achim LINGOTT

Einführung in Qt

Entwicklung von GUIs für verschiedene Betriebssysteme

Quellcode und Zusatzmaterial unter
plus.hanser-fachbuch.de

HANSER

HANSER

Achim Lingott

Einführung in Qt

Entwicklung von GUIs für verschiedene Betriebssysteme

Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf **plus.hanser-fachbuch.de** einfach diesen Code ein:

plus-a1m23-tr25i

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

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.

Dieses Werk ist urheberrechtlich geschützt.
Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2021 Carl Hanser Verlag München, www.hanser-fachbuch.de
Lektorat: Sylvia Hasselbach
Copy editing: Sandra Gottmann, Wasserburg
Bilder 4.1, 5.19, 9.1, 9.5: © Sven Lingott, Berlin
Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de
Umschlagrealisation: Max Kostopoulos
Titelmotiv: © gettyimages.de/a-r-t-i-s-t
Gesamtherstellung: Eberl & Koesel, Krugzell GmbH & Co. KG
Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Print-ISBN: 978-3-446-46691-3
E-Book-ISBN: 978-3-446-46903-7
E-Pub-ISBN: 978-3-446-46995-2

Inhalt

Vorwort

Dieses Buch vermittelt Einsteigern mit C++-Vorkenntnissen die Grundlagen der Qt-Programmierung. Mit der hervorragenden Qt-Bibliothek lassen sich grafische User Interfaces für die unterschiedlichsten Anwendungsfälle programmieren.

Zum Zeitpunkt der Manuskripterstellung war die Version Qt6 gerade veröffentlicht und wird im Buch prinzipiell als Grundlage für die Beispiele benutzt. Es sind aber verschiedene Module aus Qt5 noch nicht in dieser neuen Version enthalten, weshalb einige Beispiele auf der LTS-(Long Time Support-)Version 5.15 aufbauen. Falls in den kommenden Monaten Ergänzungen zu Qt6 hinzukommen, werden diese immer sofort auf der Plus.Hanser-Webseite veröffentlicht.

Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf *plus.hanser-fachbuch.de* einfach diesen Code ein:

```
plus-a1m23-tr25i
```

Auf dieser Webseite sind alle hier im Buch verwendeten Quelltexte zu finden, teilweise auch etwas umfangreicher als im Buch gezeigt. Diese Quelltexte enthalten auch Zeilen zum Löschen der in einzelnen Funktionen evtl. erstellten Zeiger. Diese delete()-Angaben sind in den Ausschnitten der Quelltexte im Buch in der Regel nicht enthalten.

Außerdem finden Sie dort weitere Beispiele, deren Beschreibung den Umfang des Buches sprengen würde. Nachfolgend finden Sie eine Auswahl:

- **Texteditor**

  Er liest Dateien ein, kann sie verändern und Text ausdrucken. Außerdem kann der geschriebene oder gelesene Text als PDF-Datei abgespeichert werden.

- **Mediaplayer**

  Er bietet Ihnen alle wichtigen Funktionen an. Insbesondere geht es um eigene Aufzeichnungen und das Abspielen vorhandener Dateien.

- **Verkehrsampel**

Natürlich wird nicht nur das Aussehen der Ampel erstellt, sondern auch ihre Steuerung, sowohl automatisch als auch von Hand.

- **Dashboard für ein Fahrzeug**

  Es entsteht ein modernes Dashbord für ein Auto.

- **Android-Applikationen**

  Hier wird die Installation von Qt für Android gezeigt sowie die Erstellung eigener Applikationen.

- **Touchscreen-Applikationen**

  Sie sind insbesondere für die Anwendung unter Android gedacht.

- **Netzwerk-Applikationen**

  Hier werden Beispiele für die Benutzung von HTML sowie von HTTP und anderen Netzwerkprotokollen gezeigt.

Danksagung

An dieser Stelle möchte ich allen Beteiligten einen großen Dank aussprechen. Er gilt allen beteiligten Mitarbeitern des Carl Hanser Verlages, insbesondere Frau Sylvia Hasselbach, die sich stets allen meinen Fragen zu Inhalt und Gestaltung gestellt hat und hilfreich mit Ideen zur Seite stand.

