

Visual C++

Einführung

Wenn das Wort 'Programmierung' fällt, denkt (fast) jeder an C++. C++ ist die populärste Programmiersprache die es derzeit gibt. Meist verbindet man mit C++ jedoch in irgendwelcher Weise eine ultraschwer zu erlernende und eigentlich nur Profis vorbehaltene Programmiersprache mit kryptischem Quelltext. Doch mit meinem Workshop möchte ich dir einen Einblick in die Programmiersprache C++ geben und dir zeigen, welches Hexenwerk hinter DER berühmt berüchtigten Programmiersprache steckt.

Was bedeutet der Name C++

Der Name C++ kommt folgendermaßen zustande: Die Programmiersprache beinhaltet einen speziellen Operator, den Postinkrementoperator '++', der die davor oder nach befindliche Variable um eins erhöht. Dieser Operator bringt C++ seinen interessanten Namen ein. Das C kommt so zustande: Die Vorgängersprache von C (und somit von C++) hieß B, und von dieser hieß der Vorgänger A (Dies ist kein Witz!). So heißt C eben C.

Vorteil von C++

Der Vorteil von C++ ist neben dem großen Funktionsumfang auch die Plattformunabhängigkeit, d.h. mit wenigen Handgriffen lässt sich das Programm auf die verschiedensten Arten von Betriebssystemen portieren (umwandeln).

Warum C++ und nicht Visual C++

Du wirst dich wundern, dass dieser Workshop von dem Thema Visual C++ handelt, ich jedoch im oberen Teil über C++ rede. Hiermit bist du einem oft gemachten Fehler auf den Leim gegangen, denn C++ ist die Programmiersprache und Visual C++ ist die Entwicklungsumgebung die dem Programmierer hilft C++-Programme zu erstellen, bei Visual C++ in dem sie dem Programmierer eine schon fertige Benutzeroberfläche, die nur noch an die Bedürfnisse angepasst werden muss, zum Verfügung stellt (in etwa wie bei Visual Basic).

Der Schwierigkeitsgrad von C++

Zugegebenermaßen ist Visual C++ jedoch noch nicht so einfach wie Visual Basic, doch VC++ (Visual C++) bietet ja auch einen größeren Funktionsumfang. Auch ist die Erstellung von visuellen Programmen nicht so gelöst wie bei VB (Visual Basic). Während VB immer automatisch ein Formular für den Programmierer lädt, muss bei VC++ erst die für die visuelle Programm-Entwicklung zuständige MFC (Microsoft Foundation Class) geladen werden. In VC++ kann auch "normales" C++ oder C programmiert werden. Die MFC ist nicht automatisch geladen, nur wenn der MFC-Anwendungsassistent ausgewählt wurde. Die MFC ist eine Menge von C++-Klassen (eine Klasse ist eine Versammlung von Funktionen (z.B. zur Anzeige)), die eine Windows-Anwendung bilden. Du siehst schon, schon die Grundlagen sind anspruchsvoller als bei VB.

Die MFC erstellt mittels eines MFC-Anwendungsassistenten eine Anwendung für dich (ähnlich wie beim VB-Anwendungsassistenten). Die dadurch entstandene Anwendung muss noch auf die eigenen Bedürfnisse abgeändert werden, und die einzelnen Funktionen eingefügt werden (Hört sich einfacher an als es ist). Der Entwurf von Benutzeroberflächen mit Steuerelementen usw. funktioniert ähnlich bei VB.

Visual C++ im Detail

Solltest du Visual C++ von Microsoft oder einen ähnlichen Compiler (Entwicklungsumgebungen) noch nicht besitzen und dich nun entschieden haben, dir Visual C++ zu kaufen, dann musst du für die Standard Edition (Version .NET) 118 € (BASIS! Stand: 6/2004) bezahlen. Aber für den Funktionsumfang den VC++ bietet ist der Preis angemessen.

Jedoch gibt es, wie oben erwähnt, nicht nur Visual C++ als Compiler, sondern auch andere Hersteller bieten C++-Compiler an, manche sogar kostenlos (von der etwas älteren Version 6 gab es sogenannte Autoreneditionen).

