausgabe (IO) beziehen.

Aufgabe 1.3 [12 Punkte]: Refaktorisieren Sie den Code, um die von Ihnen identifizierten Code-Smells zu beseitigen. Verwenden Sie “...” oder ähnliche Symbole, um den Code auf einer abstrakten Ebene zu skizzieren. Vermeiden Sie das Kopieren von Code und konzentrieren Sie sich auf die wichtigsten Änderungen. Ihre Lösung muss nicht vollständig korrekter Scala-Code sein.

Aufgabe 1.4 [2 Punkte]: Erklären Sie in wenigen Worten, warum Ihr Entwurf besser ist als das Original. Verwenden Sie Konzepte aus der Vorlesung.

Aufgabe 2 [5 Punkte]: Teilen Sie die möglichen Ein- und Ausgaben der Funktion `distinct` in mindestens 3 verschiedene Partitionen (Äquivalenzklassen) ein und beschreiben Sie diese kurz (mit dem Ziel Black-Box Tests zu schreiben – nicht Teil der Aufgabe).

```
1 /** Returns a new list of elements without any duplicates. */
2 def distinct(nums: List[Int]): List[Int] = ???
```

Frage 1

Was ist eine “Magic Number”?

- ☐ *a* Ein Zahlenliteral, das direkt im Code verwendet wird.
- ☐ *b* Die maximale Anzahl an Methoden, damit eine Klasse noch Kohäsion aufweist.
- ☐ *c* Eine `int`-Konstante in einem Programm mit einem schlecht gewählten Namen.
- ☐ *d* Eine `int`-Konstante, die mehr als einmal in einem Programm verwendet wird.

Frage 2

In welche Art von Notationsarten ist UML einzuordnen?

- ☐ *a* Formelle, graphische Modellierungssprachen
- ☐ *b* Formelle, textuelle Modellierungssprachen
- ☐ *c* Informelle, graphische Modellierungssprachen
- ☐ *d* Informelle, textuelle Modellierungssprachen

Frage 3

Gegeben ist die Subtypenbeziehung `Book <: Buyable <: Item`. Welche der folgenden Subtypenbeziehungen gilt?

- ☐ *a* `(Book => String) <: (Item => String)`
- ☐ *b* `(String => Item) <: (String => Buyable)`
- ☐ *c* `(Item => Item) <: (Book => Book)`
- ☐ *d* `(Book => Book) <: (Item => Item)`
- ☐ *e* `(String => Book) <: (String => Item)`

Frage 4

Wer ist in Design-by-Contract dafür verantwortlich die Vorbedingung zu erfüllen?

- ☐ *a* Der Nutzer
- ☐ *b* Die Implementiererin
- ☐ *c* Beide Parteien
- ☐ *d* Niemand