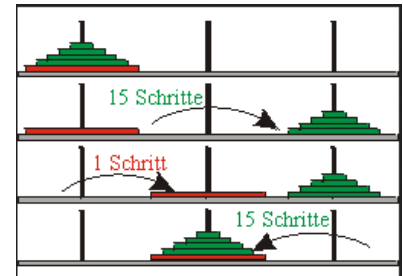


Türme von Hanoi

Java-Methoden mit Einzelschrittdurchläufen



Anzahl an benötigten Scheiben-Verschiebungen

```
public static int TvHAnzahl(int n) {
    if (n==1) return 1;
    else return TvHAnzahl(n-1) + 1 + TvHAnzahl(n-1);
    //um 1 kleinerer Turm wird verschoben
    //größte Scheibe wird verschoben
    //um 1 kleinerer Turm wird verschoben
}
```

Einzelschrittdurchlauf für 4 Scheiben

(beachte: die Methode „TvHAnzahl“ wird innerhalb der Methode immer wieder - mit veränderten Parametern - aufgerufen, weshalb mehrere Verschachtelungen erzeugt werden, bevor am Ende „rückwärtsgehend“ die ursprüngliche Anfrage gelöst wird)

TvHAnzahl(4) = 15 ☺ [= 2⁴-1]

```
n=1? nein (n=4)
else return TvHAnzahl(3)+1+TvHAnzahl(3) = 7+1+7 = 15
    n=1? nein (n=3)
    return TvHAnzahl(2)+1+TvHAnzahl(2) = 3+1+3 = 7
        n=1? nein (n=2)
        return TvHAnzahl(1)+1+TvHAnzahl(1) = 1+1+1 = 3
            n=1? ja
            return 1
```

Ausgabe der zu tätigenen Scheiben-Verschiebungen

```
public static String TvH(String a, String b, String c, int n) {
    if (n == 1) return "Scheibe von " + a + " nach " + b + ".\n";
    else return TvH(a, c, b, n-1) + TvH(a, b, c, 1) + TvH(c, b, a, n-1);
}
```

Einzelschrittdurchlauf am Beispiel (nur angedeutet, damit Logik erkennbar)

TvH(„A“, „C“, „B“, 2) = „Scheibe von A nach B.\nScheibe...“

```
n=1? nein
else - TvH(„A“, „B“, „C“, 1) + TvH(„A“, „C“, „B“, 1) + TvH(„B“, „C“, „A“, 1) = ...
    n=1? ja
    return „Scheibe von A nach B.“
```

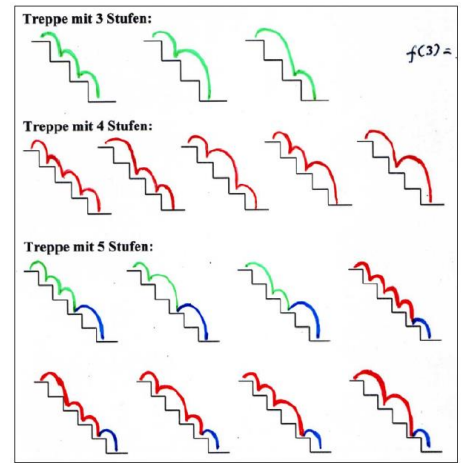
Java-Projekt (auf Konsole) - nur Ausgabe (obige Methoden - leicht verändert - eingebunden)

```
Wie viele Scheiben moechtest Du verschieben? 2
Scheibe von A nach B.
Scheibe von A nach C.
Scheibe von B nach C.
Das waren insgesamt 3 Züge.
Die rekursive Anzahl-Funktion liefert (auch) 3 Züge.
```

Treppenproblem

Java-Methoden

mit Einzelschrittdurchläufen



Anzahl an möglichen Kombinationen

```
public static int treppenAnzahl(int a) {
    if (a <= 1) return 1;
    else return treppenAnzahl(a-1)
               + treppenAnzahl(a-2);
    //Anzahl für a-1 Stufen - bei
    //Start mit Einzelschritt
    //+ Anzahl für a-2 Stufen - bei
    //Start mit Doppelschritt
}
```

Einzelschrittdurchlauf für 5 Stufen - siehe rechts

(die Abfragen/ Code-Ausschnitte sind in dem Beispiel nicht aufgeführt, aber durch die Verschachtelung eindeutig nachvollziehbar!)

Beispiel für eine Treppe mit 5 Stufen:

```
M(n) = M(n-g1) + M(n-g2)
M(5) = M(5-g1) + M(5-g2)
M(5) = M(5-1) + M(5-2)
M(5) = M(4) + M(3)
M(4) = M(4-1) + M(4-2)
M(4) = M(3) + M(2)
M(3) = M(3-1) + M(3-2)
M(3) = M(2) + M(1)
M(2) = [definiert] 2
M(1) = [definiert] 1
M(3) = M(2) + M(1) = 3
M(2) = [definiert] 2
M(4) = M(3) + M(2) = 5
M(3) = M(3-1) + M(3-2)
M(3) = M(2) + M(1)
M(2) = [definiert] 2
M(1) = [definiert] 1
M(3) = M(2) + M(1) = 3
M(5) = M(4) + M(3) = 8
```

Ausgabe der zu tätigenen Schritte

```
public static String treppen(int a, String s) {
    if (a == 0) return s + "\n";
    else if (a == 1) return "E" + s + "\n";
    else return treppen(a-1, "E" + s) + treppen(a-2, "D" + s);
}
```

Einzelschrittdurchlauf am Beispiel (nur angedeutet, damit Logik erkennbar)

Methodenablauf

```
anzahlMoeglichkeitenSTR(3, "")
= anzahlMoeglichkeitenSTR(2, "E" + "") + anzahlMoeglichkeitenSTR(1, "D" + "")
In anzahlMoeglichkeitenSTR(2, "E" + "")
anzahlMoeglichkeitenSTR(1, "E" + "E") + anzahlMoeglichkeitenSTR(0, "D" + "E")
In anzahlMoeglichkeitenSTR(1, "E" + "E")
= "E" + "EE" + "\n"
In anzahlMoeglichkeitenSTR(0, "D" + "E")
= "DE" + "\n"
....
```

Java-Projekt (auf Konsole) - nur Ausgabe (obige Methoden - leicht verändert - eingebunden)

```
Wie viele Stufen möchtest Du gehen? 3
EEE
DE
ED
Das waren insgesamt 3 Möglichkeiten.
Die rekursive Anzahl-Funktion liefert (auch) 3 Möglichkeiten.
```