

# Wechselgeldalgorithmus

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

Als Beispiel nehmen wir die Herausgabe von Wechselgeld auf Beträge unter 1€. Verfügbar sind die Münzen mit den Werten 50ct, 10ct, 5ct, 2ct, 1ct. Unser Ziel ist, so wenig Münzen wie möglich in das Portemonnaie zu bekommen. Ein Beispiel:  $78\text{ct} = 50 + 2 \cdot 10 + 5 + 2 + 1$ . Es wird jeweils immer die größte Münze unter dem Zielwert genommen und von diesem abgezogen. Das wird so lange durchgeführt, bis der Zielwert Null ist.

## Formalisierung

Gesucht ist ein Algorithmus der folgende Eigenschaften beschreibt. Bei der *Eingabe* muss gelten:

- (a) dass die eingegebene Zahl eine natürliche Zahl ist, also  $\text{betrag} > 0$
- (b) dass eine Menge von Münzwerten zur Verfügung steht  $\text{münzen} = \{c_1, \dots, c_n\}$  z.B.  $\{1, 2, 5, 10, 20, 50\}$

Die *Ausgabe* besteht dann aus ganzen Zahlen  $\text{wechselgeld}[1], \dots, \text{wechselgeld}[n]$ . Dabei ist  $\text{wechselgeld}[i]$  die Anzahl der Münzen des Münzwertes für  $c_i$  für  $i = 1, \dots, n$  und haben die Eigenschaften:

- (a)  $\text{wechselgeld}[1] \cdot c_1 + \dots + \text{wechselgeld}[n] \cdot c_n = \text{betrag}$
- (b)  $\text{wechselgeld}[1] + \dots + \text{wechselgeld}[n]$  ist minimal unter allen Lösungen für 1.

```
/**
 * <a href=
 *
 * → "https://de.wikiversity.org/wiki/Kurs:Algorithmen_und_Datenstrukturen/Vorlesung/Greedyalgorithmen
 * und Datenstrukturen/Vorlesung/Greedyalgorithmen Wechselgeldalgorithmus</a>
 */
public class Wechselgeld {

    public static int[] berechneWechselgeld(int[] münzen, int betrag) {
        int[] wechselgeld = new int[münzen.length];
        int aktuelleMünze = münzen.length - 1;
        while (betrag > 0) {
            while (betrag < münzen[aktuelleMünze] && aktuelleMünze > 0)
                aktuelleMünze--;
            if (betrag >= münzen[aktuelleMünze] && aktuelleMünze >= 0) {
                betrag -= münzen[aktuelleMünze];
                wechselgeld[aktuelleMünze]++;
            } else
                return null;
        }
        return wechselgeld;
    }

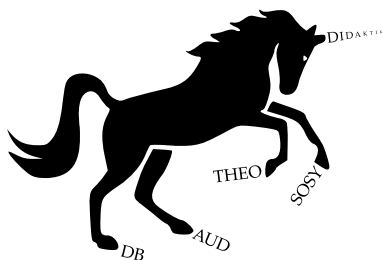
    public static void main(String[] args) {
        int[] münzen = { 1, 2, 5, 10, 20, 50 };
    }
}
```

```
int betrag = 78;
int[] wechselgeld = berechneWechselgeld(münzen, betrag);

System.out.println(String.format("Der Betrag von %s Cent wird gewechselt in:",
    ↪ betrag));

for (int i = 0; i < wechselgeld.length; i++) {
    System.out.println(String.format("%s x %s Cent", wechselgeld[i], münzen[i]));
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/muster/Wechselgeld.java](https://github.com/bschlangaul/muster/Wechselgeld.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.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: [https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30\\_AUD/60\\_Algorithmenmuster/20\\_Greedy-Algorithmen/Aufgabe\\_Wechselgeld.tex](https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30_AUD/60_Algorithmenmuster/20_Greedy-Algorithmen/Aufgabe_Wechselgeld.tex)