

Die Klasse Celebrity und die celebritiesList

Celebrity
- name: String - vorname: String - vermoegen: int - branche: String
+ Celebrity(pName: String, pVorname: String, pEinkommen: int, pBranche: String) + getName(): String + setName(pName: String) + getVorname(): String + setVorname(pVorname: String) + getVermoegen(): int + setVermoegen(pVermoegen: int) + getBranche(): String + setBranche(pBranche: String) + istAlphabetischNach(pCelebrity: Celebrity) + toString(): String

Chalamet, Timothee, 25
 Lawrence, Jennifer, 140
 Cruise, Tom, 891
 Swift, Taylor, 150
 Helene, Fischer, 13
 Lane, Khaby, 12
 Reif, Pamela, 9
 Rowling, JK, 920
 Vettel, Sebastian, 250
 Gauff, Coco, 27
 Wirtz, Florian, 19
 Schroeder, Dennis, 60

Aufgabe 1: Analyse

1. Unterteile die Methode in sinnvolle Abschnitte.
2. Erläutere, was die Abschnitte bewirken.
Erläutere dabei detailliert die unterstrichenen Zeilen.
3. Wen gibt die Methode zurück?

```

01 private Celebrity gibJemand(){
02     Celebrity ergebnis = null;
03     if(celebritiesList.isEmpty()){
04         return ergebnis;
05     }
06     celebritiesList.toFirst();
07     Celebrity ergebnis = celebritiesList.getContent();
08     while(celebritiesList.hasAccess()){
09         Celebrity c = celebritiesList.getContent();
10         if(c.getEinkommen() < ergebnis.getEinkommen()){
11             ergebnis = c;
12         }
13         celebritiesList.next();
14     }
15     return ergebnis;
16 }

```

Aufgabe 2: Implementierung

Implementiere die folgenden Methoden. Die `celebritiesList` soll dabei unverändert sein – es sein denn, die Methode verlangt eine Veränderung der Liste.

1. `public int gesamtVermoegen()`
2. `public void steuerAbzug(int pAbzug):`
Soll jedem vom Vermögen `pAbzug` abziehen.
3. `public void rausschmeissen(String pName)`
Hier braucht man die Methode `remove()`
4. `public void vermoegensGrenze(int pGrenze)`
Soll alle rausschmeissen, deren Einkommen unterhalb von `pGrenze` liegt.