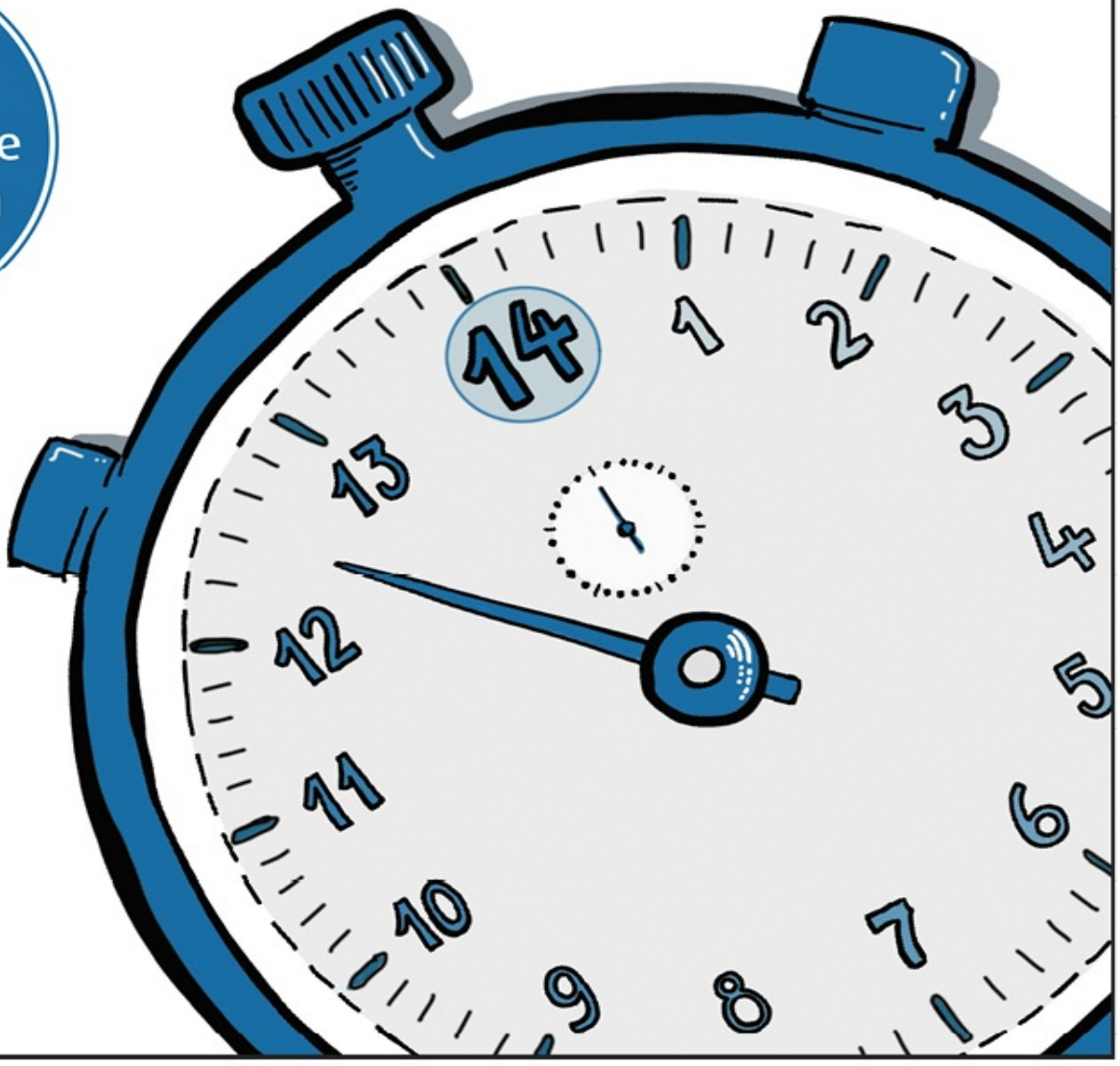
Philipp Hasper

C++ Schnelleinstieg

Programmieren lernen in 14 Tagen

++ Einfach und ohne Vorkenntnisse ++

Zahlreiche Praxisbeispiele und Übungen



mitp

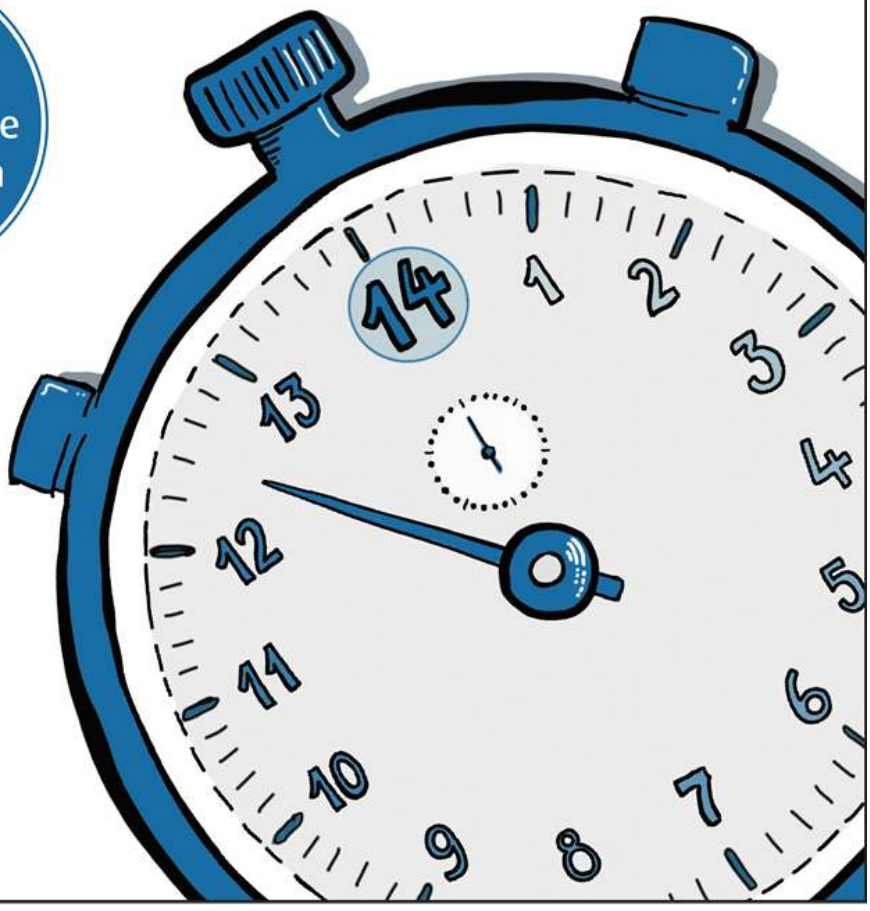
Philipp Hasper

C++ Schnelleinstieg

Programmieren lernen in 14 Tagen

++ Einfach und ohne Vorkenntnisse ++

Zahlreiche Praxisbeispiele und Übungen



mitp

Philipp Hasper

C++

Schnelleinstieg

Programmieren lernen in 14 Tagen
Einfach und ohne Vorkenntnisse

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN ISBN 978-3-7475-0324-9
1. Auflage 2021

www.mitp.de

E-Mail: mitp-verlag@sigloch.de
Telefon: +49 7953 / 7189 - 079
Telefax: +49 7953 / 7189 - 082

Lektorat: Janina Bahlmann
Sprachkorrektorat: Christine Hoffmeister
Covergestaltung: Janina Bahlmann, Christian Kalkert
Covergrafik & Icons: Tanja Wehr, sketchnotelovers
Abbildungen & Grafiken: Michael Weigend
Electronic Publishing: Petra Kleinwegen

Dieses Ebook verwendet das ePub-Format und ist optimiert für die Nutzung mit dem iBooks-reader auf dem iPad von Apple. Bei der Verwendung anderer Reader kann es zu Darstellungsproblemen kommen.

Inhalt

Einführung

1.1 Einleitung

1.1.1 Programmieren lernen in 14 Tagen

