

Übungszettel, 2015-04-20

Abgabe möglich bis Donnerstag 2015-05-07T2359

Speichern Sie Ihre Lösungen in Ihrem Ordner im Repository ab, bevor Sie sie vorführen und erklären.

22. int-Arrays [1+1+1 Punkte]

- 22.1. Schreiben Sie ein Programm, das ein Feld mit int-Werten erzeugt und initialisiert. Lesen Sie die Feldgröße vom Benutzer ein, erzeugen Sie das Feld, belegen Sie die Komponenten mit zufälligen Werten im Intervall [-100;+100] (schlagen Sie dazu die Verwendung von Math.Random() nach), und geben Sie alle Komponenten in aufsteigender Indexreihenfolge aus. Testen Sie Ihr Programm mit den Feldgrößen 10, 100, 1, 0, -1.
- 22.2. Erweitern Sie Ihr Programm, sodass am Ende zusätzlich die Summe aller Feldkomponenten ausgegeben wird sowie der Durchschnitt. Bedenken Sie, dass der Durchschnitt von int-Werten nicht unbedingt ganzzahlig ist.
- 22.3. Erweitern Sie Ihr Programm, sodass zusätzlich der kleinste Index der größten Komponente und der kleinste Index der kleinsten Komponente ausgegeben werden sowie der Wert der größten und der kleinsten Komponente.

23. Primzahlen [1+1 Punkte]

- 23.1. Schreiben Sie ein Programm, das mithilfe eines Feldes für die ersten 100 Ganzzahlen ermittelt, ob die Zahlen Primzahlen sind oder nicht. Geben Sie die Primzahlen im Intervall [1..100] aus. Geben Sie anschließend zu jeder Nicht-Primzahl den kleinsten und den größten ihrer Primfaktoren aus.
- 23.2. Erweitern Sie Ihr Programm, sodass der Benutzer eine Zahl eingeben kann und anschließend durch Suche in Ihrem angelegten Feld ermittelt wird, ob die eingegebene Zahl eine Primzahl ist. Falls die Zahl keine Primzahl ist, geben Sie die Primfaktorzerlegung der Zahl aus. Testen Sie Ihr Programm mit den Eingaben 10, 100, 17, 97, 96, 101, 2, 1, 0, -1.

24. Lookup-Tabellen [1+2+2 Punkte]

- 24.1. Schreiben Sie ein Programm, das eine Zahl im Intervall [1..365] einliest, als Tag im Jahr interpretiert und Tag und Monat im Jahr ausgibt. Verwenden Sie zunächst if-/switch-Anweisungen ohne Felder, anschließend Felder. Welcher Programmcode ist übersichtlicher und/oder kürzer und warum?
- 24.2. Schreiben Sie ein Programm, das eine Liste von Postleitzahlen und Orten einliest, eine Postleitzahl beim Benutzer abfragt und den zugehörigen Ort ausgibt. Es reicht, wenn das Programm einen passenden Ort ausgibt, falls eine Postleitzahl mehreren Orten zugeordnet ist. Sie können das Beispielprogramm `ExercisePostcode` und die CSV-kodierte Datei `OpenGeoDB-plz-ort-de.csv` heranziehen (Encoding UTF-8).
- 24.3. Schreiben Sie ein Programm, das die Adressen und (sofern bekannt) Öffnungszeiten von Bäckereien in Konstanz speichert. Der Benutzer kann eine Postleitzahl angeben und bekommt alle Bäckereien zur entsprechenden Postleitzahl ausgegeben.
<http://www.meisterbaeckerei.de/konstanz.html>
http://www.sternenbaeck.de/fileadmin/user_upload/downloads/Filialliste_Sternenbaeck.pdf
<http://www.paradiesbaeckerei-menge.de/html/filialen.html>
<http://baeckerei-konstanz.de/unsere-filialen/>
<http://www.reginbrot.de/ueber-uns/>
http://www.holsteins-backhaus.de/index.php?option=com_content&view=article&id=4&Itemid=3
<http://www.k-und-u.com/filialsuche.php>
Tipp: Wenn Sie Arbeit sparen wollen, schließen Sie sich zusammen, tippen die Informationen in eine Datei, die Sie dann in Ihren jeweils eigenen Programmen einlesen.