Der Funktionsumfang von Visual C++ 6

Da ich selbst noch die Version 6 besitzen möchte, möchte ich nun stellvertretend dafür auf den Funktionsumfang von Visual C++ zu sprechen kommen (bei der .NET-Version sind noch einige Features hinzugekommen):

VC++ erzeugt mit Hilfe der MFC 32bit Programme, es sind jedoch auch DOS-basierende C++ oder C Programme möglich, ähnlich wie VB lassen sich auch in VC++ beliebig viele ActiveX-Steuerelemente in das Projekt einbinden.

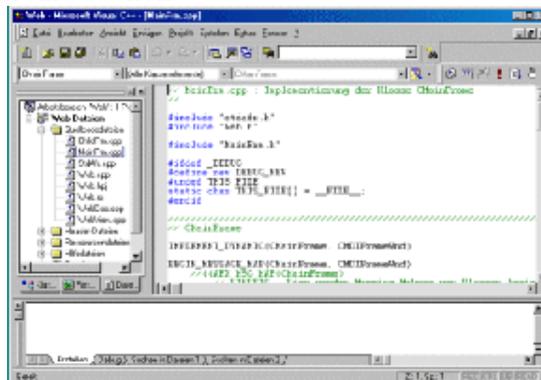
Visual C++ ist sehr vielseitig und zählt sicher zu den am meisten verbreiteten und verwendeten Programmiersprachen der Welt.

Nicht ganz so wie bei VB steht es mit der Dokumentation, es liegt zwar auch bei VC++ nur das elektronische Handbuch der msdn-Hilfe bei, jedoch liegt der Entwicklungsumgebung ein Buch zur Einführung bei (Ivor Horton's Einführung in MS Visual C++ 6.0). Einziger Hacken bei dem Buch: Es setzt voraus, dass man C++ beherrscht! Bei der Installation wird auch gleich der Wrox Press C++ Lehrgang mitinstalliert. Einziger Hacken bei dem Lehrgang: Er ist in englisch!

Die IDE

In der IDE (*integrated development environment*: Integrierte Entwicklungsumgebung = Programmierumgebung) befinden sich angedockte Symbolleisten und Fenster. So das Arbeitsbereichsfenster ganz links, in dem sind alle zum Projekt gehörigen Dateien, Klassen und bei MFC-Anwendungen Ressourcen aufgelistet. Daneben befindet sich das Quellcodefenster in das der C++ Quelltext eingegeben wird oder die Ressourcen angezeigt werden. In den meisten Fällen ist außerdem das Debugfenster zu sehen, das einem bei der Kompilierung (Übersetzung des Quelltextes in Maschinsprache) Fehler mitteilt.

Über diesen Fenstern befinden sich die Symbolleisten, eine für Einstellungen für Klassen, eine für das Starten der Anwendung, Die oberste Leiste umfasst die Funktionen speichern, öffnen und suchen.



Das Arbeitsbereichsfenster

Das Arbeitsbereichsfenster besteht aus 2-3 Registerkarten (Klassen-, Dateien- und Ressourcenansicht). Hier kannst du auswählen welche Datei, Klasse oder Ressource du bearbeiten willst. Die ausgewählte Datei, Klasse oder Ressource erscheint dann im Hauptfenster (neben dem Arbeitsbereichsfenster).

Es lässt sich streiten welche Ansicht sich am besten für die Arbeit mit Visual C++ eignet, jedoch vereinfacht so manche Anzeige die Arbeit mit den verschiedenen Dateien erheblich.

Das Hauptfenster

In dem Hauptfenster wird der Quelltext der Datei oder Klasse angezeigt und kann dort auch bearbeitet werden, gleiches gilt für die Ressourcen von MFC-Anwendungen.

