1.1: Erste Schritte

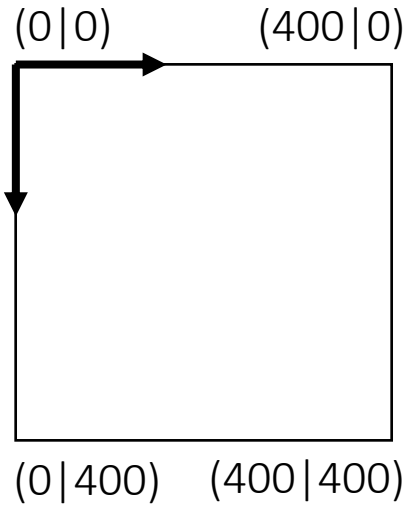
Einstieg

Voraussetzung: -

Mit den Befehlen `joe.move(Pixel)` und `joe.turn(Winkel)` kann man die Schildkröte alles zeichnen lassen, was man will.

➲Aufgabe:
Die Schildkröte soll ein Quadrat mit einer Seitenlänge von 100 Pixeln zeichnen.

Ein neu erstelltes Fenster hat die Maße 400 x 400 Pixel.
Es gleicht einem Koordinatensystem, das den Ursprung in der oberen linken Ecke hat. Die x-Koordinaten verlaufen nach rechts, die y-Koordinaten nach unten (siehe Skizze).



Der Befehl `joe.toStartingPoint(x, y)` setzt die Schildkröte auf den Startpunkt (x|y).
Mit `joe.setDirection(Winkel)` gibt man den Anfangswinkel an.

➲Aufgabe:
Die Schildkröte soll nun ein Quadrat mit einer Seitenlänge von 300 Pixeln zeichnen. Das Quadrat soll vollständig im Fenster zu sehen sein.

1.2: Stifte hoch!

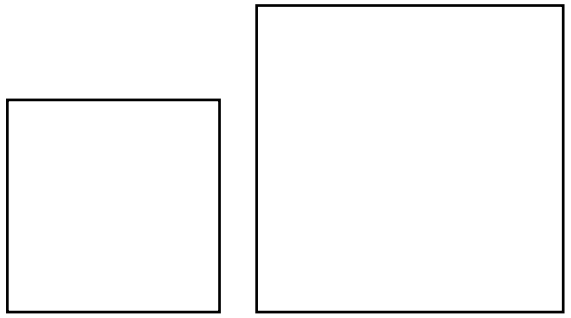
Einstieg

Voraussetzung: `move`, `turn`

Wenn die Schildkröte sich über den Bildschirm bewegt, zeichnet sie unentwegt ihren Weg nach. Man kann sich vorstellen, dass sie einen Stift in der Hand hält, den sie auf das Papier absetzen und vom Papier hochnehmen kann.

Die Befehle dazu lauten `joe.penDown()` und `joe.penUp()`.

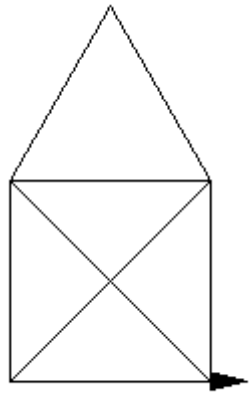
➲Aufgabe:
Die Schildkröte soll zwei Quadrate zeichnen, die sich nicht berühren.



1.3: Ist dies schon Wahnsinn, so hat es doch Methode.

Voraussetzung: -

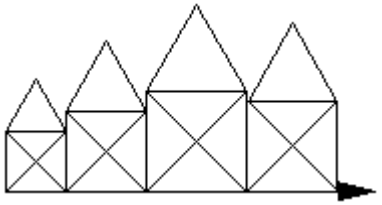
➲Aufgabe:
Die Schildkröte soll das Haus vom Nikolaus zeichnen. Die Hauswand und das Dach haben eine Länge von 100 Pixeln. Die Diagonale ist 141 Pixel lang.



