

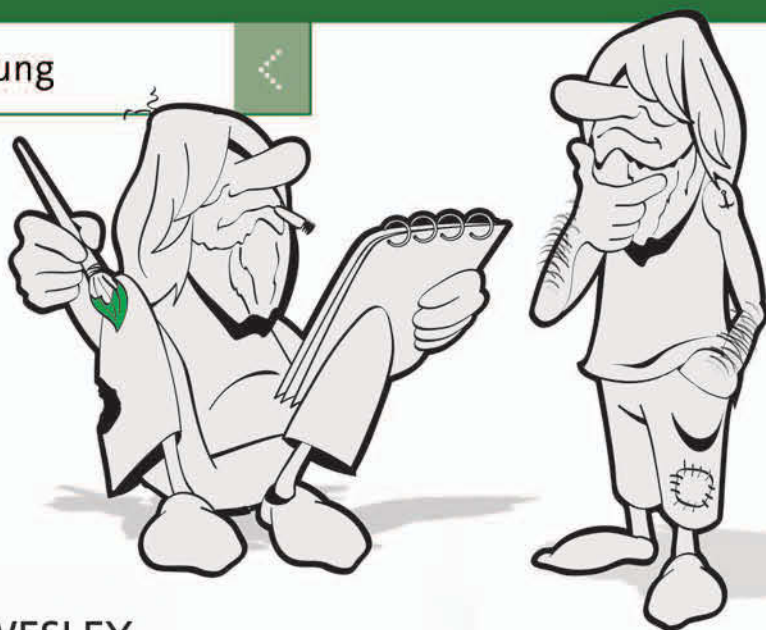

Jasmin Blanchette


Mark Summerfield

C++ GUI Programmierung mit Qt4

Die offizielle Einführung

2. Auflage



ADDISON-WESLEY

Inhalt

Über die Autoren	11
Vorwort	13
Einleitung	17
Danksagungen	21
Eine kurze Geschichte von Qt	23
Teil I: Qt für Einsteiger	27
1 Erste Schritte	29
1.1 Hello Qt	29
1.2 Verbindungen herstellen	32
1.3 Widgets gestalten	33
1.4 Die Referenzdokumentation	38
2 Dialogfelder erstellen	41
2.1 Subklassen in QDialog	42
2.2 Signale und Slots ausführlich betrachtet	49
2.3 Dialogfelder schnell entwerfen	52
2.4 Dialogfelder mit veränderlicher Gestalt	63
2.5 Dynamische Dialogfelder	71
2.6 Integrierte Widget- und Dialogfeldklassen	72
3 Hauptfenster erstellen	79
3.1 Subklassen von QMainWindow	79
3.2 Menüs und Symbolleisten erstellen	85
3.3 Die Statusleiste einrichten	91
3.4 Das Datei-Menü implementieren	93
3.5 Dialogfelder verwenden	101
3.6 Einstellungen speichern	108
3.7 Mehrere Dokumente	110
3.8 Startbildschirme	113

4	Anwendungsfunktionalität implementieren	117
4.1	Das zentrale Widget	118
4.2	Subklassen von QTableWidgetItem	119
4.3	Laden und Speichern	126
4.4	Das Bearbeiten-Menü implementieren	129
4.5	Die restlichen Menüs implementieren	134
4.6	Subklassen von QTableWidgetItem	139
5	Benutzerdefinierte Widgets erstellen	149
5.1	Qt-Widgets anpassen	149
5.2	Subklassen von QWidget	152
5.3	Benutzerdefinierte Widgets in Qt Designer integrieren	163
5.4	Double Buffering	168
Teil II: Qt für Fortgeschrittene		191
6	Layout-Verwaltung	193
6.1	Widgets in einem Formular anordnen	194
6.2	Gestapelte Layouts	200
6.3	Splitter	203
6.4	Bildlaufbereiche	207
6.5	Dockfenster und Symbolleisten	208
6.6	MDI-Anwendungen	212
7	Ereignisverarbeitung	223
7.1	Ereignishandler reimplementieren	224
7.2	Ereignisfilter installieren	230
7.3	Reaktivität bei intensiver Verarbeitung	233
8	2D-Grafik	237
8.1	Mit QPainter malen	239
8.2	Koordinatensystemtransformationen	244
8.3	Hochwertiges Rendering mit QImage	253
8.4	Elementbasiertes Rendering mit Graphics View	257
8.5	Drucken	283
9	Drag&Drop	293
9.1	Drag&Drop aktivieren	293
9.2	Benutzerdefinierte Drag-Typen unterstützen	299
9.3	Umgang mit der Zwischenablage	305

