

mediendynamik UG (haftungsbeschränkt)

Konzeption/Beratung, Software-Lösungen, Schulungen

Qt5 GUI-Programmierung mit C++

Form Inhouse-Schulung

Zeitungfang 3 - 5 Tage

Zielgruppe C++-Programmierer, Software-Entwickler, Projektleiter
Qt-Anfänger (C++-Schulungen werden ebenfalls angeboten)
Qt-Anwender mit Vertiefungsbedarf

Vorkenntnisse Objektorientierte Programmierung mit C++

Ziele Die Teilnehmer erwerben die Kompetenz, einen weiten Umfang der Qt-Bibliotheken systematisch und effizient für die Applikationsentwicklung zu nutzen.

Neben den basalen Qt-Verfahren und einfachen GUI-Konstruktionen werden auch komplexe Model-View-Muster und erweiterte Techniken wie Graphics-Programmierung, DB/XML-Zugriff, Qt-Container und Thread-Programmierung vorgestellt und vertieft.

Die im Schulungsprogramm vorgestellten Techniken werden in ihrer systematischen Anlage von den Teilnehmern verstanden und können in der Praxis kreativ und produktiv eingesetzt werden.

Sitz in Berlin • Registergericht Berlin-Charlottenburg • HGB 170134 • USt: DE 196171970
Geschäftsführer: Dr. Lothar Köster, 12051 Berlin, Warthestraße 49 • 030 / 62 84 55 92
Commerzbank Berlin • IBAN DE26 1004 0000 0130 8733 00 • BIC: COBADE33XXX

Gegenstand

Qt ist seit Mitte der 1990er von der norwegischen Firma TrollTech entwickelt worden und realisiert den Urtraum aller Applikationsentwickler:

Qt liefert eine portable Fensterbibliothek, die von den konkreten Fensterbibliotheken der Betriebssysteme (u.a. Win/Mac/XWindows) abstrahiert.

Zudem ist Qt in modernem C++ geschrieben, wodurch die ganze Welt der C/C++-Bibliotheken verfügbar ist.

Qt-Projekte werden nur einmal geschrieben und je auf den Zielsystemen kompiliert, wo sie auf Anhieb laufen.

Qt-Portierungen für Android und iOS sind in einer fortgeschrittenen Testphase.

Seit der Öffnung zur LGPL ist Qt auf dem Weg zum Standard-Framework auf den verbreiteten Rechner-/Betriebsystem-Varianten, im Embedded- und im mittlerweile auch im Mobilbereich.

Qt ist nur deshalb nicht einfach, weil es mittlerweile sehr komplex geworden ist. Für alle zentralen Programmierprobleme hat man Lösungen integriert, die man sich z.B. jahrelang in der Windows-API und MFC vergeblich gewünscht hatte.

Die Qt-Bibliothek-Module sind nicht nur konsequent objektorientiert, sondern auch durchweg modern und systematisch konstruiert. Von hierarchischer Speicherverwaltung, Thread-Unterstützung, SQL und XML über 2D/3D-Unterstützung bis hin zu Multimedia- und Webkit-Modulen bleiben hier kaum Wünsche offen.

Trotz der Vielfalt bleibt Qt ein homogenes Konstrukt mit verlässlicher Systematik.

Qt ist die erste Wahl gerade für komplexe, zeitkritische und langfristig angelegte Projekte, die für jede Betriebssystem-Wahl offen bleiben sollen.

Der aktuelle Portierungs-Boom bestätigt: Der kurzfristige Aufwand rechnet sich schnell und garantiert langfristig stabile, wartungsarme Projekte mit nunmehr genau einem einzigen Kode!

- Der Software-Hersteller gewinnt mit Effizienz und Kreativität
- Der Kunde gewinnt mit Flexibilität und kostengünstiger Leistung

Die angebotene Qt-Schulung vermittelt die systematische Kompetenz, um in der Praxis mit diesen Bibliotheken kreativ und effizient zu arbeiten.

Grundlagen

- IDE-Integration und QtCreator
- Bibliotheksmodule und Qt-Tools, insb. qmake
- Signal-Slot-Verfahren und Meta-Object-Compiler
- Objekt-Verwaltungs-Bäume
- Lokalisation von Formatierungen
- Unicode-Unterstützung und Codecs
- Mehrsprachigkeit von Applikationen, Qt-Linguist

Basale GUI-Techniken

- QWidget-Klasse und Verschachtelung
- Event-Abstraktion
- GUI-Programmierung mit dem QtDesigner
- QPainter, Varianten der Datenzeichnung, 2D-Transformationen
- QMainWindow und CentralWidget
- QMenu in QMainWindow und als Kontextmenü
- Statusbar, Toolbar, Dockbar
- Dialog-Varianten und einfache Eingabe-Widgets
- Layout-Management in Widget-Verschachtelungen
- Scroll- und Splitter-Widgets

Fortgeschrittene GUI-Techniken

- Drag&Drop-Unterstützung, Interprozess-D&D, Clipboard
- QComboBox und Daten-Listen, Referenz-Verwaltung
- Item-basierte Widgets (Liste, Tabelle, Baum)
- Item-basierte Datenverwaltung
- Model-View-Widgets (Liste, Tabelle, Baum)
- Model-basierte Datenverwaltung, QModelIndex
- 2D-Graphics-Darstellung (View, Scene, Item)
- Drag&Drop in den Graphics-Ebenen

Unterstützende Techniken

- Qt-Container-Klassen und Iterationsformen
- SQL-Zugriffe mit Qt-Klassen
- SQL-Widgets und SQL-Models
- XML-Zugriffe über StreamReader, SAX und DOM
- Thread-Programmierung in Varianten
- Inter-Thread-Kommunikation und Synchronisation

(Der Stoff wird ggf. nach Zeit und Interesse angepaßt.)