

Anleitung: Erstellen von C++-Konsolenanwendungen mit Visual Studio Community 2017 und 2019

Visual Studio Community 2019 kann kostenlos heruntergeladen und installiert werden (Link: <https://www.visualstudio.com/de/>). Um allerdings dauerhaft damit zu arbeiten, muss man sich bei Microsoft registrieren. Nach der Installation ist es sinnvoll, ein Desktop-Icon für Visual Studio anzulegen.

1. Visual Studio starten und neues Projekt anlegen

- Visual Studio durch Doppelklicken des Visual-Studio-Icons auf starten.
- Im Menü "Datei" auf "Neues Projekt/" klicken (alternativ auf das "Neues Projekt"-Icon).
- Im neu geöffneten Fenster links unter "**Visual C++**" den Unterpunkt "Allgemein" anklicken, dann in der Mitte "**Leeres Projekt**" auswählen. Unten einen Projektnamen und einen Speicherort eingeben, dann auf OK klicken. *Hinweis:* Unten kann man mit Hilfe eines Häkchens angeben, ob das Projekt innerhalb eines Projektmappenverzeichnisses erstellt werden soll oder nicht.

Ein leeres Projekt ist jetzt angelegt. Im Teilfensters "Projektmappen-Explorer" erscheint eine Projektübersicht (Externe Abhängigkeiten, Headerdateien, Quelldateien, Ressourcendateien).

- Falls der Projektmappen-Explorer nicht angezeigt wird: Im Menü "Ansicht" auf "Projektmappen-Explorer" klicken.

2. Programm-Quelltext editieren

Das angelegte Projekt enthält bisher noch keinen Quelltext.

- Im Projektmappen-Explorer mit der rechten Maustaste auf Quelldateien klicken, "Hinzufügen" auswählen und dann auf "Neues Element" klicken.
- Im neu geöffneten Fenster "**C++-Datei**" auswählen, unten einen Namen eingeben und dann auf "Hinzufügen" klicken.

Im Hauptfenster wurde jetzt ein Editor-Tab mit der neu hinzugefügten Quelldatei geöffnet.

- Man kann jetzt den Quelltext eintippen oder den Quelltext oder Teile davon von irgendeiner anderen Stelle kopieren und im Editor einfügen.

3. Existierende Quelltext-Dateien zum Projekt hinzufügen

Falls schon Quelltext-Dateien existieren, die man zum Projekt hinzufügen möchte, kann man folgendermaßen vorgehen:

- Mit dem Explorer die existierenden Quelltext-Dateien in das Projekt-Quellverzeichnis kopieren. Dieses Verzeichnis kann man sehen, wenn man mit der Maus über den Namen der Quelldatei aus Punkt 2 in dem Tab des Editors fährt. Wir gehen hier als Beispiel von den beiden existierenden Quelltext-Dateien GrafClient.h und GrafClient.cpp aus.
- Zum Hinzufügen einer Headerdatei (z. B. GrafClient.h) klickt man im Projektmappen-Explorer mit der rechten Maustaste auf Headerdateien und wählt "Hinzufügen" aus, dann "Vorhandenes Element". Es erscheint ein Fenster, in dem man die Datei auswählen und dann "Hinzufügen" klicken kann. (Es ist hier auch möglich, mehrere Dateien auszuwählen.) Die hinzugefügte Datei wird dann im Projektmappen-Explorer unter Headerdateien angezeigt.
- Zum Hinzufügen einer C++-Datei (z. B. GrafClient.cpp) klickt man im Projektmappen-Explorer mit der rechten Maustaste auf Quelldateien und wählt "Hinzufügen" aus, dann "Vorhandenes Element". Es erscheint ein Fenster, in dem man die Datei auswählen und dann "Hinzufügen" klicken kann. (Es ist hier auch möglich, mehrere Dateien auszuwählen.) Die hinzugefügte Datei wird dann im Projektmappen-Explorer unter Quelldateien angezeigt.

- Es ist auch möglich, gleichzeitig Header- und C++-Dateien zum Projekt hinzuzufügen. Dazu klickt man im Projektmappen-Explorer mit der rechten Maustaste auf den Projektnamen und wählt "Hinzufügen", dann "Vorhandenes Element". Es erscheint ein Fenster, in dem man alle gewünschten Dateien (mit Endung .h und .cpp) auswählen kann. Nach Klicken von "Hinzufügen" werden die ausgewählten Dateien im Projektmappen-Explorer an der richtigen Stelle angezeigt (je nach Endung unter Headerdateien oder Quelldateien).

