

Übungsaufgaben zu „Einführung in die Programmierung“ , WS18/19, Georg Ringwelski

Die Bearbeitung der Übungsaufgaben zählt als Prüfungsvorleistung für die Studiengänge IIb und IWb. Zur Prüfung am Ende des Semesters wird nur zugelassen, wer:

- Mindestens 11 der 13 der Übungsaufgaben in ausreichender Qualität eigenständig angefertigt und
- ihre/seine Lösungen in der angegebenen Woche persönlich (möglichst) in der Übungszeit präsentiert. Spätere Präsentationen werden nur mit Attest akzeptiert

1. Hello World (fällig in KW 41)

- a) Schreiben Sie ein Java-Programm, übersetzen es und führen es in Ihrer Arbeitsumgebung aus
- Starten Sie eine Editor (zB JEdit, Notepad, emacs...) und geben dort folgendes Programm ein:

```
class Hello{  
  
    public static void main(String[] args){  
  
        System.out.println("Hello World");  
  
    }  
  
}
```

- Speichern Sie dieses Programm unter dem Dateinamen „Hello.java“ ab
 - Installieren Sie ggf. das Java Developer Kit (JDK) auf ihrem Rechner
 - Kompilieren Sie Ihre Programm in der Eingabeaufforderung (oder shell) mit dem Befehl „javac Hello.java“
 - Führen Sie Ihr Programm aus mit dem Befehl „java Hello“
- b) Schreiben Sie ein Java-Programm, das ein an der Kommandozeile eingegebenes Wort wieder ausgibt. Nennen Sie das Programm „Echo“.
- Zum Beispiel soll auf eine Eingabe „java Echo Fritz“ soll die Ausgabe „Fritz“ folgen.
- Hinweis:** Auf das erste Argument der Kommandozeile hat man in Java mit der Variable „args[0]“ vom Typ String Zugriff. Auf das i-te Argument mit „args[i]“.

2. Variablen, Typen, Zuweisungen (fällig in KW 42)

- a) Schreiben Sie ein Java Programm, das an der Kommandozeile zwei Zahlen einliest. Diese beiden Zahlen werden dann in unterschiedlichen Variablen vom Typ byte, long, und double kopiert. Führen Sie dann die Division der Zahlen unterschiedlichen Typs aus und geben das Ergebnis aus.
- b) Schreiben Sie ein Java Programm, das für den Quotienten 11111/333 drei verschiedene Ergebnisse liefert.

Hinweis 1: In der Methode `public void main(String[] args)` hat man in Java Zugriff auf die Werte, die Nutzer in der Kommandozeile eingegeben haben durch den Parameter `args`, wobei der erste Parameter in `args[0]`, der zweite in `args[1]` usw. steht.

Hinweis 2: Die Parameter in der Kommandozeile sind Zeichenketten (Strings) und müssen zunächst in Zahlen umgewandelt werden. Um in Java eine Zeichenkette `a` in eine Integer-Zahl `x` umzuwandeln verwendet man in Java den Ausdruck `int x = Integer.parseInt(a)`; Für andere Typen gibt es in der Java-Bibliothek entsprechende Befehle.

3. Steueranweisungen (fällig in KW 43)

- a) Schreiben Sie ein Java Programm „HalloX“, das eine Zahl `x` von der Kommandozeile einliest und das Wort „Hallo“ genau `x` mal ausgibt.
- b) Schreiben Sie ein Java Programm, das die Zeichenketten in der Kommandozeile einliest und zählt, wie oft das Wort „Hallo“ eingegeben wurde.
 - Erstellen Sie eine Klasse „HalloCounter“ in einer entsprechenden neuen Datei analog zu `Hello.java`
 - Schreiben Sie einen Algorithmus zur Analyse des Arrays der Eingabeparameter „args“, indem Sie es mit einer Schleife (z.B. `while`) durchiterieren. Zählen Sie dabei in einer Variable vom Typ `long`, wie oft das Wort „Hallo“ darin vorkommt. Geben Sie anschließend diese Zahl aus.
 - Probieren Sie die Funktionalität Ihres Programms indem Sie es mit unterschiedlichen (auch leeren) Parameterlisten aufrufen.
- c) Schreiben Sie ein Programm „Dreieck“, das eine Zahl `x` einliest und dann an der Konsole ein Dreieck ausgibt, das aus „*“ besteht und dessen Seitenlänge beider Katheten `x` ist.

