

Programmierung I Klausur WiSe 2018/2019

Aufgabe 1 (schriftlich):

Man bekommt Rechnungen mit unterschiedlichen Zahltypen gegeben und muss das Ergebnis, den Datentyp des Ergebnisses und eine kurze Begründung dafür abgeben, in dieser Aufgabe mit eher untypischen Typen, z.B. Hexadezimal, Long, Float etc.

Aufgabe 2 (schriftlich):

Man bekommt wie in Aufgabe 1 eine Rechnung/Zuweisung, bestehend aus 2-4 Zahlen verschiedener Typen und muss das Ergebnis angeben. Falls die Rechnung nicht funktionieren würde (also zu einem Error führen würde), muss man eine kurze Begründung angeben, warum das Programm hier abstürzt. Ein paar Beispiele für die Rechnungen:

- 1) `int a = 1.999;`
- 2) `int double = 3.0;`
- 3) `"12"+10`
- 4) `int b = 1 / 3.0 * 2;`

Aufgabe 3:

Teil a) Schriftlich, Code auf Papier gegeben, man muss kurz erklären, was der Code macht. Der Code in der Klausur war eine Methode mit einem `int[] feld` und einem `int parameter`. Die Methode hatte eine `int summe` gegeben. Mit dem `parameter` wurde ein `switch` aufgerufen, falls `parameter == 0`, wurde mit einer `for-each-Schleife` der jeweilige Feldeintrag zur `summe` addiert, falls `parameter == 1`, dann wurde mit einer `for-each-Schleife` zuerst der Feldeintrag quadriert und dann zur `summe` addiert. Am Ende der Methode wurde `summe` zurückgegeben. Code:

```
public int methode(int[] feld, int parameter) {
    int summe = 0;
    switch (parameter) {
        case 0: {
            for(int i : feld) {
                summe += i;
            }
        }
        case 1: {
            for(int i : feld) {
                summe += i*i;
            }
        }
    }
    return summe;
}
```

Teil b) Den gleichen Code implementieren, aber ohne `for`-Schleifen

Aufgabe 4:

Man soll eine Methode schreiben, die einen String `str` einliest, falls dieser String zweimal genau aus dem gleichen String `t` besteht, also `str = t + t`, dann soll String `t` zurückgegeben werden, ansonsten `null`. (Beispiel: "TestTest" soll "Test" zurückliefern, "Auto" soll `null` zurückliefern) Lösung:

```
public static boolean stringTest(String str) {
    int haelfte = str.length() / 2;
    String teil1 = str.substring(0, haelfte);
    String teil2 = str.substring(haelfte, str.length());
    return teil1.equals(teil2);
}
```

Aufgabe 5:

Feld mit Integers wird in eine Methode übergeben, falls ein Feldeintrag **genau** zweimal im Feld vorkommt, soll *TRUE* zurückgegeben werden, sonst *FALSE*. Lösung:

```
public static boolean feldTest(int[] feld) {
    for(int i=0; i < feld.length; i++) {
        int eintrag = feld[i];
        int counter = 0;
        for(int j=0; j < feld.length; j++) {
            if(feld[j] == feld[i]) counter++;
        }
        if(counter == 2) return true;
    }
    return false;
}
```

Aufgabe 6:

Man muss zwei Klassen Professor und Student schreiben. Professor hat einen Namen, Fachbereich und eine int-Variable **zeit** (diese werden im Konstruktor gesetzt, wird von der Evaluation automatisch getan). Jeder Student bekommt im Konstruktor einen Namen, Fachbereich und einen Professor **prof** übergeben. Falls **prof** den gleichen Fachbereich besitzt, wie der gerade konstruierte Student, dann wird dieser Student in eine ArrayList des Typs Student in der Professor-Klasse eingefügt. Man soll dann eine Methode **beratung** schreiben, die die ArrayList durchläuft und einen String der Form „Student xyz wird beraten“ ausgibt, falls der Professor noch **zeit** hat, ansonsten wird der String „Professor xy hat keine Beratungszeit mehr übrig“ ausgegeben.

Aufgabe 7:

Man bekommt Strings der Form „1234;Harry Potter; This is a book“ übergeben, die erste Zahl soll die Isbn-Nummer darstellen, der Mittelteil ist der Name des Buchs und der letzte Teilstring ist ein Kommentar. Für jede Eingabe sollte die Ausgabe wie folgt aussehen (am Beispiel):

Isbn:	1234
Name:	Harry Potter
Abstrct:	This is a book

Falls ein Teilstring leer ist oder zu wenige Teilstrings vorhanden sind, soll eine Exception geworfen und gecatched werden, die dann „Invalid input“ ausgibt.

Aufgabe 8:

Man bekommt eine abstrakte Klasse Kamera gegeben, davon abgeleitet sind die Klassen Smartphone und Spiegelreflexkamera. Jede der Klassen hat eine int **bilder**, die von 1 hochzählen soll wenn ein Bild gemacht wird (dafür gibt es Methode takePicture()). Zudem hat jede neue Instanz (von Smartphone oder Spiegelreflexkamera) eine individuelle ID.

Aufgabe 9:

Man bekommt eine verkettete Liste **L**, ein Element **x** und eine int **n** gegeben und soll eine Methode schreiben, die **x** an der **n**-ten Stelle der Liste einfügt (darauf achten, dass **n** nicht größer ist als die Liste)