

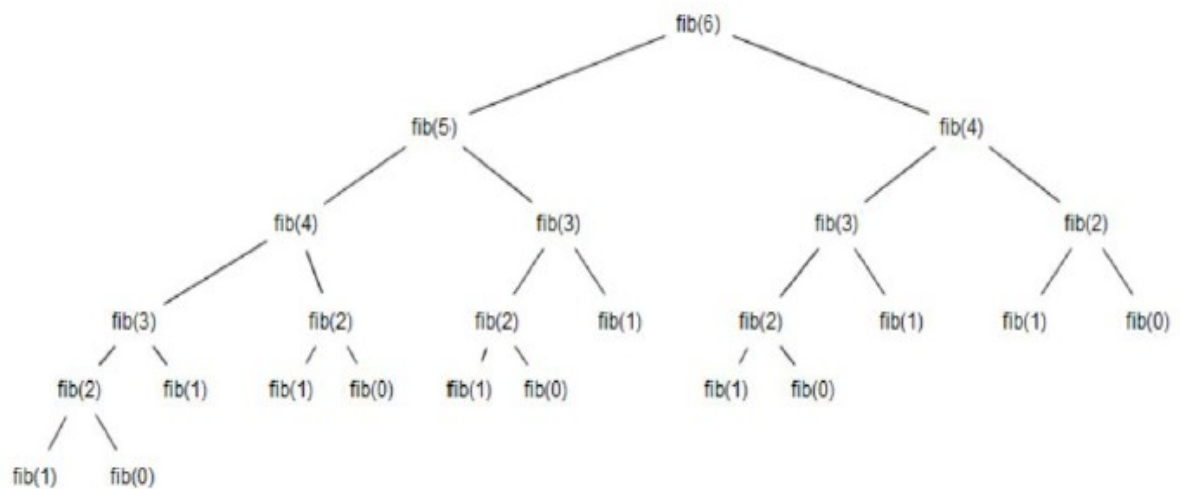
# Hausaufgabe „Lernaufgabe zur Rekursion“

## 1. Fibonacci Zahlen und kaskadierende Rekursion

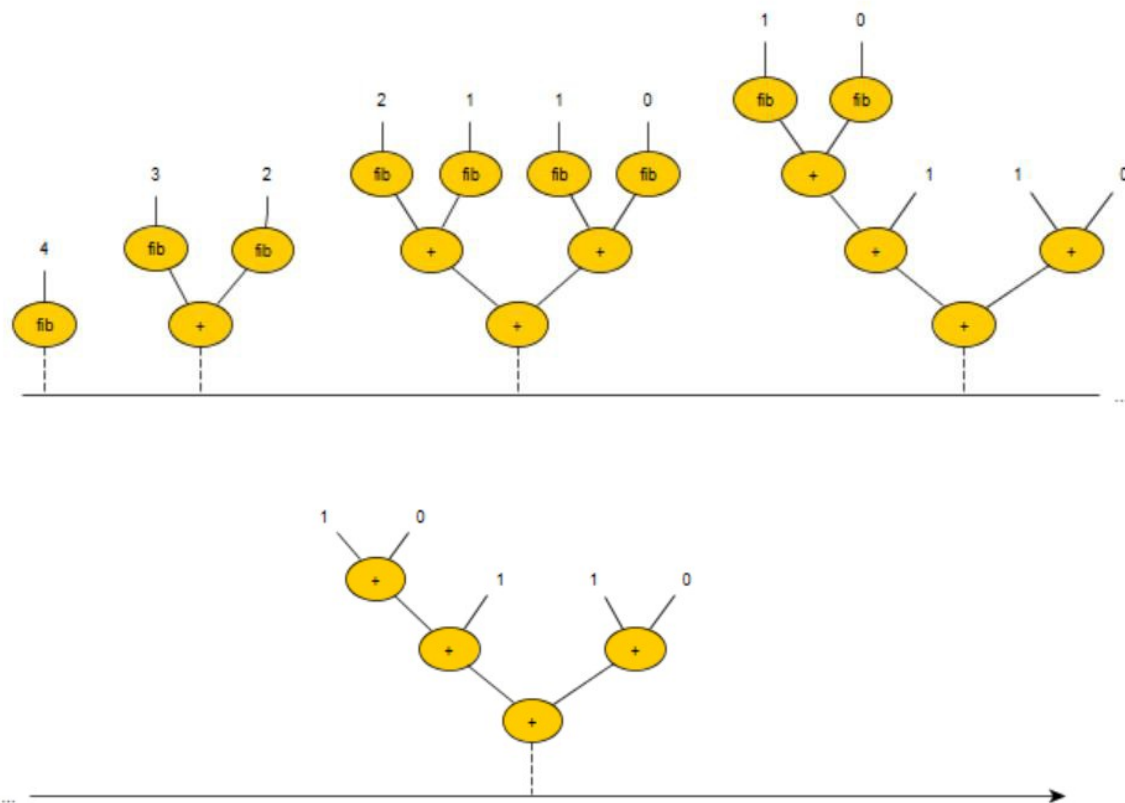
a,

```
public int fib(int n){  
    if (n<0) {  
        throw new IllegalArgumentException();  
    }  
    else if (n==0){  
        return 0;  
    }  
    else if (n==1){  
        return 1;  
    }  
    else {  
        return fib(n-1)+fib(n-2);  
    }  
}
```

b,



c,



## Aufgabe 2,

(Es wird in dieser Aufgabe davon ausgegangen, dass die SuS die Iteration mit Hilfe einer Zählschleife bereits verstanden haben)

Ihr seid im Urlaub mit eurer Familie und euer kleines Geschwisterchen hat Probleme in Mathematik. Leider ist es nun an euch das Multiplizieren zu erklären. Euer Bruder/Schwester beherrscht bereits die Addition und ihr soll darauf eingehend die Multiplikation erklären.

1. Versucht einen Alltagsgegenstand zu finden, bei den man das Konzept der Multiplikation zum Zählen verwenden kann.
2. Zeichnet ein Modell dieses Gegenstandes und versucht euch zu überlegen, wie diese Multiplikation auch als Addition geschrieben werden könnte.
3. Findet eine allgemeine Regel, wie Additionen in Multiplikationen umgeformt werden können.

Euer Geschwisterchen hat nun das Konzept verstanden und ihr kommt ins überlegen, ob man nicht diesen Ansatz verwenden kann um eine Multiplikation zu programmieren. Hierbei gibt es somit immer eine „Additionszahl“, welche immer wieder mit sich selbst addiert wird. Ebenso gibt es eine „Anzahl“, welche angibt wie oft die „Additionszahl“ mit sich selbst addiert wird.

4. An welchem Punkt des Programms kann ich die Zahl ausgeben? Welchen Wert muss die „Anzahl“ annehmen? Versucht mit den Punkten 1-3 zu arbeiten.

5. Wir möchten immer wieder die selbe Funktion verwenden, welche einfach nur mehrmals aufgerufen wird. Wie muss man die Parameter verändern, dass das Programm am Ende terminiert?

5. Wie muss der Methodenaufruf, der bei der Berechnung der n-ten Zahl der Folge benötigt wird, aussehen?

6. Programmieren nun die Methode mit Hilfe der Iteration aus den letzten Stunden:

```
public int mult(int add, int anz)
```

7. Versuche nun die Methode ohne eine Iteration (nur mit wiederholtem Funktionsaufruf zu programmieren: `public int mult(int add, int anz)`)

Für schnellere SchülerInnen: Überlegt euch, wie ihr euer Konzept auf negative Zahlen erweitern könnt.

Was passiert, wenn die „Anzahl“ 0 ist ?