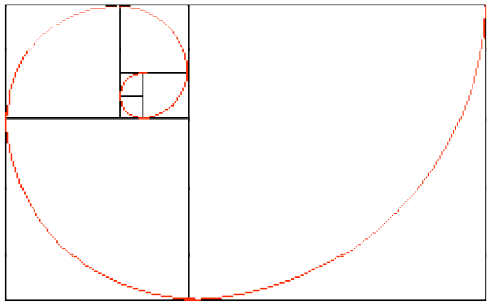
Beispiel 3



Definition der Fibonacci-Folge:

$f_n = f_{n-1} + f_{n-2} \; \forall \; n \in \mathbb{N}, n > 1$
$f_0 := 0, f_1 := 1$

In der Fibonacci-Spirale haben die Radien der Viertelkreise zueinander das durch die Fibonacci-Folge definierte Verhältnis.

„Top-Down–Verfahren:“

Erster Schritt:

```
Start "Hauptprogramm"
  berechne Fibonacci-Folge (bis z. B. 20)
  gib berechnete Folge aus
  zeichne Spirale mit Folge
Stop
```

Zweiter Schritt:

```
Start "gib Folge f aus"
  für i von 0 bis (Länge der Folge f) - 1
    gib das i-te Element der Folge f aus
Stop
```

(Die Folge f ist hierbei in Java ein Array.)

[weitere Schritte…]

Tutorium Programmiersprachen I
Arne Johannessen
Hochschule Karlsruhe – Technik und Wirtschaft

Beispiel 4

```
/* Dateiname: FibonacciNaiv.java
 * Programmiersprachen I, Hochschule Karlsruhe
 * Beispiel 4 zum Tutorium, Handzettel 2
 * Autor: Arne Johannessen
 * geschrieben am 2006-04-28
 */

/**
 * Berechnet die ersten 20 Fibonacci-Zahlen und gibt sie auf dem Terminal aus.
 * Verwendet wird ein "naiver" Algorithmus, der keine Arrays verwendet und daher
 * ein klein wenig unuebersichtlicher und unflexibler ist.
 *
 * Definiton von Fibonacci-Zahlen:
 * Die n-te Fibonacci-Zahl f(n) ist die Summe aus den beiden ("vorhergehenden")
 * Fibonacci-Zahlen an den Stellen n - 1 und n - 2. f(0) sei 0, f(1) sei 1.
 */
public class FibonacciNaiv
{

    public static void main (String args [])
    {

        // Laenge der zu druckenden Fibonacci-Folge
        int anzahl = 20;  // willkuerlich

        int a;  // f(n - 2)
        int b;  // f(n - 1)
        int f;  // f(n)

        // Startwerte (lt. Definition)
        a = 0;
        b = 1;

        System.out.print("0 1 ");

        // die Reihe ist definiert ab Stelle 2:
        for (int n = 2; n < anzahl; n++)
        {

            // diese Fibonacci-Zahl berechnen
            f = a + b;
            System.out.print(f+" ");

            // Zwischenwerte speichern
            a = b;
            b = f;
        }

        System.out.println();
    }

}
```