Wir möchten dieses Haus vom Nikolaus mehr als einmal und in unterschiedlichen Größen zeichnen. Um nicht jedes Mal denselben Quelltext schreiben zu müssen, definieren wir eine Methode:

```
public void zeichneHausVomNikolaus(int pGroesse) {
    joe.move(pGroesse);
    ...
}
```

➲Aufgabe:
Entwickle die Methode `zeichneHausVomNikolaus(int pGroesse)`.
Die Diagonale hat die Länge `groesse*1.41`.
Zeichne eine Reihe Häuser mit unterschiedlichen Größen.



2.1: n-Ecke

while-Schleife

Voraussetzung: Methoden

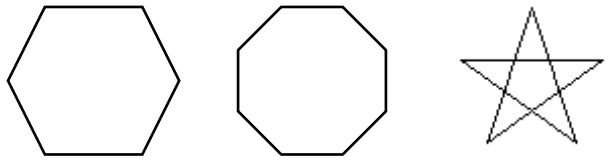
➲Aufgabe:

Zeichne auf Papier die Figur, die bei dem Aufruf von `zeichne(50)` entsteht.

```
public void zeichneMuster(int pLaenge) {
    int a = 1;
    while(a <= 4) {
        joe.move(pLaenge);
        joe.turn(90);
        a++;
    }
}
```

➲Aufgabe:

Die Schildkröte soll mithilfe von while-Schleifen die folgenden Figuren zeichnen.



➲Zusatz-Aufgabe:

Schreibe eine Methode `zeichneNEck(int pN, int pLaenge)`, die ein beliebiges n-Eck zeichnet.

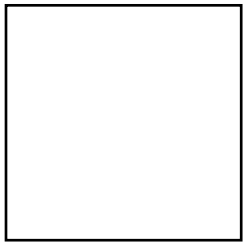
2.2: Ein 9-zackiger Stern

for-Schleife

Voraussetzung: -

Mit einer Zählschleife kann man eine Variable sehr komfortabel alle Werte von einem Startwert bis zu einem Endwert mit einer beliebigen Schrittweite durchzählen lassen: **`for(Startwert; Bedingung; Schrittweite)`**. Ein Quadrat lässt sich auf diese Weise mit wenigen Zeilen zeichnen:

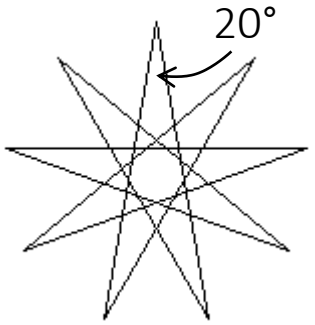
```
for(int i = 1; i <= 4; i++) {
    joe.move(100);
    joe.turn(90);
}
```



➲**Aufgabe:**

Die Schildkröte soll die nebenstehende Figur mit einer for-Schleife zeichnen.

Zeichne weitere Sterne mit unterschiedlicher Zackenanzahl.

