

Übungen zur Einführung in die Programmiersprache Java

Universität Regensburg
NWF II - Physik
Dominik Köppl

Wintersemester 2011/12
Blatt 5

18 Norm des \mathbb{R}^n

Der n -dimensionale Raum \mathbb{R}^n hat die Standardnorm $|x| = \sqrt{\sum_{i=1}^n x_i^2}$, wobei $x = (x_1, x_2, \dots, x_n)$ ein n -Tupel ist. Diesen Tupel fassen wir in Java als `double-Array` auf. Sie können also in Java eine Funktion `double getNorm(double [] array)` schreiben, welche die Norm dieses Arrays berechnet. Die Dimension des Arrays können Sie mit `array.length` abfangen, s.d. Sie $n \in \mathbb{N}$ abstrakt halten können/sollen.

19 Palindrom

Palindrome sind Wörter, die von vorn- und rückwärts-gelesen den selben Ausdruck ergibt. Bekannte Palindrome sind beispielsweise "Lagerregal" oder "Rentner". Schreiben Sie ein Prädikat `boolean istPalindrom(String str)`; welche einen String daraufhin prüft, ob es sich bei diesem um ein Palindrom handelt. Strings können mit der Methode `char charAt(int index)`; ähnlich einem gewöhnlichen Array durchwandert werden. Die String-Methode `String toLowerCase()` kopiert den String und konvertiert alle Zeichen der Kopie in Kleinbuchstaben, woraufhin die Kopie schließlich zurückgegeben wird.

20 Pascalsches Dreieck

Das Pascalsche Dreieck ist eine geometrische Darstellung der Binomialkoeffizienten $\binom{n}{k}$. Das Dreieck ist so angeordnet, dass jeder Eintrag die Summe seiner beiden oberen Einträge ist, wobei jeder Rand den Wert 1 erhält.

```
          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
```

- Erstellen Sie eine Funktion `int[][] getDreieck(int laenge)`, welches das Pascal'sche Dreieck bis zur Zeile `laenge` berechnen soll.
- Die Funktion soll ein Array erzeugen und in dieses das Pascal'sche Dreieck abspeichern. Hierbei ist auf den Speicherplatz zu achten:
 - Man kann das Dreieck in eine rechts-untere Dreiecksmatrix transformieren.
 - Man soll nur so viel Speicher allozieren wie man braucht - bitte kein Quadrat o.ä. erstellen!

- Das zurückgegebene Array soll nun auf dem Bildschirm ausgegeben werden. Hierzu kann man z.B. die `for-each`-Schleife verwenden.

21 Selection-Sort

Der Selection-Sort Algorithmus sortiert ein Array (mit sortierbaren Elementen) nach folgenden Schema: Der Algorithmus durchwandert jedes einzelne Element, beginnend mit dem ersten

- Finde das kleinste Element aus dem Array
- Vertausche dieses mit dem aktuellen Element
- Wandere zum nächsten Element

Anschließend ist das Array aufsteigend sortiert. Implementieren Sie diesen Sortieralgorithmus als Funktion `void sortiere(int[] array);`, welches das Array daraufhin sortiert. Warum werden die Werte in dem `array` verändert, während eine Funktion `void manipulate(int i) { i = 0}` die Variable `i` nicht verändern kann?

22 Die Drachenkurve

Verwenden Sie erneut die `Pencil`-Klasse, um damit nun interessantere (schwierigere ;-)) Objekte zu erschaffen: Wir wollen uns an die Drachenkurve heranwagen, die wie folgt definiert ist:

- Eine Drachenkurve ist durch ihre Ordnung und Drehwinkel gegeben.
- Eine Drachenkurve 0. Ordnung besteht nur aus einer Strecke, die immer konstant ist/bleibt.
- Eine Drachenkurve n .ter Ordnung erhält man rekursiv:
 - Zeichne eine Drachenkurve $(n - 1)$.ter Ordnung mit 90 Grad Rechtsdrehung
 - Drehe um den eigenen Drehwinkel
 - Zeichne eine Drachenkurve $(n - 1)$.ter Ordnung mit 90 Grad **Linksdrehung**

Also ist eine Drachenkurve 1. Ordnung eine Rechtsdrehung um 90 Grad gefolgt einer Strecke (die Linksdrehung am Schluss wird ja nicht gezeichnet). Schreiben Sie eine Funktion `static void drachenkurveZeichnen(Pencil p, int ordnung, int drehwinkel)` die sich rekursiv aufruft und somit die Drachenkurve der Ordnung `ordnung` mit Startwinkel `drehwinkel` zeichnet.