

252-0027-00: Einführung in die Programmierung

Übungsblatt 2

Abgabe: 10. Oktober 2023, 23:59

In dieser Übung werden Sie endlich selber Java-Programme schreiben! Checken Sie mit Eclipse die neue Vorlage aus, indem Sie wie in Übung 1 im Menü zuerst auf ihr Repository in der Repository Ansicht rechtsklicken und *Pull* wählen. Danach Importieren Sie das neue Projekt durch Rechtsklick auf *u02* und Auswahl von *Import Projects...*

Beachten Sie, dass Sie in dieser Übung mehrere unabhängige Programme im selben Eclipse-Projekt haben werden. Bevor Sie ein Programm starten, achten Sie deshalb darauf, dass Sie die richtige Datei im Package Explorer ausgewählt oder im Editor geöffnet haben.

Aufgabe 1: Fehlerbehebung


Finden und beheben Sie alle Fehler im Programm "FollerVehler.java". Eclipse hilft Ihnen dabei, indem es anzeigt, wo die Fehler sind (und eine mehr oder weniger hilfreiche Fehlermeldung dazu ausgibt), aber Sie müssen selber herausfinden, was das Problem ist und wie Sie es beheben können. Wenn Sie alle Fehler behoben haben, sollte das Programm folgendes ausgeben:

```
Hello world
Gefällt Ihnen dieses Programm?

Ich habe es selbst geschrieben.
```

Aufgabe 2: Schweizerfahne (Konsole)

Statt eine Vorlage zu benutzen, schreiben Sie in dieser Aufgabe ein Programm von Grund auf selbst. Eclipse wird Ihnen allerdings etwas Schreibarbeit abnehmen.

1. Erstellen Sie eine neue Java-Datei "SwissFlag.java". Wählen Sie dafür im Menü *File* → *New* → *Class* oder klicken Sie auf das  Symbol in der Symbolleiste. Geben Sie im Dialog bei *Name* "SwissFlag" ein und drücken Sie "Finish".
2. Erstellen Sie zuerst eine leere *main*-Methode, so wie Sie es bei anderen Programmen gesehen haben. (Tipp: Sie können in Zukunft auch die Option *public static void main(String[] args)* im Dialog für neue Java-Klassen auswählen, um sich ein wenig Arbeit zu sparen.)

3. Erweitern Sie das Programm so, dass es die Schweizerfahne in der Konsole ausgibt. Die Fahne könnte ungefähr wie folgt aussehen, Sie dürfen aber auch eine grössere oder schönere Version entwerfen:

```
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
```

```
      +++++++
      +++++++
      +++++++
+++++
+++++
+++++
```

```
      +++++++
      +++++++
      +++++++
```

Teilen Sie das Programm in mehrere Methoden auf, welche von der `main`-Methode aufgerufen werden. Damit sorgen Sie dafür, dass weniger Wiederholungen von Code-Stücken vorkommen, was das Ändern des Programms deutlich einfacher macht.

Aufgabe 3: Berechnungen

In dieser Aufgabe implementieren Sie verschiedene Berechnungen.

1. Vervollständigen Sie “SharedDigit.java”. In der Main-Methode sind zwei int Variablen a und b deklariert und mit einem Wert zwischen 10 und 99 (einschliesslich) initialisiert. Das Programm soll einer int Variablen r einen bestimmten Wert zuweisen. Wenn a und b eine Ziffer gemeinsam haben, dann wird r die gemeinsame Ziffer zugewiesen (wenn a und b beide Ziffern gemeinsam haben, dann kann eine beliebige Ziffer zugewiesen werden). Wenn es keine gemeinsame Ziffer gibt, dann soll -1 zugewiesen. Sie brauchen für dieses Programm keine Schleife.

Beispiele:

- Wenn $a: 34$ und $b: 53$, dann ist $r: 3$
- Wenn $a: 10$ und $b: 22$, dann ist $r: -1$
- Wenn $a: 66$ und $b: 66$, dann ist $r: 6$
- Wenn $a: 34$ und $b: 34$, dann ist $r: 3$ oder 4

Testen Sie Ihre Loesung mit a gleich 34 und b gleich 43. Was liefert Ihr Programm?

2. Vervollständigen Sie "SumPattern.java". In der Main-Methode sind drei int Variablen a, b, und c deklariert und mit irgendwelchen Werten initialisiert. Wenn die Summe von zwei der

Variablen die dritte ergibt, nehmen wir an dass $a + c == b$, so soll die Methode "Moeglich. $a + c == b$ " ausgeben (wobei die Werte für a, b, und c einzusetzen sind). Wenn das nicht der Fall ist, dann soll die Methode "Unmoeglich." ausgeben.

Beispiele:

- Wenn a: 4, b: 10, c: 6, dann wird "Moeglich. $4 + 6 == 10$ " oder "Moeglich. $6 + 4 == 10$ " ausgegeben.
 - Wenn a: 2, b: 12, c: 0, dann wird "Unmoeglich." ausgegeben.
3. Vervollständigen Sie "AbsoluteMax.java". In der Main-Methode sind drei int Variablen a, b, und c deklariert und mit irgendwelchen Werten initialisiert. Das Programm soll einer int Variable r den grössten absoluten Wert von a, b, und c zuweisen.

Aufgabe 4: Postconditions

Geben Sie für die folgenden Programmsegmente die Postcondition an, welche nach der Ausführung gilt, wenn zuvor die angegebene Precondition gilt. Verwenden Sie Java Syntax. Alle Anweisungen sind Teil einer Java Methode. Alle Variablen sind vom Typ int und es gibt keinen Overflow.

1.

P: { $x > 10$ }
S: $y = x + 5;$
Q: { ?? }

2.

P: { $x == 2$ }
S: $y = x + 5;$
Q: { ?? }

3.

P: { $x > 0 \ \&\& \ z > 0$ }
S: $y = z * x$
Q: { ?? }