Mit diesem Buch erhalten Sie einen praktischen Einstieg in das Programmieren mit C++. Sicherlich endet Ihre Lernkurve nicht mit der letzten Seite, aber bis dahin werden Sie Einblicke in die verschiedenen Funktionalitäten, Stärken und Fallstricke der Sprache bekommen haben. Dieses Buch bringt Ihnen nicht nur reines C++ bei, sondern zeigt Ihnen auch, wie Sie es auf einfache Weise mit Open-Source-Bibliotheken erweitern, sodass Sie grafische Benutzeroberflächen erstellen oder mit Internetservern kommunizieren können. Es endet mit einem Ausblick darauf, wie Sie Ihr gewonnenes Wissen im Programmieralltag einsetzen und selbstständig erweitern können.

Wenn Sie Zeit genug haben, können Sie jeden Tag ein neues Kapitel durcharbeiten und so innerhalb von zwei Wochen Programmieren lernen. Alle Erklärungen sind leicht verständlich und setzen keine Vorkenntnisse voraus. Sie werden schnell erste Erfolge erzielen und Freude an der Programmierung finden.

1.1.2 Wie man dieses Buch liest

Dieses Buch gibt Ihnen einen Einblick in C++, und zwar in der aktuell verfügbaren Version C++17. Die Unterschiede der verschiedenen Versionen sind für dieses Buch nicht sonderlich relevant – die meisten hier verwendeten Konzepte gibt es schon seit C++11, welches meiner Meinung nach ein Quantensprung für die Modernisierung von C++ war.

