

Grundlagen der Programmierung

Rekursive Programmierung

Aufgabe 1: Quersumme

Schreiben Sie eine rekursive Klassenmethode, die die Quersumme einer Zahl berechnet. Z.B. sollte die Klassenmethode für die Zahl 1293404 das Ergebnis 23 liefern. Ihre Klassenmethode muss die folgende Signatur haben:

```
public static int quersumme(int zahl)
```

Sie können zur Umsetzung alle Ihnen bisher bekannten Datentypen und Operatoren verwenden. Falls dem Parameter `zahl` eine Zahl kleiner als 0 übergeben wird, dann soll die Klassenmethode `-1` liefern.

Aufgabe 2: Zählen von Ziffern

Schreiben Sie eine rekursive Klassenmethode, die das Vorkommen einer Ziffer in einer Zahl zählt. Z.B. sollte die Klassenmethode für die Zahl 1213414 und die Ziffer 1 das Ergebnis 3 liefern. Ihre Klassenmethode muss die folgende Signatur haben:

```
public static int zaehleZifferInZahl(int ziffer, int zahl)
```

Sie können zur Umsetzung alle Ihnen bisher bekannten Datentypen und Operatoren verwenden. Falls dem Parameter `zahl` eine Zahl kleiner 0 übergeben wird oder falls dem Parameter `ziffer` keine Ziffer übergeben wird, dann soll die Klassenmethode `-1` zurückgeben.

Aufgabe 3: Blumenbäume malen

In dieser Aufgabe sollen Sie Blumenbäume, wie Sie sie in der Abbildung 1 sehen, mit Hilfe einer rekursiven Klassenmethode malen. Zum Malen einer einzelnen Blume sollen Sie die Klassenmethode `maleBlume` vom letzten Praktikum verwenden.

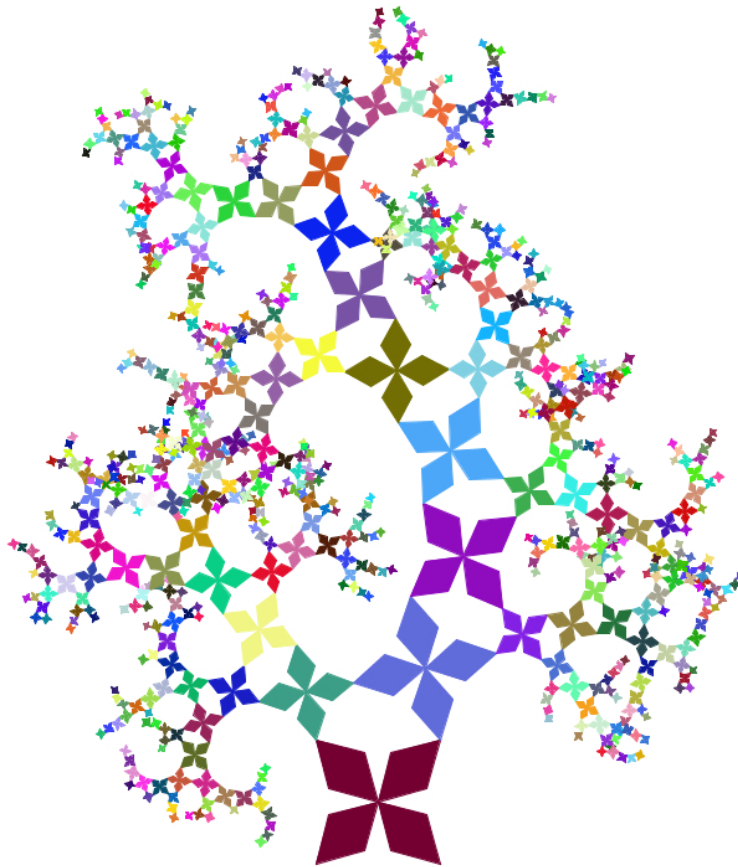


Figure 1: Rekursiv gemalter Blumenbaum

a) Schreiben Sie eine rekursive Klassenmethode

```
public static boolean maleBlumenBaum(int depth, double ax, double ay,  
                                     double bx, double by, double r),
```

