

Jürgen Wolf

Aktuell
zu
C11

Grundkurs C

- ▶ Eine kompakte Einführung in die Programmiersprache C
- ▶ Vom ersten Schritt bis zum komplexen Programm
- ▶ Mit Übungen und Musterlösungen zur Lernkontrolle

2., aktualisierte Auflage

 Rheinwerk
Computing

1.4 Übersetzen mit der Entwicklungsumgebung

Sobald Sie Ihre Entwicklungsumgebung heruntergeladen und installiert haben, können Sie ein erstes Projekt anlegen und aus einem einfachen Quelltext ein ausführbares Programm erstellen. Ich zeige Ihnen beides anhand von Pelles C, aber der Vorgang ist in der Regel bei anderen Entwicklungsumgebungen vergleichbar – nur lautet hier und da ein Befehl anders oder findet sich an einer anderen Stelle.

1. Am einfachsten ist es, wenn Sie ein neues Projekt anlegen, weil sich dann die Entwicklungsumgebung gleich um viele kleinere Details kümmert und Sie sich leichter tun, wenn Sie später weitere Dateien zum Projekt hinzufügen wollen. Ein neues Projekt starten Sie mit **DATEI • NEU • PROJEKT**.

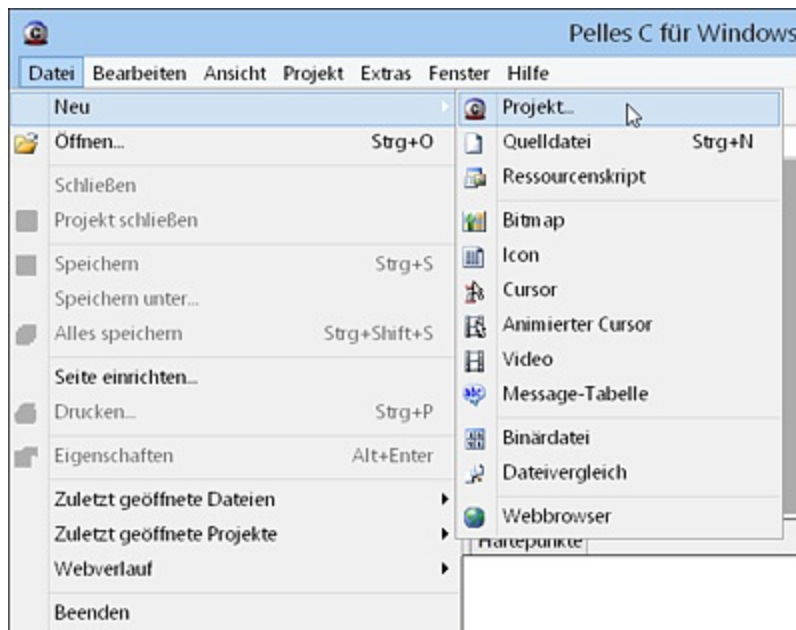


Abbildung 1.3 Zunächst sollten Sie ein neues Projekt anlegen.

2. Gewöhnlich folgt jetzt ein Dialog mit der Abfrage, welche Art von Projekt Sie erstellen wollen, wie sein Name lauten und wo es gespeichert werden soll. Häufig ist es am einfachsten, ein leeres C-Projekt oder (wie hier) eine Konsolenanwendung (hier: Win64-Konsolenanwendung (EXE)) zu erzeugen. Wird kein passendes Projekt als Vorlage aufgelistet, bieten Entwicklungsumgebungen häufig auch an, entsprechende Vorlagen herunterzuladen.

3. Abhängig von der Entwicklungsumgebung finden Sie vielleicht schon ein C-Grundgerüst mit Quellcode vor. Bei anderen Entwicklungsumgebungen müssen Sie erst noch ein neues Element bzw. eine neue Datei hinzufügen. Neue Elemente bzw. Dateien werden Sie auch hinzufügen müssen, wenn Sie den Quellcode in mehrere Module aufteilen. Hierbei ist immer darauf zu achten, dass Sie die neue Datei zum Projekt hinzufügen und nicht einfach nur eine neue, leere Datei anlegen.