Insbesondere Universitäten oder berufliche Fortbildungen beginnen das Lehren von C++ oft über den Vorgänger C und dessen sehr hardwarenahen Konzepte. Dieses Buch verfährt jedoch andersherum: Sie als Leser sollen ohne Vorkenntnisse in die Lage versetzt werden, möglichst schnell erste Resultate zu erreichen. Dafür ist es nicht relevant, sich von Tag 1 an mit dem Speichermanagement und anderen Grundlagen auseinanderzusetzen. Der Vollständigkeit halber werde ich diese Dinge auch ansprechen, allerdings erst in den späteren Kapiteln.

Verweise auf Kapitel weiter hinten im Buch werden Ihnen häufiger begegnen. Dies ist zweierlei Dingen geschuldet

- Sie sollen einen Ausblick bekommen, in welche Richtungen das bereits Erlernte noch erweitert wird.
- Einige Konzepte bedingen sich gegenseitig, dennoch möchte ich sie Ihnen der Reihe nach präsentieren.

Lassen Sie sich daher nicht davon beirren, es ist ein bisschen wie beim Lernen einer normalen Sprache: Das Erlernen der Grammatik ist schwer, ohne Vokabeln zu kennen, reines Vokabellernen ohne die Anwendung macht aber auch keinen Spaß. Ich empfehle Ihnen, sich nicht den Anspruch aufzuerlegen, jedes Programmbeispiel auf Anhieb Zeichen für Zeichen zu verstehen, sondern im Zweifel Dinge erst mal hinzunehmen und gegebenenfalls später wiederaufzugreifen.

Am Ende eines jeden Kapitels finden Sie kleine Übungsaufgaben, mit denen Sie Ihr Wissen praktisch anwenden und vertiefen können. Manchen Übungen ist ein relativ eindeutiger Lösungshinweis beigelegt – dieser wird dann auf dem Kopf geschrieben, sodass Sie auch erst ohne ihn versuchen können, die Aufgabe zu lösen. Genauso finden Sie auch kurze Zwischenfragen innerhalb der Kapitel, deren Antwort ebenfalls auf dem Kopf geschrieben ist.

Neben den Übungsaufgaben möchte ich Sie ermuntern, auch schon innerhalb der Kapitel ein bisschen herumzuprobieren und den Pfad des Buches mal links und rechts zu verlassen. Ein Hinweis dazu: Leider ist C++ nicht immer besonders experimentierfreudig, sodass gerade in den ersten Kapiteln vielleicht noch weniger Spielraum für eigenständige Versuche ist.

Die Beispiele der ersten Kapitel werden allesamt Konsolenprogramme sein, denn diese sind leichter zu erstellen. Hierzu noch eine Vorwarnung: Die Konsole unter Windows hat standardmäßig Probleme mit Umlauten. Die Lösung hierfür bekommen Sie in Abschnitt 5.1 präsentiert – bis dahin verzichten Sie bei Ihren eigenen Experimenten auf Umlaute oder schreiben Sie ae, oe, ue.

Wie ich später in einem eigens dafür angelegten Abschnitt 2.5 noch ausführen möchte, ist die Welt des Programmierens sehr englischlastig. Viele Fachbegriffe werden auch in deutschen Gesprächen in ihrer englischen Variante verwendet. Zulasten der Sprachschönheit werde ich in diesem Buch die tatsächliche Sprechweise deutscher Programmierer nachahmen, die sich durch einen Mix aus englischen und eingedeutschten Begriffen auszeichnet. Generell ergibt es auch Sinn, sich die englischen Begriffe bevorzugt zu merken, denn das macht die Problemlösung via Internetsuche deutlich einfacher – dazu mehr im Kapitel 14, »Der Alltag eines Programmierers«.

Aufgrund der besseren Lesbarkeit wird im Buch das generische Maskulinum verwendet. Gemeint sind jedoch immer alle Geschlechter.

Wenn Sie nicht sowieso die physikalische Ausgabe dieses Buches erworben haben, so hoffe ich, dass Sie das E-Book zumindest auf einem separaten Lesegerät aufrufen können, denn das ermöglicht es Ihnen, das Buch beim Programmieren der Beispielprogramme nebenher offen auf dem Schreibtisch zu haben, Informationen anzustreichen und noch mal zurückblättern zu können. Kleben Sie Marker an wichtige oder interessante Stellen, sodass Sie auch nach dem Durchlesen schnell die hilfreichen Stellen wiederfinden, sobald Sie sich an Ihre eigenen Projekte machen.

Die gedruckte Ausgabe bewahrt Sie auch vor einer verlockenden Abkürzung, nämlich die Programmbeispiele einfach 1:1: über die Zwischenablage zu kopieren. Das geht zwar schneller, als sie selbst abzutippen, ist aber deutlich weniger lehrreich. Um Verständnisfragen zu klären, finden Sie dennoch sämtliche Programmbeispiele des Buchs unter:

https://cpp.hasper.info/code_im_buch

1.1.3 Programmtexte, Lösungen und Glossar zum Download

Sie erhalten alle Begleitmaterialien des Buches wie Codebeispiele oder die Musterlösungen der Übungen hier zum Download:

https://cpp.hasper.info

Sie werden an den relevanten Stellen auch auf den vollständigen Link hingewiesen.