Diesen Dank möchte ich aber auch an diejenigen richten, die, vermutlich ohne es zu ahnen, zur einen oder anderen Idee beigetragen haben: Das sind die Teilnehmer meiner Seminare,

deren Fragen mich dazu anregten, das eine oder andere Problem etwas umfassender oder überhaupt anzugehen.

Und nun viel Spaß beim Lesen und bei der Arbeit mit Qt!

Achim Lingott

1 Einführung und erste Schritte

Dieses Buch wurde hauptsächlich für Personen geschrieben, die Vorkenntnisse in der Programmiersprache C++ besitzen, aber bisher wenig oder nichts mit Qt zu tun hatten. Qt ist übrigens keine Abkürzung, sondern wird ausgesprochen wie das englische Wort *cute*.

Es soll Ihnen die Qt-Grundlagen nahebringen und Sie in die Lage versetzen sich selbstständig weiter mit Qt zu beschäftigen. Sie haben mit Qt eine ausgezeichnete Möglichkeit, grafische User Interfaces für verschiedene Programme zu erstellen. Vielleicht haben Sie schon C++-Programme, die jetzt statt einer Konsolenbedienung ein mit Qt erstelltes GUI erhalten sollen. Dieses Buch wäre ein Anfang dafür. Es soll zwar eine Einführung sein, wird aber an einigen Stellen doch weiter in die Tiefen der Qt-Programmierung, insbesondere die Qt-Bibliothek, einsteigen, um Ihnen wesentliche Zusammenhänge zu zeigen. Darauf können Sie dann später weiter aufbauen und Ihre eigenen Programme entwickeln.

Alle Beispiele verwenden die Bibliothek Qt6, sollten aber bis auf wenige Ausnahmen auch mit der LTS-Version (Long Term

Support-Version) Qt5.15 funktionieren.

Qt entstand Anfang der 1990er-Jahre und ist in C++ geschrieben. Es ist mehrfachlizenziert und kann sowohl für die Open-Source-Programmierung als auch kommerziell genutzt werden.

Qt6 baut vollständig auf die Standardversion C++17 auf. Deshalb ist zur Erstellung auch ein C++17-kompatibler Compiler nötig. Qt für Windows unterstützt das Betriebssystem Windows 10 und die Architektur x86_64. Die unterstützten Compiler sind MSVC 2019 und MinGW 8.1. Ansonsten ist es auch für Linux, macOS, Android und iOS erhältlich.

Ab der Version 2013 wird als Entwicklungsumgebung Visual Studio von Microsoft unterstützt (siehe Abschnitt 1.7). Die Hauptmerkmale von Plug-ins für Visual Studio sind:

- Assistenten zum Erstellen neuer Qt-Projekte und Qt-Klassen
- Verwendung des *Meta Object Compilers (moc)*, des *User Interface Compilers (uic)* und des *Resource Compilers (rcc)*
- Import und Export von Qt-Projektdateien (.pro) und Projekt-Include-Dateien (.pri)
- Automatische Konvertierung eines Qt-Visual Studio-Projekts in ein *qmake*-Projekt und umgekehrt
- Eingebautes Qt-Ressourcenmanagement
- Erstellen einer Übersetzungsdatei (.ts)
- Benutzung des *Qt Linguist*
- Qt-Dokumentation

Ebenso wie in Microsoft Visual Studio lassen sich Qt-Plug-ins auch in *Eclipse*, *Netbeans* oder in *Code::Blocks* verwenden.

Das Kompilieren und Linken von Qt-Quelltext wird wie üblich auf der Grundlage eines *Makefiles* durchgeführt, das von verschiedenen Programmen erstellt werden kann. In Qt existieren *qmake* und *CMake*. Es gibt Unterschiede. Da *qmake* einfacher zu erlernen und bestens in Qt integriert ist, benutzen wir in den Beispielen dieses Buches *qmake*. *CMake* ist schwieriger zu erlernen, bietet aber einige Vorteile, insbesondere beim Einbinden fremder Programme.

