

6. Rekursive Funktionen

Definition: Eine Funktion heißt **rekursiv**, wenn ihr Anweisungsteil mindestens einen Aufruf von sich selbst enthält.

Aufgabe 13:

- a) Was bewirkt folgende rekursive Funktion?

```
void lies(char z)
{
    cin >> z;
    if (z != '#') lies(z);
    cout << z;
}
```

- b) Erstellen Sie ein Programm, das diese Funktion aufruft.
 c) Gehen Sie mit dem Cursor in die Zeilen, die mit Hilfe des Debuggers schrittweise abgearbeitet werden sollen, und klicken Sie auf das Symbol mit der Hand , betätigen Sie anschließend wiederholt den Schalter  links neben der Hand → Programm wird Schritt für Schritt abgearbeitet.
 d) Beobachten Sie den Parameter z und die Aufrufliste (*Ansicht / Debug-Fenster / Aufrufliste*).

Aufgabe 14:

- a) Was berechnet folgende rekursive Funktion?

```
unsigned long int f(unsigned int n)
{
    if (n < 2) return 1;
    else return n * f(n - 1);    //tail recursion
}
```

- b) Erstellen Sie ein Programm, das nach Eingabe einer natürlichen Zahl m den Wert von $f(m)$ auf dem Bildschirm ausgibt.
 c) Arbeiten Sie das Programm mit Hilfe des Debuggers schrittweise ab.

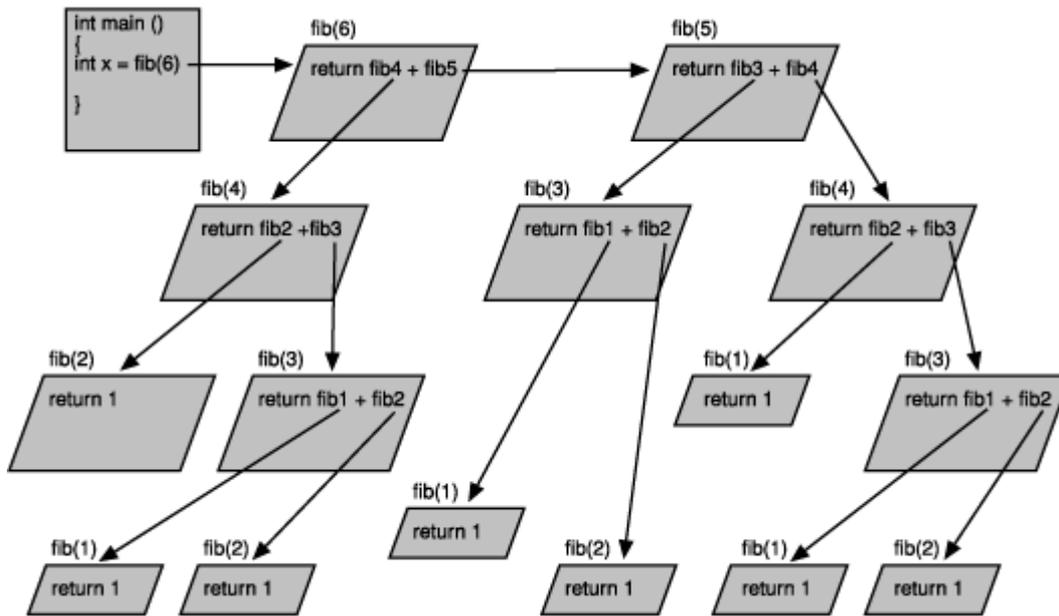
Aufgabe 15:

Die **Fibonacci-Folge** ist wie folgt rekursiv definiert:

$$\begin{aligned} fib(1) &= 1 \\ fib(2) &= 1 \\ \text{und für } n > 2: \quad fib(n) &= fib(n - 2) + fib(n - 1) \end{aligned}$$

- a) Berechnen Sie $fib(10)$ nach obiger Vorschrift.
 b) Geben Sie die C++-Definition dieser rekursiv definierten Funktion an.
 c) Schreiben Sie ein Programm zur Berechnung und Ausgabe der Folgeglieder nach Eingabe einer ganzen Zahl n .
 d) Arbeiten Sie das Programm mit Hilfe des Debuggers schrittweise ab. Beobachten Sie dabei die Variable n .

Grafische Veranschaulichung der Rekursion¹



Aufgabe 16:

Schreiben Sie eine **rekursive** Funktion ohne Rückgabewerte, die eine natürliche Zahl x in eine Du-
alzahl durch wiederholte Division durch 2 umrechnet.

Beispiel $x = 25$:

$$\begin{aligned}
 x &= q * 2 + r \\
 25 &= 12 * 2 + 1 \\
 12 &= 6 * 2 + 0 \\
 6 &= 3 * 2 + 0 \\
 3 &= 1 * 2 + 1 \\
 1 &= 0 * 2 + 1
 \end{aligned}$$

Der Algorithmus wird abgebrochen, wenn man als Quotienten den Wert 0 erhält. Die gesuchte Du-
alzahl ergibt sich aus den von unten nach oben gelesenen Resten r , also für das Beispiel 11001.

Hinweis: Stellt man die Ausgabe an den Schluss der rekursiven Funktionsdefinition, so erzielt man
diese Ausgabe „von unten nach oben“.

Aufgabe 17:

Folgende Funktion ermittelt den größten gemeinsamen Teiler von natürlichen Zahlen rekursiv.

```

int ggtrek(int m, int n)
{
    if (n == 0) return m;
    return ggtrek(n, m % n);
}
  
```

Ersetzen Sie in dem Programm aus Aufgabe 3 (S. 11) die Funktion durch obige Funktion.

Aufgabe 18:

Definieren Sie eine Funktion, die die *Ulamfolge* aus dem Kapitel *Schleifen* rekursiv berechnet.

¹ Quelle: Jesse Liberty, C++ in 21 Tagen, Tag 5, S. 23, ISBN: 3-8272-5624-1