Neben dem Stichwortverzeichnis am Ende des Buches finden Sie ein Glossar mit Erklärungen der wichtigsten

Begriffe dieses Buchs unter:

https://cpp.hasper.info/glossar

1.1.4 Fragen und Feedback

Unsere Verlagsprodukte werden mit großer Sorgfalt erstellt. Sollten Sie trotzdem einen Fehler bemerken oder eine andere Anmerkung zum Buch haben, freuen wir uns über eine direkte Rückmeldung an *lektorat@mitp.de*.

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter

https://cpp.hasper.info/korrekturen/

Ich bedanke mich beim mitp-Lektorat und bei Manuel Landsmann für die Unterstützung und wünsche Ihnen viel Spaß beim Lesen!

Philipp Hasper

1.2 Über C++

C++ ist ein richtiger Altmeister unter den Programmiersprachen. Die Sprache wurde Anfang der Achtzigerjahre entwickelt und seitdem kontinuierlich erweitert. Der aktuellste einsetzbare Sprachstandard heißt C++17[1], und C++20 ist schon in der Umsetzung. Die Sprache hat sich wohl deshalb so lange gehalten und rangiert immer noch auf den Bestenlisten der meistverwendeten Programmiersprachen, weil sie gleichzeitig sowohl effiziente als auch verständlich geschriebene Programme erlaubt. Verständlichkeit ist natürlich immer relativ, und sicherlich kann man sagen, dass es mittlerweile intuitivere Sprachen gibt. Aber bezogen auf die Vielseitigkeit und die sehr tiefgreifenden

Kontrollmöglichkeiten für den Programmierer ist und bleibt C++ ein Meilenstein in Bezug auf die Erlernbarkeit.

Wenn Sie einen Vergleich mit anderen, leichter zugänglichen Sprachen wie beispielsweise Python oder Java ziehen wollen, könnte man sagen: »C++ macht exakt das, was Sie sagen. Andere Sprachen machen das, von dem sie denken, dass Sie es wollen«. Offensichtlich hat beides seine Vor- und Nachteile.

C++ wurde zuerst als »C with Classes« (C mit Klassen) bezeichnet, da die Sprache als Erweiterung der auch heute ebenfalls noch populären Sprache C entstand. Dadurch ermöglicht C++ das Schreiben von objektorientierten Programmen, eine Technik, die im Verlauf des Buchs erklärt wird. C++ findet man in der Programmierung von Spielen, Betriebssystemen, Datenbanken, wissenschaftlichen Simulationen, Anwendungen der künstlichen Intelligenz, Raketen[2]... und sie wird von Start-ups bis hin zu Großkonzernen eingesetzt.

Wenn Sie sich jetzt fragen, warum 40 Jahre später sowohl die Erweiterung C++ als auch die anscheinend erweiterungsbedürftige Sprache C gleichzeitig populär sind: Es wird niemals »die eine Programmiersprache« geben, die sich gegen alle durchsetzt. Vielmehr sind sie alle als Teil eines Werkzeugkastens zu begreifen, aus der sich ein Programmierer für jedes neue Projekt das jeweils passende Hilfsmittel aussucht.

1.3 Einrichten der Programmierumgebung

Da C++ eine so weitverbreitete Sprache ist, gibt es auch eine Menge an unterschiedlichen Werkzeugen zur Entwicklung. Sie benötigen hierzu zwei Dinge: Einen

Texteditor zum Schreiben des Codes und einen sogenannten *Compiler* zum Übersetzen in Maschinensprache.

Man kann sich diese beiden Programme selbst zusammenstellen, oder aber man greift zu einer *integrierten Entwicklungsumgebung* (geläufige englische Abkürzung: IDE, für Integrated Development Environment), die beides in sich vereint und obendrein noch weitere Werkzeuge zur Verfügung stellt: zum Beispiel eine automatische Codevervollständigung, die oft getippte Zeichenfolgen vorschlägt, oder einen sogenannten *Debugger*, der bei der Fehlersuche hilft (hierzu mehr gegen Ende des Buchs in Kapitel 13, »Fehlersuche leicht gemacht: Debugging«).

In diesem Buch verwenden wir Microsoft Visual Studio in der kostenlosen Community-Edition, welches Sie hier für Windows herunterladen können: *https://visualstudio.microsoft.com/de/vs/community/*. Falls Sie ein anderes Betriebssystem verwenden, benötigen Sie eine andere IDE, die Inhalte diese Buchs bleiben jedoch gültig.

Was, wenn Sie nicht mit Windows arbeiten?