4. Programm übersetzen und starten

Hat man einen gültigen Quelltext eingegeben oder kopiert und ggf. noch andere Quelldateien zum Projekt hinzugefügt, kann man das Programm übersetzen und starten. Einfachste Möglichkeit:

- Auf den kleinen grünen Pfeil (Play-Button) in der Toolbar klicken (Beschriftung: Lokaler Windows-Debugger) oder man wählt im Menü "Debuggen" den Punkt "Debugging starten". Alternativ kann man die F5-Taste drücken. Es erscheint dann ggf. ein kleines Fenster mit der Frage: "Das Projekt ist veraltet. Erstellen?". Hier muss man auf "**Ja**" klicken.
- Falls eine Meldung "Fehler beim Erstellen. ..." kommt, wird abgefragt, ob man die letzte fehlerfreie Version starten möchte. Hier sollte man auf "**Nein**" klicken. In diesem Fall liegt ein Syntax- oder Linkerfehler vor und man muss sich die Fehlermeldungen (unten) ansehen und den Fehler im Quelltext beseitigen.

Das Programm wird nun, falls keine Fehler aufgetreten sind, kompiliert und gestartet.

5. Projekt speichern

Explizites Speichern eines Projekts ist normalerweise nicht notwendig, da das Projekt automatisch beim Kompilieren gespeichert wird. Um eine C++-Datei xyz zu speichern kann man auf den Reiter der Datei rechtsklicken und "xyz.cpp speichern" auswählen, oder dies im Datei-Menü auswählen. Im Datei-Menü kann man auch alle Dateien unter "Alle Speichern" speichern.

6. Abgespeichertes Projekt öffnen


Um ein Projekt öffnen, öffnet man am besten die Projektmappe:


- Dazu geht man im Menü "Datei" auf "Öffnen" und dann auf "Projekt/Projektmappe". Es öffnet sich ein Fenster zur Dateiauswahl. Es können jetzt im entsprechenden Projektverzeichnis Projektmappendateien (Endung .sln) geöffnet werden.
- War ein Projekt schon einmal geöffnet, kann man im Menü "Datei" auf "Öffnen" und dann auf "Zuletzt geöffnete Projekte und Projektmappen" klicken. Es wird dann eine Liste mit den entsprechenden .sln-Dateien angezeigt und man kann die entsprechende Projektmappe durch Anklicken öffnen.

In allen Fällen wird beim Öffnen einer Projektmappe die aktuell geöffnete Projektmappe geschlossen. Es ist aber auch möglich, mehrere Instanzen von Visual Studio zu öffnen und dadurch mehrere Projektmappen geöffnet zu haben.

7. Alternative Quelldateien in einem Projekt

Möchte man verschiedene Versionen eines Programms in einem Projekt haben und diese in verschiedenen Quelldateien speichern, so gibt es die Möglichkeit, schnell einzelne Quelldateien zum Projekt hinzuzufügen oder aus dem Projekt auszuschließen.

- Man klickt in der Toolbar des Projektmappen-Explorers auf das Icon  ("Alle Dateien anzeigen"). Dadurch werden jetzt nicht mehr nur die Header- und Quelldateien angezeigt, die dem Projekt hinzugefügt wurden (d. h. aktiv sind), sondern alle Dateien, die im Projektverzeichnis vorhanden sind. Die aktiven Dateien haben ein C++- oder h-Icon vor dem Namen, die nicht aktiven Dateien ein "Einfahrt verboten"-Icon.
- Jetzt besteht die Möglichkeit, mit einem Rechtsklick auf eine der angezeigten Dateien diese zum Projekt hinzuzufügen (wenn sie nicht aktiv ist) oder aus dem Projekt auszuschließen (wenn sie aktiv ist).

- Man kann jederzeit durch einen Klick auf das Icon  ("Alle Dateien anzeigen") den Projektmappen-Explorer wieder auf die normale Ansicht mit Header- und Quelldateien umschalten.

8. Bei Programmabbruch

Bricht das laufende Programm mit einer Fehlermeldung ab (Laufzeitfehler), so wird das Programm an der Stelle angehalten, an der der Fehler auftrat. Nach Klick auf "Unterbrechen" muss der Debug-Modus beendet werden, indem man im Menü "Debuggen" auf "Debugging beenden" klickt, oder einfacher: in der Toolbar auf das kleine rote Quadrat (Stop-Button) klickt.