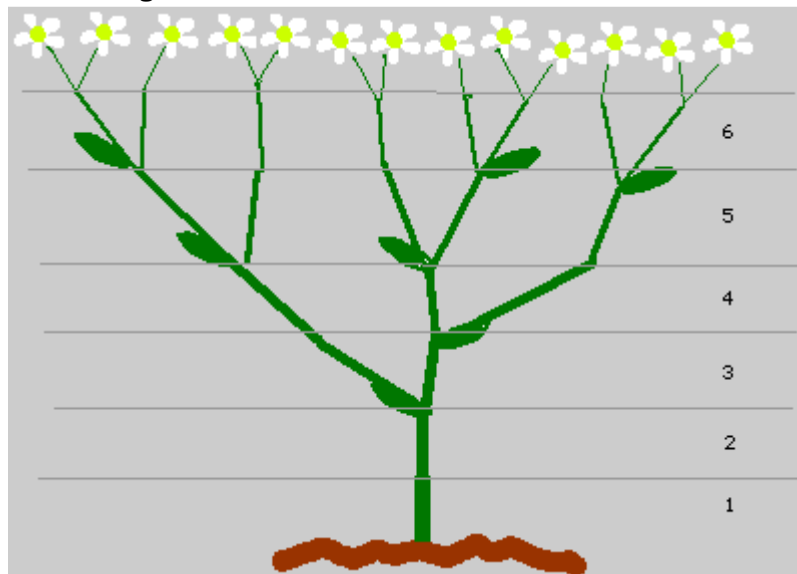


Übungsblatt 16 Java FX

1. Achilea Ptarmica – Sumpf-Schafgabe

Bei der Sumpf-Schafgabe wächst ein neuer Trieb zwei Monate lang, bevor er einen Seitentrieb bilden kann. Der alte Trieb bringt hingegen im weiteren Wachstumsverlauf pro Monat einen neuen Trieb hervor. Die Anzahl $A(n)$ der Trieb-Enden am Ende des n -ten Monats lässt sich aus der Anzahl der Trieb-Enden am Ende der beiden vorangegangenen Monate berechnen. Wie viele Trieb-Enden hat die Sumpf-Schafgabe am Ende des siebten Monats? Wie sieht der allgemeine Term für $A(n)$ aus?



a) Zeichnen Sie eine eigene acht Monate alte Sumpf-Schafgabe.

Es sollen die Funktionswerte $A(n)$ für $1 \leq n \leq 90$ berechnet und ausgegeben werden.

b) Schreiben Sie eine Methode zur iterativen Lösung des Problems.

c) Implementieren Sie alternativ eine rekursive Lösung.

$$A(n) = \begin{cases} A(n-1) + A(n-2), & \text{falls } n > 1 \\ 1, & \text{falls } n = 1 \\ 0, & \text{falls } n \leq 0 \end{cases}$$

d) Vergleichen Sie die Laufzeit.

n	A(n)
1	1
2	1
3	2
4	3
5	5
6	8
...	
88	1100087778366101931
89	1779979416004714189
90	2880067194370816120

2. Peano Axiome für natürliche Zahlen

Die Rekursion wurde aus der Mathematik übernommen, wo mit ihrer Hilfe unendliche Mengen durch endliche Aussagen charakterisiert werden können. Das Musterbeispiel ist die Menge N_0 der natürlichen Zahlen. Sie wird durch die Axiome von Peano gekennzeichnet:

1. 0 ist eine natürliche Zahl.
2. Zu jeder natürlichen Zahl n gibt es eine natürliche Zahl $n+1$ als Nachfolger von n .
3. 0 ist nicht Nachfolger einer natürlichen Zahl.
4. Natürliche Zahlen mit gleichem Nachfolger sind gleich.
5. Enthält eine Menge die Zahl 0 und mit jeder natürlichen Zahl n auch ihren Nachfolger, so enthält sie alle natürlichen Zahlen.

Schreiben sie eine Klasse mit Namen Peano.

