

Übung 2a [8]: Pascalsches Dreieck

Schreiben Sie ein Programm, mit dem ein Pascalsches Dreieck beliebiger Größe berechnet und ausgegeben wird. Die Größe (=Zeilenanzahl) ist als Konstante N festzulegen. Dies darf die EINZIGE Stelle sein, die verändert werden muß, um eine andere Größe zu berechnen!

Definition:

Das Pascalsche Dreieck ist eine Dreiecksmatrix, in dem das Element i,j ($0 \dots N-1$) nur dann existiert, wenn $j \leq i$ ist und den Wert $i!/(j!(i-j)!)$ besitzt. Eine einfachere Formel ist: Das erste und letzte Feld ist immer 1 und jeder innere Wert ist die Summe der beiden über ihm stehenden Werte.

Implementierungshinweis:

Zur formatierten Ausgabe können Sie die Klassen NumberFormat und FieldPosition verwenden.

Beispiel:

```

          1
        1  1
       1  2  1
      1  3  3  1
     1  4  6  4  1
    1  5 10 10  5  1
    
```

Übung 2b [8+8]: Primzahlensieb

Schreiben Sie zwei Programme, die alle Primzahlen kleiner als N berechnen, speichern und anschließend ausgeben. Der Grenzwert ist wie im ersten Beispiel eine Konstante, die als einzige für einen anderen Bereich verändert werden darf.

Speichern Sie die Primzahlen beim ersten Programm in einem Array (überlegen Sie sich die minimal notwendige Größe!) und beim Zweiten in einem Vektor (Verwenden Sie zum Durchlauf eine Enumeration und keinen Zugriff per `elementAt()`!).

- Berechnen Sie jeweils den Füllgrad des Arrays/Vektors.
- Überlegen Sie sich, mit welchen Werten sie Ihre Programme zumindest testen müssen!

Beispiel:

Ausgabe der Primzahlen bis 100

```
Array:  2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83 89 97
```

```
Vektor: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83 89 97
```

```
Anzahl der Primzahlen bis 100: 25
```

```
Fuellgrad Array:  50.0 %
```

```
Fuellgrad Vektor: 62.5 %
```

```
Fuellgrad Vektor nach trimToSize: 100.0 %
```