

Greedy-Münzwechsler

*(Münzwechsler)***Stichwörter:** Greedy-Algorithmus

- (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

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

die diese Aufgabe mit einem Greedy-Algorithmus löst und für den Betrag von n Cent die Anzahl $c5$ der 5-Cent-Münzen, die Anzahl $c2$ der 2-Cent-Münzen und die Anzahl $c1$ 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

Lösungsvorschlag

```
public static void wechsle(int betrag) {
    int rest;
    int c5 = betrag / 5;
    rest = betrag % 5;
    int c2 = rest / 2;
    int c1 = rest % 2;

    System.out.println(String.format("Für den Betrag von %s Cent werden \n" +
→ "%s Fünf-Cent-Münzen, \n"
    + "%s Zwei-Cent-Münzen und \n" + "%s Ein-Cent-Münzen ausgegeben.", betrag,
→ c5, c2, c1));
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/greedy/Muenzwechsler.java](https://github.com/bschlangaul/aufgaben/aud/muster/greedy/Muenzwechsler.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 Greedy-Algorithmus mit den drei Münzen nicht mehr optimal ist? Begründen Sie Ihre Antwort.

Lösungsvorschlag

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.

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

Additum

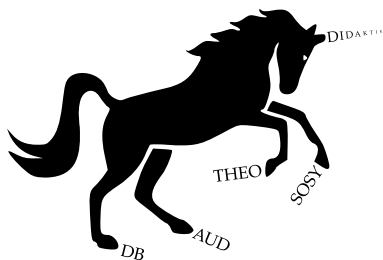
```
/**
 * Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen:
 * Aufgabenblatt 3: Algorithmenmuster.
 *
 * <a href="https://www.studon.fau.de/file2521908_download.html">Angabe: AB_3
 * Greedy_DP_Backtracking.pdf</a>
 * <a href="https://www.studon.fau.de/file2521907_download.html">Lösung: AB_3
 * Greedy_DP_Backtracking_Lsg.pdf</a>
 */
public class Muenzwechsler {

    /**
     * Wechsle einen Cent-Betrag in die Münzen 5-Cent, 2-Cent and 1-Cent.
     *
     * @param betrag Geldbetrag in Cent.
     */
    public static void wechsle(int betrag) {
        int rest;
        int c5 = betrag / 5;
        rest = betrag % 5;
        int c2 = rest / 2;
        int c1 = rest % 2;

        System.out.println(String.format("Für den Betrag von %s Cent werden \n" +
→ "%s Fünf-Cent-Münzen, \n"
→ "      + %s Zwei-Cent-Münzen und \n" + "%s Ein-Cent-Münzen ausgegeben.", betrag, c5,
→ c2, c1));
    }

    public static void main(String[] args) {
        wechsle(1);
        wechsle(20);
        wechsle(23);
        wechsle(42);
    }
}
```

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



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bsclangaul@gmx.net. Der \LaTeX -Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: https://github.com/bsclangaul-sammlung/examens-aufgaben-tex/blob/main/Module/30_AUD/60_Algorithmenmuster/20_Greedy-Algorithmen/Aufgabe_Muenzwechsler.tex