

Übungen 10.11.-14.11.2025
Blatt 4

Abgabe der Hausaufgaben bis spätestens **Freitag, 14.11.2025 um 23:59 Uhr** via git.
Besprechung der Hausaufgaben in der nächsten Woche.

Hinweise:

- Zur Erinnerung: Erstellen Sie ein neues Repository für jedes Übungsblatt. Folgen Sie dabei den Anweisungen aus der git-Starthilfe.
- Die Bewertung jeder Aufgabe erfolgt unabhängig; ein Fehler in Hausaufgabe 3 beeinflusst beispielsweise nicht die Bewertung von Hausaufgabe 1.
- Um die Hausaufgaben zu testen, weisen wir den Variablen diverse Werte zu. Stellen Sie also sicher, dass Ihre Lösung auch unabhängig von den Werten funktioniert, mit denen sie Sie initialisieren.
- Denken Sie daran den SSH und nicht den https Link zu Ihrem Repository in Ilias hochzuladen:
`git@github.com:<USERNAME>/<REPOSITORY_NAME>.git`.



JAKE-CLARK.TUMBLR

Präsenzaufgabe 1 [Fahrstuhlsimulation]

Erstellen Sie im Repository ein Package **p1** und fügen Sie diesem Package die Klasse **P1_main** hinzu, welche die **main**-Methode enthält.

Schreiben Sie ein Programm zur Fahrstuhl-Display-Simulation. Lösen Sie folgende Aufgaben, wobei Sie ausschließlich die Kontrollstruktur **switch** benutzen dürfen:

1. Wenn der Wert einer int-Variable **stockwerk** 0 ist, soll nur das Wort „Angebote“ auf die Konsole ausgegeben werden. Bei Wert 1 soll nur das Wort „Damenbekleidung“ ausgegeben werden. bei Wert 2 „Herrenbekleidung“, bei Wert 3 „Kinderbekleidung“, bei Wert -1 „Parkhaus“. Hat **stockwerk** keinen Wert aus $\{-1, 0, 1, 2, 3\}$, dann soll nur einmal das Wort „unbekannt“ ausgegeben werden.
2. Wenn der Wert von **stockwerk** 0 ist, soll nur das Wort „Angebote“ auf die Konsole ausgegeben werden. Bei Wert 1 soll erst das Wort „Damenbekleidung“, dann, eine Zeile darunter, das Wort „Angebote“ ausgegeben werden. Bei Wert 2 erst das Wort „Herrenbekleidung“, eine Zeile darunter „Damenbekleidung“, eine Zeile darunter „Angebote“ ausgegeben werden. Analoges bei Wert 3 mit „Kinderbekleidung“. In der Ausgabe steht also ganz oben in der Konsole die höchste Etage.
Bei Wert -1 soll nur das Wort „Parkhaus“ ausgegeben werden. Hat **stockwerk** keinen Wert aus $\{-1, 0, 1, 2, 3\}$, dann soll nur einmal das Wort „unbekannt“ ausgegeben werden.

Präsenzaufgabe 2 [Typumwandlungen und Ausdrücke]

Erstellen Sie im Repository ein Package **p2** und fügen Sie diesem Package die Klasse **P2_main** hinzu, welche die **main**-Methode enthält.

Gegeben seien die folgenden Deklarationen im Quellcode.

- **int a = 3, b = 5, c = 6;**
- **double x = 1.5, y = 2.3;**
- **int intResult;**
- **double doubleResult;**

Unter den Deklarationen findet sich jeweils eine der folgenden Anweisungen:

- | | |
|-----------------------------------|--------------------------------------|
| (a) intResult = c - a * 3; | (d) doubleResult = c / b; |
| (b) intResult = c / b; | (e) doubleResult = c + a / b; |
| (c) intResult = a + b / c; | (f) doubleResult = x + y * b; |

Beantworten Sie die folgenden Fragen für jede der Teilaufgaben:

- Frage 1: Welchen Wert und welchen Typ hat der Ausdruck auf der rechten Seite des Zuweisungsoperators?
- Frage 2: Ist die Zuweisung fehlerfrei (also ohne Kompilierfehler) möglich?

Sie müssen in der Lage sein, Ihre Antwort zu erklären.

Schreiben Sie weiterhin ein Programm, um Ihre Aussagen bezüglich des Werts der Ausdrücke zu überprüfen. Insbesondere sollten die Zeilen (d), (e) und (f) in Ihrem Code vorkommen.

Ändern Sie nun eben diese oben genannten Zeilen (falls nötig) ab, so dass in **doubleResult** der Wert gespeichert wird, der auch in der Mathematik angenommen werden würde. Ändern Sie hierbei möglichst sparsam. In anderen Worten: Gelingt es Ihnen, die Änderung durch Hinzufügen von höchstens einem Operator pro Anweisung durchzuführen?

Präsenzaufgabe 3 [Wahrheitstabelle]

Erstellen Sie im Repository ein Package **p3** und fügen Sie diesem Package die Klasse **P3_main** hinzu, welche die **main**-Methode enthält.

