



Die Hausaufgaben müssen von den Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog <https://seblog.cs.uni-kassel.de/ws2223/einfuehrung-in-die-informatik/> zu berücksichtigen.

Abgabefrist ist der 10.11.2022 - 23:59 Uhr

Vorbereitung

Zur Abgabe dieser Hausaufgabe muss zunächst ein neues Respository angelegt werden. Mit dem folgenden Link kannst du das Respository für Hausaufgabe 3 anlegen bzw. einsehen.

https://classroom.github.com/a/6vCLJrr_

Achte bei jeder Aufgabe auf das verlangte Dateiformat. Andere Formate werden nicht akzeptiert und folglich mit **0** Punkten gewertet. Handschriftliche Abgaben werden nicht akzeptiert.

Nicht oder zu spät abgegebene (Teil-)Aufgaben werden mit 0 Punkten bewertet.

Aufgabe 1 - Fehler (6P)

a) Gegeben ist der folgende Code. Hier haben sich einige Fehler eingeschlichen.

```
/open EidiInput.jsh

String Name = EidiInput.readLine();
int age = EidiInput.readLine();

if (age >= 18) {
    System.out.println(Name + " is an adult.")
}
else {
    System.out.println(Name + " is a child.");
}

/exit
```

Finde die Fehler und beschreibe kurz das Problem. Schreibe deine Antworten in Form eines Kommentars am Ende der jeweiligen Zeile.

Beispiel:

```
String beispiel = 5; // Fehler: einem String kann kein Integer zugewiesen werden
```

b) Korrigiere die Fehler! Korrigiere sie direkt in der selben Datei. Achte darauf, dass deine Kommentare aus (a) erhalten bleiben. Nach der Korrektur sollte das Programm fehlerfrei ausführbar sein.

Beispiel:

```
int beispiel = 5; // Fehler: einem String kann kein Integer zugewiesen werden
```

Gib deine Lösung in einer Datei mit dem Namen `errors.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 2 - While (3P)

Schreibe ein Programm mit drei Absätzen, welche jeweils eine `while`-Schleife enthalten. Nachdem der User eine positive Zahl `x` eingegeben hat, sollen folgende Ausgaben nacheinander passieren:

- Ein Countdown beginnend von `x` bis inklusive 0 wird ausgegeben
- Es wird von 0 bis exklusive `x` hochgezählt
- Die Zahl `x` wird `x`-mal ausgegeben

Das Grundgerüst des Programms ist vorgegeben. Befülle die Bedingungen und Rumpfe der `while`-Schleifen, um das vorgegebene Verhalten zu erzielen. Es dürfen zusätzliche Hilfsvariablen verwendet werden.

Folgende Konstrukte dürfen **nicht** verwendet werden:

- `if`
- `break`
- Variablen vom Typ `boolean`

Gegeben ist folgender Code:

```
/open EidiInput.jsh

System.out.print("Please enter positive number: ");
int x = EidiInput.readInt();

System.out.println();
System.out.println("Count from x down to (inclusive) 0:");
// helper variables
while (/* TODO */) {
    // TODO
}

System.out.println();
System.out.println("Count from 0 up to (exclusive) x:");
// helper variables
while (/* TODO */) {
    // TODO
}

System.out.println();
System.out.println("Print x times the number x:");
// helper variables
while (/* TODO */) {
    // TODO
}

System.out.println();
System.out.println("Bye!");

/exit
```

Programm-Ablauf mit User-Eingabe 5:

```
Please enter positive number: 5

Count from x down to (inclusive) 0:
5
4
3
2
1
0

Count from 0 up to (exclusive) x:
0
1
2
3
4

Print x times the number x:
5 5 5 5 5

Bye!
```

Programm-Ablauf mit User-Eingabe 8:

```
Please enter positive number: 8

Count from x down to (inclusive) 0:
8
7
6
5
4
3
2
1
0

Count from 0 up to (exclusive) x:
0
1
2
3
4
5
6
7

Print x times the number x:
8 8 8 8 8 8 8 8

Bye!
```