Keine Sorge, Sie können mit diesem Buch trotzdem C++ lernen, nur ist das Einrichten der Programmierumgebung etwas komplizierter. Ich empfehle Ihnen die Installation des kostenlosen Editors *Visual Studio Code*, ebenfalls von Microsoft. Trotz der Namensähnlichkeit zu Visual Studio ist dieses Programm nur ein Texteditor, den man aber mit Erweiterungen zu einer vollwertigen Entwicklungsumgebung machen kann. Eine englische Installationsanleitung finden Sie hier: *https://code.visualstudio.com/docs/languages/cpp*. Die Benutzeroberfläche ist anders als die Visual-

Studio-Screenshots in diesem Buch, Sie sollten sich allerdings dennoch gut zurechtfinden.

Führen Sie folgende Schritte aus, um Visual Studio zu installieren:

1. Laden Sie das Installationspaket von *https://visualstudio.microsoft.com/de/vs/community/* herunter und führen Sie es aus.
2. Visual Studio kann für verschiedene Programmiersprachen benutzt werden, für die man vorher die entsprechenden Module installieren muss. Setzen Sie daher den Haken in der Kachel *Desktopentwicklung mit C++* (Abbildung 1.1).

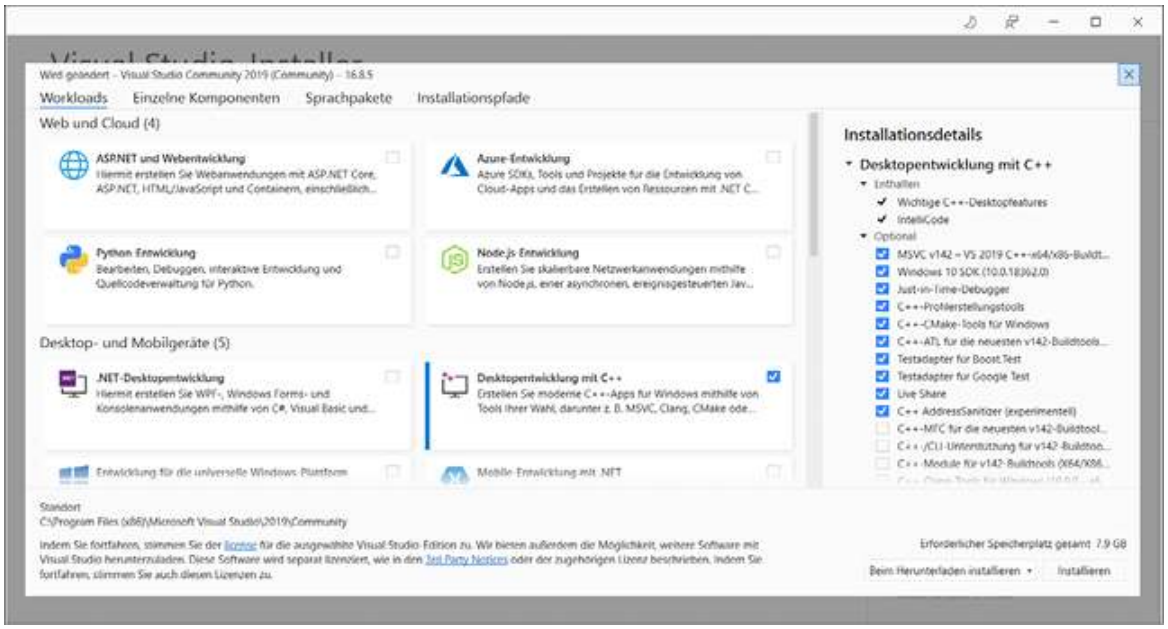


Abb. 1.1: C++-Modul im Installationsdialog aktivieren

3. Wechseln Sie am oberen Fensterrand den Tab zu *Sprachpakete*.
4. Aktivieren Sie unbedingt das englische Sprachpaket, das wir später aus technischen Gründen brauchen werden. Das deutsche Paket ist hilfreich, weil die Erklärungen und

Screenshots in diesem Buch auf Deutsch sind. Sie sollten es daher ebenfalls installieren (Abbildung 1.2).

