TECHNISCHE HOCHSCHULE AUGSBURG – FAKULTÄT FÜR INFORMATIK

PRAKTIKUM EINFÜHRUNG IN DIE PROGRAMMIERUNG

BLATT1

Wintersemester 2025/2026

Bonuspunkte für die Praktikumsklausur:

- Anwesenheit in jedem Praktikumstermin gibt jeweils einen Bonuspunkt
- Vorstellung einer Aufgabe in Praktikum gibt einen Bonuspunkt
- Alle Tests auf dem Server für ein Blatt lauffähig gibt einen Bonuspunkt pro Blatt

Deadline zur Abgabe von Blatt 1: 23.10.2025 (nach dem Praktikum in der nächsten Woche)

AUFGABE 1 - RAHMENPROGRAMM

Laden Sie sich das neue Rahmenprojekt *blatt1* vom Git-Server, indem sie das folgende Repository clonen:

`ssh://git@prog.informatik.tha.de:3333/prog1/<username>/blatt1`

Wie im letzten Blatt müssen sie <username> durch ihren Usernamen ersetzen.

Das Projekt enthält eine JAR-Datei mit dem Namen ”progTools.jar” mit einer Bibliothek zum einfachen einlesen von Daten von der Kommandozeile. Um die Methoden zur Eingabe von der Konsole benutzen zu können, müssen wir diese erst in wie folgt in jede Klasse importieren:

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

Im Anschluss können die Methoden verwendet werden. Ein Beispiel für die Verwendung finden Sie in der Datei *InputDemo.java* im Paket *demo*. Führen sie diese Klasse *InputDemo* aus und machen Sie sich mit den Methoden vertraut. Probieren Sie auch aus was passiert, wenn Sie fehlerhafte Daten eingeben.

Hinweis: Diese Teilaufgabe dient der Vorbereitung der anderen Aufgaben. Sie können die Methoden der Datei nutzen, um in einfacher Weise Daten von der Kommandozeile einzulesen. Um die Methoden in ihren Klassen der nachfolgenden Aufgaben zu nutzen, müssen sie die Import-Anweisung von oben in ihre Klasse einfügen.

AUFGABE 2 - ERSTE SCHRITTE MIT METHODEN UND BEDINGTEN ANWEISUNGEN

Erstellen sie ihre Lösung im Paket *Aufgabe2*.

a. Legen sie in eine Klasse mit dem Namen *Decisions* an. Diese soll eine Main-Methode haben. In der Main-Methode sollen sie die korrekte Funktionsweise der Methoden der folgenden Teilaufgaben durch Beispielaufrufe demonstrieren.

b. Legen sie in der Klasse *Decisions* eine Klassenmethode mit dem Namen *substractNumbers* an, die jeweils zwei *int*-Parameter akzeptiert. Die Methode zieht den zweiten Wert vom ersten Wert ab und gibt das Ergebnis zurück.

   Demonstrieren sie durch verschiedene Aufrufe der Methode in ihrer Main-Methode, dass sie die Methode richtig implementiert haben.

c. Legen sie eine zweite Klassenmethode mit dem Namen *isPositive* in *Decisions* an. Die Methode nimmt einen Parameter vom Typ *float* entgegen. Sie gibt zurück ob die übergebene Zahl positiv ist.

   Demonstrieren sie durch verschiedene Aufrufe der Methode in ihrer Main-Methode, dass sie die Methode richtig implementiert haben.

d. Legen sie in der Klasse *Decisions* eine Klassenmethode mit dem Namen *isProductEven* an. Die Methode nimmt zwei Parameter vom Typ *int*. Sie gibt zurück, ob das Produkt der beiden Parameter gerade ist.

   Demonstrieren sie durch verschiedene Aufrufe der Methode in ihrer Main-Methode, dass sie die Methode richtig implementiert haben.

e. Legen sie in der Klasse *Decisions* eine Klassenmethode mit dem Namen *calcCircleArea*, die einen Radius als *double* entgegen nimmt und die Fläche berechnet (Formel: $A = r^2 * \pi$). Das Ergebnis wird zurückgegeben. Sie können die Variable *Math.PI* für PI verwenden.

   emonstrieren sie durch verschiedene Aufrufe der Methode in ihrer Main-Methode, dass sie die Methode richtig implementiert haben.

AUFGABE 3 - BERECHNUNG DER MEHRWERTSTEUER

Im Paket *Aufgabe 3* finden sie eine Klasse mit dem Namen *VATCalculator*. Diese Klasse soll am Ende der Aufgabe zu einem beliebigen Netto-Preis den Preis inklusive Mehrwertsteuer berechnen.

a. Die Main-Methode der Klasse soll einen float-Wert von der Kommandozeile einlesen. Erweitern Sie das Programm so, dass der Wert von der Konsole eingelesen und wieder auf der Konsole ausgegeben wird.

   Experimentieren Sie mit diesem Programm:

   - Geben Sie eine Kommazahl ein und betrachten Sie die Ausgabe. Sind diese gleich?
   - Müssen Sie die Kommazahl mit einem Komma oder mit einem Punkt als Trennzeichen eingeben?
   - Welche Zahl gibt der Code zurück, wenn Sie keine Zahl eingeben sondern irgendeinen Text?