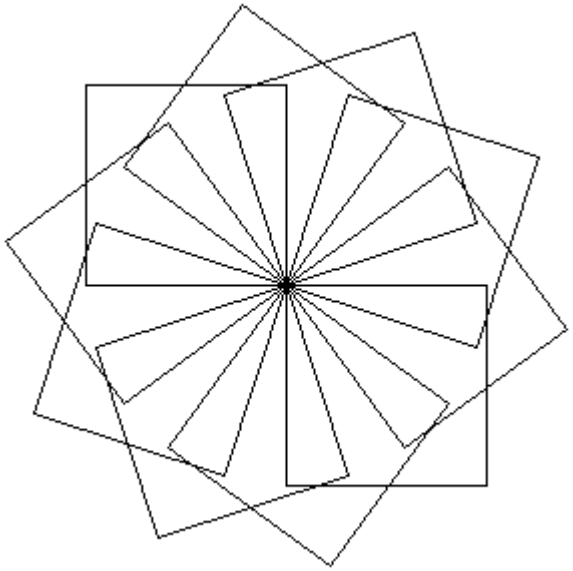


2.3: Flowers from outer space

verschachtelte for-Schleifen

Voraussetzung: for-Schleifen

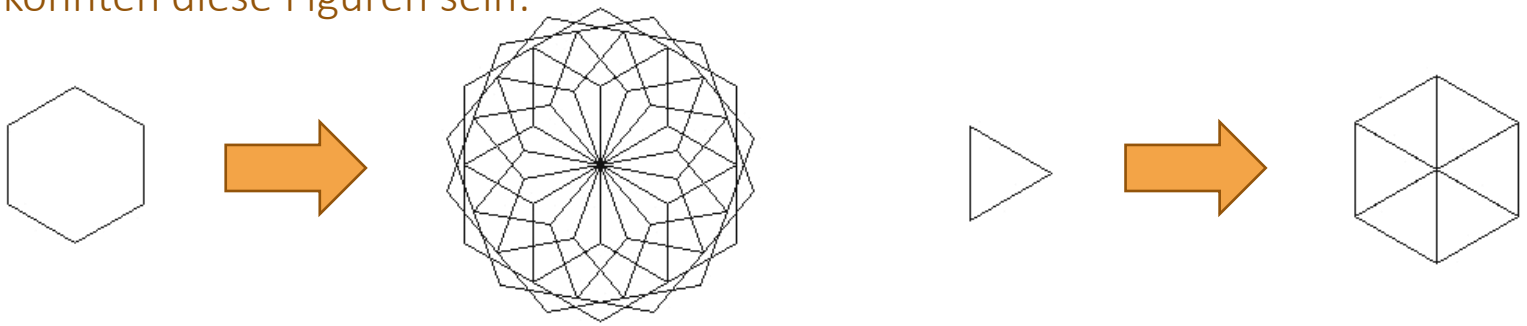
Schleifen können auch ineinander verschachtelt werden. Die nebenstehende Figur wurde erstellt, indem mit der inneren Schleife jeweils ein Quadrat gezeichnet wird. Mit der äußeren Schleife werden 10 Quadrate zeichnet, wobei nach jedem Quadrat eine Drehung um 36 Grad erfolgt.



➲**Aufgabe:**

Die Schildkröte soll die oben beschriebene Figur zeichnen.

Experimentiere mit den Parametern und erstelle weitere exotische außerirdische Blumen. Ein Anreiz könnten diese Figuren sein:



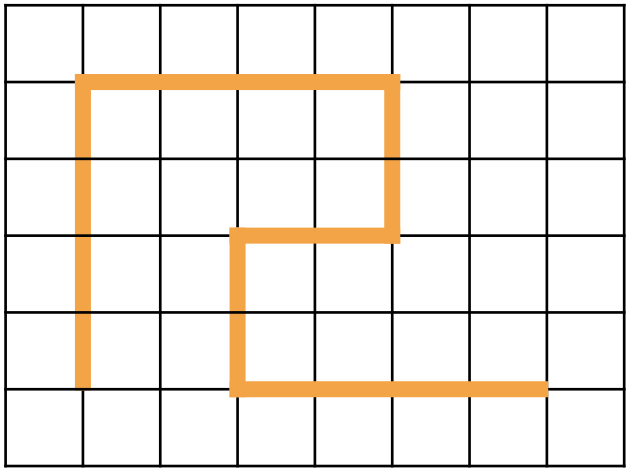
2.4: Mäander

Übung Schleifen

Voraussetzung: for-Schleife

Der Mäander ist ein seit der Jungsteinzeit verwendetes orthogonales Ornament. Der Name entstand in Anlehnung an die gleichnamigen Flussschlingen. Eine Sonderform des Mäanders ist der sogenannte Doppelmäander. Dieser besteht aus zwei entgegengesetzt verlaufenden Mäandern.

➲**Aufgabe:**
Die Schildkröte soll das nebenstehende Mäander zeichnen. Die längere Strecke ist 100 Pixel, die kürzere 50 Pixel lang. Schreibe nun eine Methode, die einen Parameter **pLaenge** annimmt und zeichne mit dieser ein Mäanderband.



➲**Aufgabe:**
Recherchiere im Internet nach weiteren Beispielen für Mäander und setze sie mit der Turtle um.

3.1: Jetzt wird‘s bunt!