- a) Legen sie eine statische Methode `int succ(int x)` an (successor=Nachfolger), die den Nachfolger einer Zahl x zurückgibt.
- b) Legen sie eine statische Methode `int pred(int x)` an (predecessor=Vorgänger), die den Vorgänger einer Zahl x zurückgibt, falls $x=0$ soll 0 zurückgegeben werden. Man gehe davon aus, dass x keine negative Zahl sei.

Die fünf Operatoren $+$, $-$, $*$, $/$ und potenzieren sollen mit Hilfe obiger Funktionen rekursiv nachgebildet werden.

- c) Legen sie eine statische Methode `int add(int x, int y)` an, die die Summe von x und y ohne Verwendung des $+$ Operators berechnet.
- d) Legen sie eine statische Methode `int sub(int x, int y)` an, die die Differenz von x und y ohne Verwendung des $-$ Operators berechnet.
- e) Legen sie eine statische Methode `int mul(int x, int y)` an, die das Produkt von x und y ohne Verwendung des $*$ Operators berechnet. Nutzen sie die Methoden `add` und `pred`.
- f) Legen sie eine statische Methode `int div(int x, int y)` an, die den Quotient von x und y ohne Verwendung des $/$ Operators berechnet. Nutzen sie die Methoden `sub` und `pred`.
- g) Legen sie eine statische Methode `int pot(int x, int y)` an, die den Wert von x^y ohne Verwendung des $*$ Operators berechnet. Nutzen sie die Methoden `mul` und `pred`.

Rechnen nach den Peano Axiomen mit natürlichen Zahlen

```
=====
0 Ende
1 Addition
2 Subtraktion
3 Multiplikation
4 Division
5 Potenzierung
Was wünschen Sie zu tun? 1
Geben Sie zwei positive ganze Zahlen mit Komma getrennt ein:4, 7
--> 4 + 7 = 11
```

Rechnen nach den Peano Axiomen mit natürlichen Zahlen

```
=====
...
2 Subtraktion
...
Was wünschen Sie zu tun? 2
Geben Sie zwei positive ganze Zahlen mit Komma getrennt ein:9,2
--> 9 - 2 = 7
```

Rechnen nach den Peano Axiomen mit natürlichen Zahlen

```
=====
...
5 Potenzierung
Was wünschen Sie zu tun? 4
Geben Sie zwei positive ganze Zahlen mit Komma getrennt ein:2,5
--> 2 ^ 5 = 32
```

3. Umwandeln einer Dezimalzahl in Dualzahl

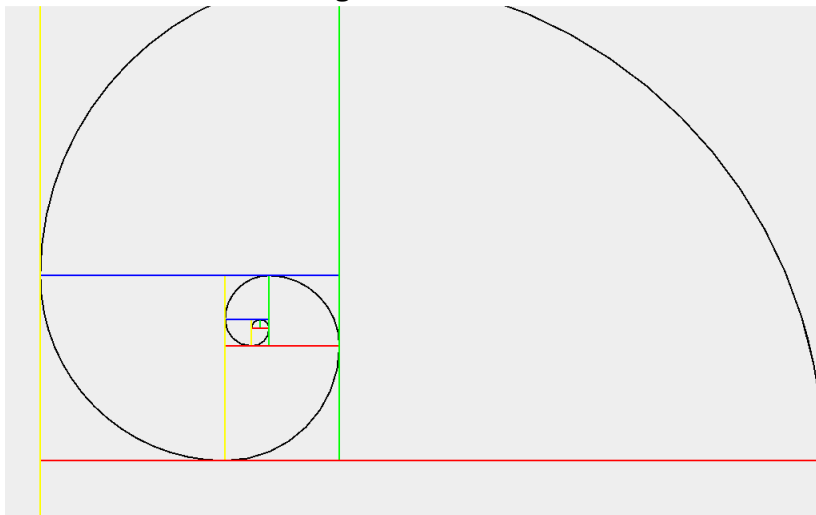
Formulieren Sie den Zigeuneralgorithmus in rekursiver Form.

