

Projektblatt P1 (19 P)

Abgabe: Donnerstag 19. September 2024, 12:00h

Entpacken Sie zunächst die Archiv-Datei `vorgaben-P1.zip`, in der sich neben mehreren Rahmendateien für die zu lösenden Aufgaben auch die Hilfsklasse `IOTools` befindet. Ergänzen Sie zunächst alle Dateien mit Ausnahme von `IOTools.java` in der 1. Zeile durch einen Kommentar, der Ihren Namen beinhaltet. Ergänzen Sie die Dateien dann durch Ihre Lösungen gemäß der Aufgabenstellung unten. Der abgeänderte bzw. hinzugefügte Java-Sourcecode in den Dateien `Aufgabe_1_1.java`, `Aufgabe_1_2.java`, `Aufgabe_1_3.java` und

`Aufgabe_1_4.java` sollte syntaktisch richtig und vollständig formatiert sein. Antworten in Textform müssen stets in Form von (Block-)Kommentaren erfolgen. Alle Dateien mit der Endung `.java` sollten am Ende fehlerfrei übersetzt werden können.

Verpacken Sie die Dateien für Ihre Abgabe in einem ZIP-Archiv mit dem Namen `IhrNachname.IhrVorname.P1.zip`, die Sie auf Ilias hochladen.

Führen Sie dazu in dem Verzeichnis, in dem Sie die Dateien bearbeitet haben, folgenden Befehl auf der Kommandozeile aus:

```
zip IhrNachname.IhrVorname.P1.zip Aufgabe_1_*.java
```

Aufgabe 1: Fehler finden

5P

Die folgende `main`-Methode enthält zehn auskommentierte Zeilen, in denen jeweils eine Variable deklariert und initialisiert wird. Manche der Zeilen enthalten syntaktische Fehler.

Ergänzen Sie die Datei `Aufgabe_1_1.java`, die diese `main`-Methode enthält, wie folgt:

Fügen Sie nach jeder Zeile einen (mehrzeiligen) Kommentar ein. Wenn die jeweilige Zeile einen Fehler enthält, beschreiben Sie kurz, worin dieser Fehler besteht. Ist die Zeile dagegen syntaktisch korrekt (wenn auch vielleicht stilistisch unschön), geben Sie den Wert der Variablen in der Form an, wie er mit Hilfe der Methode `System.out.println` ausgegeben werden würde.

Beispiel:

```
// int x = 1.0;
// 1.0 ist eine Gleitkommazahl und kann daher nicht einer
// int Variablen zugewiesen werden.
```

```
// int z = 'A';
// Die Variable z hat den Wert 65
```

```
1  public static void main(String [] args){
2      // char c = " ";
3      // char d = ' ';
4      // double d1 = .1f1;
5      // double d2 = 12.5F-12;
6      // int i = 1E2;
7      // int j = 0b11;
8      // long k = 0xFEDCBA;
9      // long l = 1l;
10     // int x = (0x1 < 0x2);
11     // float f = (-1 > 0) ? 1.5 * 2 : 1.5 % 2;
12 }
```

Hinweis: Verwenden Sie den Compiler, um die Fehler zu finden. Übersetzen und verändern Sie dazu das (inhaltlich völlig sinnlose) Programm in der Datei `Aufgabe_1_1.java` schrittweise, indem Sie jeweils die Kommentarzeichen vor einer der Zeilen entfernen und dann das Programm übersetzen. Analysieren Sie den angezeigten Fehler und ergänzen Sie anschließend die Datei wie oben beschrieben.

Aufgabe 2: Operationen/Ausdrücke nachvollziehen 6P

Gegeben seien die ganzzahligen (Typ `int`) Variablen $a = 1, b = 2, c = 3$ und $d = 4$. Geben Sie für jeden der folgenden Ausdrücke an, welchen Typ und Wert das Ergebnis hat. Begründen Sie Ihre Antwort, indem Sie die bei der Auswertung des Ausdrucks nacheinander ausgeführten Operationen und die dazugehörigen Zwischenergebnisse angeben. Falls eine Variable in dem jeweiligen Ausdruck verändert wird (z.B. durch einen Inkrement/Dekrement-Operator), schreiben Sie zusätzlich die Wirkung dieses Operators auf die Variable dazu. Sie dürfen davon ausgehen, dass bei jeder Teilaufgabe zu Beginn die vier Ausdrücke ($a == 1$), ($b == 2$), ($c == 3$) und ($d == 4$) zu `true` ausgewertet würden.

Ergänzen Sie hierzu die Datei `Aufgabe_1_2.java` wie im Folgenden beschrieben.

