Übungsblatt 10

zu bearbeiten bis 15.07.2010

Das letzte Übungsblatt (für dieses Semester)

Wir beginnen dieses Blatt mit zwei guten Nachrichten: dieses Blatt ist erstens das letzte für dieses Semester und zweitens freiwillig zu bearbeiten. Freiwillig? Nun, einige von Euch haben ja noch einen Aufhol-Bedarf bei den Übungspunkten, um die erforderlichen Vorleistungen zu erbringen (siehe Kasten). Mit diesem Blatt helfen wir hier ein wenig nach: wer es macht, bekommt die hier erreichten Punkte noch gutgeschrieben (bis zu 70 können erreicht werden!), wer es nicht macht, hat aber keinen Nachteil, da es in die 50%-Regelung nicht mehr hineingerechnet wird.

Heisst das, dass dieses Blatt nun uninteressant ist? Keineswegs, denn die neuen Inhalte (z.B. der Umgang mit den Kommandozeilenparametern) sind klausurrelevant. Außerdem ist das Blatt nochmals eine schöne Übungsmöglichkeit für die anstehende Klausur, insbesondere Aufgabe 2! Insofern lohnt sich die Beschäftigung mit diesen Aufgaben auf jeden Fall auch als Klausurvorbereitung.

Aufgabe 1 – Das Collatz-Problem (15 Punkte)

Am vergangenen Dienstag (06.07.) hätte ein Mathematiker seinen 100. Geburtstag gefeiert, wenn er nicht schon tot wäre: Lothar Collatz. Dieser Mathematiker hatte ein Problem. „Gut,“, denkt Ihr euch jetzt, „nach zwei Semestern an der Uni haben wir schon mitbekommen, dass viele Mathematiker ein Problem haben...“. Mag sein, aber das Problem von Collatz ist eine bis heute ungelöste (d.h. unbewiesene bzw. unwiderlegte) Vermutung.

Deswegen lautet die Aufgabe in dieser Woche: *Beweist die Collatz'sche Vermutung!*

Hallo? Seid Ihr noch da? Keine Panik: das war nur'n Witz! Natürlich braucht Ihr diese Vermutung *nicht* zu beweisen, aber anhand eines kleines Java-Programms sollt Ihr sie zumindest nachvollziehen.

Was ist die Collatz'sche Vermutung? Sie ist ein einfacher Algorithmus, der wie folgt funktioniert:

Schritt 1: Man nehme eine beliebige natürliche Zahl.

Schritt 2: Ist diese Zahl gerade, so wird sie halbiert. Ist sie ungerade, so wird sie verdreifacht und zu diesem Ergebnis eins addiert.

Schritt 3: Wenn das Ergebnis der Berechnung aus Schritt 2 „1“ ist, endet der Algorithmus, ansonsten geht es unter Verwendung dieses Ergebnisses erneut in Schritt 2.

Die Collatz'sche Vermutung besagt nun, dass dieser Algorithmus stets endet, d.h. egal, mit welcher natürlichen Zahl man beginnt: irgendwann erhält man die 1 und damit das Ende des Algorithmus.

Schreibt also ein Java-Programm, das eine natürliche Zahl als Parameter übergeben bekommt und den vorgestellten Algorithmus darauf anwendet. Jedes erhaltene Zwischenergebnis soll dabei ausgegeben werden.

Beispiel: Angenommen die Eingabe ist 6. Dann liefert der Algorithmus die folgende Ausgabe:

```
6, 3, 10, 5, 16, 8, 4, 2, 1
```

(Übrigens: wenn Ihr eine Zahl findet, für die der Algorithmus *nicht* funktioniert, habt Ihr den Beweis erbracht, dass die Vermutung falsch ist, denn in der Mathematik genügt ein Gegenbeweis, um eine solche Aussage als falsch gelten zu lassen.)

Aufgabe 2 – Klausurvorbereitung (30 Punkte)

Auch das letzte Übungsblatt in diesem Semester soll das Nasensymbol tragen, und deshalb geben wir Euch mit dieser Aufgabe typische und *absolut klausurrelevante* Aufgaben. Wir empfehlen dringend, diese Aufgaben eigenständig zu bearbeiten! Jede von ihnen gibt maximal 5 Punkte. Zum Testen Eurer Methoden könnt Ihr das Hauptprogramm verwenden, das Ihr auf unserer Homepage findet.