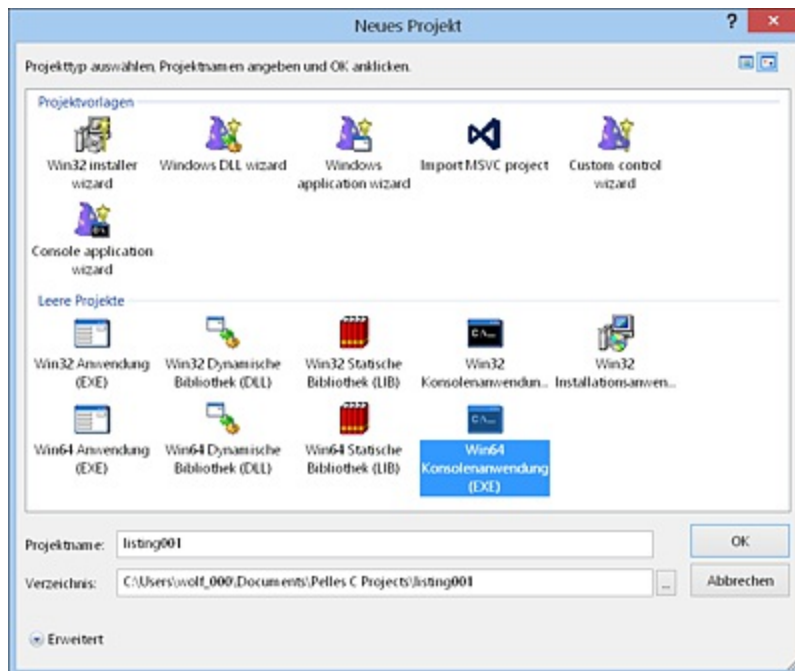


Abbildung 1.4 Normalerweise hilft Ihnen ein Assistent dabei, das Projekt zu erstellen.

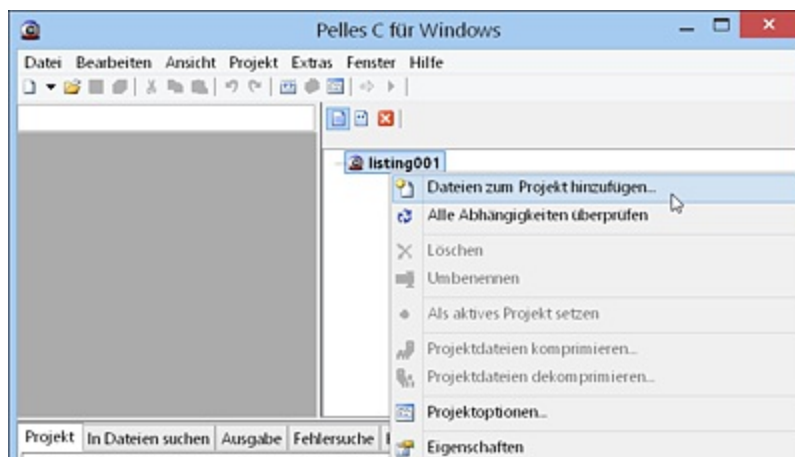


Abbildung 1.5 Eine neue Datei zum Projekt hinzufügen

4. Häufig hilft Ihnen auch hier ein weiterer Dialog, eine Datei zum Projekt hinzuzufügen. Bei Ihren ersten Beispielen werden Sie ohnehin eine

Quelldatei mit der Endung *.c hinzufügen wollen. Bei späteren Projekten werden Sie des Öfteren neben einer C-Datei auch eine eigene Headerdatei mit der Endung *.h hinzufügen. Bei Pelles C müssen Sie nur den Dateinamen (hier: *listing001.c*) eintippen und auf die Schaltfläche **ÖFFNEN** klicken.

