

## Übungsaufgabe 1 (Überblickswissen):

- Wann ist eine `for`-Schleife einer `while`-Schleife gegenüber vorzuziehen? Wann sollte man eher eine `while`-Schleife nutzen?
- Was sind die Vor- und Nachteile der `switch`-Anweisung gegenüber einer `if ... else ...`-Anweisung?

## Übungsaufgabe 2 (Casting):

Bestimmen Sie den Typ und das Ergebnis der folgenden Java-Ausdrücke und begründen Sie Ihre Antwort. Sollte der Ausdruck nicht typkorrekt sein, begründen Sie, worin der Fehler besteht.

Dabei seien die Variablen `x`, `y` und `z` wie folgt deklariert: `int x = 1; int y = 2; int z = 3;`

- `10 / 3`
- `10 / 3.`
- `x == y ? x > y : y < z`
- `(byte) (127 + 1)`
- `'x' + y + z`
- `1 || 0`

## Übungsaufgabe 4 (Programmierung):

In dieser Aufgabe geht es um die Ein- und Ausgabe in Java. Dafür soll die von Java bereitgestellte Klasse `IO`<sup>1</sup> genutzt werden. Um einen String `str` auszugeben, nutzen Sie `IO.println(str)`. Benutzen Sie `IO.readLine("Bitte geben Sie einen String ein: ")`, um einen Wert vom Typ `String` einzulesen. Sie können die von Java vordefinierte Methode `Integer.parseInt(str)` benutzen, um den String `str` in eine Zahl umzuwandeln. Mit `Integer.parseInt(IO.readLine("Bitte geben Sie eine Zahl ein: "))` können Sie also einen Wert vom Typ `int` einlesen.

Schreiben Sie ein Java-Programm, welches einen einfachen Taschenrechner darstellt. Der Taschenrechner liest initial eine Zahl ein. Auf diese initiale Zahl (und eine später eingelesene weitere Zahl) kann nun eine der zwei folgenden Rechenoperationen angewandt werden:

- **ADD:** Addiere die aktuelle Zahl und eine weitere eingelesene Zahl.
- **SUB:** Subtrahiere eine weitere eingelesene Zahl von der aktuellen Zahl.

Nachdem eine dieser zwei Operationen ebenfalls eingelesen wurde, wird eine zweite Zahl eingelesen und das Ergebnis entsprechend der Operation berechnet. Danach wird das eben berechnete Ergebnis zwischengespeichert, eine weitere Operation und eine weitere Zahl werden eingelesen. Das ursprüngliche Zwischenergebnis wird nun benutzt, um ein neues Zwischenergebnis zu berechnen. Dieser Prozess wird solange wiederholt, bis die Eingabe **STOP** getätigt wurde. Danach wird das Endergebnis ausgegeben und das Programm wird beendet. Bei fehlerhaften Eingaben kann sich Ihr Programm beliebig verhalten.

Ein Ablauf des Programms könnte z.B. so aussehen:

<sup>1</sup>Siehe hierzu auch <https://docs.oracle.com/en/java/javase/25/docs/api/java.base/java/lang/IO.html>.

Bitte geben Sie eine Zahl ein: 12  
Bitte geben Sie eine Rechenoperation (ADD oder SUB) oder STOP ein: ADD  
Bitte geben Sie eine Zahl ein: 10  
Aktuelles Ergebnis: 22  
Bitte geben Sie eine Rechenoperation (ADD oder SUB) oder STOP ein: SUB  
Bitte geben Sie eine Zahl ein: 5  
Aktuelles Ergebnis: 17  
Bitte geben Sie eine Rechenoperation (ADD oder SUB) oder STOP ein: STOP  
Endergebnis: 17

#### Hinweise:

- Im Lernraum befindet sich ein Java-Template für Ihre Lösung.
- Verwenden Sie `str1.equals(str2)` um zu überprüfen, ob zwei Strings `str1` und `str2` die gleiche Sequenz von Zeichen beinhalten.
- Sie können `System.exit(1)` verwenden, um den `main`-Methodenaufruf im Falle von fehlerhaften Eingaben zu beenden.