25. Schachbrett [1+1+1+1 Punkte]

- 25.1. Schreiben Sie ein Programm, das den Zustand einer Schachpartie speichern kann. Wählen Sie eine geeignete Darstellung der unterschiedlichen Figuren, sowohl was Datentyp als auch Ausgabe auf der Konsole angeht. Der Zustand der Schachpartie darf im Programm festgelegt sein, muss also nicht vom Benutzer oder aus einer Datei eingelesen werden.
- 25.2. Erweitern Sie das Schachprogramm, sodass die Positionen der Figuren zeilenweise ausgegeben werden, also z.B. "Weißer König auf d1". Der Benutzer soll wählen können, ob die Ausgabe auf Deutsch oder auf Englisch erfolgt.
<http://de.wikipedia.org/wiki/Schach>
<http://en.wikipedia.org/wiki/Chess>
- 25.3. Schreiben Sie ein Programm, das die Anzahl der Reiskörner auf den Feldern eines Schachbretts speichert. Dabei sollen auf a1 ein Reiskorn gelegt werden, auf b1 zwei, auf c1 vier, also auf das nächste Feld jeweils die doppelte Menge des vorhergehenden. Wählen Sie eine geeignete Darstellung des Schachbretts und der Menge der Reiskörner.
- 25.4. Schreiben Sie ein Programm, das eine Lösung für das Acht-Damen-Problem berechnet und ausgibt. Beim Acht-Damen-Problem geht es darum, auf einem Schachbrett acht Dame-Figuren so zu platzieren, dass keine Figur eine andere schlagen kann. Eine Dame kann eine Figur schlagen, wenn die andere Figur in derselben Zeile oder in derselben Spalte steht oder über eine Diagonale erreicht werden kann.
Falls die Berechnung lange dauert, woran könnte es liegen? In welchen Situationen ist Ihr Programmcode ggf. nicht effizient?

26. Buchstaben [1 Punkt]

- 26.1. Schreiben Sie ein Programm, das eine Textdatei einliest und den Benutzer nach einem Buchstaben fragt. Geben Sie anschließend aus, wie häufig der eingegebene Buchstabe in der Textdatei vorkommt.
Tipp: Sie können die Datei `faust.txt` (Encoding ISO-8859-1) verwenden. Schlagen Sie die Benutzung von `java.util.Scanner.nextLine()` und von `String.charAt()` nach.

27. Freiwillige *-Aufgabe mit 5+5 Punkten

- 27.1. Erweitern Sie die Bäckerei-Aufgabe, sodass Sie den Standort der Bäckereien auf einer Karte anzeigen.
- 27.2. Zeigen Sie die Standorte abhängig von Ihrer aktuellen Position an. Die Positionsbestimmung soll dabei automatisch erfolgen. Auf Wunsch sollen nur die Filialen angezeigt werden, die am gleichen Tag noch geöffnet haben.

28. Freiwillige *-Aufgabe Vier gewinnt mit 4 Punkten

Schreiben Sie ein Programm, das ein Vier-Gewinnt-Spielfeld anzeigt (gerne auf der Konsole mit 6 Zeilen und 7 Spalten, x und o und Leerzeichen). Die Spieler sollen ihre Namen eingeben und dann abwechselnd die Spalte angeben, in die sie einen Spielstein stecken. Nach jedem Zug soll der Zustand des Spielfelds ausgegeben werden. Ihr Programm soll die Spielsteine an der richtigen Position speichern, feststellen, wann eine Spalte voll ist und ungültige Züge zurückweisen. Hat ein Spieler am Ende seines Zuges vier Spielsteine in einer waagerechten, senkrechten oder diagonalen Reihe angeordnet, soll das Programm das Spiel beenden und den Gewinner beglückwünschen.

```
xoxo x
oxooxxo
xxxooxx
oxooooo
-----
1234567
Spieler <Spielernamen> (''x'') gewinnt!
```