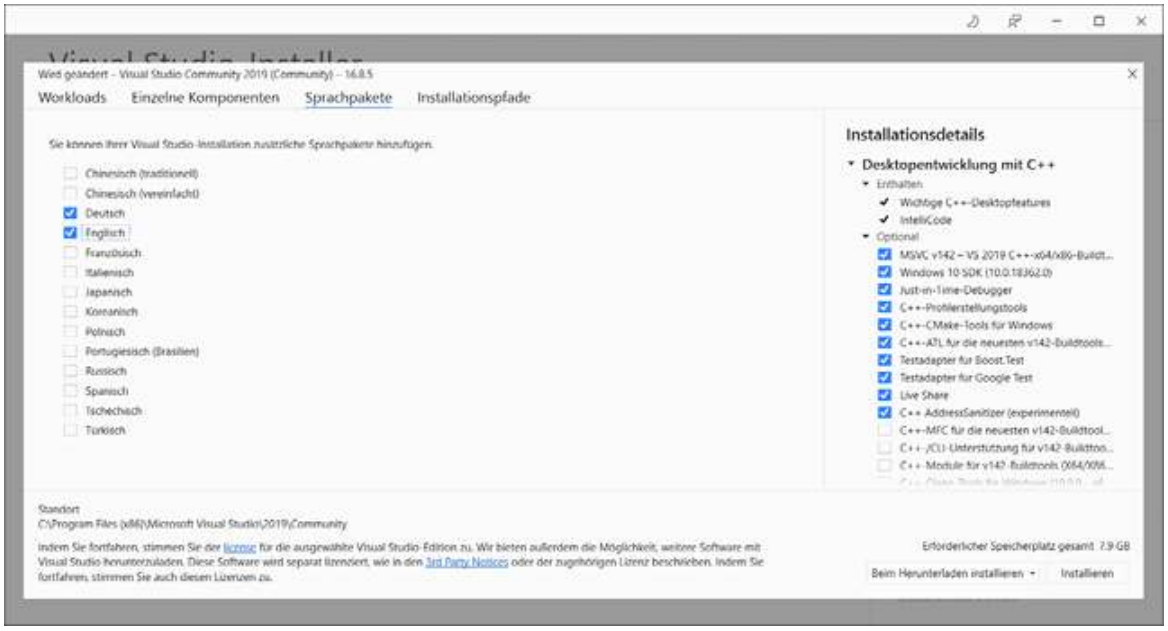


Abb. 1.2: Das englische und deutsche Sprachpaket im Installationsdialog aktivieren

5. Klicken Sie auf *Installieren* in der rechten unteren Ecke und warten Sie, bis der Vorgang abgeschlossen wurde. Das kann eine Weile dauern.
6. Nun werden Sie gebeten, einen Visual-Studio-Account anzulegen. Sie können diesen Schritt zwar fürs Erste überspringen, nach spätestens 30 Tagen werden Sie jedoch erneut dazu aufgefordert, um das Programm weiterhin kostenlos nutzen zu können (Abbildung 1.3).

Abb. 1.3: Entwicklerkonto einrichten oder überspringen

7. Zuletzt können Sie noch ein Farbschema auswählen und dann über die gleichnamige Schaltfläche Visual Studio starten.

Sie werden mit einem Startfenster begrüßt (Abbildung 1.4), welches Sie im nächsten Kapitel nutzen, um ein neues Projekt zu erstellen.

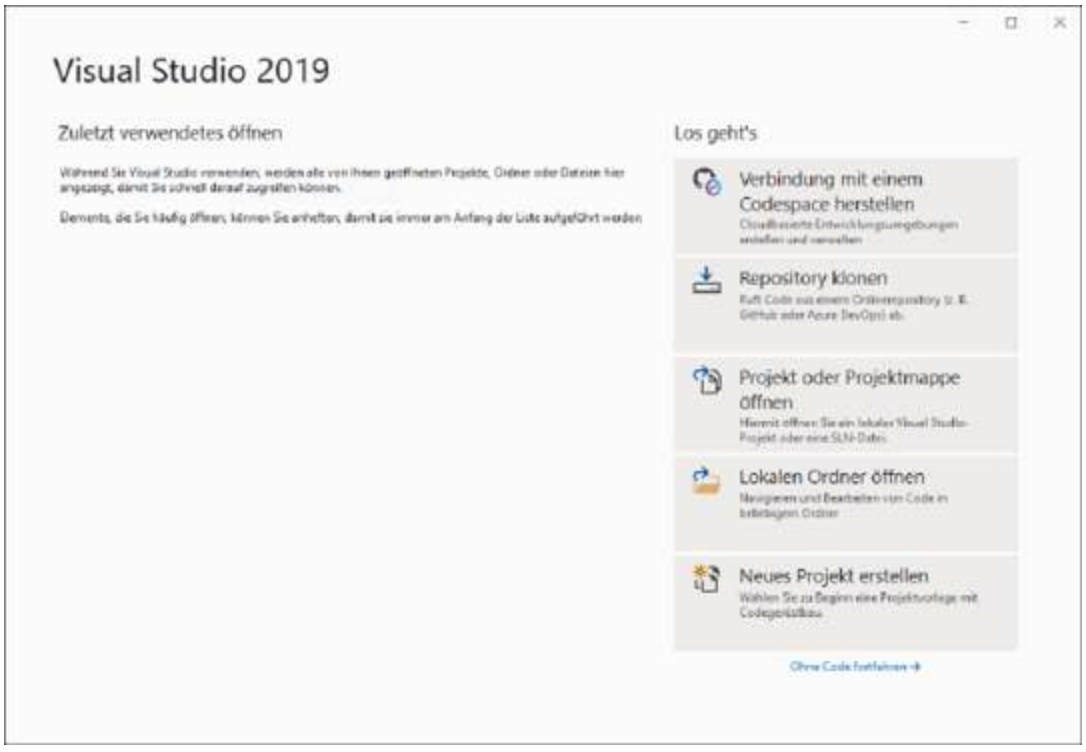


Abb. 1.4: Das Visual-Studio-Startfenster

---

[1] Stand: April 2021

[2] Raketenfirmen sind entweder älter als C++ oder besonders geheimniskrämerisch. Allerdings haben Mitarbeiter von SpaceX ihre Verwendung von C++ bestätigt.

