

C++ Programme

Grundstruktur und Funktionen

1 Ein Programm, dass absolut nichts tut.

Ein C++ Programm besteht mindestens aus der Funktion main, die einen Wert vom Typ

```
int
```

zurückgibt. Die Funktion benötigt keine Parameter.

Eine Funktion ist wie in der Mathematik definiert

$$f : X \rightarrow Y, f(x) \mapsto y.$$

In C++ sind X und Y Typen, z.B.

```
int long int double
```

aber auch der leere Typ

```
void
```

oder aber auch

```
()
```

Syntax:

```
Ergebnistyp funktions_name(ParameterTyp)
```

```
{
```

```
    Folge von Anweisung;
```

```
}
```

Man kann eine nichtstuende Funktion main und somit ein nichtstuendes Programm so definieren:

```
int main(void)
```

```
{
```

```
}
```

Es macht noch weniger als das schon absolut nichtsnutzige "Hello World". Wir benötigen auch keinerlei Präprozessor-Anweisungen wie

```
#include<iostream>
```

2 Ein nützliches Programm

Im folgenden werden wir uns eine Lösung zur ersten “richtigen” Programmieraufgabe genauer anschauen. Die wichtigen Zeilen sind mit Nummern versehen. Siehe die Erklärung zum Schluss.

```
(1) #include <iostream>
(2) // ein erster Kommentar
(3) long int sumFormel0(int n)
(4) {
(5)     long int wert = 0;
(6)     for (int i = 1; i <= n; i++)
(7)     {
(8)         wert = wert + i;
(9)     }
(10)     wert = wert*wert;
(11)     return wert;
(12)}
```

```
long int sumFormel1(int n)
{
    long int wert = 0;
    for(int i=1;i<=n;i++)
    {
        wert = wert+i*i*i;
    }
    return wert;
}
```

```

(13) int main(void)
(14) {
(15) long int wert = 0;
(16) int n;
(17) std::cout << " Gib eine natuerliche Zahl ein: \n";
(18) std::cin >> n;
(19) wert = sumFormel0(n);
(20) std::cout << "Summe der ersten " << n << " Kubikzahlen nach Methode 1: ";
    std::cout << wert << '\n';
    wert = sumFormel1(n);
    std::cout << "Summe der ersten " << n << " Kubikzahlen nach Methode 2: ";
    std::cout << wert << '\n';
    std::cout << "Fertig!\n";
(21) }
(22)

```

Bemerkungen

- (1) Präprozessoranweisung: Einbinden der Bibliothek `iostream`, benötigt man für Ein- und Ausgabe.
- (2) Definition einer (lokalen) Funktion `sumFormel`, die Werte vom Typ `long int` liefert und einen Parameter vom Typ `int` verlangt.
- (3) Kommentarzeilen werden durch `//` eingeleitet
- (4) Beginn des Funktionenrumpfes
- (5) Definition der lokalen Variablen `wert` vom Typ `long int` und gleichzeitige Initialisierung mit dem Wert `0`
- (6) eine `for`-Schleife, `i` ist hier wieder lokal definiert, die Abbruchbedingung ist das `n` aus der Parameterliste.
- (7) Beginn des Rumpfes der `for`-Schleife
- (8) Zuweisung
- (9) Ende des Schleifenrumpfes
- (10) Zuweisung, `wert` hat vorher den Wert $\sum_{i=1}^n i$, dannach den Wert $(\sum_{i=1}^n i)^2$
- (11) als Ergebnis der Funktion wird `wert` definiert und die Funktion verlassen

- (12) Ende des Funktionenrumpfes
- (13) die Hauptfunktion main
- (14) Beginn des Rumpfes von main
- (15) Definition von wert als long int und Initialisierung mit 0
- (16) Definition von n
- (17) Ausgabe auf die Standardausgabe (Bildschirm) \n ist ein Zeilenumbruch, std:: steht für den Standard-Namensraum. Wenn using namespace std in das Programm eingefügt wird, reicht cout <<...
- (18) Einlesen eines Wertes in n von der Standarteingabe (Tastatur), beachte die Richtung der "Pfeile" >>
- (19) Aufruf der Funktion sumFormel0 und Zuweisung an wert
- (20) Ausgabe auf die Standardausgabe (Bildschirm): << Reiht verschiedene Typen (String, long int String) zu einem Datenstrom
- (21) Ende der main-Funktion
- (22) der C++ Compiler erwartet einen Zeilenumbruch als letztes Zeichen in der Programmdatei