Auswahl

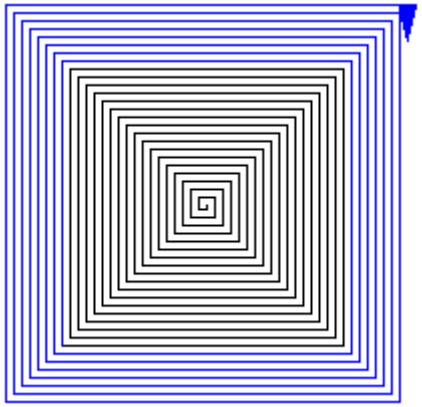
Voraussetzung: Schleifen

Die Schildkröte hat nicht nur einen schwarzen Stift dabei, sondern gleich einen ganzen Malkasten. Mit der Methode `joe.setColor(Farbe)` kann eine der folgenden Stiftfarben ausgewählt werden (Farbe ist ein Integer-Wert):

0 – schwarz, 1 – grün, 2 – blau, 3 – gelb, 4 – rot, 5 – grau, 6 – hellgrau, 7 – weiß.

➲Aufgabe:

Die Schildkröte soll eine quadratische Spirale zeichnen. Sie bewegt sich in einer Wiederholstruktur vorwärts und dreht sich um 90° nach rechts. Dabei wird die Schrittlänge um 2 vergrößert. Falls die Schrittlänge gleich 140 ist, wechselt die Stiftfarbe auf blau.



3.2: Farbexplosion

switch-case

Voraussetzung: Schleifen

➲Aufgabe:

Interpretiere die Methode `explodiereFarbe()`, implementiere anschließend.

```
public void explodiereFarben() {
     joe.hideTurtle();
     for (int i = 0; i < 120; i++) {
          if (i % 3 == 0) {
               joe.setColor(4);
          }
          if (i % 3 == 1) {
               joe.setColor(1);
          }
          if (i % 3 == 2) {
               joe.setColor(3);
          }
          joe.move(150);
          joe.move(-150);
          joe.turn(-3);
     }
}
```

➲Aufgabe:

Schreibe die Methode `explodiereFarbe()` so um, dass statt einseitiger Auswahl eine mehrseitige Auswahl verwendet wird.

4.1: Quadratschnecke

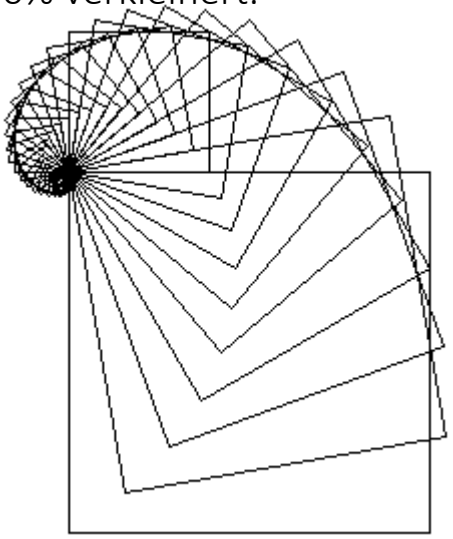
Variablen

Voraussetzung: Schleifen

Variablen sind Speicherbereiche, in welchen man Daten, die im Verlauf des Programms benötigt werden, speichern kann. Vor der ersten Verwendung müssen Variablen deklariert und initialisiert werden. Die Deklaration legt den Wertebereich der Variablen fest (`int` für ganze Zahlen, `double` für Dezimalzahlen usw.). Die Initialisierung legt den Startwert der Variablen fest (z. B. `i = 0`). In der Regel werden diese beide Schritte zusammen gemacht
(z. B. `int i = 0`, `double a = 15.5`).

In folgendem Beispiel zeichnet die Turtle Quadrate unterschiedlicher Größe. Die Quadratseite a hat zu Beginn die Länge 180 und wird nach jedem gezeichneten Quadrat auf 90% verkleinert.

```
double a = 180;
while (a > 5) {
    for (int i = 0; i < 4; i++) {
        joe.move(a);
        joe.turn(90);
    }
    joe.turn(-10);
    a = a * 0.9;
}
```



➲Aufgabe:

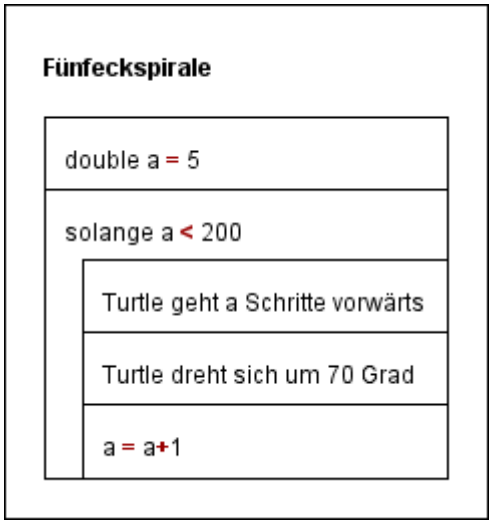
Passe das Programm so an, dass nicht ein Quadrat, sondern ein gleichseitiges Dreieck (ein Sechseck, ein Kreis, …) die sich drehende Figur ist.

4.2: Fünfeck-Spirale

Struktogramm

Voraussetzung: Variablen, Schleifen

Dieses Struktogramm erzeugt die nebenstehende Figur.



➲Aufgabe:

Setze das Struktogramm in Java-Code um.

4.3: Turtleception

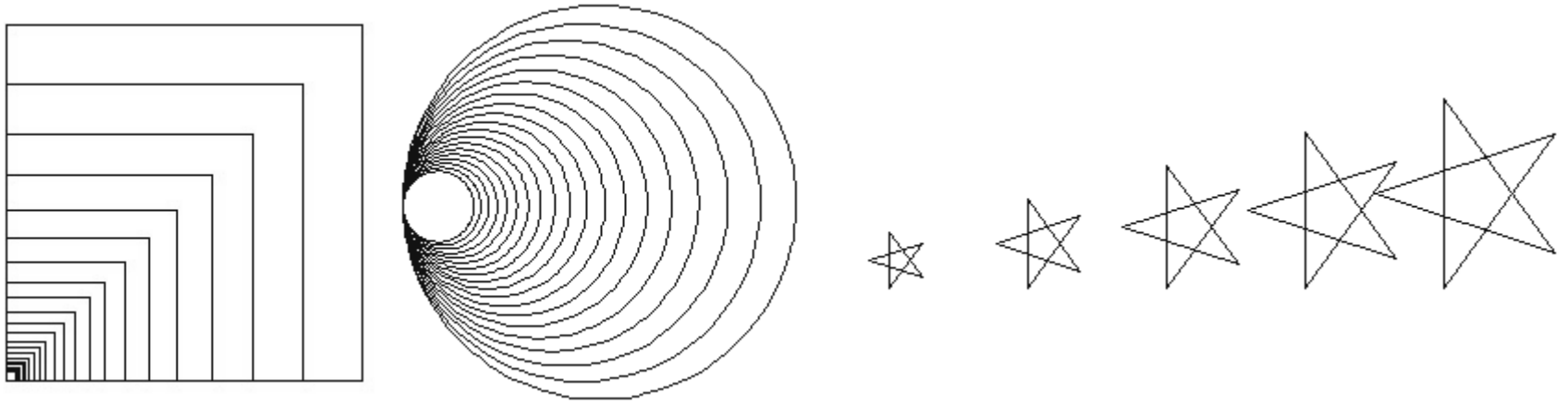
Übung Variablen

Voraussetzung: Variablen, Schleifen

Mit Schleifen und sich innerhalb dieser verändernden Variablen kann man interessante Bilder erzeugen.

➲Aufgabe:

Verwende Variablen, um solche Figuren zu zeichnen:



Denke dir weitere Figuren aus, die man mit Variablen und Schleifen erzeugen kann.

4.4: Ich seh‘ den Sternenhimmel

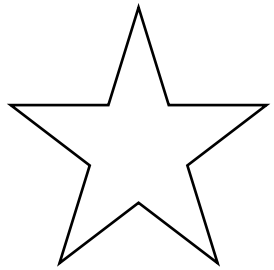
Voraussetzung: Variablen, Schleifen

Die Schildkröte kann nicht nur vorwärts laufen und sich drehen. Mit dem Befehl `joe.moveTo(x, y)` springt sie zur Koordinate (x|y).