Das Programm *qmake* benötigt eine Informationsdatei mit der Endung .pro, die verschiedene Variablen enthält sowie Hinweise auf die zu kompilierenden Dateien. Daraus erstellt *qmake* eine Datei namens *Makefile*, die schließlich vom Programm *make* verwendet wird.

Ein wichtiger Teil des Qt-Frameworks ist der *Meta-Object Compiler* (*moc*). Der *moc* durchsucht die Header-Dateien von Klassen nach C++-fremdem Quellcode und erstellt daraus C++-Quellcode, der gemeinsam mit den anderen C++-Inhalten kompiliert wird. Die .pro-Datei enthält eventuell auch Build-Regeln, die bei Bedarf den *moc* aufrufen.

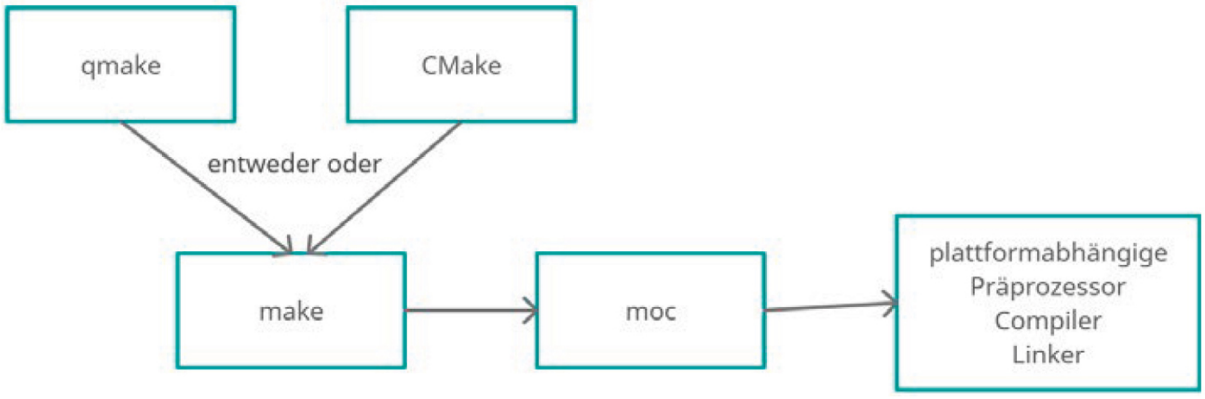

Bild 1.1 Die Erstellung eines Qt-Projekts

Aktuelle Informationen zu den gültigen Bibliotheken, aktuellen Anwendungen und auch Beispiele finden Sie unter *https://www.qt.io/*.

1.1 Download und Installation

Alle benötigten Programme und die Qt-Bibliothek erhalten Sie per Download von *https://www.qt.io/download*. Wählen Sie *Downloads for open source users*. Auf den folgenden Seiten gelangen Sie zum Download des *Qt Online Installer*. Der Punkt *Custom installation* ermöglicht Ihnen die Auswahl einzelner zu installierender Komponenten.