Ist im Arbeitsbereichsfenster die Ressourcenansicht aktiviert, so kann man im Hauptfenster neben dem Quelltext auch die Ressourcendateien des Programms bearbeiten, wie ich schon erwähnt habe. Will man ein Steuerelement auf der Form platzieren, so klickt man das Symbol an und zieht es dann auf der Form auf. Also Symbol anklicken, mit Mauszeiger zur Form gehen, Maustaste klicken und gedrückt halten und ein Quadrat aufziehen, dann wieder los lassen und fertig ist das Steuerelement. Ist die Größe nicht perfekt oder ist die Position falsch, so kann man das Steuerelement auch noch verändern. Zum Verschieben auf das Steuerelement klicken, Maustaste gedrückt halten und an die korrekte Position verschieben. Zum Verkleinern Steuerelement anklicken, dann an der rechten unteren Ecke anklicken und bei gedrückter Maustaste in die richtige Größe verkleinern. Die Eigenschaften verstecken sich etwas, man findet sie im Kontextmenü; also Rechtsklick auf das eingefügte Steuerelement und Eigenschaften auswählen. Doch so viele Eigenschaften wie beispielsweise in Visual Basic und Delphi wird man hier nicht finden.

Ausgabefenster

Das Ausgabefenster leistet einem bei der Kompilierung sehr gute Dienste. Denn es zeigt was gerade durchgeführt wird, und wenn Fehler aufgetreten sind zeigt es diese an und auch die Stelle wo der Fehler aufgetreten ist. Da man sehr einfach Fehler in C++ Programme einbauen kann (verflucht seien diese Semikolons (;)), wird einem so viel Arbeit erspart, wobei man manchmal auch die Zeilen oberhalb der Fehlerstelle untersuchen sollte, da manchmal der Fehler erst im späteren Programmverlauf zum Tragen kommt.

Wichtige Funktionen

Wenn du dich in der Ressourcen-Ansicht befindest und Dialoge bearbeitest, kannst mit dem Menüpunkt "Projekt/Dem Projekt hinzufügen/Komponenten und Steuerelemente..." neue Steuerelemente in das Steuerelemente-Fenster deines Dialogs einbinden.

Mit dem Ausrufezeichen in der Symbolleiste kannst du das Programm kompilieren lassen und danach ausführen lassen. Die Anwendung ist danach schon als .exe-Datei in dem Projektordner aufzufinden, und kann so auch ohne die IDE ausgeführt werden.

Hilfe

Auf der msdn-Hilfe-CD-ROM findest du zu jedem Element eine meist ausführliche Beschreibung. Gehe dafür, wenn du in der msdn-Hilfe bist (In VC++ über Hilfe (?) aufrufbar), auf den Registerreiter "Suchen" am linken Bildschirmrand. Es werden

dann die Treffer aufgelistet.

Weitergabe

Wenn du keine zusätzlichen Steuerelemente in dein Projekt integriert hast, dann ist die Datei auch auf anderen Computern ohne zusätzliche Dateien lauffähig. Die Ausnahme bildet hier die Datenbankprogrammierung, die noch eine zusätzliche Weitergabe von Datenbankdateien zum Programm erfordert, doch darauf will ich an dieser Stelle nicht weiter eingehen.

Eingeschränkt gilt dies auch für die MFC-Programme, denn eigentlich müsste man auch hier Dateien mitliefern, jedoch ist auf den meisten PC bereits eine Version der MFC-Laufzeitdateien.

Um den Umfang der Programme zu reduzieren, sollte man, wenn das Programm fertig gestellt und fehlerfrei ist, das Programm mit der Einstellung der Compilereinstellung Release nochmals kompilieren. Mit dieser unter "Erstellen/Aktive Konfiguration festlegen.../[Projekt]-Win32 Release" zu findenden Einstellung werden überflüssige Debug-Befehlszeilen entfernt und das Programm ist kleiner und schneller lauffähig.

Programmierung mit C++: Begriffserklärungen

Da wäre die Fähigkeit der 'Vererbung', diese Technik macht es möglich, Funktionen und Klassen in dem aufeinander folgenden Quelltext wieder verwenden zu können. So kann man beispielsweise aus einer Fenster-Anzeige-Funktion die Fähigkeit des Zeichnens verwenden.

