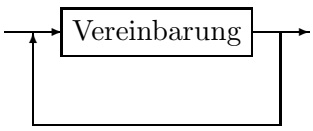
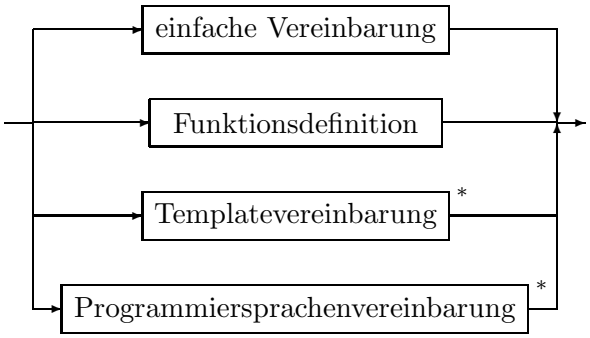
Programmaufbau

- Der Zeichensatz eines C++-Quellprogramms besteht aus den druckbaren Zeichen des ASCII-Zeichensatzes mit Ausnahme von `$`, `‘`, `@`. Dazu kommen noch das Leerzeichen, Horizontal- und Vertikaltabulator, Seitenvorschub, und eine Darstellung für das Zeilenende. In Zeichen(ketten)literalen und Kommentaren können weitere druckbare Zeichen vorkommen. Groß- und Kleinschreibung ist signifikant.
- Vor der eigentlichen Übersetzung des C++-Programms, das auf mehrere Dateien verteilt sein kann, wird ein Präprozessorlauf durchgeführt, der Textersetzungen vornimmt, Dateien einfügt und das Programm in Eingabesymbole (Namen, reservierte Wörter, Literale, Operatoren, Trennzeichen) zerlegt. Dabei werden sukzessive jeweils die längstmöglichen[†] Eingabesymbole gebildet.
- Namen werden aus Buchstaben, Ziffern und dem Unterstrich `_` gebildet und müssen mit einem Buchstaben oder dem Unterstrich beginnen.[‡]
- Zwischenraum (Leerzeichen, Zeilenende, Tabulatorzeichen, Seitenvorschub, Kommentare) kann zwischen Eingabesymbolen zur Erhöhung der Lesbarkeit eingefügt werden. Er muß direkt benachbarte Namen, reservierte Wörter und Literale trennen.
- Kommentare werden durch `/*` eingeleitet und durch `*/` abgeschlossen, oder erstrecken sich von `//` bis zum Zeilenende. Sie werden vom Präprozessor noch vor Interpretation der `#`-Anweisungen durch ein Leerzeichen ersetzt.
- Das Ergebnis des Präprozessorlaufs sind eine oder mehrere Übersetzungseinheiten.

Übersetzungseinheit :

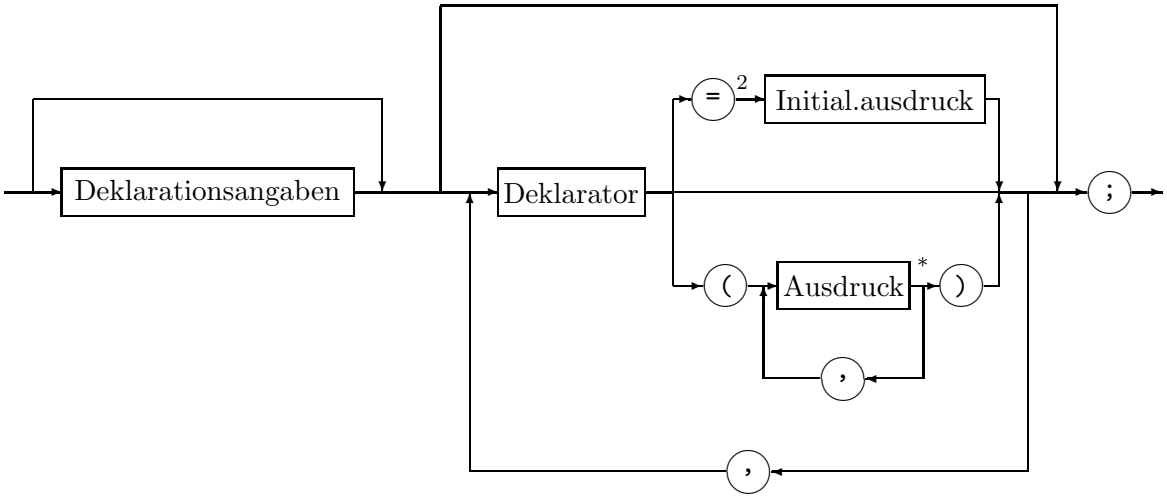


Vereinbarung [1]:

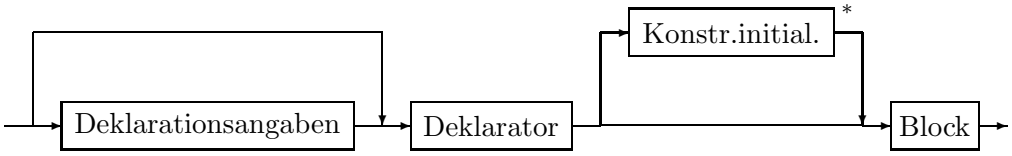


[†] mit bestimmten sinnvollen Ausnahmen ab C++11
[‡] für Nichtziffern sind Unicodezeichen in einer Ersatzdarstellung möglich
[*] nur C++
[1] vereinfacht, insb. ohne Namensräume, ASM-Definitionen und Attributangaben

