

Greedy-Münzwechsler

- (a) Nehmen Sie an, es stehen beliebig viele 5-Cent, 2-Cent und 1-Cent-Münzen zur Verfügung. Die Aufgabe besteht darin, für einen gegebenen Cent-Betrag möglichst wenig Münzen zu verbrauchen. Entwerfen Sie eine Methode

```
1 public void wechselgeld (int n)
```

die diese Aufgabe mit einem Greedy-Algorithmus löst und für den Betrag von n Cent die Anzahl c_5 der 5-Cent-Münzen, die Anzahl c_2 der 2-Cent-Münzen und die Anzahl c_1 der 1-Cent-Münzen berechnet und diese auf der Konsole ausgibt. Sie können dabei den Operator $/$ für die ganzzahlige Division und den Operator $\%$ für den Rest bei der ganzzahligen Division verwenden.¹

```
3 public class GreedyMuenzwechsler {
4
5     /**
6      * Wechsle einen Cent-Betrag in die Münzen 5-Cent, 2-Cent and
7      * ↪ 1-Cent.
8      *
9      * @param betrag Geldbetrag in Cent.
10     */
11     public static void wechsle(int betrag) {
12         int rest;
13         int c5 = betrag / 5;
14         rest = betrag % 5;
15         int c2 = rest / 2;
16         int c1 = rest % 2;
17
18         System.out.println(String.format(
19             "Für den Betrag von %s Cent werden \n" +
20             "%s Fünf-Cent-Münzen, \n" +
21             "%s Zwei-Cent-Münzen und \n" +
22             "%s Ein-Cent-Münzen ausgegeben.",
23             betrag, c5, c2, c1));
24     }
25
26     public static void main(String[] args) {
27         wechsle(1);
28         wechsle(20);
29         wechsle(23);
30         wechsle(42);
31     }
32 }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/ab_3/GreedyMuenzwechsler.java](https://github.com/org/bschlangaul/aufgaben/aud/ab_3/GreedyMuenzwechsler.java)

- (b) Es kann gezeigt werden, dass der Greedy-Algorithmus für den obigen Fall der Münzwerte 5, 2 und 1 optimal ist, d. h. dass er immer die Gesamtzahl der Münzen minimiert. Nehmen Sie nun an, es gibt die Münzwerte 5 und 1. Ist es dann möglich, einen dritten Münzwert so zu wählen, dass der

¹Quelle möglicherweise von <https://www.yumpu.com/de/document/read/17936760/ubungen-zum-prasenzmodul-algorithmen-und-datenstrukturen>

Greedy-Algorithmus mit den drei Münzen nicht mehr optimal ist? Begründen Sie Ihre Antwort.

Falls der dritte Münzwert 4 ist, ist der Greedy-Algorithmus nicht mehr optimal. Der Greedy-Algorithmus benutzt zunächst so viele 5-Cent-Münzen wie möglich und dann so viele 4-Cent-Münzen wie möglich. Ein Betrag von 8 Cent wird also in eine 5-Cent und drei 1-Cent-Münzen aufgeteilt. Optimal ist aber die Aufteilung in zwei 4-Cent-Münzen.