Da wäre darüber hinaus die Fähigkeiten von 'Klassen', diese Fähigkeit macht es möglich, mehrere Funktionen und Objekte in einem Überobjekt zusammen zufassen. So kann man diese leichter vererben und sie sind zentral zusammengefasst. Die MFC ist eine Sammlung vieler Klassen.

Die Fähigkeit 'Kapselung' macht es möglich, dass man bestimmte Variablen und deren Inhalte nur für Klassen frei gibt, die für den Zugriff zugelassen sind; so können die Inhalte nicht in falsche Hände fallen.

Mit der Fähigkeit 'Polymorphie' kann man die Fähigkeiten der vererbten Funktionen oder Objekte verändern, sozusagen deren Charakter.

Warum muss ich C++ können

Da man mit Visual C++ auch "normales" C++ programmieren kann möchte ich ein kleines C++-Programm schreiben. Denn die MFC, die Visual C++ zur visuellen Programmiersprache macht, ist nur eine Klassenbibliothek. Mit dem kleinen Programm kannst du erste Erfahrungen in der Programmierung mit C++ sammeln. Es handelt sich um ein DOS-basierendes Zinsberechnungsprogramm.

Neues Projekt anlegen

Starte VC++ (Visual C++) wie gewohnt und gehe dann auf den Menüpunkt "Datei/Neu", nun erscheint ein Fenster mit 4 Registerreitern, stelle sicher, dass der Registerreiter "Projekte" angewählt ist. Wähle dann Win32-Konsolenanwendung. Im rechten Bereich des Fensters befindet sich ein Texteingabefeld für den Dateinamen, gib den gewünschten Projektnamen ein (z.B. Zinsen). Darunter befindet sich ein Feld für die Pfadauswahl, wähle dort das richtige Verzeichnis aus. Klicke dann auf "OK", der Assistent fragt dich jetzt, welche Art von Konsolenanwendung erstellt werden soll. Wähle "Eine einfache Anwendung" aus und klicke auf "Fertigstellen" um das Projekt zu erstellen. Nun werden dir noch "Informationen zum neuen Projekt" angezeigt, klicke auf "OK" um die Erstellung abzuschließen. Nach dem kurzen Erstellungsvorgang

kann es jetzt zur Programmierung gehen.

Welche Ansicht ist die beste zum C++-Programmieren?

Ich persönlich arbeite am liebsten in der Dateienansicht, deshalb wähle den Registerreiter für die Dateienansicht aus. Klicke auf das Pluszeichen neben *Zinsen* Dateien, wobei *Zinsen* für den Projektnamen steht. Es öffnet sich ein Baum aus Dateiartern, die das Projekt enthält. Klicke auf das Pluszeichen neben "Quellcode-Dateien", es öffnet sich ein weiterer Baum aus den Quellcodedateien. Doppelklicke nun auf "*Zinsen.cpp*", wobei *Zinsen* für den Projektnamen steht. Die Datei öffnet sich als Quellcodeansicht, nun geht es an das Einfügen von Quellcode.

Der Quelltext

Ich werde erst mal den gesamten Quelltext abdrucken und anschließend die einzelnen Anweisungen beschreiben (Nur die fett geschriebenen kursiven Quelltextfragmente übernehmen!):

```
// Zinsen.cpp : Definiert den Einsprungpunkt für die Konsolenanwendung.
//
#include "stdafx.h"
#include <iostream.h>
int main(int argc, char* argv[])
{
    float Kapital;
    float Zinssatz;
    float Tage;
    float Ergebnis1;
    float Ergebnis2;
    int End;
    cout << " |-----|" << "\n";
    cout << " |ZINSBERECHNUNGSPROGRAMM|" << "\n";
    cout << " |-----|" << "\n";
    cout << "" << "\n";
    cout << " Willkommen beim Zinsberechnungsprogramm!" << "\n";
    cout << "" << "\n";
    cout <<
    "
    _____
    " << "\n";
    cout << "" << "\n";
    cout << "Geben Sie das Kapital ein (in Euro): "; cin >> Kapital;
    cout << "Geben Sie den Zinssatz ein (in %): "; cin >> Zinssatz;
    cout << "Geben Sie die Tage ein: "; cin >> Tage;
    cout << "" << "\n";
    Ergebnis1 = Kapital * Zinssatz * Tage;
    Ergebnis2 = Ergebnis1 / 36000;
    cout <<
    "
    _____
    " << "\n";
    cout << "" << "\n";
    cout << "Das Ergebnis sind " << Ergebnis2 << " Euro" << "\n";
    cout << "" << "\n";
}
```

```

cout << "B drücken zum Beenden "; cin >> End;
cout << "" << "\n";
return 0;
}

