

Die Kursunterlagen finden Sie unter der angegebenen Website:

<https://github.com/ThorbjornSchulz/ProgrammingCourse2018>

**Aufgabe 1. ▲** Der Geldautomat ! Überlegt euch ein Programmablauf, dass das Verhalten eines **Geldautomat** simuliert. Genauer soll simuliert werden, wenn ein Kunde Geld abheben möchte. Dazu stellen wir ein Programmgrundgerüst zur Verfügung, damit der Einstieg leichter fällt. Macht euch mit der algorithmischen Denkweise vertraut und überlegt, was alles geprüft werden muss, um Geld abzuheben.

Der Einfachheit halber wird das Prüfen der Geldkarte 'simuliert'. Dazu ist es lediglich nötig, eine ID zu vergleichen.

Es soll möglich sein, dass man mit dem Geldautomat:

- 10 Euro
- 25 Euro
- 50 Euro
- 100 Euro
- 200 Euro
- 500 Euro

abgeben kann. Bitte bezieht dieses Kriterium mit in eure Logik ein.

Hinweis: Die Aufgabe ist etwas praxisnaher als die abstrakten 'Mathe'-Aufgaben. Daher ist das Ziel der Aufgabe, sich mit den Gegebenheiten der Programmiersprache auseinander zu setzen. Also denkt bitte nicht zu kompliziert und geht an die Aufgabe mit gesundem Menschenverstand heran.

Die Vorlage der Aufgabe ist 'Geldautomat.c'

**Aufgabe 2. ▲** Berechne die Summe der ersten  $n$  ungeraden Zahlen mit einer `while`-Schleife. Wie ist die Ausgabe für  $n = 1, \dots, 15$ . Was fällt dir auf? Könnte man diese Aufgabe nun also effizienter programmieren?

**Aufgabe 3. ▲**

- a) Implementiere den Primzahltest (Algorithmus 1) von gestern.
- b) Schreibe ein Programm, dass jeweils die nächste Primzahl nach 20000, 30000 und 40000 findet.

**Aufgabe 4. ▲ ▲** Implementiere den Algorithmus 2 von gestern, welcher den größten gemeinsamen Teiler zweier Zahlen berechnet.

**Aufgabe 5. ▲ ▲** Schreibe ein Programm, um den Wert der Reihe

$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

zu berechnen. Er sollte  $\frac{\pi^2}{6}$  sein. Wichtig ist, sich ein geeignetes Abbruchkriterium zu überlegen.