

Bachelor-Studiengang Informatik
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2010/2011

Übungsblatt 4: Arrays und Ausdrücke

Ausgabe am: 21.10.2010
Abgabe am: 5.11.2010

Aufgabe 1: Fibonacci-Folge **15 Punkte**

Die Fibonacci-Folge ist eine unendliche Folge von Zahlen (den Fibonacci-Zahlen), bei der sich die jeweils folgende Zahl durch Addition der beiden vorherigen Zahlen ergibt.

Zahl	0	1	2	3	4	5	6	usw.
Fibonacci	0	1	1	2	3	5	8	usw.

[Quelle: Wikipedia]

Schreiben Sie unter Verwendung von while-Schleifen ein Programm "Fibonacci", das die Fibonacci-Folge berechnet. Der Anwender soll zu Beginn eine ganze Zahl eingeben; diese bestimmt dann, wieviele Zahlen aus der Folge berechnet und in ein passendes Array gespeichert werden. Sind alle Zahlen berechnet soll der Inhalt des Arrays auf der Konsole ausgegeben werden. Bei einer ungünstigen Eingabe sollte ein Array der Länge eins mit dem Inhalt -1 erzeugt und dann auch ausgegeben werden.

Beispiele:

Eingabe: 0
Ausgabe: 0

Eingabe: 4
Ausgabe: 0, 1, 1, 2, 3

Aufgabe 2: Schachbrett-Legende **10 Punkte**

Einer alten Legende nach lebte einst in Indien ein König namens Sher Khan. Während seiner Regentschaft erfand jemand das Spiel, das heute Schach heißt. Der König war von diesem Spiel so begeistert, dass er seinen Erfinder zu sich an den Hof rufen ließ, weil er ihn für diese vortreffliche Idee belohnen wollte. Da er reich und mächtig war versprach er dem weisen Mann, ihm jeglichen Wunsch zu erfüllen. Dieser dachte eine geraume Weile nach und bat dann um ein einziges Reiskorn, das er auf das erste Feld des Schachbrettes legte. Der König lachte und fragte ihn, ob das wirklich alles sei, er könne sich doch viel mehr wünschen? Da antwortete der Mann, der König solle ihm auf jedes folgende Feld doppelt so viele Körner legen, wie auf dem Feld davor wären. Also auf dem zweiten Feld zwei Reiskörner, auf dem dritten vier, auf dem vierten acht, auf dem fünften Feld sechzehn Reiskörner usw. Leicht verärgert willigte der König ein, hielt es aber trotzdem für einen dummen Wunsch, da er der Ansicht war, dass der Erfinder sich eine viel größere Belohnung verdient hätte.

Schreiben sie ein Programm "SchachLegende", das die Summe der Reiskörner auf einem beliebigen Schachbrett berechnet. Zu Beginn soll der Anwender die Anzahl der Felder des imaginä-

ren Schachbretts angeben. Dann wird ein int-Array der entsprechenden Größe angelegt und nach der Regel der Legende gefüllt. Wenn das Füllen abgeschlossen ist, soll die Summe aller Reiskörner berechnet und ausgegeben werden.

Bei einer ungünstigen Eingabe soll als Summe 0 ausgegeben werden.

Aufgabe 3: Kleines Adressregister

8 + 8 = 16 Punkte

Ein kleines Adressregister bietet folgende Operationen an:

- Namen löschen (über die Position)
- Namen an einer gewünschten Position hinzufügen

Schreiben sie ein Programm "AdressRegister", das die oben beschriebenen Operationen ermöglicht. Verwenden sie hierfür ein String-Array, das mit den nachfolgenden fünf Namen initialisiert ist:

Position	1	2	3	4	5
Name	Andreas	Bernd	Carsten	Dennis	Emil

Zu Beginn soll der Benutzer eine natürliche Zahl eingeben, um die Operation zu bestimmen.

1 = Löschen
2 = Einfügen

Bei einer ungültigen Eingabe soll der aktuelle Inhalt des Registers mit der Anzahl der enthaltenen Namen ausgegeben werden.

Beispiel:

Eingabe: 3
Ausgabe: Andreas, Bernd, Carsten, Dennis, Emil; 5

Wählt der Benutzer die Operation "Löschen" aus, wird er aufgefordert, die Position (ganze Zahl), die gelöscht werden soll, einzugeben. Ist die Zahl größer oder gleich der letzten Position, soll der Name gelöscht werden. Ist die Zahl kleiner oder gleich eins, soll der erste Name gelöscht werden.

Nun wird ein neues String-Array angelegt mit passender Größe und allen Namen, außer dem zu löschenden. Dann wird das neue Array mit der Anzahl der enthaltenen Namen ausgegeben.

Beispiel:

Eingabe: 1
Eingabe: 3
Ausgabe: Andreas, Bernd, Dennis, Emil; 4

Wählt der Benutzer die Operation "Einfügen" aus, wird er aufgefordert, die Position (ganze Zahl), an welcher der neue Name eingefügt werden soll, einzugeben. Danach wird er aufgefordert, den neuen Namen (als String) einzugeben.

Nun wird ein neues Array mit einer zusätzlichen Position erzeugt, in dem der neue Name an seiner gewünschten Position steht. Das neue Array wird dann mit der Anzahl der enthaltenen Namen ausgegeben. Beispiel:

Eingabe: 2
Eingabe: 4

Eingabe: Jack

Ausgabe: Andreas, Bernd, Carsten, Jack, Dennis, Emil; 6

Aufgabe 4

15 Punkte

Das Osterdatum ist festgelegt auf den ersten Sonntag nach dem ersten Vollmond nach der Tag- und-Nacht-Gleiche im Frühling. Das Datum ergibt sich für das Jahr y (4-stellige Angabe) mit der folgenden Berechnung (gültig für Jahre ab 1582):

$$g = y \bmod 19$$

$$c = \frac{y}{100}$$

$$h = \left(c - \frac{c}{4} - \frac{8c + 13}{25} + 19g + 15 \right) \bmod 30$$

$$i = h - \frac{h}{28} \cdot \left(1 - \frac{29}{h + 1} \cdot \frac{21 - g}{11} \right)$$

$$j = \left(y + \frac{y}{4} + i + 2 - c + \frac{c}{4} \right) \bmod 7$$

$$l = i - j$$

$$m = 3 + \frac{l + 40}{44}$$

$$d = l + 28 - 31 \cdot \frac{m}{4}$$

Die Divisionen sind ganzzahlig und ignorieren den Divisionsrest.

Erstellen Sie ein Programm (Klasse "Easter"), das eine vierstellige Jahreszahl von der Konsole abfragt. Der Zahlenbereich ist zu prüfen, die Jahreszahl bei Bedarf wiederholt zu erfragen.

Dann berechnet das Programm anhand obiger Formel das Datum des Ostersonntags und gibt es aus ("Der Ostersonntag im Jahr ... fällt auf den x.y.zzzz.").

Hinweise

- Die Aufgaben sind in Eclipse zu bearbeiten. Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung04* an.
- Von allen Aufgaben sind Listings abzugeben, *keine* Testläufe.
- In den Laborstunden sollen Ihre Programme automatisch getestet werden. Damit Sie vorab prüfen können, ob Ihr jeweiliges Programm äußerlich korrekt ist (was nicht heißt, dass es korrekt funktioniert!), finden Sie im Wiki für jedes Ihrer Programme ein JUnit-Testprogramm.