

# PRAKTIKUM EINFÜHRUNG IN DIE PROGRAMMIERUNG

## BLATT3

Wintersemester 2025/2026

**Abgabe des Blattes bis zum Donnerstag 13.11 - 0:00 Uhr**

### AUFGABE 1 - UMGANG MIT ZAHLEN UND STRINGS

Im Paket *aufgabe1* finden sie zwei Klassen. Implementieren sie die vorgegebenen Methoden wie folgt:

- Die Klasse *NumberSort* hat drei private Klassenvariablen *small*, *medium* und *big*. Diese können sie in der Methode *setValues* verwenden, genauso wie sie lokale Variablen verwenden können. Entwickeln sie die Methode *setValues*, die drei Parameter *a*, *b* und *c* nimmt. Weisen sie die Werte *a*, *b*, und *c* jeweils den Klassenvariablen *small*, *medium* und *big* der Klasse so zu, dass gilt *small*  $\leq$  *medium*  $\leq$  *big*.

Testen sie die korrekte Funktionalität ihrer Methode, in dem sie eine Main-Methode hinzufügen, dort *setValues* mit verschiedenen Beispielen aufrufen und anschließend mit Hilfe der Methoden *getSmall()*, *getMedium()* und *getBig()* die Werte der drei Klassenvariablen ausgeben.

- Die Klasse *StringCompare* besitzt die Methode *compareString*. Diese nimmt einen String entgegen, und soll prüfen ob dieser String dem String "Programmieren" entspricht.

Führen Sie den Test auf Gleichheit der Strings mit zwei verschiedenen Ausdrücken durch:

- `word == "Programmieren"`
- `word.equals("Programmieren")`

Versuchen sie in ihrer Implementierung herauszufinden, welcher der beiden Tests das richtige Ergebnis liefert. Lesen sie dazu einen String von der Konsole mit Hilfe der *progTools* ein (Achtung: Wenn sie den String "Programmieren" jeweils zwei mal hart in den Quellcode schreiben, dann sehen sie den Effekt nicht). Binden sie die *progTools* wie folgt ein:

```
import de.tha.prog.tools.Input.*;
```

Recherchieren sie den Grund für das Verhalten ihres Codes.

### AUFGABE 2 - SCHALTJAHRE UND WOCHENTAGE

- Legen sie im Paket *aufgabe2* eine Klasse mit einer Methode mit dem Namen "DateHandler" und einer Klassenmethode mit dem Namen "isLeapYear" an. Diese Methode soll ein Jahr als *int* als Parameter nehmen und mittels einer *boolean* zurück geben, ob das Jahr ein Schaltjahr ist oder nicht.

- Schaltjahre sind Jahre, die durch vier teilbar sind (aber nicht alle).
- Ist ein Jahr durch vier und einhundert teilbar ist, dann ist es in der Regel kein Schaltjahr.
- Ein Jahr das durch vier und einhundert und vierhundert teilbar ist, ist aber dennoch ein Schaltjahr.

Testen Sie ihr Programm mit verschiedenen Jahren (z.B. 2020 (Schaltjahr), 1900 (kein Schaltjahr), 2000 (Schaltjahr), 2022 (kein Schaltjahr)) in einer Main-Methode.

- b. Christian Zeller hat im Jahr 1882 folgende Formel zur Berechnung des Wochentages eines Datums definiert<sup>1</sup>:

$$w = (d + \lfloor 2.6 * m - 0.2 \rfloor + y + \lfloor \frac{y}{4} \rfloor + \lfloor \frac{c}{4} \rfloor - 2c) \bmod 7$$

Die eckigen Klammern  $\lfloor x \rfloor$  stellen die Gaußklammern dar, die den Wert innerhalb der Klammern abrunden. Für diese Berechnung brauchen Sie keine spezielle Methode, sondern können dies mit den aus der Vorlesung und den vorherigen Praktika bekannten Eigenschaften der Berechnungen einfacher Datentypen erreichen.

Die einzelnen Variablen der Formel sind wie folgt festgelegt:

- **d**: Tagesdatum (1 bis 31)
- **m**: Monat gemäß der unten stehenden Tabelle. Demzufolge währt ein Jahr von März bis Februar. Da für Januar und Februar  $m = 11$  bzw.  $m = 12$  gilt, muss für diese beiden Monate die Jahreszahl bei der Berechnung um 1 vermindert werden.

März	April	Mai	Juni	Juli	August	September	Oktober	November	Dezember	Januar	Februar
1	2	3	4	5	6	7	8	9	10	11	12

- **y**: Die beiden letzten Stellen der Jahreszahl, bei den Monaten Januar und Februar die letzten Stellen des Vorjahrs (für Dez. 1907 also 7, für Jan 1907 6, für Jan 1900 99)
- **c**: Die beiden ersten Stellen der Jahreszahl, bei den Monaten Januar und Februar die ersten Stellen des Vorjahrs (für Dez. 1907 also 19, für Jan 1907 19, für Jan 1900 18)
- **Ergebnis: w**: Wochentag (dabei gilt: Sonntag ist die 0, Montag die 1, Dienstag die 2, ...)

Fügen sie in die Klasse *DateHandler* eine neue Klassenmethode mit dem Namen "calculateWeekDay". Diese nimmt drei Parameter entgegen: den Tag des Monats als *byte*, den Namen des Monats als *String* und das Jahr als *short*. Die Methode soll dann den Wochentag zu diesem Datum berechnen und zurückgeben. Die Wochentage und Monate sollen dabei jeweils in deutscher Sprache übergeben werden.