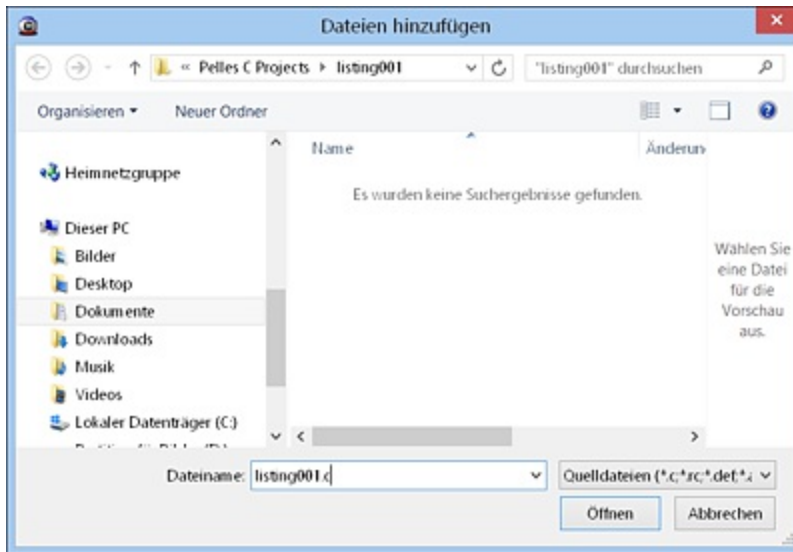


Abbildung 1.6 Den Dateinamen für die neue Projektdatei eingeben

5. Sobald Sie die Datei zum Projekt hinzugefügt haben, können Sie anfangen, den Quelltext einzutippen. Nach dem Abspeichern des Quellcodes können Sie diesen übersetzen (kompilieren) und ausführen. Bei Pelles C können Sie den Quelltext über **PROJEKT • KOMPILIERE LISTING001.C** nur kompilieren, über **PROJEKT • ERZEUGE LISTING001.EXE** kompilieren, linken und ein ausführbares Programm daraus machen oder ihn mit **PROJEKT • LISTING001.EXE AUSFÜHREN** kompilieren, linken und auch gleich ausführen.

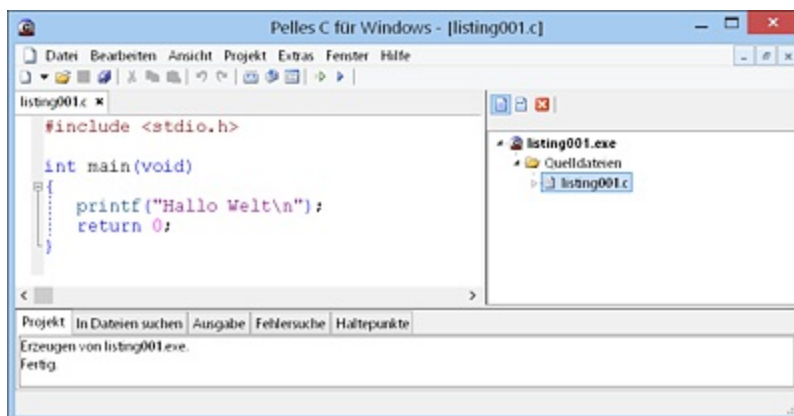


Abbildung 1.7 Quelltext eintippen, dann das Programm übersetzen und starten

1.5 Übersetzen mit gcc und clang

Die Verwendung von Kommandozeilen-Compilern wie `gcc` oder `clang` soll hier auch kurz beschrieben werden. Hierbei schreiben Sie Ihren Quelltext mit einem beliebigen ASCII-Editor und speichern ihn mit der Endung `*.cpp` ab. In der Kommandozeile übersetzen Sie den Quelltext dann mit `gcc` oder `clang`. Gerade bei kleineren Listings wie in diesem Buch ist dieser Vorgang weniger aufwendig, als den Quelltext mit einer Entwicklungsumgebung zu übersetzen.

Dabei gehe ich davon aus, dass Sie die Kommandozeile kennen und sich mit unterschiedlichen Kommandos durch die Verzeichnisse bewegen können. Wenn Sie den Quelltext geschrieben und abgespeichert haben, öffnen Sie die Kommandozeile, und wechseln Sie in das Verzeichnis, in dem Sie den Quelltext gespeichert haben. Tippen Sie Folgendes ein (*listing001.c* sei der abgespeicherte C-Quelltext):

```
$ gcc -o listing001 listing001.c
$ ./listing001
...
```

Mit dem Schalter `-o` wird hier veranlasst, dass der Compiler aus der Quelldatei *listing001.c* die ausführbare Datei *listing001* macht. Sie können statt *listing001* natürlich auch einen anderen Programmnamen verwenden. In der folgenden Zeile wird daraufhin die ausführbare Datei in der Kommandozeile mit dem Programmnamen gestartet.

In der Praxis empfiehlt es sich aber, mehrere solcher Flags wie `-o` zu verwenden, um dem Compiler bei einer Warnung oder Fehlern mehr Informationen zu entlocken. Meine minimale Zeile zum Übersetzen eines Quelltextes lautet daher:

```
$ gcc -Wall -o listing001 listing001.c
$ ./listing001
...
```

Der Schalter `-Wall` gibt mir alle sinnvollen Warnungen des Compilers aus. *Wall* setzt sich aus **Warning** und **all** zusammen. Sollten Sie den Quellcode nur kompilieren (und nicht zusätzlich linken) wollen, müssen Sie statt `-o` den Schalter `-c` verwenden.

Schalter für C99 und C11

Wenn der Compiler nicht schon von Haus aus den C11-Standard verwendet, können Sie diesen mit dem Flag `-std=c11` aktivieren. Dasselbe gilt auch für

den vorherigen C99-Standard, den Sie mit `-std=c99` aktivieren können –
beispielsweise:

```
$ gcc -Wall -o prgname prgname.c -std=c11  
$ ./prgname  
...
```