die einen *Blumenbaum* entsprechend der Abbildung 2 malt. Der Parameter `depth` entspricht der Rekursionstiefe, die der *Blumenbaum* haben soll. Ein *Blumenbaum* der Tiefe 0 besteht also lediglich aus genau einer Blume. Die unterste Blume wird über den Werten (ax, ay, bx, by) gemalt.

Die Positionen, der jeweils auf eine Blume folgenden zwei *Blumenbäume*, berechnet sich wie folgt aus der Position der aktuellen Blume (siehe Abbildung 3).

```
sx = ax + (ay - by)  
sy = ay + (bx - ax)  
tx = ax + (ay - by) + r * (bx - ax) + Math.sqrt(r * (1.0 - r)) * (ay - by)  
ty = ay + (bx - ax) + r * (by - ay) + Math.sqrt(r * (1.0 - r)) * (bx - ax)  
ux = bx + (ay - by)  
uy = by + (bx - ax)
```

Dabei sind ax und ay die Koordinaten des Punkts A, bx und by die Koordinaten des Punkts B, sx und sy die Koordinaten des Punkts S, tx und ty die Koordinaten des Punkts T und ux und

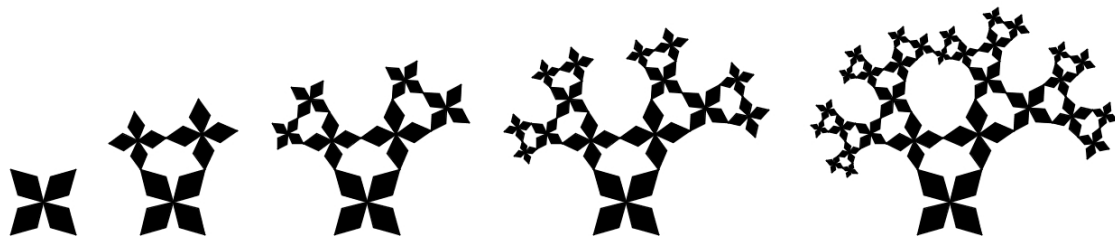


Figure 2: *Blumenbäume* mit Rekursionstiefe 0, 1, 2, 3 und 4. $r=0.4$.

uy die Koordinaten des Punkts U. Der Wert r wird Ihrer Methode als Parameter übergeben und kann zwischen 0.0 und 1.0 gewählt werden.

Tipp: Sie können sich die angegebenen Berechnungen der Punkte einfach in Ihr Java-Programm kopieren.

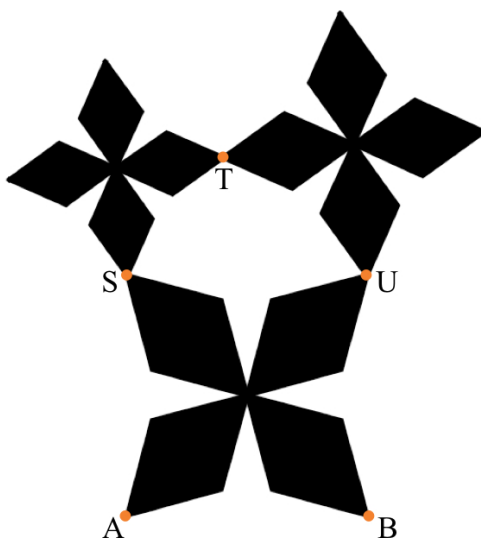


Figure 3: Punkte der rekursiven Blumenbäume

Der Rückgabewert der Klassenmethode `maleBlumenBaum` soll `true` sein, falls alle Blumen erfolgreich gemalt wurden, ansonsten `false`.