Bsp:

```
C:\Java Dreieck 4
```

```
****
```

```
***
```

```
**
```

```
*
```

```
C:\
```

Hinweis: um in Java zwei Zeichenketten `a` und `b` zu vergleichen benutzt man den Ausdruck `„a.equals(b)“`, der wahr ist, wenn `a` und `b` gleich sind.

Hinweis: Die Anzahl der Elemente eines Arrays, wie z.B. den Kommandozeilenparametern „String[] args“ kann man mit dem Ausdruck `„args.length“` abfragen

4. Arrays (fällig in KW 44)

Schreiben Sie ein Programm, das **ohne Verwendung eines Sortieralgorithmus**

- ein Array mit 87 zufälligen ganzen Zahlen zwischen 1 und 1000 füllt,
- das kleinste und das größte Element findet und ausgibt,

- das arithmetische Mittel der Zahlen ausgibt,
- den Medianwert der Zahlen ausgibt.

Hinweis: Zufallszahlen können Sie mit der Klasse Random aus der Java API erzeugen. Wie das geht finden Sie in der API Dokumentation dazu.

5. Ein erstes brauchbares Programm (fällig in KW 45)

Schreiben Sie ein Java Programm Bruchrechner, das Brüche addiert, subtrahiert, multipliziert oder dividiert und (ungekürzt) ausgibt. Zur Ein- und Ausgabe soll die Eingabeaufforderung verwendet werden. Der Aufruf des Programms erfolgt dann mit einer Rechenoperation (add, sub, mul, div), gefolgt von vier ganzen Zahlen, also zum Beispiel „java Bruch add 1 2 3 4“. Die Ausgabe in diesem Beispiel wäre dann $1/2 + 3/4 = 10/8$.

- Schreiben Sie ein Java-Programm, das die Parameter der o.g. Eingabe einliest.
- Implementieren Sie nun für alle vier Rechen-Operationen Algorithmen, die die korrekten Ergebnisse berechnen. Lassen Sie sich die Ergebnisse ausgeben und prüfen Sie, ob Ihre Algorithmen richtig arbeiten. Probieren Sie das auch mit negativen Zahlen und der 0.
- Implementieren Sie eine Fehlerbehandlung, die das Programm mit einer Meldung abbricht, wenn eine Division durch 0 auftritt.

6. Strukturierung des Programms (fällig in KW 46)

- Strukturieren Sie Ihr Programm aus der vorigen Woche so, dass jede Rechenoperation in einer separaten Methode durchgeführt wird
- Implementieren Sie in einer weiteren Methode das Kürzen von Brüchen (mit dem Euklidischen Algorithmus, den Sie im Internet nachlesen können). Rufen Sie nach jeder Berechnung nun noch diese Methode auf, so dass nur noch vollständig gekürzte Brüche ausgegeben werden.

Realisieren Sie das ohne die Verwendung globaler Variablen.

7. Automatisches Testen (fällig in KW 47)

Schreiben Sie ein Programm, das die Prozeduren Ihres Bruchrechner-Programms testet.

- Erstellen Sie ein Programm BruchrechnerTest, mit den Testmethoden addTest, mulTest, divTest, subTest und kuerzenTest ohne Parameter oder Rückgabewert.
- Implementieren Sie in BruchrechnerTest eine main-Methode, die alle Testmethoden jeweils einmal aufruft.
- Die Testmethoden rufen jeweils ihre entsprechende Methode (also zB addTest die Methode add) in Bruchrechner auf und überprüft, ob das Ergebnis richtig ist. Verwenden Sie zur Überprüfung des Ergebnisses eine Methode void erwaerteGleichheit(...), die zwei (oder vier) Parameter hat und ausgibt, ob die beiden Werte (oder Wertpaare) gleich sind oder nicht.
- Definieren Sie in den fünf Testmethoden jeweils mindestens drei interessante Testfälle, wie negative Werte, sehr große oder sehr kleine Werte und die Null. Probieren Sie, ob Ihr Programm in allen Fällen richtig arbeitet, indem Sie alle Tests automatisch durchführen. Ihr Ziel als Tester ist es, scheiternde Tests zu schreiben, also Fälle zu finden, in denen Ihr Programm nicht richtig arbeitet.

