

# Informatik Q1 Abels



Was ist **Rekursion**?

# Was passiert hier?

```
public static int iterative(int n) {  
    int x = 0;  
    for (int i = 0; i <= n; i++) {  
        x += i;  
    }  
    return x;  
}
```

# Was passiert hier?

```
public static int recursive(int n) {  
    if (n == 1) return 1;  
    return n + recursive(n-1);  
}
```

# Iteration

```
public static int sum(int n) {  
    int s = 0;  
    for (int i = 0; i <= n; i++) {  
        s += i;  
    }  
    return s;  
}
```

# Rekursion

```
public static int sum (int n) {  
    if (n == 1) return 1;  
    return n + sum (n-1);  
}
```



# Übung 1

Schreibe eine Klasse namens **Fakultaet.java**, die folgende Funktionen enthält:

- `public static int fak_iterative(int n)`
- `public static int fak_recursive(int n)`

# Fibonacci



*Leonardo von Pisa (ca. 1170 – 1250) besser unter dem Namen Fibonacci (Sohn des Bonacci, wobei Bonacci bedeutet: „von gutem Wesen o.ä.) bekannt, stellte in seinem wichtigsten Werk, dem Liber Abaci von 1202 die harmlos erscheinende Kaninchenaufgabe:*

*„Jemand brachte ein frisch geborenes Kaninchenpaar in einen allseits von Wänden umgebenen Ort, um herauszufinden, wie viele Paare aus diesem Paar in einem Jahr entstehen würden. Es sei die Natur der Kaninchen, pro Monat ein neues paar hervorzubringen und im zweiten Monat nach der Geburt erstmals zu gebären. Todesfälle sollen nicht eintreten.“*

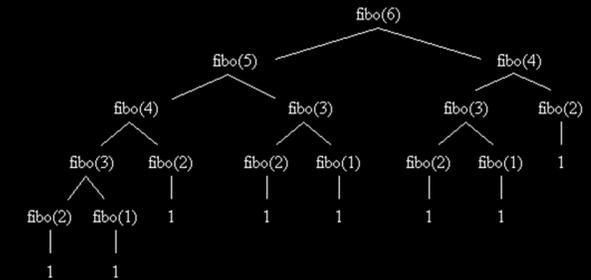


# Übung 2

$$\begin{array}{cccccc} 1 & 1 & 2 & 3 & 5 & 8 \\ \hline a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \quad \dots \quad a_n = a_{n-1} + a_{n-2}$$

a) Schreibe eine Klasse namens **Fibonacci.java**, die folgende Funktionen enthält:

- `public static int fib_iterative(int n)`
- `public static int fib_recursive(int n)`



b) Stelle in einer Visualisierung (**Ableitungsbaum**) dar, welche Funktionsaufrufe bei der rekursiven Variante für den Fall  $n=6$  getätigt werden.

c) Gehe anhand dessen auf die **Ineffizienz** ein des rekursiven Algorithmus ein.

d) Erkläre folgende Unterscheidung: **lineare Rekursion** vs **Mehrfachrekursion**



## Übung 3



*Der Legende nach begeisterte das Schachspiel den König von Indien so sehr, dass er den Erfinder des Spiels suchen ließ. Aus Dankbarkeit wollte er einem Untertan (ein Mathematiklehrer namens Buddhiram) jeden Wunsch erfüllen. Dieser wünschte sich daraufhin ein Reiskorn auf dem ersten Feld, zwei auf dem zweiten Feld, vier auf dem dritten Feld, 8 auf dem vierten Feld usw. Der König lächelte zunächst über die Einfalt seines Untertanen, stellte jedoch bald fest, dass der Wunsch nicht zu erfüllen war.*

- a) Zur Berechnung der Reiskörner auf dem Feld  $n$  entwickelte Harry folgendes Programm. Wieso funktioniert sein Programm nicht?

```
public static long potenzVonZwei(int n) {  
    return 2 * potenzVonZwei(n-1);  
}
```

- b) Schreibe eine Klasse namens **Reiskorn.java**, die folgende Funktionen enthält:
- `public static long reisAufFeld(int n)`
  - `public static long reisBisFeld(int n)`



## Übung 4

- a) Welche Funktionswerte liefert die folgende Funktion bei den Parametern HANNAH, SARAH und OTTO ?

```
public static boolean textprobe(String s) {
    if ((s.length() == 0) || (s.length() == 1)) {
        return true;
    }
    return textprobe(s.substring(1, s.length()-1)) && (s.charAt(0) == s.charAt(s.length()-1));
}
```

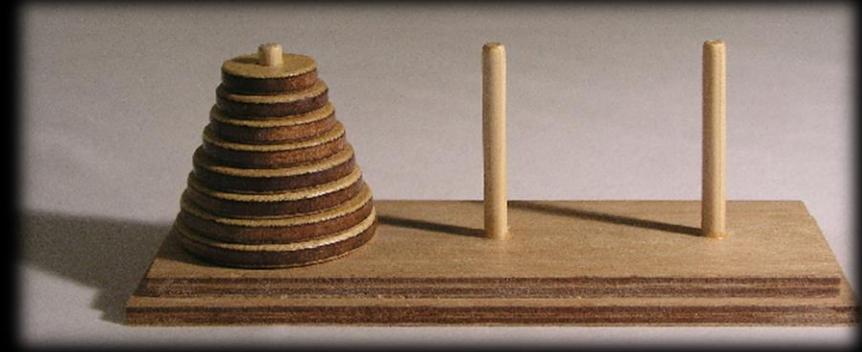
- b) Beschreibe die Funktionalität der Funktion.
- c) Schreibe eine Klasse namens **RekursiveStrings.java**, die folgende Funktion enthält:  
**public static String umdrehen(String word)**
- d) Vereinfache obige Funktion durch deine neue Funktion.



# Tagebucheintrag

Rekursion vs. Iteration

# Türme von Hanoi



*Das Spiel wird von einer Person gespielt. Es besteht aus drei gleich großen Stäben, auf die mehrere gelochte Scheiben gesteckt werden, alle verschieden groß.*

*Zu Beginn liegen alle Scheiben auf dem linken Stab, der Größe nach geordnet, mit der größten Scheibe unten und der kleinsten oben (siehe Bild).*

*Ziel des Spiels ist es, den kompletten Scheiben-Stapel vom linken Stab auf den rechten Stab zu versetzen. Hierbei gelten zwei Regeln:*

- 1. Es darf immer nur eine Scheibe gleichzeitig bewegt werden.*
- 2. Die bewegte Scheibe darf nicht auf eine kleinere Scheibe gesteckt werden.*

*Folglich sind zu jedem Zeitpunkt des Spieles die Scheiben auf jedem Feld der Größe nach geordnet.*



# Wochenübung

- Probiere selbst aus: Wie viele Züge benötigst du auf jeden Fall für 3 Scheiben? Wie viele für 4? Wie viele benötigst du allgemein für n Scheiben?
- Beschreibe deine Strategie in einem Struktogramm.
- Entwickle eine Klasse **Hanoi.java**, in der folgende rekursive Funktion implementiert ist:

```
public static void move(int n, char a, char b, char c)
```

Die Funktion soll die Zugfolge in der Console ausgeben:

*Verschiebe die oberste Scheibe von a nach c.*

*Verschiebe die oberste Scheibe von a nach b.*

*Verschiebe die oberste Scheibe von c nach b.*

*USW.*