Teilaufgabe a): Minimum-Bestimmung

Schreibt eine Methode, die ein Array von `int`-Zahlen übergeben bekommt und das Minimum der in diesem Array enthaltenen Zahlen zurückliefert.

Teilaufgabe b): Gerade oder ungerade

Schreibt eine Methode „`istGerade`“, die eine `int`-Zahl übergeben bekommt. Die Methode soll `true` zurückliefern, wenn die übergebene Zahl gerade ist, andernfalls `false`.

Teilaufgabe c): Berechnung des Absolut-Betrags

Schreibt eine Methode „`abs`“, die eine `int`-Zahl übergeben bekommt. Die Methode soll den Absolut-Betrag der übergebenen Zahl zurückliefern. Die entsprechende Logik dazu soll von Euch geschrieben werden, d.h. die Verwendung von `Math.abs` und ähnlichen Methoden ist nicht zulässig.

Teilaufgabe d): Sind Ziffern enthalten?

In der Klasse `Character` ist eine Methode namens „`isDigit`“ enthalten, die in der Dokumentation wie folgt erklärt wird:

```
static boolean isDigit(char ch)
```

Determines if the specified character is a digit.

Schreibt eine Methode „`enthaeltZiffer`“, die einen String übergeben bekommt und mit Hilfe der Methode „`isDigit`“ der Klasse `Character` ermittelt, ob der String (mindestens eine) Ziffer(n) enthält. Ist dies der Fall, soll sie `true` zurückgeben, andernfalls `false`.

Teilaufgabe e): Überladen von Methoden

Macht Verwendung von dem Prinzip des Überladens von Methoden und spendiert Eurer Methode aus Aufgabe d) einen weiteren Parameter, diesmal vom Typ `int`. Die Methode soll zurückliefern, ob mindestens so viele Ziffern im String enthalten sind, wie dies durch den zweiten Parameter angeben wird.

Beispiel: Der Aufruf mit dem String „`abc123`“ und der „2“ als zweitem Parameter würde `true` zurückliefern, da der String mehr als zwei Ziffern enthält. Hingegen würde der Aufruf mit dem String „`abc123`“ und dem int-Wert 4 auf den Rückgabewert `false` führen, da nicht mindestens 4 Ziffern im String enthalten sind.

Teilaufgabe f:) Vereinfachung

Angenommen, Ihr habt die Methode aus Teilaufgabe e) geschrieben. Wie kann man nun jene Methode aus Teilaufgabe d) drastisch vereinfachen?

Aufgabe 3 – Das Sieb des Eratosthenes (25 Punkte)

In der letzten Aufgabe in diesem Semester beschäftigen wir uns mit Primzahlen. Wir möchten alle Primzahlen im Bereich von 1 bis n (wobei n eine natürliche Zahl ist, die der Benutzer eingibt) ermitteln und ausgeben lassen.

Eine einfache Methode, dies zu erreichen ist das sogenannte „Sieb des Eratosthenes“. Auf der entsprechenden Wikipedia-Seite[1] ist die Funktionsweise dieser Methode sehr anschaulich beschrieben, mit einer Animation versehen und sogar mit Pseudo-Code erklärt.

Schreibt ein Java-Programm, das zunächst die gewünschte Obergrenze vom Benutzer entgegennimmt und dann die Berechnungen durchführt.

Ihr könnt bzw. solltet Euch dabei am Pseudo-Code des Wikipedia-Eintrags orientieren. Denn dies ist die einfachste Möglichkeit, das Gewünschte zu erreichen. Andere Wege führen zwar auch zum Ziel, sind aber wesentlich komplizierter.

1 http://de.wikipedia.org/wiki/Sieb_des_Eratosthenes

DIE MAGISCHE ZAHL

Die magische Zahl dieses Übungsblattes ist die 155. Warum? Weil Ihr die Vorleistungen genau dann bestanden habt, wenn Ihr mindestens 155 Übungspunkte erreicht habt! A propos Vorleistungen: sofern

155

noch nicht geschehen, meldet Euch bitte umgehend zu diesen im Hochschulportal an. Ab der 29. KW (d.h. ab 19.07.) läuft dann die Anmeldung zur Klausur. Bitte denkt daran, Euch ***rechtzeitig*** *– ebenfalls über das Hochschulportal – dazu anzumelden!*