Notieren Sie sich eine Tabelle, z.B. auf einem Schmierzettel, die wie folgt strukturiert ist:

x	y	Ausdruck A
true	true	Wert des Ausdrucks A
true	false	Wert des Ausdrucks A
false	true	Wert des Ausdrucks A
false	false	Wert des Ausdrucks A

Ersetzen Sie dabei den Ausdruck *A* mit

- (a) `!(x!=y && x==y)` (b) `x || true` (c) `x && false` (d) `(x&&!y) || (!x)`

Überlegen Sie sich, welche Werte in der rechten Spalte zu erwarten sind, und seien Sie folglich in der Lage, den Ergebniswert erklären zu können. Schreiben Sie zudem ein Programm, mit dem Sie Ihre theoretischen Ergebnisse aus dem ersten Aufgabenteil überprüfen können.

Hausaufgabe 1 [Fahrkartautomat]

40 Punkte

Erstellen Sie im Repository ein Package **h1** und fügen Sie diesem Package die Klasse **H1_main** hinzu, welche die **main**-Methode enthält.

Ein Fahrkartautomat verkauft Tickets je nach Anzahl der befahrenen Zonen. Der Preis steigt mit jeder Zone, die zusätzlich durchfahren wird.

- Deklarieren Sie die **int**-Variable **zone**, sowie die **double**-Variable **price** und weisen Sie ihnen beliebige Werte zu.
- Schreiben Sie ein Java-Programm, das den Preis in Abhängigkeit des Wertes von **zone** bestimmt und in **speichert**. Nutzen Sie die folgende Übersicht zur Preisbestimmung:

Zonen	Zusatzkosten	Gesamter Preis
1	2,00€ Basispreis (kein weiterer Zuschlag)	2,00€
2	+0,35€	2,35€
3 & 4	+0,50€	2,85€
5	+0,70€	3,55€
ab 6	pauschal 4.00 € (kein weiterer Zuschlag)	4,00€

Hinweise:

- Verwenden Sie ausschließlich die Kontrollstruktur **switch** zur Erstellung Ihrer Lösung.
- Sie dürfen davon ausgehen, dass **zone** stets positiv ist.

Hausaufgabe 2 [Wahrheitswerte]

40 Punkte

Erstellen Sie im Repository ein Package **h2** und fügen Sie diesem Package die Klasse **H2_main** hinzu, welche die **main**-Methode enthält.

- Deklarieren Sie die **boolean**-Variablen **x,y** sowie **a,b,c** und die **int**-Variable **input**. Allen Variablen weisen Sie beliebige Werte zu. Sie dürfen bei **input** davon ausgehen, dass nur einer der folgenden Werte eingegeben wird: **0,1,10** oder **11**.
- Wir definieren folgende Eigenschaften:
 - E1: **x** ist wahr (hat also den Wert **true**).
 - E2: **y** ist wahr.
 - E3: die Werte von **x** und **y** sind unterschiedlich.
- Weisen Sie **x** nur dann den Wert **true** zu, falls **input** den Wert **10** oder **11** hat. In den beiden anderen Fällen erhält **x** also den Wert **false**. Weisen Sie **y** nur dann den Wert **true** zu, falls **input** den Wert **11** oder **1** hat; in den anderen beiden Fällen wird **y** der Wert **false** zugewiesen.

Weiterhin erhält **a** nur dann den Wert **true**, falls E1 und E2 (gleichzeitig) erfüllt sind. Die Variable **b** erhält nur dann den Wert **true**, falls mindestens eine der Eigenschaften E1 und E3 erfüllt sind. Die Variable **c** verhält sich entsprechend der folgenden Wahrheitstabelle:

x	y	c
true	true	false
true	false	true
false	true	false
false	false	true

- Beispiel: Hat **input** den Wert 10, dann hat **x** den Wert **true**, **y** den Wert **false**, **a** den Wert **false**, **b** den Wert **true** und **c** den Wert **true**.

Hausaufgabe 3 [Temperaturkonvertierung]

20 Punkte

Erstellen Sie im Repository ein Package **h3** und fügen Sie diesem Package die Klasse **H3_main** hinzu, welche die **main**-Methode enthält.

Deklarieren Sie die **float**-Variable **celsiusFloat** und **fahrenheitFloat**, sowie die **double**-Variablen **celsiusDouble** und **fahrenheitDouble** und initialisieren Sie diese mit beliebigen Werten.

Schreiben Sie Programm, das die Konvertierung von Celsius in Fahrenheit ermöglicht, indem Sie jeweils ausschließlich mit den **float** bzw. **double**-Variablen arbeiten.

Hinweise:

- Die Formel für die Konvertierung lautet: $Celsius * \frac{9}{5} + 32$.
- Wir möchten Ihnen bei dieser Aufgabe zeigen wie schnell es zu Rundungsfehlern kommen kann. Beispielsweise führt ein Wert von 14,2 Celsius zu 57.55999965667725 Fahrenheit bei der Verwendung von **celsiusFloat** und **fahrenheitFloat**. Fangen Sie diese Rundungsfehler nicht ab.
- Testen Sie selbstständig.