- Ergänzen Sie die Erklärungen zur Auswertung der Ausdrücke und welche Werte der Ausdruck sowie die Variablen a, b, c und d am Ende haben, in den Blockkommentaren zu den Aufgabenteilen (a) - (f).
- Entfernen Sie die Kommentarzeichen vor den Ausdrücken, deklarieren Sie eine Variable mit dem für das jeweilige Ergebnis passenden Typ und weisen Sie den Ausdruckswert dieser Variablen zu.
- Ergänzen Sie die Ausgabeanweisungen so, dass der Ausdruckswert ausgegeben wird.

Beispiel: Für den Ausdruck $c++ * (d - b)$ ergibt sich:

```
/* Aufgabenteil (x)
Ergebnistyp: int
Ergebniswert: 6
1.) c++    -> Ausdruckswert 3, c = 4,
1.) d - b  -> Ausdruckswert 4 - 2 = 2
2.) 3 * 2  -> 6
*/
int e = c++ * (d - b);
System.out.println("e = " + e);
```

- (a) `a + b * c - d / b;`
- (b) `a + a++ + ++a + a;`
- (c) `a -= b += ++c;`
- (d) `a * d < b * c || a + d != b + c;`
- (e) `!(-1 * a + b < d - c) ? ++a * ++a : c++ + ++c;`
- (f) `(h1 = ((a > b) ? b : a)) > (h2 = ((c > d) ? d : c)) ? h2 : h1;`
 Hierbei seien *h1* und *h2* Variablen vom Typ `int`.

Anmerkung: Die Codestücke enthalten zum Teil stilistisch sehr schlechten Code und dienen nur dazu, zu testen, inwieweit Sie mit komplexen Ausdrücken zurechtkommen.

Aufgabe 3: Arithmetische Ausdrücke

3P

Ergänzen Sie die Datei `Aufgabe_1_3.java` zu einem Programm, welches einem Nutzer mitteilt, wie sich sein Kapital bei einer Verzinsung mit einem bestimmten Zinssatz nach einer frei wählbaren Anzahl von Jahren entwickelt. Der Nutzer soll zunächst nach seinem Grundkapital K_0 , dem Zinssatz p und der Anzahl von Jahren j gefragt werden. Unter stetiger Verzinsung lässt sich dann der Kapitalwert K_j nach j Jahren mit folgender Formel berechnen:

$$K_j = K_0 * (1 + \frac{p}{100})^j$$

Wählen Sie zur Repräsentation des Grundkapitals, des Zinssatzes und der Anzahl der Jahre Variablen eines geeigneten Typs und lesen Sie Werte für diese mit Hilfe der `IOTools` Klasse ein. Berechnen Sie das Ergebnis nach obiger Formel und geben Sie das Ergebnis aus.

Verwenden Sie die Methode `Math.pow` als Hilfe bei der Berechnung von a^b für zwei `double` Variablen `a` und `b`.

Ein Beispiel für einen Programmlauf könnte wie folgt aussehen (Benutzereingaben hier in grün hervorgehoben):

Bitte geben Sie folgende Daten ein:

Grundkapital: 10000

Zinssatz: 2.0

Anzahl Jahre: 10

Das Ergebnis nach 10 Jahren beträgt 12189.944199947573

Aufgabe 4: Ganzzahldivision mit Rest

5P

Ergänzen Sie die Datei `Aufgabe_1_4.java` zu einem Programm, dass zunächst zwei Uhrzeiten einliest (dabei sollen die Stunden und die Minuten getrennt eingegeben werden). Die beiden Uhrzeiten sollen den Beginn und das Ende einer Parkzeit repräsentieren.

Berechnen Sie aus den beiden Zeitpunkten die Parkdauer und eine Parkgebühr (mit 1,50 € pro angefangener Stunde). Geben Sie dann beide Resultate auf der Konsole aus.

Hinweis: Wandeln Sie zunächst die beiden Zeitpunkte in Minuten um und berechnen Sie die Differenz. Aus dem Differenzwert können Sie dann die Anzahl der angefangenen Stunden ableiten. Bei einer glatten Stundenzahl (mit $n \geq 0$ Stunden und 0 Minuten sollen $n * 1.50$ € berechnet werden. Nutzen Sie den ternären Bedingungsoperator, um festzulegen, ob Sie $n * 1.5$ oder $(n + 1) * 1.5$ rechnen müssen.

Der Unicode (UTF-16) für das Eurozeichen ist 20AC

Sie dürfen in dem Programm voraussetzen, dass die Eingabewerte korrekt und beide Zeitpunkte am gleichen Tag sind.

Ein Programmlauf könnte dann beispielsweise wie folgt aussehen (Benutzereingaben hier in grün hervorgehoben):

```
Parkbeginn, Stunde: 8
Parkbeginn, Minute: 15
Parkende, Stunde: 11
Parkende, Minute: 10
Parkdauer: 2 Stunde(n) 55 Minuten
Parkgebühr: 4.5 €
```