**Hinweis:** Wenn ihr Code mit einigen der Daten in der Test-Datei zuverlässig funktioniert, mit anderen aber ein illegaler Wert für den Wochentag herauskommt, dann werfen sie einen Blick in die Vorlesung zu Ausdrücken auf die Operatoren (insbesondere der Modulo-Operator)

### AUFGABE 3 - GLEICHUNGSSYSTEME

In dieser Aufgabe werden wir ein Programm schreiben, dass die folgende Gleichung löst:

$$a * x^2 + b * x + c = 0$$

Nutzen Sie dazu die Mitternachtsformel:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

<sup>1</sup><https://de.wikipedia.org/wiki/Wochentagsberechnung>

- a. Legen Sie im Paket *aufgabe3* eine Klasse mit dem Namen "MidnightFormula" an. Diese soll eine Klassenmethode mit dem Namen "printFormula" haben, die drei Parameter vom Typ *int* entgegen nimmt und einen String liefert. Bei den Parametern handelt es sich um die Parameter *a*, *b*, *c* der Mitternachtsformel.

Die Methode soll Gleichung als String zurückgeben. Achten Sie darauf, dass sich die Vorzeichen der drei Variablen in der Gleichung widerspiegeln. Beispiel:

- $a = 2, b = 2, c = 3 \rightarrow 2 * x^2 + 2 * x + 3 = 0$
- $a = -2, b = -2, c = -3 \rightarrow -2 * x^2 - 2 * x - 3 = 0$

Geben sie den entstandenen String dann als Rückgabewert der Methode zurück.

- b. Fügen sie eine zweite Klassenmethode mit dem Namen "calculateFormula" in die Klasse ein. Diese soll ebenfalls einen String als Ergebnis liefern.

Berechnen sie in der Methode das Ergebnis der Gleichung. Machen sie sich dazu Gedanken wie viele Lösungen die Gleichung bei verschiedenen Parametern haben kann.

Bedenken Sie insbesondere, dass der Nutzer für den Parameter *a* auch den Wert 0 eingeben kann. Geben Sie die Anzahl der Lösungen und die berechneten Lösungen auf zwei Kommastellen gerundet aus. Eine Beispielerückabe für folgende Formel:

$$2.0*x^2 - 5.0*x + 1.0 = 0$$

kann wie folgt aussehen:

Die Gleichung hat zwei reelle Lösungen:

$x_1 = 2.28$   
 $x_2 = 0.22$

In *blatt3-test* werden auch die anderen Fälle für *a*, *b*, *c* getestet. Orientieren sie sich bei der Formatierung ihrer Ergebnisse an der Ausgabe der Testfälle. Beachten sie: der zurückgegebene String soll am Ende des Strings keine Newline haben. Die Zeilenumbrüche innerhalb des Strings sollen sie mit '\n' umsetzen.

Hinweise: Mit Hilfe der Methode `Math.sqrt(x)` können Sie die Quadratwurzel von *x* berechnen. Mit der Methode `Math.abs(x)` bekommen Sie den absoluten Wert von *x*.

#### AUFGABE 4 - GELDAUTOMAT

Ein Geldautomat soll einen gewünschten Betrag in vorhandenen Scheinen ausgeben. Folgende Scheinarten befinden sich im Geldautomaten: 500 Euro, 200 Euro, 100 Euro, 50 Euro, 20 Euro, 10 Euro und 5 Euro.

Schreiben sie im Paket *aufgabe4* die Klasse *CashMachine*. Diese soll eine main-Methode haben, die mit Hilfe der Methoden aus *progTools* einen Geldwert von der Konsole einliest.

Das Programm soll danach den gewünschten Betrag mit möglichst wenig Scheinen ausgeben. Die Ausgabe soll wie folgt formatiert sein:

```
Bitte gewünschten Ausgabebetrag an: 948
500-Euro Scheine: 1
200-Euro Scheine: 2
100-Euro Scheine: 0
50-Euro Scheine: 0
20-Euro Scheine: 2
```

---

10-Euro Scheine: 0

5-Euro Scheine: 1

Rest: 3

#### AUFGABE 5 - PARKKOSTEN

Sie parken Ihr Auto in einem Parkhaus. Das Parkhaus berechnet 1,75 € pro angefangene Stunde. Schreiben sie ein Programm, dass einem Nutzer die kumulierten Kosten pro Stunde für eine gewünschte Parkdauer anzeigt.

Erstellen sie dazu im Paket *aufgabe5* eine neue Klasse *ParkingGarage*. Diese soll mit Hilfe der Methoden aus den *progTools* den Nutzer anfragen, wie viele Stunden geparkt werden soll. Die kumulierten Kosten pro Stunde sollen im Anschluss ausgegeben werden. Eine Ausgabe soll wie folgt aussehen:

Wie viele Stunden möchten Sie parken? 10

Berechnung:

Nach 1 Stunde: 1.75 Euro

Nach 2 Stunden: 3.5 Euro

Nach 3 Stunden: 5.25 Euro

Nach 4 Stunden: 7.0 Euro

Nach 5 Stunden: 8.75 Euro

Nach 6 Stunden: 10.5 Euro

Nach 7 Stunden: 12.25 Euro

Nach 8 Stunden: 14.0 Euro

Nach 9 Stunden: 15.75 Euro

Nach 10 Stunden: 17.5 Euro

Gesamtkosten: 17.5