

Programmierung I – 1. Klausur WS 21/22 – bei Herr Prof. Dr. Müller

100 von 112 Punkte für 1.0 erreichen

Aufgabe 1 (7P)

7 Aufgaben: Programmschnipsel: Frage was System.out.println(...) ausgegeben wird

Bsp.: System.out.println("10" + 20) Ausgabe: 1020

Aufgabe 2 (6P)

6 Aufgaben: Initialisierung von Datentypen, Frage nach korrekter Syntax, wenn ja, Wert angeben, wenn nein, den Grund angeben

Bsp.: string test = "abc"; (Fehler, String muss großgeschrieben werden)
Double d = 6,5; (Fehler, double Zahlen werden mit einem Punkt initialisiert)

Aufgabe 3 (14P)

- a) Programm auf Blattpapier schriftlich erklären – (6P)
- b) Programm ohne while-Schleife schreiben, z.B. mit einer for-Schleife (8P)

Aufgabe 4 (8P) – Stringverarbeitung

Eine Methode bekommt String s1 und String s2 übergeben. Zu überprüfen war, ob der String s2 in s1 enthalten ist und dann der Teilstring t vor s2 und hinter s2 wieder stand, also quasi die Form hat:

s1 = (t + s2 + t) - **Bsp.:** s1 = abcxyzabc , s2 = xyz

Wenn das gestimmt hat, sollte t übergeben werden, ansonsten null und wenn t = "" war, dann sollte "" zurückgegeben werden

Aufgabe 5 (8P) – Rekursion

Rekursion: einfach nur die verschiedenen Fälle abschreiben und wieder rekursiv aufrufen lassen

Aufgabe 6 (12P) – Arrays

Es wurde ein Array einer Methode übergeben und man sollte wieder ein Array zurückgeben, allerdings sollten dann alle Werte umgedreht sein.

Falls das Array eine Null-Referenz ist, sollte auch null zurückgegeben werden

Aufgabe 7 (15P) – Collections

Man hatte eine ArrayList <ExamResults> (ExamResults waren wiederum Objekte, bestehend aus Person, Note und Fach) und sollte dafür 2 Methoden schreiben:

1. Eine Methode die neue ExamResults einfügt
2. Eine Methode, welche eine „**int mark**“ übergeben bekommt, die alle Objekte der ArrayList durchläuft (Iterator bietet sich da an) und dann alle Personen in eine neue ArrayList einträgt, welche die Note „**mark**“ haben.
Anschließend sollte die ArrayList<String> mit den Personennamen zurückgegeben werden

Aufgabe 8 (15P) – Exceptions

Man bekam einen String der Form : „Zahl“ gefolgt von „> oder <“ und dann wieder eine „Zahl“. Getrennt waren die Strings nur mit Leerzeichen, sodass sich ein StringTokenizer anbietet, in Kombination mit der trim() Funktion.

Man sollte überprüfen, ob der String diesem Format entspricht, wenn nein, sollte eine RuntimeException geworfen werden, wenn ja, dann sollte überprüft werden, ob die Werte der beiden Zahlen auch mit dem „> oder <“ Zeichen übereinstimmen. Wenn das der Fall war, dann sollte „valid“ zurückgegeben werden, ansonsten zwei Texte, wie „links ist nicht größer als rechts“ oder analog „links ist nicht kleiner als rechts“.

Aufgabe 9 (15P) – Vererbung

Man hatte 4 Klassen und musste ein Interface schreiben, wovon eine abstrakte Klasse „Song“ geerbt hat und dann sollte man eine Klasse „Cover“ schreiben, welche von Song erbt.

Es wurde ein Konstruktor verlangt, sowie Initialisierung von den abstrakten Methoden, sowie ein @Override einer Stringmethode.

Aufgabe 10 (12P) – Liste

Abgespeckte Version von der Vorlesung, man hatte nur die Klasse Elem und List (mit String Objekten) und musste folgende Methoden schreiben:

1. Eine Methode, die einmal durch die Liste läuft und alle Stringwerte zu einem String zusammenfasst, sodass man jeden Stringwert durch ein „+“ getrennt hat. Ein Beispiel für die Form ist: was+wie+wer+wo+wann
Verlang war nur, dass hinter dem letzten String kein Plus mehr gesetzt wird
2. Eine Methode size(), welche beim Durchlaufen einmal zählen sollte, wie viele Objekte in der Liste drinnen sind