

JDK, Ein-/Ausgabe, Berechnungen

Die folgenden Aufgaben werden Sie mit dem Java Development Kit (JDK), der Ein- und Ausgabe auf der Konsole, sowie einfachen Berechnungen vertraut machen.

Aufgabe 1: Fehlersuche (6 P.)

Tippen Sie folgendes Java Programm ab, kompilieren Sie es (`javac`) und führen Sie es anschließend aus (`java`). Gehen Sie dabei folgendermaßen vor:

1. Installieren Sie das Java Development Kit (JDK) auf Ihrem System. Folgen Sie dazu den Anweisungen in den Folien am Moodle.
2. Erstellen Sie ein Projektverzeichnis auf Ihrem Rechner (beispielsweise `/somewhere/SW1/UE01_MyMatrikelnummer`) und kopieren Sie die beiden Dateien `In.java` und `Out.java` in diesen Ordner (diese können vom Moodle heruntergeladen werden).
3. Starten Sie einen Texteditor Ihrer Wahl (beispielsweise Notepad++ auf Windows) und tippen Sie folgendes Programm ab. Es enthält absichtlich Fehler, die in den nächsten Schritten behandelt werden.

```
class MyFirstProgram {
    public static main(String[] args) {
        Out.println("Hello user, how old are you (in years)?");
        in age = In.readint();
        Out.println(String.format("So you're %f years old?", age);

        int days = age x 365;
        Out.println("Wow! That's around " + days + " days!");

        int Hours = days + 24
        Out.printl("Or, in terms of hours, over " + Hours " hours!");
    }
}
```

4. Speichern Sie das Programm unter dem Namen `MyFirstProgram.java` im selben Projektverzeichnis wie die Dateien `In.java` und `Out.java` ab. Öffnen Sie dann eine Kommandozeile und navigieren Sie mit `cd /somewhere/SW1/UE01_MyMatrikelnummer` in dieses Verzeichnis.
5. Kompilieren Sie das Programm mit dem Befehl `javac MyFirstProgram.java` (inklusive Dateierweiterung `.java`). Der Compiler wird Meldungen ausgeben und Sie auf Fehler im Programm aufmerksam machen. Lesen Sie Fehlermeldungen *immer* aufmerksam. Sie enthalten wertvolle Informationen zur Behebung der Fehler! Lernen Sie aus Fehlern und merken Sie sich ihre Ursprünge für die Zukunft! Im Laufe dieser Lehrveranstaltung werden Sie höchstwahrscheinlich noch häufig ähnliche Fehlermeldungen zu sehen bekommen. Es ist daher wichtig, dass Sie lernen, wie diese zu lesen und zu verstehen sind.
6. Beheben Sie die vom Compiler gemeldeten Fehler, speichern sie die geänderte `.java`-Datei wieder ab und wiederholen Sie den Kompiliervorgang, bis keine Fehler mehr vorhanden sind.
7. Starten Sie das Program mit dem Befehl `java MyFirstProgram` (nicht `java MyFirstProgram.java`) und geben Sie ein Alter (Jahre) auf der Konsole ein. Das Programm hat auch semantische ("logische") Fehler, die der Compiler nicht erkennen kann. Beheben Sie diese, damit das Programm eine korrekte Ausgabe liefert.
8. Finden Sie außerdem eingebaute Stilfehler (nach den Stilrichtlinien) und beheben Sie diese.

Eine korrekte Ausführung bei Eingabe von 26 sollte zu folgender Ausgabe führen:

```
Hello user, how old are you (in years)? // Ausgabe
27 // Eingabe
So you're 27 years old? // Ausgabe
Wow! That's around 9855 days! // Ausgabe
Or, in terms of hours, over 236520 hours! // Ausgabe
```

Aufgabe 2: Berechnungen (6 P.)

Entwickeln Sie ein Programm `Calculations.java`, das vier Ganzzahlen einliest und daraus folgende Werte berechnet und ausgibt (Gleitkommazahlen sollen auf zwei Kommastellen genau ausgegeben werden):

1. Die Summe der vier Zahlen.
2. Das Produkt der vier Zahlen.
3. Die Summer der ersten zwei Zahlen, multipliziert mit der Summer der letzten zwei Zahlen.
4. Den Quotienten der ersten Zahl (Dividend) und der vierten Zahl (Divisor). Achten Sie darauf, dass die Division als Gleitkommadivision, nicht als Ganzzahldivision, ausgeführt wird.
5. Den Rest (Modulo) der Ganzzahldivision aus der ersten Zahl und der dritten Zahl.
6. Die Quadratwurzel der vierten Zahl.
7. Die Potenz aus der zweiten Zahl hoch der vierten Zahl (als Ganzzahl).
8. Unter der Annahme, dass die Zahlen der Reihe nach **a**, **b**, **c** und **d** heißen: $\frac{a}{4} \cdot \frac{b^{\pi \cdot c}}{d-1}$ (als Gleitkommazahl).

Beispielausgabe:

```
1st number: 2      // Eingabe
2nd number: 3      // Eingabe
3rd number: 4      // Eingabe
3rd number: 5      // Eingabe
Sum: 14
Product: 120
Sum of first two times sum of last two: 45
Quotient of first and last: 0,40
Rest of first and third: 2
Square-root of last: 2,24
Second to the power of the fourth: 243,00
Complex equation: 1010,89
```

Aufgabe 3: Grundstücksberechnung (6 P.)