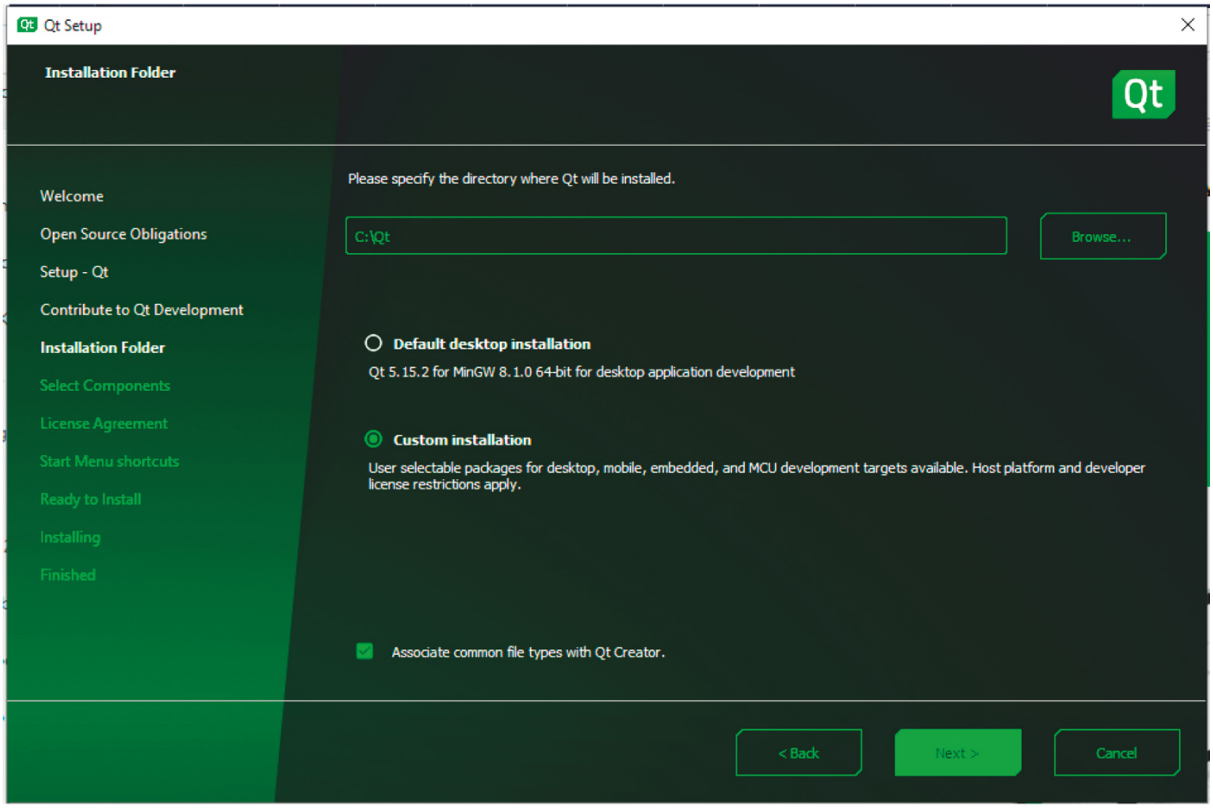

Bild 1.2 Qt Setup und Installer

Wählen Sie mindestens die gezeigten Komponenten aus (Bild 1.2). Die Bibliothek Qt 5.15 benötigen Sie eventuell bei einigen Beispielen, da im Augenblick in der zurzeit erhältlichen Version Qt 6.0 noch nicht alle Module zur Verfügung stehen. Da die Bibliotheken recht umfangreich sind, ist es auch möglich, diese Version später noch zu installieren. Weitere Komponenten stehen zur Verfügung, sind aber für die einführenden Beispiele in diesem Buch nicht nötig. Aber auch diese können Sie später jederzeit noch installieren.

Da sowohl die Bibliothek als auch die bereitgestellten Programme ständig weiterentwickelt werden, sehen die

Installationsoberflächen und die Versionsangaben in Zukunft sicher anders aus. Arbeiten Sie also sinngemäß.