Umwandlung eine Dezimalzahl in eine Dualzahl

```
=====
Geben Sie bitte eine Dezimalzahl ein: 12
1100
Geben Sie bitte eine Dezimalzahl ein: 12345
11000000111001
Geben Sie bitte eine Dezimalzahl ein: -1
11111111111111111111111111111111
```

4. Fibonacci-Spirale

Erstellen Sie ein Java Programm zum Zeichnen der Fibonacci-Spirale.



Erlauben Sie dem Benutzer per Mausklick den Wechsel zwischen linksdrehender- und rechtsdrehender Spirale.

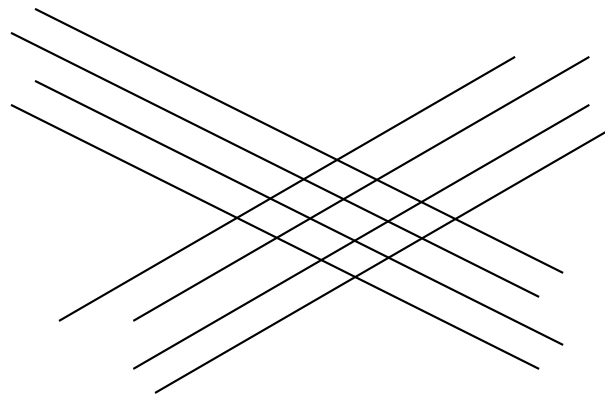
Drehen am Mausrad erlaubt eine Skalierung der Spirale zwischen $1 \leq x \leq 20$.

Wie lang ist die Spirale nach n -Umläufen? Geben Sie eine Wertetabelle aus.

Umlauf	Länge der Spirale
1	10.995574287564276
2	84.82300164692441
3	590.6194188748811
4	4057.366912111218
5	27818.802947537617
6	190682.10770228607
7	1306963.804950099

5. Gitterpunkte

Ein Paar aus jeweils n zueinander parallelen Geraden schneidet sich. Wie viele Schnittpunkte gibt es für $n=37$, wenn für $n=36$ die Anzahl A der Schnittpunkte $A(36)=1296$ beträgt? Wie könnte man allgemein die Anzahl der Schnittpunkte von n Geradenpaaren aus der Anzahl $n-1$ bestimmen? Finden Sie eine rekursive Funktion $A(n)$.

