

Aufgabe 1: Code Verstehen

Datentypen nennen oder Fehler erkennen (Value durch den richtigen Datentypen ersetzen):

Beispiel:

int a = 20;

Double b = 3.10d

Value = a+b

Value muss vom Typ Double sein

Int a = 7;

String b = "Hello"

Value c = a+b;

Value = String

int i = "0"

Error: String kann nicht einem int zugewiesen werden

Aufgabe 2: Quellcode verstehen

a) Beschreibe eine Methode F in 2-3 Sätzen:

```
public Static int[] f(int[] a, int b){  
    if(a.length == 0) return null;  
  
    int[] c = new int[a.length];  
    for(int i = 0; i < a.length; i++){  
        c[i] = a[i] * b  
    }  
    Return c;  
}
```

b) schreibe die F Methode als Methode G um. Verwende keine For Schleifen mehr, sondern eine andere Schleifenart: (z.B. while)

Aufgabe 3: Prefix bestimmen

Prefix von 2 Strings finden

z.B

"abcd" und "abcf" haben "abc" gemeinsam als Prefix

„abcd“ und „aef“ = „a“

„abcd“ und „efgh“ = „“

```
public Static String greatestCommonPrefix(String s1, String s2)
```

Aufgabe 4: Recursion

Recursion Variante von Aufgabe 3 selbst schreiben

```
public Static String greatestCommonPrefixRecursive(String s1, String s2)
```

Aufgabe 5: Array

Gegeben sind 3 Test-Case von 2D Arrays.

Vereinfachung: Jedes Array hat 12 Zeilen (für jeden Monat) und 5 Temperaturen.

Die Aufgabe ist es eine Methode zu schreiben

getArrayMinIndex(int[][] Temperatures), dass ein Array der Größe 12 zurückgibt und darin den Index des kleinsten Wertes im Monat speichert

Aufgabe 6: String II: Funktion replace selber schreiben

String replace selber schreiben, ohne string.replace() zu nutzen

```
public String replaceAll(String Str, String seq, String newSeq)
```

Aufgabe 7: OOP

OOP Konzepte

(a)Tupel (int id, Object left, Object right) + static int Variable Counter

(b) Exceptions:

Arithmetik Exception werfen + Test Case schreiben:

Methode:

```
public double squareDiv(Double a, Double b) {
```

```
if(b != 0)

    return a/(b*b)

throw new ArithmeticException("b is not allowed to be Zero")
```

Dann eine Test Methode selbst schreiben:

Solange wie squareDiv eine Exception wirft: les 2 neue Zahlen ein, probiere erneut, und wenn keine Exception geworfen wird, gebe Ergebnis aus

Aufgabe 8: Collections

Es war etwas wie: Movies sind gegeben und mit jedem Movie Objekt, das erstellt wird, soll eine id automatisch hochgezählt werden. Zusätzlich soll eine Methode (etwas mit get all Movie Titles of one genre)

- Movie (String Title, List<String> Genres)
 - StreamingService (List<Movie>)
 - Konstruktor, Getter für beide Klassen:
 - o StreamingService hat außerdem:
 - o void addMovie(String Title, List<String> Genres)
 - o Map<String, List<Movie>> getGenreMap()
-

Aufgabe 9: Vererbung

Klasse Customer, BusinessCustomer, Interface LineParser, CustomerParse
BusinessCustomer extends Customer, CustomerParse implements LineParser

Aufgabe:

- Customer: Variablen schreiben (id, name, street, city), Getter/Setter und Konstruktoren
- BusinessCustomer(company)Getter, Konstruktoren
- LineParser (Method: parseCustomer)
- CustomerParse(Create customer)

Drauf achten, das in der Abfrage die länge der Elemente stimmen soll, Zeile einlesen, .split() und dann anzahl vergleichen und Customer/ BusinessCustomer objekte erstellen:

Input Format:

"Id;Name;Street;City"

"id;Name;Street;City;Company"