```

Was bedeutet der Quelltext?

Die erste Anweisung (`#include <iostream.h>`) bedeutet folgendes: Mit `#include` bindet man Header-Dateien, die wichtige Deklarationen für Funktionen, Objekte und Klassen enthalten, die für die Quelltextdatei benötigt werden, ein. `iostream.h` ist eine Header-Datei, die Funktionen zur Aus- und Eingabe von Text auf dem Bildschirm beinhaltet.

Bei `float` handelt es sich um einen C++ Datentyp der Gleitkommazahlen (Zahlen mit Nachkommastellen) aufnimmt. Mit `float Kapital` wurde die Variable `Kapital` deklariert, d.h. festgelegt. Außer dem Datentypen `float` gibt es noch `int`, `long`, `short` (Integer-Datentypen); `char` (Zeichenfolgen); `bool` (Ja-Nein); `double` und `long double` (Gleitkommazahlen). Die Variablen werden alle wie `float` deklariert (z.B. `long Var`). Damit man den Datentyp der Variable besser wieder erkennt, werden folgende Präfixe verwendet: `b` (= `bool`), `c` oder `ch` (= `char`), `d` oder `dbl` (= `double`), `f` oder `fl` (= `float`), `L` oder `l` (= `long`) und `n` oder `i` (= `integer`).

Mit `cout` wird Text auf dem Bildschirm ausgegeben. Dabei ist `cout` folgendermaßen anzuwenden: `cout << " [Text] " << "\n";`. "`\n`" heißt Zeilenumbruch, d.h. nach dieser Anweisung wird der Cursor eine Zeile tiefer gelegt. In manchen `cout`-Anweisungen taucht noch `cin` auf, während `cout` für die Ausgabe zuständig ist, ist `cin` für die Eingabe zuständig. `cout` schreibt die Eingabebeschreibung und `cin` wartet dann auf die Eingabe die mit [ENTER] abgeschlossen wird. Die Eingabe wird dann an die Variable (`>> Kapital`) übergeben und gespeichert.

Achte beim Abtippen auf das Abschließen der Zeilen mit einem Semikolon!

Programm ausführen

Nun kannst du das Programm ausführen, sollten im Debug-Fenster während der Kompilierung Fehler auftreten, und du weißt den Grund dafür nicht, untersuche, ob du auch alle Anweisungen mit dem Strichpunkt (Semikolon) (;) beendet hast!