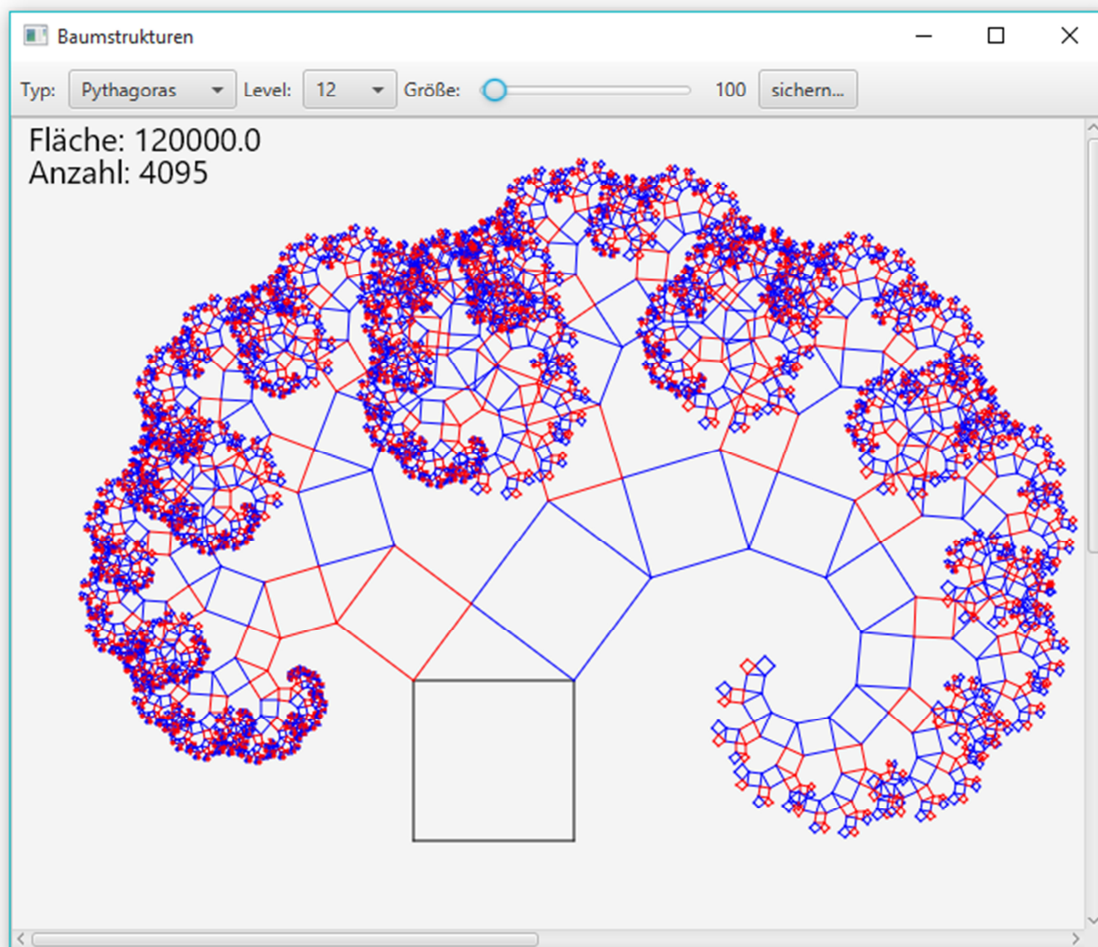
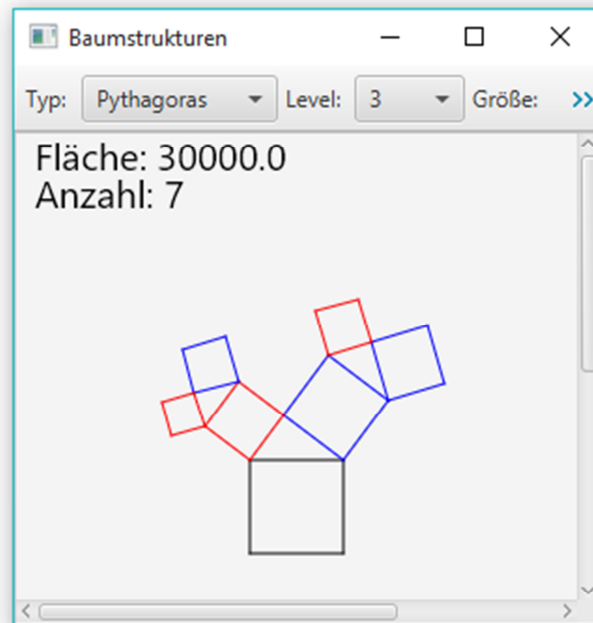