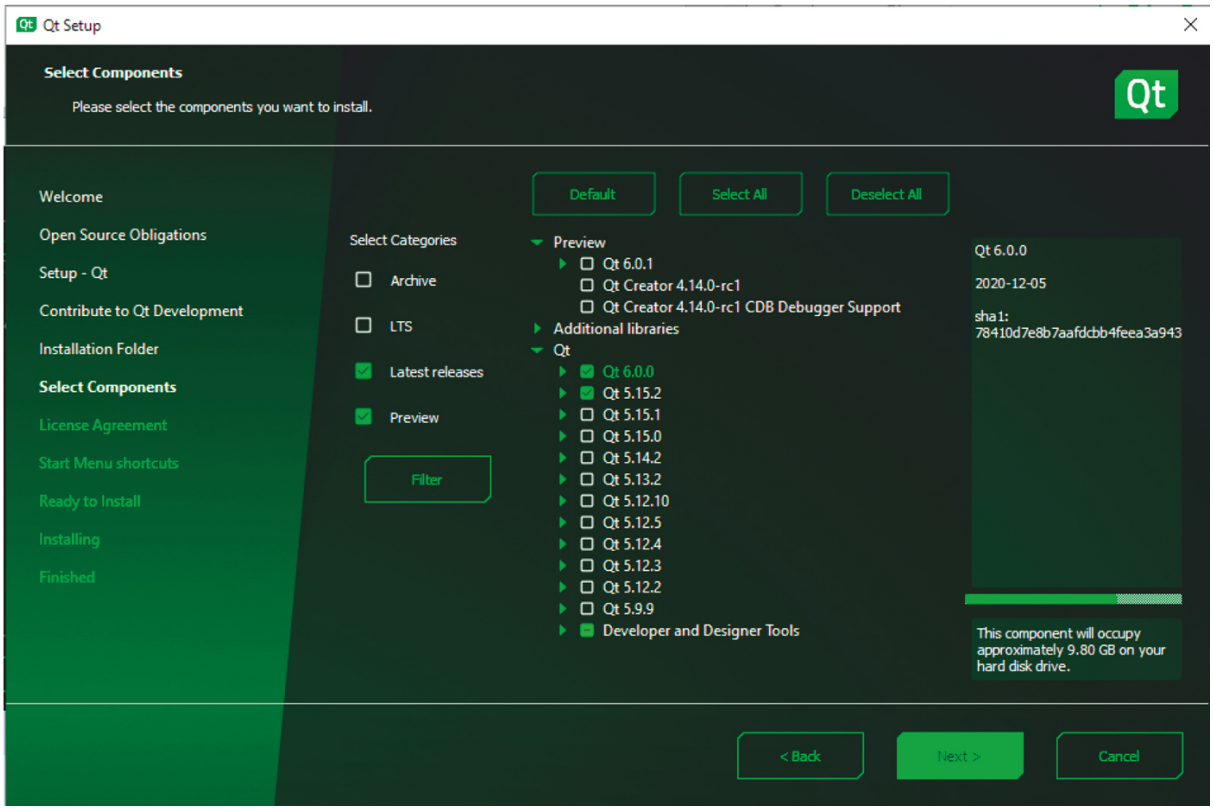

Bild 1.3 Auswahl im Qt Installer

Der Download- und Installationsprozess dauert sicher einige Zeit. Nach Fertigstellung starten Sie den installierten Qt Creator einmal und sehen sich das Ergebnis an.

