Algorithmen und Datenstrukturen

Rekursion

Aufgabe 1: Türme von Hanoi [2 Punkte]

Sie haben in den Unterlagen einen Lösungsansatz für die Problemstellung der *Türme von Hanoi*. Implementieren Sie einen `HanoiServer` mit der rekursiven Methode `moveDisk`. Geben Sie als Resultat einen Text aus, der einer Anleitung für die Lösung entspricht:

```
move A to C
move C to B
```
…

Die Anzahl der Scheiben soll als Parameter der `execute` Methode mitgegeben werden.

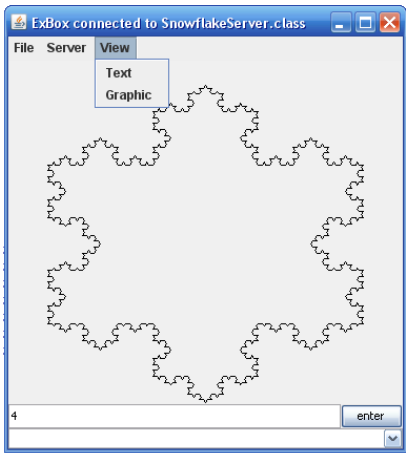
Aufgabe 2: Koch'sche Schneeflocke [4 Punkte]

Zum Zeichnen wird die ExBox verwendet. Wobei nicht direkt gezeichnet, sondern ein CommandExecutor `SnowflakeServer` entwickelt werden soll, der die Zeichnung mit Hilfe der Klasse `Turtle` erstellt, welche die fertige Graphik in `getTrace()` als XML-String retourniert.

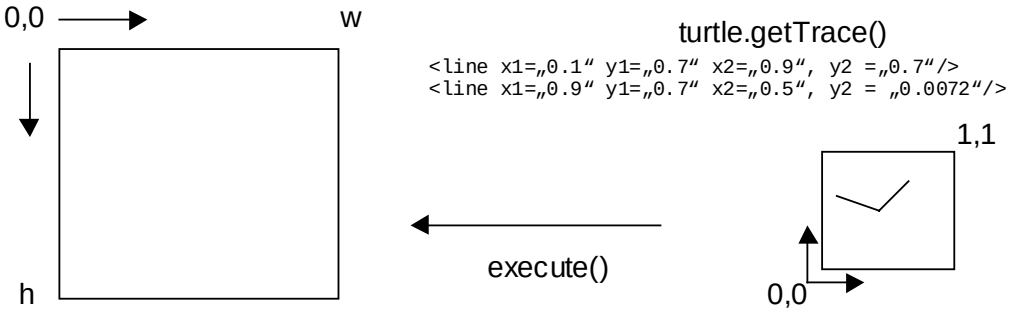
Hinweise:

- Es soll die in der Vorlesung besprochene Schneeflockenkurve mittels *Turtlegraphic* gezeichnet werden.
- Instanzieren Sie in Ihrer `execute` Methode eine `Turtle` Klasse, mit der Sie die Schneeflocke nach der Vorlage aus dem Skript zeichnen können.
- Mittels `getTrace()`kann der zurückgelegte Weg der `Turtle` als String abgefragt und als Resultat der `execute` Methode zurückgegeben werden. Die Klasse `Turtle` liefert die Spur des zurückgelegten Weges in der unten beschriebenen Form.

Hinweise zur Implementation:



Mittels dem `GraphicPanel` können Zeichnungen dargestellt werden. Dieses kann mittels `Views→Graphic` aktiviert werden. Dadurch kann zwischen graphischer und textueller Darstellung umgeschaltet werden. In der ExBox wird der Resultat-String der `execute` Methode einfach mittels `setFigure()`dem `GraphicPanel` übergeben, falls die graphische Ansicht aktiviert ist.



View Koordinaten-System

Modell Koordinaten-System

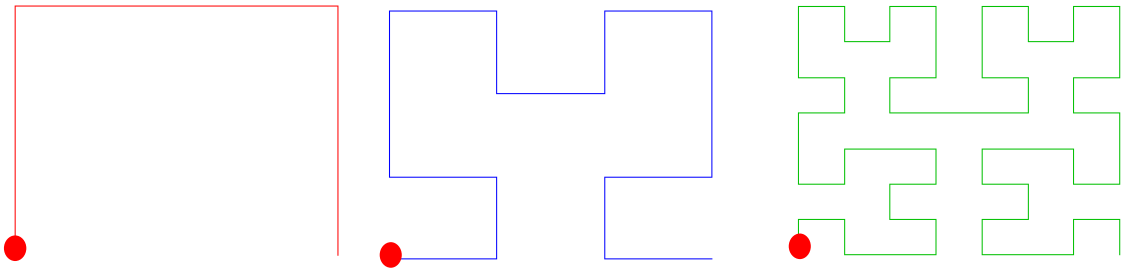
Die Graphik wird als XML-String erzeugt, wobei jeweils Gruppen von vier Koordinaten, denen das Wort "`<line`" vorangestellt ist, eine Linie definieren (siehe oben). Dies entspricht auch dem SVG-Format, das im Browser für Vektorgraphiken verwendet wird.

Dabei sind die Koordinaten der Modell-Zeichenfläche so normiert, dass die linke untere Ecke bei 0.0,0.0 liegt und die rechte obere Ecke bei 1.0,1.0. Die Normierung hat den Vorteil, dass für die Berechnung keine Annahmen über das View Koordinaten-System (z.B. aktuelle Grösse des Fensters) gemacht werden muss. Das Einheitsquadrat wird immer in maximaler Grösse dargestellt.

Aufgabe 3: Hilbertkurve [4 Punkte]

In der Mathematik ist die Hilbert-Kurve eine stetige Kurve, die – durch Wiederholung ihres Konstruktionsverfahrens – jedem beliebigen Punkt einer quadratischen Fläche beliebig nahekommt und die Fläche vollständig ausfüllt. Die Hilbert-Kurve ist eine sogenannte raumfüllende oder FASS-Kurve. Sie wurde 1891 von dem deutschen Mathematiker David Hilbert entdeckt (ref: Wikipedia). Die Möglichkeit, mit einer stetigen eindimensionalen Kurve ein zweidimensionales Gebiet komplett abdecken zu können, war den Mathematikern des neunzehnten Jahrhunderts neu.

Hinweise:



n = 0 **n = 1** **n = 2**

Der Code für die Hilbert-Kurve Stufe 0 sieht wiefolgt aus:

```
private void hilbert(int depth, double dist, double angle) {
      turtle.turn(-angle);
      // draw recursive
      turtle.move(dist);
      turtle.turn(angle);
      // draw recursive
      turtle.move(dist);
      // draw recursive
      turtle.turn(angle);
      turtle.move(dist);
      // draw recursive
      turtle.turn(-angle);
}

...
int depth = Integer.parseInt(command);
double dist = 0.8 / (Math.pow(2,depth+1)-1);
turtle = new Turtle(0.1, 0.1);
hilbert(depth, dist, -90);
```

Hinweis:

- Überlegen Sie sich, wie Sie mittels rekursiven Aufrufen von der roten Kurve (=Stufe 0) zur blauen und dann zur grünen etc. kommen. Der rote Punkt in der Zeichnung soll der Startpunkt sein. Dabei kann mit den *angle* Parameter jeweils die Ausrichtung der U-förmigen Kurve festgelegt werden (alternierend 90 und -90 Grad).