Visual C++ - Projekt erstellen

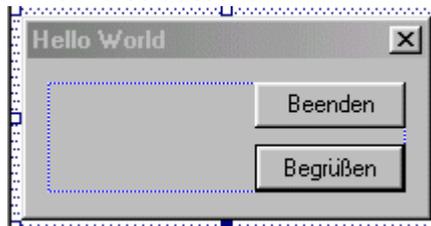
Starte VC++ (Visual C++) wie gewohnt und wähle dann "Datei/Neu" aus, stelle dann sicher, dass der Registerreiter "Projekte" ausgewählt ist, wähle dann den Menüpunkt "MFC-Anwendungs-Assistent (exe)". Wähle dann unter Pfad rechts im Fenster den Pfad für die Anwendung aus, trage dann den Projektnamen in das Feld für den "Projektnamen" ein. Bestätige dann mit "OK". Der Anwendungsassistent ist in 4 Schritte eingeteilt, die man schrittweise durcharbeiten muss. In dem ersten Schritt wird die Art der Anwendung bestimmt, markiere den Menüpunkt "Dialogfeldbasierend" aus, in dem du auf den Radio-Button daneben klickst, lasse das Fenster ansonsten unberührt. Drücke auf "Weiter >" um einen Schritt weiter zu gelangen. Bei dem zweiten Schritt gib unter "Geben Sie bitte einen Titel für Ihr Dialogfeld ein: 'Hello World'" ein. Sonst muss nichts verändert werden. Sodass man getrost "Weiter >" gehen kann. Bei den nächsten Schritten muss nichts verändert werden, also lasse alles unverändert und arbeite dich bis zum vierten Schritt vor. Hier wird dir angezeigt, welche Klassen alle für dich erstellt werden. Klicke auf "Fertigstellen" um die Erstellung des Projektes abzuschließen. Es werden dir noch "Informationen zum neuen Projekt" gegeben. Mit einem Klick auf "OK" leitest du die Erstellung ein. Dieser Vorgang kann etwas

dauern, da einige Dateien für dich geschrieben werden müssen. Diese enthalten dann das Grundgerüst für deine Anwendung, d.h. die Fenster.

Form-Design

Es öffnet sich automatisch die "Ressourcen-Ansicht" mit dem Dialogfeld. Klicke auf den Text "ZU ERLEDIGEN: ..." um ihn zu markieren und drücke dann auf [Entf] um den Text zu löschen. Klicke auf den Button mit der Aufschrift "OK" um ihn zu markieren, klicke dann mit der rechten Maustaste auf den Button und wähle im Kontextmenü "Eigenschaften" aus. Ändere in dem Fenster dann den Titel von "OK" in "Beenden" um. Wähle dann den Registerreiter "Formate" aus und entferne durch einen Klick darauf das Häkchen neben "Standartschaltfläche". Diese Eigenschaft legt fest, ob die Schaltfläche beim Drücken der [ENTER]-Taste ausgelöst wird. Klicke dann außerhalb des Eigenschaftsfensters oder drücke [ENTER] um das Fenster zu schließen. Markiere den "Abbrechen"-Button und lösche ihn dann. Wähle dann in dem Steuerelemente-Fenster das Steuerelement Schaltfläche () und zeichne einen Button der etwa die gleiche Größe wie der "OK"-Button hat. Markiere den neuen Button und öffne mit Hilfe der rechten Maustaste das Kontextmenü indem du den Menüpunkt "Eigenschaften" auswählst. Ändere in dem Fenster den Titel in "Begrüßen". Wechsel dann in den Registerreiter "Formate" und mache neben "Standartschaltfläche" ein Häkchen. Dieser Button soll ausgelöst werden, wenn die [ENTER]-Taste gedrückt wird.

Ändere die Größe des Dialogfeldes, indem du auf eine leere Fläche am Rand des Dialogfeldes klickst. Es erscheinen Ziehpunkte, mit denen du die Größe ändern kannst. Das Dialogfeld sollte dann in etwa so aussehen:



Das Coding

Nun geht es an das Einfügen von Quelltext, damit das Programm auch etwas macht, wenn man auch den "Begrüßen"-Button klickt. Doppelklicke auf den "Begrüßen"-Button, es erscheint eine Eingabeaufforderung, behalte die Voreinstellung von "OnButton1" bei und klicke auf "OK". VC++ springt nun automatisch in die Quelltext-Ansicht, an den Punkt, an den der Funktionsaufruf "OnButton1" eingefügt wurde. Gebe nun unter das Kommentar "// TODO: ..." die Quelltextzeile `AfxMessageBox("Hello World");` ein. Diese Zeile bewirkt, dass beim Drücken der "Begrüßen"-Taste ein Meldungsfeld mit der Aufschrift "Hello World" erscheint.

**Diesen und viele andere Workshops gibt es auf
www.abbyter.de**