

Probeklausur	_	Textaufgabe 1:		
zur Vorlesung Software Engineer im Wintersemester 2023	ing	Textaufgabe 2:		
Dauer: 90min		Multiple Choice:		
Vorname(n):				
Nachname(n):				
Matrikelnummer:				
Studiengang:			z.B.	Bachelor
Studienfach:			z.B.	Medizininformatik
Teamprojekt bereits belegt?	Nein Nein			
	☐ Ja, im Semester: ☐			
Diese Klausur besteht aus offenen A Fragen hat genau eine richtige Antweine Antwort, oder mehrere Antwor	wort. Für jede Multiple-	-Choice-Frage können Sie k	eine	Antwort,
Wir benutzen die Konvention dass bedeuten "Antwort nicht gewählt".				s auch = bedeutet

Hinweise zur Ankreuzstrategie:

ebenfalls "Antwort ausgewählt".

- 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]:

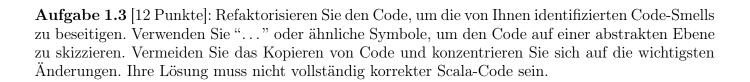
Aufgabe 1.1 [3 Punkte]: Lesen Sie folgenden Code. Wenn Sie den Code von einer höheren Abstraktionsebene aus betrachten, was sind die wesentlichen 4 Berechnungsschritte? Folgen Sie dem Beispiel, markieren und beschriften Sie die restlichen Berechnungsschritte.

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

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 1.4 [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 ${\tt Book}<:{\tt Buyable}<:{\tt Item}.$ 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?
$a \square$ Der Nutzer $b \square$ Die Implementiererin $c \square$ Beide Parteien $d \square$ Niemand