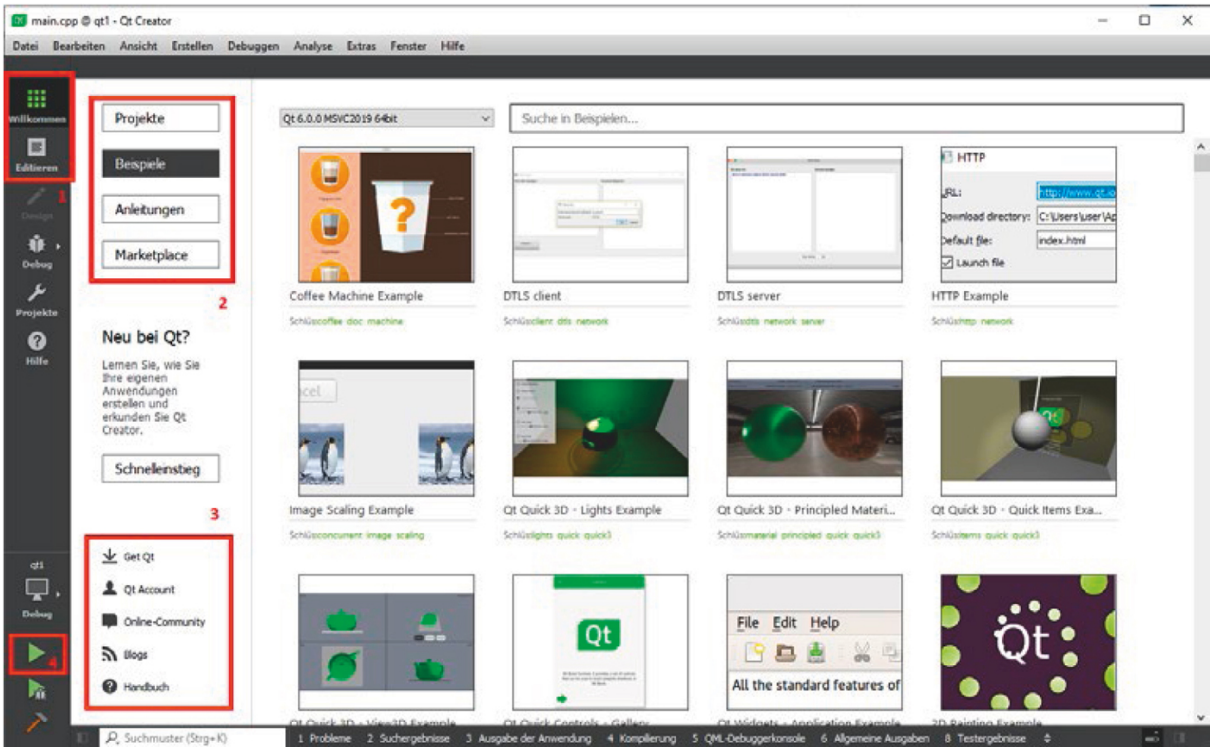

Bild 1.4 Qt Creator nach dem Start

Es fällt sofort auf, dass sehr viele Beispielprogramme zu verschiedenen Themen mit installiert wurden. Sie erreichen diese Beispiele durch Einschalten der Willkommen-Seite (1) und anschließende Betätigung des Buttons *Beispiele* (2). Weiterhin gibt es Zugänge zu sehr vielen Anleitungen und dem Marktplatz, über den Sie viele Zusatzprodukte beziehen können (2). Zugänge zur Online-Community und zum Handbuch des Qt Creators ergänzen diese Seite (3). Über den dreieckigen Button (4) wird ein Programm gestartet. Der darunter liegende Button startet den Debugger (siehe Kapitel 11).

1.2 Die Qt-Bibliothek

Die Bibliothek finden Sie unter *https://doc.qt.io/qt-6/*. Unter *All Qt C++ Classes* sind die Informationen zu mehr als 900 Klassen und unter *All Qt Modules* die Informationen zu den Modulen zu finden. Die Module unterteilen sich in *Qt Essentials* und *Qt Add-Ons*.

Die in den *Qt Essentials* enthaltenen Klassen sind grundlegend für alle Plattformen und werden für die meisten Qt-Anwendungen benötigt.

Die Add-on-Module werden eventuell zusätzlich für bestimmte Einsatzzwecke benötigt. Sie stehen aber möglicherweise nicht auf allen Plattformen zur Verfügung.

Wer die Bibliothek von Qt5 kennt, stellt fest, dass bei Qt6 mit bisherigem Stand weniger Module existieren. Es wurden noch nicht alle übernommen und stehen erst in späteren Versionen von Qt6 zur Verfügung.

Das Installationsprogramm bietet die Möglichkeit, weitere Module herunterzuladen und zu installieren. Das können Sie auch bei bereits erfolgter Installation über den Punkt *Uninstall Qt* im Start-Verzeichnis von Windows unter den installierten Programmen tun. Lassen Sie sich nicht vom Namen abschrecken.

1.2.1 Die Klassen der Bibliothek

In der Bibliothek sind Klassen enthalten, die zur Erstellung und Gestaltung von GUIs benötigt werden, aber auch solche Klassen, die die Erstellung der dazugehörigen Logik ermöglichen. Das sind z. B. Klassen wie `QSqlDatabase`, die Sie zum Verbindungsaufbau zu einer Datenbank benötigen, oder `QXmlReader` zum Auswerten einer XML-Datei.