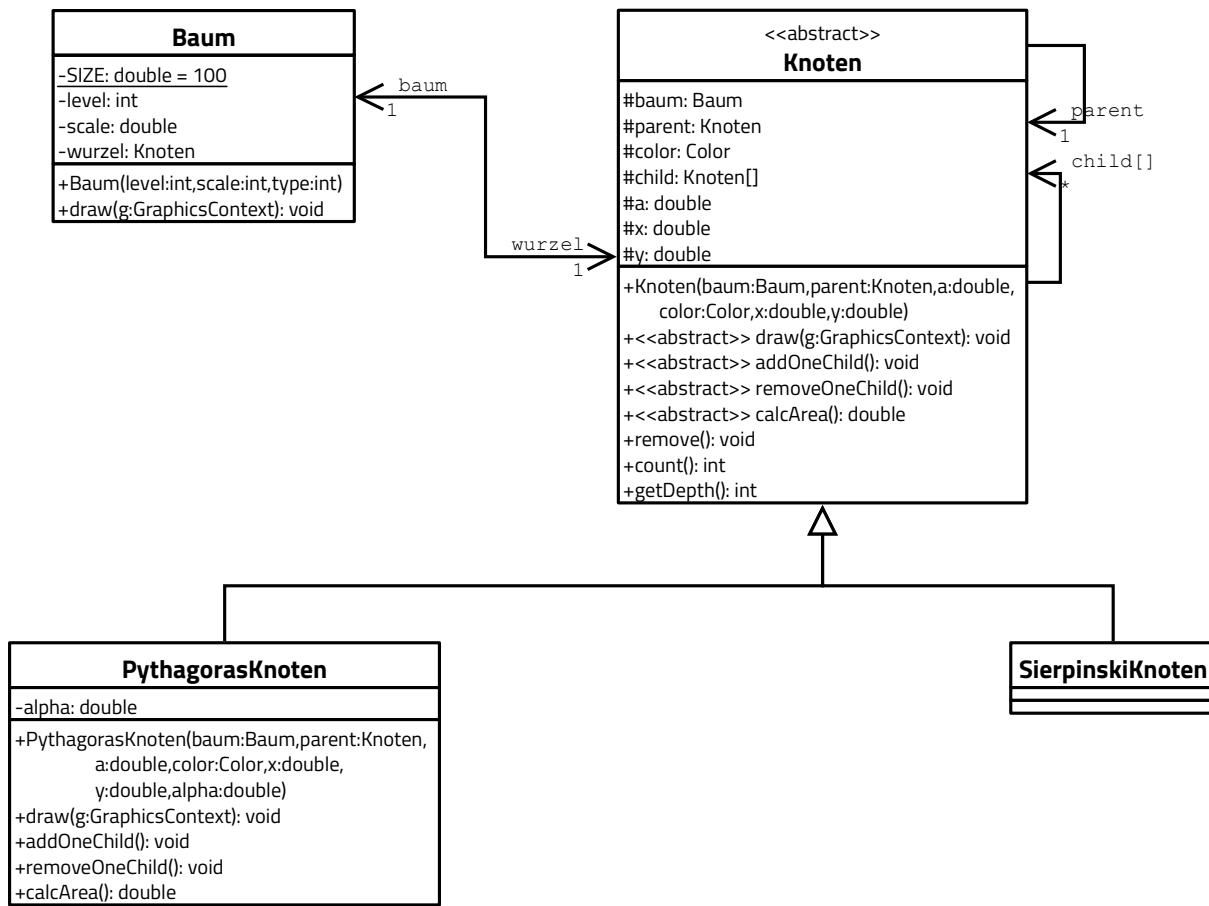
Geben Sie eine Wertetabelle aus.

Gitterpunkte	
n Geradenpaare	Anzahl der Schnittpunkte
1	1
2	4
...	
36	1296
37	1369
...	

6. Baumstrukturen

Entwerfen Sie ein Modell zum Zeichnen von Baumstrukturen: Pythagorasbaum, Sierpinski-Dreieck und Von Koch-Kurve.





7. Anzeige des Scene Graph

Schreiben sie eine Klasse zur rekursiven Anzeige der Baumstruktur des Scene Graphs. Testen Sie an unterschiedlichen Beispielen. Sehen Sie eine Möglichkeit zur Beeinflussung der Durchlaufrichtung vor z.B.:

Vorwärts:

```

BaumPane
  Top:
    ToolBar
      Label
      ComboBox
      Label
      ComboBox
      Label
      Slider
      Label
      Button
    Center:
      ScrollPane
        BaumCanvas
  
```

Rückwärts:

```

Label
ComboBox
Label
ComboBox
Label
Slider
Label
Button
ToolBar
Top:
  BaumCanvas
ScrollPane
Center:
  BaumPane
  
```