8. Minesweeper - Daten (fällig: KW 48)

- Erstellen Sie eine Klasse „Feld“, die für ein Feld im Spielplan steht. Die Eigenschaften der Objekte dieser Klasse sind: „ist Mine?“, „geöffnet?“, „Anzahl der benachbarten Minen“ ...
- Erstellen Sie dann eine Klasse Spielfeld, in der ein zweidimensionales Array von Objekten der Klasse Feld referenziert wird. Die Größe des Spielfeldes und die Anzahl der Minen sollen dabei im Konstruktor als Parameter angegeben werden können. Der Konstruktor verteilt dann die Minen zufällig im Spielfeld und definiert bei den Nachbarfeldern die Anzahl der angrenzenden Minen.

9. Minesweeper Test der Daten (fällig in KW 49)

Erstellen Sie ein Programm zum Testen der Klasse Spielfeld. Dabei ist darauf zu achten, dass die Zufallszahlen im Programm keinen Einfluss auf das Testergebnis haben. Sie können dazu z.B. die Minen durch einen Befehl ihres Testprogramms an eine konkrete Stelle legen und dann prüfen, ob das die gewünschten Auswirkungen hatte:

- Sind wirklich genauso viele Minen vorhanden, wie angegeben wurden?
- Stimmen die Zahlen in den Feldern?
- Testen Sie mit unterschiedlichen Eingabewerten

Beseitigen Sie alle Fehler aus ihrem Programm, bis der Test erfolgreich ist.

10. Minesweeper Logik(fällig in KW 50)

- Implementieren Sie in der Klasse Spielfeld eine Methode „linksKlick(int x, int y)“, die das Anklicken eines Feldes implementiert. Dabei soll das angeklickte Feld und evtl. benachbarte Felder offengelegt werden. Orientieren Sie sich bei der Funktionalität am bekannten Vorbild.
- Testen Sie Ihre Implementierung, indem Sie für linksKlick und ggf. weitere Prozeduren geeignete Testmethoden in ihre Testprogramm aus der vorigen Woche schrieben. Dazu brauchen Sie Situationen, die wiederherstellbar sind. Hier ist das zufällige Verteilen der Minen natürlich nicht geeignet...

11. Minesweeper GUI (fällig in KW 2)

Implementieren Sie eine graphische Oberfläche mit javafx (als Erweiterung von `javafx.application.Application`) mit einem Button (`javafx.scene.control.*`) für jedes Feld.

Implementieren Sie auch ein Menü, in dem man das Spiel verlassen, ein neues Spiel beginnen oder die Anzahl der Minen oder die Spielfeldgröße definieren kann. Das Menü und die Buttons auf dem Spielfeld brauchen in dieser Woche noch nicht mit irgendwelchen Funktionen verknüpft zu werden. Es geht nur um die Implementierung einer ansprechenden Benutzeroberfläche.

- Entwerfen Sie Ihre Oberfläche (mit Menüs) zunächst auf Papier
- Erstellen Sie eine Klasse MinesweeperGUI, in der sie dann die GUI so implementieren, wie Sie sie skizziert hatten.

12. Minesweeper Interaktivität (fällig in KW 3)

- Implementieren Sie nun EventHandler für die Buttons im Spielfeld, die die Methode „linksKlick“ einer Spielfeld-Instanz aufruft und anschließend die graphische Darstellung der Felder aktualisiert. Geöffnete Felder enthalten dann entweder ein Minen-Symbol oder eine

Zahl, die die benachbarten Minen angibt. Öffnet der Spieler einen Button mit Mine, so hat er verloren und wird durch ein Geräusch darauf aufmerksam gemacht. Es werden dann auch alle vorhandenen Minen angezeigt. Die von ihm geöffnete wird dabei rot hinterlegt

- Implementieren Sie EventHandler für Ihre Menüeinträge aus der vorigen Woche, die den gewünschten Effekt haben.

13. Minesweeper Deluxe (fällig in KW 4)

- Implementieren Sie eine weitere Methode „rechtsKlick“, mit der Buttons als Minenverdächtig markierbar sind. Hier soll sich nur das Aussehen des Buttons ändern. Bei einem weiteren Klick wird das ursprüngliche Aussehen wieder hergestellt
- Implementieren Sie einen tollen Effekt für die Situation, in der der Spieler gewonnen hat