

Fibonacci-Zahlen

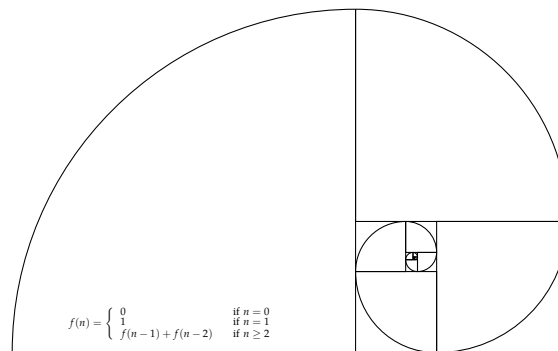
(Fibonacci-Zahlen)

Stichwörter: Mehr-Adress-Befehl-Assembler

Die Fibonacci-Folge ist die unendliche Folge natürlicher Zahlen, die mit zweimal der Zahl 1 beginnt. Im Anschluss ergibt jeweils die Summe zweier aufeinanderfolgender Zahlen die unmittelbar danach folgende Zahl:

$$\text{fib}_n = \text{fib}_{n-1} + \text{fib}_{n-2}$$

Dabei bezeichnet n die n -te Zahl dieser Reihe. Die darin enthaltenen Zahlen heißen Fibonacci-Zahlen. Benannt ist die Folge nach Leonardo Fibonacci, der damit im Jahr 1202 das Wachstum einer Kaninchenpopulation beschrieb. Die Folge war aber schon in der Antike sowohl den Griechen als auch den Indern bekannt. Gleichmaßen lassen sich Quadratgrößen damit beschreiben:



Die Lösung der Berechnung soll zum Schluss in R5 liegen.

Lösungsvorschlag

```

-- Fibonacci-Zahlen

-- Die Fibonacci-Folge ist die unendliche Folge natuerlicher Zahlen, die mit zweimal der
-- Zahl 1 beginnt. Im Anschluss ergibt jeweils die Summe zweier aufeinanderfolgender
-- Zahlen die unmittelbar danach folgende Zahl:
-- fib_n = fib_n-1 + fib_n-2
-- Dabei bezeichnet n die n-te Zahl dieser Reihe. Die darin enthaltenen Zahlen heissen
-- Fibonacci-Zahlen. Benannt ist die Folge nach Leonardo Fibonacci, der damit im Jahr
-- 1202 das Wachstum einer Kaninchenpopulation beschrieb. Die Folge war aber schon
-- in der Antike sowohl den Griechen als auch den Indern bekannt. Gleichmassen lassen
-- sich Quadratgroessen damit beschreiben:
-- Die Loesung der Berechnung soll zum Schluss in R5 liegen.

-- public static int fib(int n) {
--     if (n <= 0)
--         return 0;
--     else if (n == 1 || n == 2)
--         return 1;
--     else
--         return fib(n - 1) + fib(n - 2);
-- }

fibonacci:

```

```

SEG
    MOVE W I H'0000FFFF', SP
    JUMP einstieg

fib:
    PUSHR
    MOVE W 64+!SP, R0
    CMP W R0, I 1
    JLT gibNull
    CMP W R0, I 2
    JLT gibEins
    -- n - 1
    SUB W I 1, R0, R0
    -- n - 2
    SUB W I 1, R0, R1
    MOVE W I -1, -!SP
    MOVE W R0, -!SP
    CALL fib
    ADD W I 4, SP
    -- Ergebnis von fib(n - 1) nach R3
    MOVE W !SP+, R3
    MOVE W I -1, -!SP
    MOVE W R1, -!SP
    CALL fib
    ADD W I 4, SP
    -- Ergebnis von fib(n - 2) nach R4
    ADD W !SP+, R4
    -- fib(n - 1) + fib(n - 2)
    ADD W R4, R3, R2
    JUMP rueckgabe

gibNull:
    MOVE W I 0, R2
    JUMP rueckgabe

gibEins:
    MOVE W I 1, R2

rueckgabe:
    MOVE W R2, 68+!SP
    POPR
    RET

einstieg:
    MOVE W n, R0
    MOVE W I -1, -!SP
    MOVE W R0, -!SP
    CALL fib
    ADD W I 4, SP
    -- bei n = 7: 13
    MOVE W !SP+, R5
    JUMP abschluss

abschluss:
    HALT

n:
    DD W 7

-- n:
    DD W 0 -- 0
-- n:
    DD W 1 -- 1

```

```

-- n:          DD W 2 -- 1
-- n:          DD W 3 -- 2
-- n:          DD W 4 -- 3
-- n:          DD W 5 -- 5
-- n:          DD W 6 -- 8
-- n:          DD W 7 -- 13
-- n:          DD W 8 -- 21
-- n:          DD W 9 -- 34
-- n:          DD W 10 -- 55
END

```

```

public class FibonacciRekursiv {

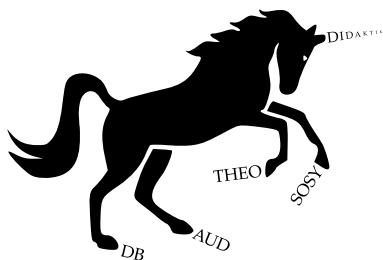
    public static int fib(int n) {
        if (n <= 0)
            return 0;
        else if (n == 1 || n == 2)
            return 1;
        else
            return fib(n - 1) + fib(n - 2);
    }

    public static void main(String[] args) {
        int n = 7;
        System.out.println(fib(n));

        System.out.println(fib(0)); // 0
        System.out.println(fib(1)); // 1
        System.out.println(fib(2)); // 1
        System.out.println(fib(3)); // 2
        System.out.println(fib(4)); // 3
        System.out.println(fib(5)); // 5
        System.out.println(fib(6)); // 8
        System.out.println(fib(7)); // 13
        System.out.println(fib(8)); // 21
        System.out.println(fib(9)); // 34
        System.out.println(fib(10)); // 55
    }
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/tech_info/assembler/mehr_adress/FibonacciRekursiv.java](https://github.com/bschlangaul/aufgaben/tech_info/assembler/mehr_adress/FibonacciRekursiv.java)



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der \TeX -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50_TECH/20_Mehr-Adress/Aufgabe_11-Fibonacci.tex