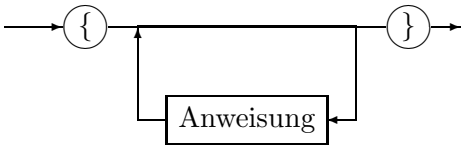
einfache Vereinbarung :



Funktionsdefinition [3]:

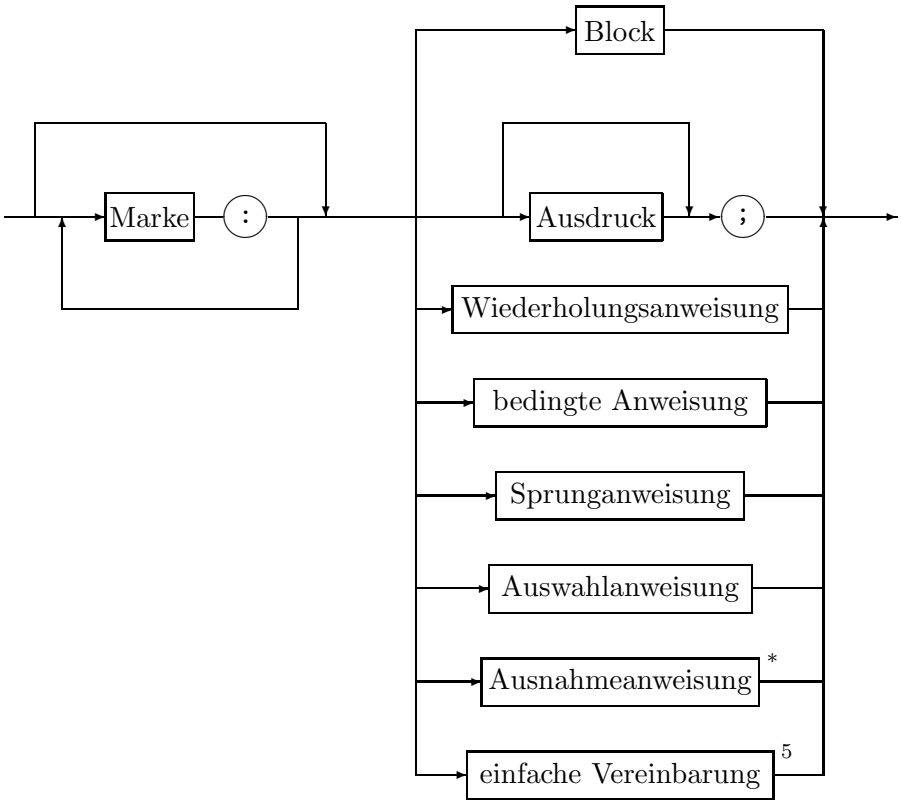


Block :



[2] Kann in C++11/14 in bestimmten Fällen weggelassen werden
[3] ohne try-Funktionsblock

Anweisung[4] :



- Innerhalb eines C++-Programms muß es genau eine Funktionsdefinition mit Namen `main` geben. Diese wird beim Programmstart automatisch aufgerufen.
- Namen müssen vor ihrer Verwendung vereinbart werden (Ausnahme: Marken). Der Gültigkeitsbereich erstreckt sich vom Vereinbarungspunkt bis zum Ende des innersten umschließenden Blocks bzw. bei Vereinbarungen außerhalb aller Funktionen bis zum Ende der Übersetzungseinheit (Ausnahmen: Klassen, Funktionsparameter, Marken).
- Namen, die in einem Block vereinbart werden, haben ab dem Vereinbarungspunkt Vorrang vor den Namensvereinbarungen außerhalb des Blocks.
- Zu den Ausdrücken zählt auch die Zuweisung.

Beispielprogramme

- **Hello, world!**

```
#include <iostream>
using namespace std;

int main()
{
  cout << "Hello, world!" << endl;
  return 0;
}
```

[4] ohne Attributangaben
[5] in ANSI C nur direkt am Blockanfang möglich

- $\sum_{i=1}^{n} i \quad (n \in \mathbb{N})$ *Tabellarische Ausgabe der Partialsummen*

```
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
  int i,n;
  double s;
  cout << "n: "; cin  >> n;
  i = 1; s = 0;
  while (i<=n) {
    s = s+i;
    // 2 Nachkommastellen bei Ausgabe von double-Zahlen
    // Feldbreiten 2 und 6
    cout << fixed   << setprecision(2)
         << setw(2) << i << setw(6) << s << endl;
    i = i+1;
  }
  return 0;
}
```

- $\sum_{k=0}^{n} \frac{x^k}{k!} \quad (x \in \mathbb{R}, n \in \mathbb{N})$

```
#include <iostream>
#include <cmath>
using namespace std;

double exp_reihe(double x, int n)
{
  int k; double s,a;
  k = 1; a = 1; s = 1;
  while (k<=n) {
    a = a*x/k;
    s = s+a;
    k = k+1;
  }
  return s;
}

int main()
{
  double x; int n;
  cout << "x n: ";
  cin  >> x >> n;
  cout << "x=" << x << "  n=" << n
       << "  exp_reihe(x,n)=" << exp_reihe(x,n)
       << "  exp(x)=" << exp(x) << endl;
  return 0;
}
```