➲Aufgabe:

Die Schildkröte soll zum Punkt (100|200) springen und einen Stern zeichnen. Der folgende Programmausschnitt ist fehlerhaft. Korrigiere ihn und erweitere ihn um den Sprungbefehl.

```
for(int i=1; i<5; i++) {
   joe.move(10);
   joe.turn(-72);
   joe.move(10);
   joe.turn(144);
}
```



Nun soll ein zufälliger Sternenhimmel gezeichnet werden. Die Zeile `int zufallszahl = (int)(Math.random() * 17) + 13;` generiert eine von 17 Zahlen im Bereich von 13 bis 29.

➲Aufgabe:

Erstelle zwei Zufallszahlen x und y jeweils aus dem Bereich von 0 bis 400 und zeichne einen Stern an der Position (x|y). Generiere auf diese Weise 30 Sterne mit zufälligen Koordinaten.

5.1: Kochkurve

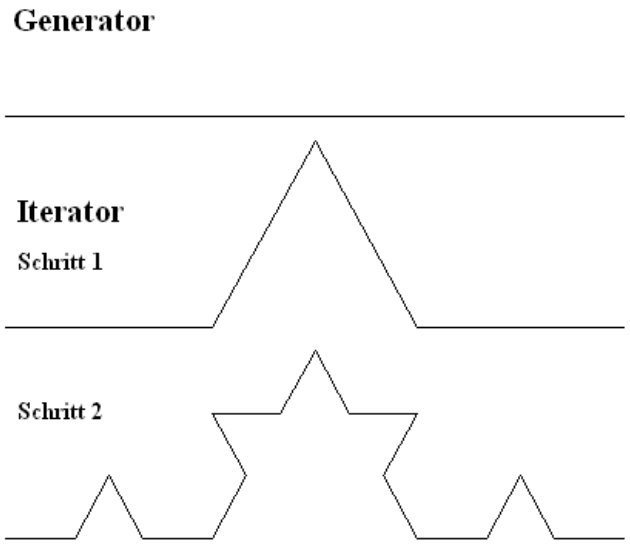
Voraussetzung: Auswahl

➲Aufgabe:

Informiere dich über das Prinzip der Rekursion.

Die Kochkurve ist ein einfaches Beispiel der grafischen Rekursion. Die Koch-Kurve wird durch eine Iteration folgendermaßen beschrieben:

Zunächst sei eine Strecke mit vorgegebener Länge gegeben (Generator). Dann entfernt man das mittlere Drittel der Strecke und errichtet darüber ein gleich-seitiges Dreieck mit der Länge der entfernten Strecke (Iterator). Im nächsten Schritt wird der Iterator auf jeden der vier entstandenen Streckenabschnitte angewandt. Diese Iteration wird nun beliebig oft wiederholt.



➲Aufgabe:

Die Schildkröte soll in einer Methode `zeichneKoch(int pLaenge)` die Figur des 1. Iterationsschrittes zeichnen.

5.2: Forts. Kochkurve

Voraussetzung: Auswahl

Die Methode sollte in etwa so aussehen wie im Kasten links.

```
public void zeichneKoch(int pLaenge) {
    joe.move(pLaenge);
    joe.turn(-60);
    joe.move(pLaenge);
    joe.turn(120);
    joe.move(pLaenge);
    joe.turn(-60);
    joe.move(pLaenge);
  }
```

Hier durch eine Koch-Kurve ersetzen.

Mit fortlaufender Iteration soll nun jeder Streckenabschnitt wieder durch eine Kochkurve der Länge **pLaenge/3** gezeichnet werden. Außerdem braucht die Methode einen Parameter für den Iterationsschritt, in dem sich die Rekursion gerade befindet.

Damit die Rekursion nicht ins Unendliche geht, ist eine Abbruchbedingung nötig (falls der Iterationsschritt 0 erreicht wurde, zeichne eine Linie).

```
public void zeichneKoch(int pI, int pLaenge)
{
  if (pI == 0) {
    joe.move(pLaenge);
  }
  else {
    zeichneKoch(pI-1, pLaenge/3);
    joe.turn(-60);
    zeichneKoch(pI-1, pLaenge/3);
    joe.turn(120);
    zeichneKoch(pI-1, pLaenge/3);
    joe.turn(-60);
    zeichneKoch(pI-1, pLaenge/3);
  }
}
```

➲**Aufgabe:**

Nutze die Methode, um eine Koch-Schneeflocke zu zeichnen.