   HINWEIS: Die Eingabe von Kommazahlen ist abhängig von der Spracheinstellung des Systems. Diese entscheidet ob ein Punkt oder ein Komma das Dezimaltrennzeichen ist.

b. Schreiben sie eine Klassenmethode mit dem Namen *calculateVATPrice*, die zu einem gegebenen Netto-Preis und einem gegebenen Steuersatz den Brutto-Preis berechnet.

   Ein Aufruf der Methode kann wie folgt aussehen:

   ```
   calculateVATPrice(100.0f, 19.0f);
   ```

   Die Methode soll die Parameter als *floats* entgegen nehmen und einen *float* zurückgeben (Hinweis: Achten sie auf die Datentypen, da sie sonst andere Rundungsfehler bekommen, als die Tests auf dem Server).

c. Erweitern sie den Code in der **Main-Methode so**, dass er einen Netto-Preis auf der Kommandozeile einliest und einen Brutto-Preis mit regulärem Mehrwertsteuersatz von 19% und 7% berechnet. Geben sie die berechneten Werte aus. Eine mögliche Ausgabe kann wie folgt aussehen:

```
Netto-Preis: 12.0
Preis mit 19% Mehrwertsteuer: 14.28
Preis mit  7% Mehrwertsteuer: 12.84
```

Probieren Sie ihren Code mit unterschiedlichen Werten aus. Finden Sie Werte, bei denen der Brutto-Betrag nicht auf ganze Cents berechnet wird? Finden Sie zusätzlich auch Rundungsfehler wie wir sie in der Vorlesung kennen gelernt haben?

d. Erstellen sie die Methode *calculateVATPriceRounded* mit den gleichen Parametern und dem gleichen Rückgabetyp wie *calculateVATPrice* so, dass in der Berechnung mit Hilfe der Funktion `Math.round()`[1] alle Beträge auf ganze Cent gerundet werden.

Hinweis: Die Methode kann nur auf die nächste ganze Zahl runden. Überlegen sie sich geeignete (sehr einfache) mathematische Berechnung, die das Komma in der Zahl geschickt vor und nach dem Runden verschieben können.

AUFGABE 4

a. Mit der Methode `System.out.println()` haben wir Ausgaben auf der Konsole erzeugt. Die Methode `String.format()` erlaubt die Erzeugung von formatierte Zeichenketten. Im Paket *Aufgabe4* finden sie die Klasse *NumberFormat*, die die Funktionalität der Methode demonstriert. Versuchen Sie sich anhand des Beispiels zu erschließen, wie die Formatierung funktioniert. Sie können natürlich auch die Dokumentation unter `https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Formatter.html#syntax` lesen (Achtung: Die Format-Optionen sind sehr umfangreich)

Erstellen Sie eine Klasse *TimeExamples* im Paket *Aufgabe4*. Fügen sie eine Klassenmethode mit dem Namen *formatTime* ein, die drei Parameter *hours*, *minutes* und *seconds* akzeptiert.

Die Methode soll aus den drei Parametern:

```
hours=1, minutes=2, seconds=3;
```

folgenden String erzeugen und zurückgeben:

```
Aktuelle Uhrzeit: 01:02:03 Uhr
```

Schreiben sie eine main-Methode, die demonstriert, dass *formatTime* korrekt funktioniert.

b. Fügen sie in die Klasse *TimeExamples* eine neue Klassenmethode mit dem Namen *formatTimeFromSeconds* ein. Diese nimmt einen Parameter vom Typ *int*. Dieser übergibt die seit 0:00 Uhr eines Tages vergangenen Sekunde. Die Methode gibt einen String mit der Uhrzeit in Stunden, Minuten und Sekunden zurück. Aus dem Wert:

```
int x = 18101; // Entspricht den seit 0:00Uhr vergangenen Sekunden
```

wird der String:

```
Aktuelle Uhrzeit: 05:01:41 Uhr
```

Testen sie *formatTimeFromSeconds* in der Main-Methode.

[1] `https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/Math.html#round(float)`

c. Im folgenden wollen wir die Differenz in Sekunden zwischen zwei Zeitpunkten A und B berechnen. Gegeben sei z.B. folgende Definition:

```
int hoursA = 14, minutesA = 20, secondsA = 33;
int hoursB = 18, minutesB = 31, secondsB = 43;
```

Schreiben Sie in der Klasse *TimeExamples* eine Klassenmethode mit dem Namen *timeDifference* das die Differenz in Sekunden berechnet und wie folgt als String zurück gibt:

```
Zwischen 14:20:33 Uhr und 18:31:43 Uhr sind  15070 Sekunden vergangen
```

Die Methode soll die Zeitpunkte als sechs Parameter übergeben bekommen.

Funktioniert ihr Programm auch wenn eine Stelle (z.B. die Sekunden) zum Zeitpunkt B kleiner ist als zum Zeitpunkt A?

Beispiel (Sekunden in B sind kleiner als Wert für Sekunden in A):

```
int hoursA = 14, minutesA = 20, secondsA = 33;
int hoursB = 14, minutesB = 21, secondsB = 21;
```

Schreiben sie einen Test für ihre *timeDifference* in der Main-Methode. Erklären Sie warum ihr Programm die richtige Differenz berechnet.