Entwickeln Sie ein Programm `Area.java`, das die Länge und die Breite eines rechteckigen Grundstücks als Gleitkommazahlen einliest. Berechnen Sie folgende Werte und geben Sie diese anschließend auf zwei Kommastellen genau aus:

1. Die Fläche des Grundstücks.
2. Die Länge der Grundstücksgrenze (also der Rechtecksumfang).
3. Die Diagonallänge des Grundstücks.
4. Die Menge an Rasensamen (in kg), die benötigt werden, um das gesamte Grundstück zu sähen, wenn pro Quadratmeter 25 Gramm benötigt werden.
5. Der Preis, der gezahlt werden muss, wenn $\frac{3}{4}$ des Grundstücks zu einem Meterpreis von 30.50€ eingezäunt werden.
6. Die kürzere Seite des Grundstücks (mittels `Math.min`).
7. Stellen Sie sich vor, sie wollen nun ein halbkugelförmiges Iglu bauen. Der Radius des Iglus beträgt $\frac{1}{4}$ der kürzeren Seitenlänge. Wie groß ist das Volumen¹ des Iglus?

Achten Sie darauf, dass sie Werte, die sie für weitere Berechnungen benötigen, in Variablen zwischenspeichern (beispielsweise die Grundstücksgröße).

Beispielausgabe:

```
Area width (in meters): 15.5    // Eingabe
Area height (in meters): 20.5   // Eingabe
Area: 317,75 square meters
Circumference: 72,00 meters
Diagonal: 25,70 meters
Seeds: 7,94 kg
Fence price: 1647,00 Euro
Short side: 15,50 meters
Iglu: 121,86 cubic meters
```

¹<https://de.wikipedia.org/wiki/Kugel>

Aufgabe 4: Aquastore (6 P.)

Aquastore ist eine Tierhandlung, die verschiedene Produkte für Aquarienbesitzer anbietet. *Aquastore* bietet den Service, nur die Aquariumsgröße bekannt geben zu müssen, und dem Käufer wird ein passendes Einkaufspaket mit typischen Produkten zusammengestellt. Schreiben Sie ein Java-Programm `Aquastore.java`, das eine Bestellmaske für diesen Service implementiert. Gehen Sie dabei folgendermaßen vor:

1. Lesen Sie die Größe des Aquariums (in Liter) ein. *Hinweis: Sie müssen keine Fehlerkontrolle bei den Eingaben durchführen. 0 bzw. negative Mengenangaben sind erlaubt.*
2. Berechnen Sie die Mengen folgender Produkte: 5 Gramm Flockenfutter je Liter Beckengröße, dazu 1.8 Milliliter Unterwasserpflanzendünger je Liter Beckengröße, und 2 Gramm Filterwolle je Liter Beckengröße.
3. Berechnen Sie den Preis der einzelnen Produkte: Flockenfutter kostet 10 Euro je Kilogramm, Wasseraufbereiter 4 Cent je Milliliter, und Filterwolle 30 Euro je Kilogramm.
4. Berechnen Sie den Gesamtpreis.
5. Geben Sie eine formatierte Rechnung mit sämtlichen Informationen der Bestellung aus.

Eine Ausführung mit formatierter Ausgabe könnte auf der Konsole wie folgt aussehen:

```
=====
Aquastore - Order
=====
Aquarium size (in liter): 620                      // Eingabe
=====
Food          3,10 kg x 10,00 =      31,00 EUR
Fertilizer    1116,00 ml x  0,04 =    44,64 EUR
Filter Wool   1,24 kg x 30,00 =     37,20 EUR
-----
TOTAL                               112,84 EUR
=====
```

Zusätzliche Anweisungen

- Implementieren Sie alle Programme ohne Verzweigungen oder Schleifen.
- Verwenden Sie zur Implementierung der Ein- und Ausgabe für alle Programme nur die Funktionen der beiden bereitgestellten Klassen `In` und `Out`, wie in der Übung gezeigt. Die entsprechenden Java-Dateien samt HTML-Dokumentation sind in Moodle unter *InOut.zip* zu finden.
- Implementieren Sie formatierte Ausgaben, indem Sie `String.format` in Kombination mit `Out.print` und `Out.println` verwenden, wie in der Übung gezeigt. Verwenden Sie als Referenz den Foliensatz `StringFormat.pdf` im Moodle.
- Vermeiden Sie Codeduplikation.
- Verwenden Sie für Variablen passende Datentypen Ihrer Wahl.
- Alle Namen (Klassennamen, Variablennamen, Methodennamen, etc.) sind in Englisch zu wählen.
- Formatierung, Namenswahl, etc. fließen in die Bewertung mit ein.
- Verwenden Sie, wo nötig, `Math`-Funktionen der Java-Bibliothek, wie in Vorlesung und Übung gezeigt.

Abzugeben

Geben Sie *eine .zip Datei* mit dem Namen kxxxxxxx_UExx.zip ab (Beispiel: k01234567_UE02.zip). Darin muss enthalten sein:

- Der *Source Code* aller Aufgaben der aktuellen Übung (alle `.java` Dateien, keine `.class` Dateien).
- Zu jeder Aufgabe ist ein Testprotokoll in Form von mehreren Testausgaben anzufertigen. Testen Sie nicht nur einen Fall, sondern mehrere allgemeine Fälle sowie typische Grenzfälle (negative Zahlen, null, etc.). Die Testprotokolle dürfen entweder gesamt in einer Datei zusammengefasst werden (`Testprotokoll.txt` oder `Testprotokoll.pdf`), oder in eine Datei pro Aufgabe aufgeteilt werden (`Testprotokoll_BspXX.txt` oder `Testprotokoll_BspXX.pdf`).