Das erste Programm: »Hello World«

Es hat sich etabliert, beim Lernen einer neuen Programmiersprache immer mit einem sogenannten »Hello World« zu beginnen. Das ist das einfachste mögliche Programm, welches nur den Text »Hello World« ausgibt und sich dann wieder beendet. Es verdeutlicht kurz und knapp, was mindestens benötigt wird, um mit dem Programmieren loslegen zu können.

Und das hier ist »Hello World« in C++:

```
001 #include <iostream>
002
003 int main()
004 {
005   std::cout << "Hello World" << std::endl;
006   return 0;
007 }
```

Sieht erst mal seltsam aus? Keine Sorge, wir werden diesen sogenannten *Code* in diesem Kapitel Stück für Stück durchgehen. Sie werden das Grundgerüst eines jeden C++-Programms kennenlernen und dann das Konzept von Variablen erlernen. Im Verlauf des Kapitels erweitern wir nach und nach den obigen Code um neu gelernte Konzepte und beenden die Lerneinheit mit ein paar Übungen.

Die Nummern an der linken Seite sind übrigens nicht Teil des Programmcodes – das sind Zeilennummern, damit wir

einfacher über den Inhalt sprechen können.

2.1 Anlegen eines neuen Projekts

Jedes neue Programmierprojekt beginnt in seinem eigenen, leeren Ordner auf der Festplatte. Ganz so, als würden Sie ein neues Textdokument beginnen, nur dass es statt einer einzelnen Datei ein ganzer Ordner voller Programmierdateien werden wird. Das Aufsetzen eines Projekts läuft immer ähnlich ab, und so können Sie die folgende Liste als Spickzettel nutzen, wann immer Sie ein neues starten.

Diese Anleitung ist spezifisch für Visual Studio geschrieben und weicht bei anderen Entwicklungsumgebungen leicht ab.

1. Legen Sie einen neuen Ordner `Programmierprojekte` an, zum Beispiel als Unterordner von `Dokumente`. Sie können auch einen beliebigen anderen Ordnernamen wählen.
2. Starten Sie Visual Studio über das Startmenü.
3. Das Startfenster erscheint (Abbildung 1.4). Sollte dies nicht der Fall sein, sondern sich die Programmierumgebung direkt öffnen, können Sie das Startfenster am oberen Bildschirmrand über *DATEI|Startfenster* aufrufen.

Abb. 2.1: Das Visual-Studio-Startfenster öffnen

4. Klicken Sie im Startfenster auf *Neues Projekt erstellen*.

Wählen Sie rechts die Vorlage *Konsolen-App* aus, und klicken Sie auf *Weiter*. Die anderen Projektvorlagen werden in diesem Buch nicht verwendet.

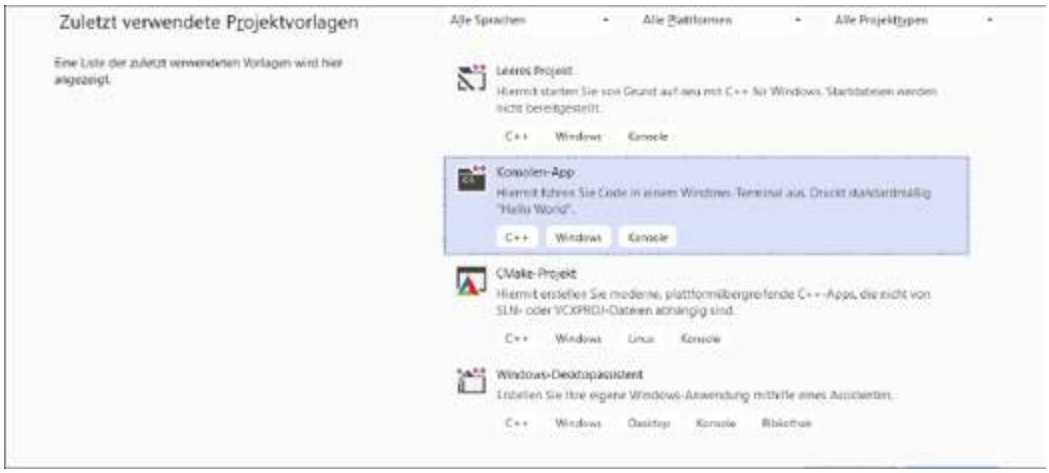


Abb. 2.2: Ein Projekt nach der Vorlage »Konsolen-App« erstellen

5. Geben Sie dem Projekt einen Namen, beispielsweise »Hello World«, und wählen Sie den im ersten Schritt erstellten Ordner `Programmierprojekte` als Speicherort aus. Aktivieren Sie den Haken neben *Platzieren Sie die Projektmappe und das Projekt im selben Verzeichnis*.