10 Klassen für die Elementpräsentation	307
10.1 Die Elementansicht-Klassen verwenden	309
10.2 Vordefinierte Modelle verwenden	316
10.3 Benutzerdefinierte Modelle implementieren	323
10.4 Benutzerdefinierte Delegates implementieren	339
11 Containerklassen	347
11.1 Sequenzielle Container	348
11.2 Assoziative Container	357
11.3 Generische Algorithmen	361
11.4 Strings, Byte-Arrays und Varianten	363
12 Ein- und Ausgabe	373
12.1 Binärdaten lesen und schreiben	374
12.2 Text lesen und schreiben	381
12.3 Verzeichnisse durchlaufen	387
12.4 Ressourcen einbetten	389
12.5 Interprozesskommunikation	390
13 Datenbanken	397
13.1 Verbinden und Abfragen	398
13.2 Tabellen anzeigen	405
13.3 Datensätze in Formularen bearbeiten	408
13.4 Daten tabellarisch darstellen	415
14 Multithreading	425
14.1 Threads erstellen	426
14.2 Threads synchronisieren	430
14.3 Kommunikation mit dem Hauptthread	437
14.4 Qt-Klassen in untergeordneten Threads verwenden	444
15 Netzwerkprogrammierung	447
15.1 FTP-Clients schreiben	447
15.2 HTTP-Clients schreiben	458
15.3 TCP-Client/Server-Anwendungen schreiben	461
15.4 UDP-Datagramme senden und empfangen	474
16 XML	479
16.1 XML-Daten mit QDomStreamReader lesen	480
16.2 XML mit DOM lesen	489
16.3 XML mit SAX lesen	494
16.4 XML schreiben	499

17 Online-Hilfe bereitstellen	503
17.1 QuickInfos, Statusleistentipps und Direkthilfe	503
17.2 Online-Hilfe mit einem Webbrowser bereitstellen	506
17.3 QTextBrowser als einfaches Hilfemodul	508
17.4 Qt Assistant als leistungsfähige Online-Hilfe	511
Teil III: Qt für Experten	515
18 Internationalisierung	517
18.1 Mit Unicode arbeiten	518
18.2 Anwendungen übersetzungsbereit machen	523
18.3 Dynamischer Sprachenwechsel	530
18.4 Anwendungen übersetzen	536
19 Erscheinungsbild anpassen	541
19.1 Qt-Stylesheets verwenden	542
19.2 Klassen von QStyle ableiten	559
20 3D-Grafik	579
20.1 Mit OpenGL zeichnen	580
20.2 OpenGL und QPainter kombinieren	586
20.3 Overlays mit Framebuffer-Objekten realisieren	594
21 Plug-Ins erstellen	601
21.1 Qt mit Plug-Ins erweitern	602
21.2 Anwendungen Plug-In-fähig machen	614
21.3 Plug-Ins für Anwendungen schreiben	618
22 Anwendungsskripts	621
22.1 Überblick über die Sprache ECMAScript	622
22.2 Qt-Anwendungen mit Skripten erweitern	633
22.3 GUI-Erweiterungen mithilfe von Skripten implementieren	637
22.4 Aufgaben per Skripting automatisieren	646
23 Plattformspezifische Merkmale	661
23.1 Schnittstellen zu systemeigenen APIs	662
23.2 ActiveX unter Windows	666
23.3 X11-Sitzungsverwaltung	680

24 Eingebettete Programmierung	689
24.1 Erste Schritte mit Qt/Embedded Linux	690
24.2 Qt/Embedded Linux anpassen	693
24.3 Qt-Anwendungen mit Qtopia integrieren	694
24.4 Die Qtopia-APIs verwenden	700
Teil IV: Anhänge	711
A Qt installieren	713
A.1 Hinweise zur Lizenzierung	714
A.2 Qt/Windows installieren	714
A.3 Qt/Mac installieren	715
A.4 Qt/X11 installieren	716
B Qt-Anwendungen erstellen	719
B.1 Das Tool qmake verwenden	720
B.2 Erstellungstools von Drittanbietern	726
B.2.1 CMake: Plattformübergreifendes Make	727
B.2.2 Boost.Build (bjam)	728
B.2.3 Das Softwarekonstruktionswerkzeug SCons	730
C Einführung in Qt Jambi	733
C.1 Erste Schritte mit Qt Jambi	734
C.2 Qt Jambi in der Eclipse-IDE verwenden	740
C.3 C++-Komponenten in Qt Jambi integrieren	746
D Einführung in C++ für Java- und C#-Programmierer	755
D.1 Erste Schritte mit C++	756
D.2 Hauptunterschiede zwischen den Sprachen	761
D.2.1 Elementare Datentypen	761
D.2.2 Klassendefinitionen	763
D.2.3 Zeiger	770
D.2.4 Referenzen	773
D.2.5 Arrays	775
D.2.6 Zeichenstrings	779
D.2.7 Enumerationen	780
D.2.8 Typdefinitionen	782
D.2.9 Typumwandlungen	783
D.2.10 Operatorüberladung	786
D.2.11 Werttypen	788
D.2.12 Globale Variablen und Funktionen	790
D.2.13 Namespaces	793
D.2.14 Der Präprozessor	795
D.3 Die C++-Standardbibliothek	798
Index	803