

Programmiersprache C++

Вінницький національний технічний університет

Анотація. У статті розглядається мова програмування C++, її опис та критика.

Ключові слова: мова програмування, код, системне програмування, комп'ютерні системи.

C++ ist eine von der ISO genormte Programmiersprache. Sie wurde ab 1979 von Bjarne Stroustrup bei AT&T als Erweiterung der Programmiersprache C entwickelt. C++ ermöglicht sowohl die effiziente und maschinennahe Programmierung als auch eine Programmierung auf hohem Abstraktionsniveau. Der Standard definiert auch eine Standardbibliothek, zu der verschiedene Implementierungen existieren.

C++ wird sowohl in der Systemprogrammierung als auch in der Anwendungsprogrammierung eingesetzt und gehört in beiden Bereichen zu den verbreitetsten Programmiersprachen[1].

Typische Anwendungsfelder in der Systemprogrammierung sind Betriebssysteme, eingebettete Systeme, virtuelle Maschinen, Treiber und Signalprozessoren. C++ nimmt hier oft den Platz ein, der früher ausschließlich Assemblersprachen und der Programmiersprache C vorbehalten war.

Bei der Anwendungsprogrammierung kommt C++ vor allem dort zum Einsatz, wo hohe Forderungen an die Effizienz gestellt werden, um durch technische Rahmenbedingungen vorgegebene Leistungsgrenzen möglichst gut auszunutzen. Ab dem Jahr 2000 wurde C++ aus der Domäne der Anwendungsprogrammierung von den Sprachen Java und C# zurückgedrängt[2].

Ein Beispiel für das Schreiben eines Codes in der Programmiersprache C ++ ist in Abbildung 1 dargestellt.

```
// Include needed header files
#include <OC_const.h>           // OC Constants
#include <Common.h>           // Basic types
#include <stdio.h>            // Printf
#include <Data.h>             // Vectors and matrices
#include <OC_neg8.h>          // NAG function

int GetData()
{
    // Declare local variable holding error code
    int iErr;

    // *** Declare and initialize input parameters ***
    // *** passed into the NAG function nag_regsn_mult_linear. ***
    int nPts;                 // Input number of rows
    int nTdx;                 // Input number of columns
    matrix mX;                // Input matrix containing data points
    vector vY;                // Input vector containing data points
    vector vWT;               // Input vector containing weights

    // There are 10 rows of data in sample data
    nPts = 10;

    // There are 3 independent variables in data
    nTdx = 3;

    // Get independent variable data sets
    Worksheet wks("Data1"); // Attach OC Worksheet object
    mX.CopyFromWks( wks, 1, 3, 0, 9); // Copy independent variables
}
```

Abbildung 1. Ein Beispiel eines Codes in der Programmiersprache C ++

Kompatibilität mit C

Die Kompatibilität mit C zwingt C++ zur Fortführung einiger dadurch übernommener Nachteile. Dazu zählt die teilweise schwer verständliche C-Syntax, der als überholt geltende Präprozessor sowie verschiedene

von der jeweiligen Plattform abhängige Details der Sprache, die die Portierung von C++-Programmen zwischen unterschiedlichen Rechnertypen, Betriebssystemen und Compilern erschweren[3].

Sprachmerkmale im Detail. C++ basiert auf der Programmiersprache C beschrieben. Zusätzlich zu den in C vorhandenen Möglichkeiten bietet C++ weitere Datentypen sowie neuartige Typumwandlungsmöglichkeiten, Klassen mit Mehrfachvererbung und virtuellen Funktionen, Ausnahmebehandlung, Templates (Schablonen), Namensräumen, Inline-Funktionen, Überladen von Operatoren und Funktionsnamen, Referenzen, Operatoren zur Verwaltung des dynamischen Speichers und mit der C++-Standardbibliothek eine erweiterte Bibliothek[4].

Oft geäußerte Kritik an der Sprache umfasst beispielsweise:

- C++ sei sehr komplex und fehleranfällig zu programmieren. Man müsse viel lernen und üben, um es gut zu beherrschen und viele Features gelten als äußerst komplex.
- C++ sei zu low-level-mäßig aufgebaut, während es viele Features von höher abstrahierenden Sprachen aufweist (Klassen, generische Klassen/Funktionen etc.), seien als wichtig empfundene Dinge, insbesondere Garbage Collection nicht vorhanden.
- Typisch in Verbindung mit C++ ist das Zitat von Bjarne Stroustrup[5].

Schlussfolgerungen

C++ gilt zwar als schnell, beispielsweise wegen der Möglichkeit, frei mit Pointern zu arbeiten, doch diese Leistung sei auf den heutigen, schnellen Computersystemen nur in Ausnahmefällen nötig: Während es sinnvoll sei, Betriebssysteme o. Ä. in C++ zu schreiben, sei es softwaretechnisch viel günstiger, Anwendungsprogramme in höheren Sprachen zu schreiben, da diese leichter zu warten seien und immer noch eine ausreichende Leistung aufwiesen.

LITERATUR

1. Bjarne Stroustrup: Proposal to Acknowledge that Garbage Collection for C++ is Possible. 1996, abgerufen am 6. Juni 2015 (PDF, englisch): „The most loudly proclaimed fundamental weakness of C++ these days is the absence of automatic garbage collection. For many applications automatic garbage collection is indeed a very powerful tool and the right tool for the job.“
2. Scott Meyers: Effektiv C++ Programmieren. Addison-Wesley, S. 22/23, 43/44 und 46, nach Google-Books, abgerufen 20. August 2014.
3. MSDN: Unterstützung für C++11/14/17-Funktionen (Modern C++) Unterstützung von C++11/14/17-Features durch Microsoft-Compiler.
4. https://www.henkessoft.de/C++/Cpp_school/_history.htm
5. <https://www.iso.org/standard/64029.html>

Варчук Ліана Вікторівна – викладач кафедри іноземних мов Вінницького національного технічного університету, Вінниця.

Душук Юліанна Ігорівна – студентка групи ІКІ-166, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: duschykyu@gmail.com.