

Aufgabe 1: Primitive Datentypen und Termauswertung
Siehe Anhang

Aufgabe 2: Deklarationen und Casts
Siehe Anhang

Aufgabe 3a: Quelltext verstehen
Siehe Anhang

Aufgabe 3b: Den Quelltext aus Aufgabe 3a so umformulieren, das er genau die gleichen Ergebnisse liefert, das Programm jedoch keine while-Schleife enthält.

Aufgabe 4: Strings

man sollte eine Methode schreiben, welche 2 Strings als Input hat.

Beispiel: s1= abcDEFabc, s2 = DEF. Man sollte bei dieser Eingabe abc zurückliefern, heißt man muss überprüfen ob der vordere String und der hintere String gleich sind. Ist dies der Fall, so sollte man diesen zurückgeben. s1 = abab, s2 = „“ → Ausgabe: ab,
weiteres Beispiel: s1: abaaba, s2 = aa → Ausgabe null.

Aufgabe 5: ein Array wird eingelesen, die Aufgabe ist es eine KOPIE dieses Arrays und dessen Elemente in der Reihenfolge vertauscht. Beispiel: [1,2,3,4,5] → [5,4,3,2,1]. Dabei darf das eingelesene Array nach dem Aufruf der Methode sich nicht verändert haben.

Aufgabe 6: Rekursion

die Aufgabe fällt mir nicht mehr ein, da die Beschriebene Rekursion sehr kompliziert war (ich weiß nichtmals was diese Rekursion gemacht hat). Die Aufgabe ließ sich jedoch innerhalb einer Minute bewältigen, dazu ist es unwichtig was die Rekursion macht, man muss diese nur stur implementieren!

Aufgabe 7: Collecitons, **Aufgabe 8:** Vererbung

→ Diese Aufgaben waren komplex und Materialbezogen, diese kann ich hier leider nicht mehr sinnvoll wiedergeben.

Aufgabe 9: Exceptions

Seltsamerweise wurde diese Aufgabe nach 2 Tagen auf Moodle wieder freigeschaltet (die anderen leider nicht). Daher habe ich die exakten Dateien, welche auch in der Klausur gegeben waren (siehe Anhang)

Aufgabe war es: Evaluation.java darf nicht geändert werden

Man muss eine Methode String checkComparison(String calculation){} schreiben. Diese Methode soll einen String auswerten. Richtige Kombinationen für einen String sind:

„0 < 1“

„1 > 0“

„-1 < 0“

„+1 > 1111“

„234 < 435“

„-233 < -345“

falsche Kombinationen sind:

„0< 1“

„+0 < 1“ //2 Leerzeichen

„0 = 0“

Die Aufgabe ist nun:

- bei einer Eingabe mit falscher Kombination soll eine RuntimeException geworfen werden, welche „syntax error“ ausgibt.
- ist die Eingabe eine richtige Kombination, jedoch macht die Eingabe logischerweise keine Sinn (also „+1 > 11111“), so ist die Rückgabe entweder
 - x is not greater than y
- oder
 - x is not smaller than y
- ist die Eingabe eine richtige Kombination und macht auch logischerweise Sinn (also „+1 < 0), so soll die Rückgabe „valid“ sein

Im Ordner „Lösungen“ befindet sich meine Lösung zu dieser Aufgabe, die Lösung hat volle Punktzahl erreicht.

Aufgabe 10: Listen

Auch diese Aufgabe ist zu Komplex, als das ich sie hier verständlich ausdrücken könnte.

Letzte Worte meinerseits:

- lese dir die Aufgaben zuerst komplett durch, bei manchen Aufgaben ist dies für das Verständnis notwendig!
- 2h Bearbeitungszeit, 112 zu erreichende Punkte
- mache dich vertraut damit wie du neue Projekte auf Eclipse/ IntelliJ anlegst, öffnest du IntelliJ auf dem Uni Rechner hat dieser kein Projekt und du musst selber eins anlegen (ich denke nicht, dass das ein Problem sein sollte, aber lieber zu viel gesagt als zu wenig)