Viele Klassen sind abgeleitet von `QObject`. Diese Klasse stellt insbesondere Funktionen zur Verfügung, die von allen abgeleiteten Klassen benötigt werden, z. B. Funktionen wie `connect()` und `disconnect()`.

Aber aufgepasst, wer andere Bibliotheken wie z. B. die von .NET oder Java kennt, hat gelernt, dass alle im Programm erzeugten Klassen automatisch von der Klasse `Object` abgeleitet sind. Das ist in Qt nicht so! Es gibt zwar eine Klasse `QObject`. Aber von dieser Klasse erfolgt keine automatische Ableitung, sondern eine solche Ableitung muss programmiert werden. Viele Klassen der Bibliothek sind von `QObject` abgeleitet.

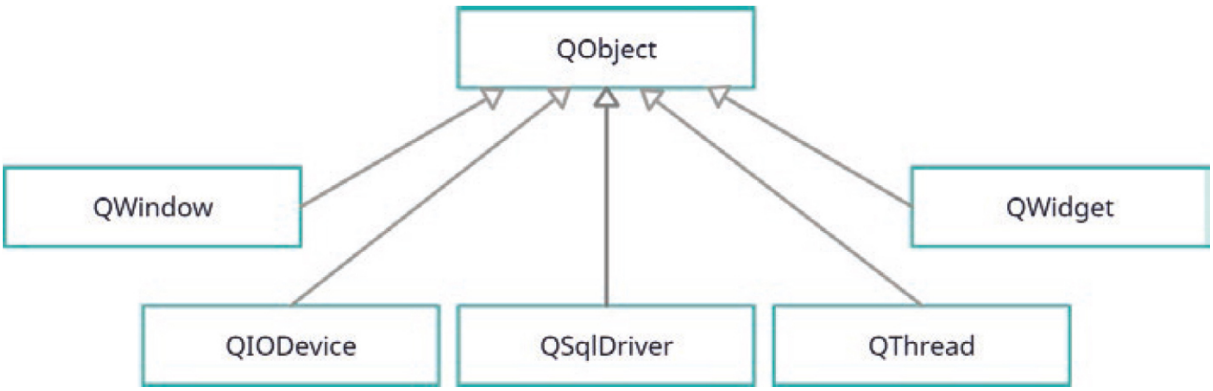

Bild 1.5 Ein kleiner Ausschnitt der Qt-Bibliothek

Weitere Informationen können Sie in der Klassenbibliothek erhalten. Rufen Sie die Klasse `QObject` auf. Dort werden alle von dieser Klasse abgeleiteten Klassen angezeigt.

1.2.2 Die Module der Bibliothek

Während die Klassennamen in der Regel mit Q beginnen (z. B. `QWidget`), beginnen die Namen der Module meist mit Qt (z. B. `QtWidgets`). An dieser Stelle ein paar Informationen zu wichtigen Modulen.

QtCore

Das Modul enthält Klassen mit nichtgrafischer Funktionalität und ergänzt C++ durch einige Eigenschaften, wie z. B.

- einen neuen Mechanismus für die Objektkommunikation, die *signals* und *slots*,
- ein Makro `Q_OBJECT`, das die Kommunikation über *signals* und *slots* sowie einige andere Eigenschaften ermöglicht,
- den *Meta Object Compiler (MOC)*, der in jeder von `QObject` abgeleiteten Klasse C++-fremden Quellcode (wie *signals, slots*) in Code umwandelt, den jeder C++-Compiler verarbeiten kann,
- ein eigenes Property-System, das über ein Makro *Q_PROPERTY* ermöglicht, einen ähnlichen Zugriff auf Variablen zu erreichen, wie sie durch die Properties bei C# bekannt sind,
- Zeiger, die automatisch auf 0 gesetzt werden, wenn das referenzierte Objekt zerstört wird (solche Zeiger werden durch das Klassentemplate `QPointer` bereitgestellt),
- Unterstützung für die Erstellung benutzerdefinierter Typen. Wichtige Klassen dieses Moduls sind z. B. `QObject`, `QPointer`, `QVariant`, `QMetaType`.