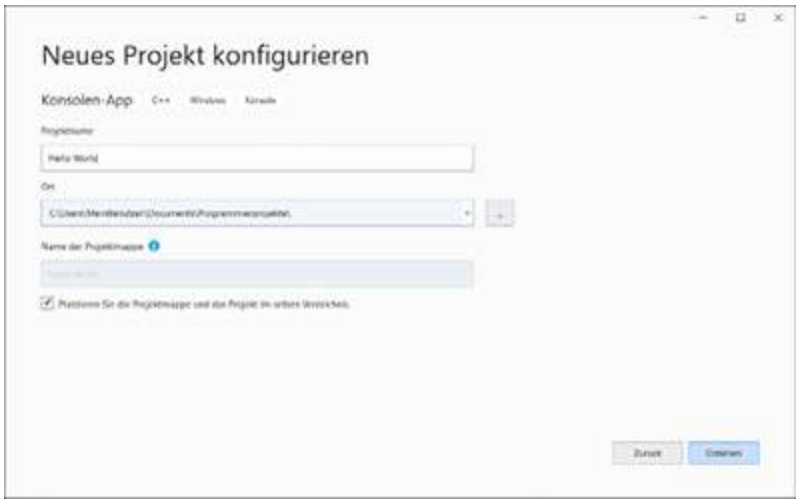


Abb. 2.3: Speicherort des neuen Projekts wählen

6. Nach dem Klick auf *Erstellen* öffnet sich die Programmierumgebung.

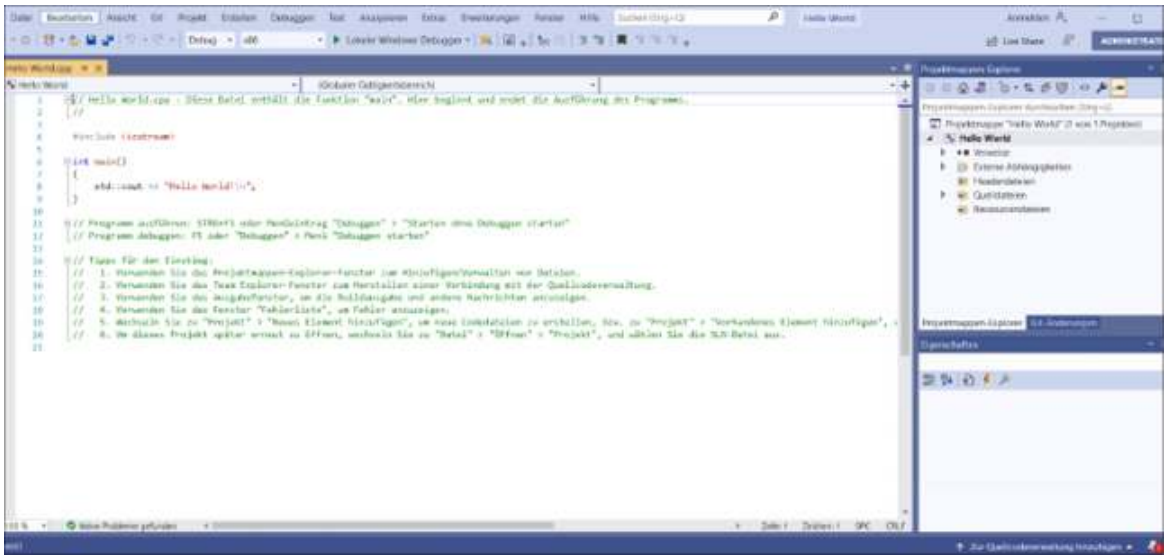


Abb. 2.4: Die Programmierumgebung nach der Erstellung eines neuen Projekts

7. Der Großteil des Fensters besteht aus einem Texteditor, in dem Sie Ihren Programmcode schreiben werden. Der Text wurde schon von Visual Studio vorausgefüllt und zeigt eine leicht andere Variante des Hello Worlds, als ich sie Ihnen weiter oben vorgestellt habe. Markieren Sie den vorhandenen Text, und löschen Sie ihn vollständig.

Nun haben Sie ein neues, leeres Projekt vorbereitet. Übertragen Sie anschließend die Codezeilen des Hello Worlds in den Texteditor.

```
001 #include <iostream>
002
003 int main()
004 {
005   std::cout << "Hello World" << std::endl;
006   return 0;
007 }
```

Sie werden merken, dass Sie beim Programmieren eine Menge an Spezialsymbolen benötigen, die Sie vermutlich