Programm-Ablauf mit User-Eingabe 0:

```
Please enter positive number: 0

Count from x down to (inclusive) 0:
0

Count from 0 up to (exclusive) x:

Print x times the number x:

Bye!
```

Gib deine Lösung in einer Datei mit dem Namen `while.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 3 - Booleans (4P)

Gegeben ist folgender Code. Ergänze bei den Variablen vom Typ boolean je einen Ausdruck, der die gewünschten Eigenschaften darstellt. Die Stellen sind mit einem „TODO“-Kommentar markiert. Verändere ansonsten **nichts** am Quellcode.

```
/open EidiInput.jsh

System.out.print("Enter first number: ");
int numberOne = EidiInput.readInt();

System.out.print("Enter second number: ");
int numberTwo = EidiInput.readInt();

boolean numberOneIsBigger = // TODO (a)
if (numberOneIsBigger) {
    System.out.println("The first number is bigger than the second one.");
} else {
    System.out.println("The second number is bigger than or equal to the first one.");
}

boolean productBiggerThanHundred = // TODO (b)
if (productBiggerThanHundred) {
    System.out.println(numberOne + " * " + numberTwo + " is bigger than 100.");
}

boolean bothEven = // TODO (c)
if (bothEven) {
    System.out.println("Both numbers are even.");
}

boolean coolNumber = // TODO (d)
if (coolNumber) {
    System.out.println("One of the numbers is 42. Very cool!");
}

System.out.println("Goodbye!");

/exit
```

Gib deine Lösung in einer Datei mit dem Namen `booleans.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 4 - Taschenrechner (13P)

Programmiere einen Taschenrechner! Zunächst soll der erste ganzzahlige Wert eingelesen werden. Danach soll die Rechenoperation ausgewählt werden, der User muss hierzu ein +, -, * oder / eingeben. Nutze hierfür `readChar()` aus `EidiInput`. Anschließend soll der zweite ganzzahlige Wert eingelesen werden. Das Programm soll nun `<Wert1><Operation><Wert2>` ausrechnen und das Ergebnis ausgeben. Anhand des Operators muss eine Fallunterscheidung durchgeführt werden, um das gewünschte Ergebnis auszurechnen.

Eine Division durch 0 soll mithilfe einer Bedingung abgefangen werden. In diesem Fall soll dem User die Fehlermeldung „Can't divide by 0“ ausgegeben werden.

Achte außerdem darauf, dass bei einer Division auch Zahlen herauskommen können, die nicht ganzzahlig sind. Deshalb sollen bei einer Division `doubles` verwendet werden.

Nach Ausgabe eines Ergebnisses bzw. der Fehlermeldung soll eine leere Zeile ausgegeben werden, um die Übersicht zu verbessern.

Nach einer Berechnung wird der User gefragt, ob er eine weitere Berechnung durchführen möchte. Hierfür wird ein String eingelesen. Antwortet der User „yes“, beginnt das Programm erneut nach der ersten Zahl zu fragen usw. Somit kann der User beliebig viele Berechnungen durchführen (Tipp: `while`). Ansonsten wird das Programm beendet. Zum Abschied wird „Have a nice day!“ ausgegeben.

Die Ausgabe soll wie folgt lauten, wobei die User-Eingaben blau hervorgehoben wurden.

Beispiel-Ablauf:

```
Please enter your first number: 6
Choose + - * / operation: *
Now enter your second number: 7
Your result: 42

Do you want to do another calculation? (yes/no) : yes
Please enter your first number: 5
Choose + - * / operation: /
Now enter your second number: 0
Can't divide by 0

Do you want to do another calculation? (yes/no) : yes
Please enter your first number: 5
Choose + - * / operation: /
Now enter your second number: 2
Your result: 2.5

Do you want to do another calculation? (yes/no) : no
Have a nice day!
```

Gib deine Lösung in einer Datei mit dem Namen `calc.js` im GitHub-Repository zur aktuellen Hausaufgabe ab.