QtGUI

In diesem Modul sind die Basisklassen für grafische User Interfaces (GUI) enthalten, Klassen für die Ereignisbehandlung, OpenGL- und OpenGL ES-Integration, für Grafiken, Schriftarten und Text.

Für die weitergehende Entwicklung von Benutzeroberflächen bietet Qt mit dem Modul `QtQuick` eventuell besser geeignete Klassen und Funktionen. Das Modul `QtGUI` (und somit die darin enthaltenen Klassen) wird standardmäßig inkludiert. Falls Sie das Modul ausschalten möchten (z. B. wenn Sie eine reine Konsolenanwendung erstellen möchten), muss die .pro-Datei Ihres Projekts folgenden Eintrag bekommen: `QT -= gui`

Die wichtigsten Klassen dieses Moduls sind die Klassen `QGuiApplication` und `QWindow`.

`QGuiApplication` (abgeleitet von `QCoreApplication`) stellt die überschriebene statische Funktion `instance()` zur Verfügung. Diese gibt einen Zeiger zurück, der dem globale Zeiger *qApp* entspricht.

QtQML

enthält alle Klassen für die Sprache *QML (Qt Meta Language)*, die von *Quick* benötigt wird (dazu mehr in Kapitel 8 dieses Buches).

Im Modul sind auch die QML-Typen definiert (siehe Kapitel 8) und folgende QML-Objekttypen:

```
Component, QtObject, Binding, Connections, Timer
```

Eine JavaScript-Umgebung ist ebenfalls integriert.

QtQuick

Dieses Modul enthält die Standardbibliothek für QML-Applikationen (eine weitere Möglichkeit zum Erstellen von UI, siehe Kapitel 8) und die erforderliche Engine. Es werden alle grundlegenden Typen zum Erstellen einer Benutzeroberfläche mit QML bereitgestellt sowie eine Zeichenfläche und Möglichkeiten zur Animation und Kontrollelemente für Benutzeroberflächen.

QtWidgets

Dieses Modul stellt eine Reihe von Elementen zur Gestaltung von GUIs zur Verfügung, wie die Klassen `QWidget`, `QPushButton`, `QFrame` und andere sowie Klassen für Styles, Layouts und die Model-View-Architektur.

Im Bereich der Add-on-Module ist vielleicht das Modul **`QtPrintSupport`** zu erwähnen. Es enthält Klassen, die ein Drucken auf jeder Plattform ermöglichen. Ebenso lassen sich damit PDF-Dateien erzeugen (siehe Kapitel 7).

Auch wichtig kann das Modul **`QtSQL`** sein. Darin sind Klassen enthalten, die eine Integration von Datenbanken ermöglichen. Man könnte die Klassen grob in die Bereiche Datenbanktreiber, SQL-Zuständigkeit und GUI-Zuständigkeit einteilen (dazu mehr in Kapitel 6).

1.3 Die installierten Programme

Qt Creator

Ein Bild davon ist weiter vorn zu finden (Bild 1.4). Qt Creator stellt eine Arbeitsumgebung für den Programmierer zur Verfügung, sowohl für Qt- als auch Quick-Applikationen. Seine Benutzung wird in Abschnitt 1.4 genauer beschrieben.

Außer dem Qt Creator gibt es inzwischen Plug-ins für andere C++-Entwicklungsumgebungen wie Visual Studio von Microsoft und Eclipse, Netbeans und Code::Blocks (siehe Abschnitt 1.7).

Alle Programme außer dem Qt Creator werden für jeden installierten Compiler separat installiert. Deshalb finden Sie auch im Windows-Start-Verzeichnis jedes der folgenden Programme so oft, wie Sie Compiler installiert haben.

Qt Assistant