

SEW3 UE08 2024/25

Rekursion, Grafik

1) Fibonacci- Zahlen

Die **Fibonacci-Folge** ist die unendliche Folge von natürlichen Zahlen, die mit zweimal der Zahl 1 beginnt. Im Anschluss ergibt jeweils die Summe zweier aufeinanderfolgender Zahlen die unmittelbar danach folgende Zahl (siehe <https://de.wikipedia.org/wiki/Fibonacci-Folge>)

Es gilt also : $f(n) = f(n-1) + f(n-2)$

Die ersten Elemente der Folge sind demnach: **1 1 2 3 5 8 13 21 ...**

- Schreibe eine rekursive Methode

```
int fibonacci(int n) { .. } ,
```

die das n-te Element der Fibonacci-Folge ermittelt und zurückliefert

$\text{fibonacci}(1) \rightarrow 1$, $\text{fibonacci}(2) \rightarrow 1$, $\text{fibonacci}(3) \rightarrow 2$,

- Schreibe eine Methode

```
void printFibonacciFolge(int anz) { .. } ,
```

die mit Hilfe der obigen Funktion **fibonacci(..)** eine Folge von Fibonacci-Zahlen mit **anz** Elementen ausgibt ...

(das Beispiel weiter oben wird von **printFibonacciFolge(8)** erzeugt ..)

2) Rekursion mit Strings

Schreibe eine rekursive Methode

```
int zaehleSelbstlaute(String s) { .. } ,
```

die zählt und zurückliefert, wie viele Selbstlaute im String s enthalten sind (Groß-/Kleinschreibung soll ignoriert werden).

[rekursiver Ansatz: Teilergebnis = 1 Zeichen + Reststring analog weiter bearbeitet]

3) Zahl hoch

Viele(9) Multiplikationen:

$$2^{10} = 2 * 2 * \dots$$

Oder: durch geschicktes Zusammenfassen – 4 Multiplikationen

$$2^{10} = 2^5 * 2^5$$

$$2^5 = 2 * 2^2 * 2^2$$

$$2^2 = 2^1 * 2^1$$

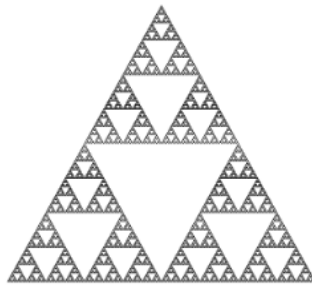
$$2^1 = 2$$

Schreibe die Methode `zahlHoch(int x, int n)` die (rekursiv) nach der zweiten Methode rechnet.

Löse: wenn man im Jahre 0 einen Euro mit 1% verzinst angelegt hätte – wie viel wäre das heute.

4) Sierpinski-Dreieck

Zeichne das Sierpinski-Dreieck.



Sierpinski-Dreieck (sieh auch: <https://de.wikipedia.org/wiki/Sierpinski-Dreieck>)

- Gehe von der Java-Datei »DemoSimpleDrawing.java« aus.
- Die eigentliche rekursive Zeichen-Methode soll `void drawSierpinskiTriangle` heißen und die folgenden Parameter haben:
 - `GraphicsContext gc`: die Zeichenfläche
 - `int size`: Abbruchkriterium (Anzahl der Pixel der kürzesten Kantenlänge, die noch gezeichnet werden soll).
 - Koordinaten für die 3 Eckpunkte (`double`) des großen Dreiecks.
- Das große Dreieck besteht aus drei kleineren Dreiecken – diese kann man durch einen rekursiven Aufruf zeichnen.
- Beispiel:
 - Die Seitenlänge des großen Dreiecks soll 580 Pixel sein.
 - Die kürzeste Seite, die noch gezeichnet wird, soll maximal 10 Pixel sein.

Hinweise:

- das Dreieck kann auch schief sein
- es hat schon jemand eine Klasse für Punkte geschrieben (`Point2D`, Teil von JavaFX)