

УДК 004.438С++
ББК 32.973.26-018.1
Ш68

Шлее М.

Ш68 Qt 5.10. Профессиональное программирование на С++. — СПб.: БХВ-Петербург, 2018. — 1072 с.: ил. — (В подлиннике)

ISBN 978-5-9775-3678-3

Книга посвящена разработке приложений для Windows, Mac OS X, Linux, Android и iOS с использованием библиотеки Qt версии 5.10. Подробно рассмотрены возможности, предоставляемые этой библиотекой, и описаны особенности, выгодно отличающие ее от других библиотек. Описана интегрированная среда разработки Qt Creator и работа с технологией Qt Quick. Книга содержит исчерпывающую информацию о классах Qt 5, и так же даны практические рекомендации их применения, проиллюстрированные на большом количестве подробно прокомментированных примеров. Проекты примеров из книги размещены на сайте издательства.

Для программистов

УДК 004.438С++
ББК 32.973.26-018.1

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Посвящается
любимой Аленушке,
моим детям,
родителям
и семейству Гоуз (Goes)

Оглавление

Предисловие Маттиаса Эттриха	1
Благодарности.....	3
Предисловие автора.....	4
Структура книги.....	5
Введение	13
ЧАСТЬ I. ОСНОВЫ QT.....	25
Глава 1. Обзор иерархии классов Qt.....	26
Первая программа на Qt.....	26
Модули Qt	27
Пространство имен Qt.....	29
Модуль <i>QtCore</i>	29
Модуль <i>QtGui</i>	30
Модуль <i>QtWidgets</i>	30
Модули <i>QtQuick</i> и <i>QtQML</i>	32
Модуль <i>QtNetwork</i>	32
Модули <i>QtXml</i> и <i>QtXmlPatterns</i>	32
Модуль <i>QtSql</i>	32
Модули <i>QtMultimedia</i> и <i>QtMultimediaWidgets</i>	32
Модуль <i>QtSvg</i>	32
Дополнительные модули Qt.....	32
Резюме	33
Глава 2. Философия объектной модели.....	35
Механизм сигналов и слотов	38
Сигналы	41
Слоты	43
Соединение объектов	44
Разъединение объектов	49
Переопределение сигналов	50

Организация объектных иерархий	51
Метаобъектная информация	53
Резюме	54
Глава 3. Работа с Qt	55
Интегрированная среда разработки	55
Программа Qt Assistant.....	55
Работа с qmake	55
Рекомендации для проекта с Qt.....	59
Метаобъектный компилятор MOC.....	60
Компилятор ресурсов RCC	61
Структура Qt-проекта	62
Методы отладки.....	62
Отладчик GDB (GNU Debugger).....	63
Прочие методы отладки	66
Глобальные определения Qt	69
Информация о библиотеке Qt.....	71
Резюме	72
Глава 4. Библиотека контейнеров	74
Контейнерные классы	75
Итераторы	76
Итераторы в стиле Java	77
Итераторы в стиле STL	78
Ключевое слово <i>foreach</i>	80
Последовательные контейнеры	80
Вектор <i>QVector<T></i>	82
Массив байтов <i>QByteArray</i>	83
Массив битов <i>QBitArray</i>	83
Списки <i>QList<T></i> и <i>QLinkedList<T></i>	84
Стек <i>QStack<T></i>	85
Очередь <i>QQueue<T></i>	86
Ассоциативные контейнеры	86
Словари <i>QMap<K,T></i> и <i>QMultiMap<K,T></i>	87
Хэши <i>QHash<K,T></i> и <i>QMultiHash<K,T></i>	89
Множество <i>QSet<T></i>	90
Алгоритмы.....	91
Сортировка	92
Поиск	93
Сравнение	94
Заполнение значениями.....	94
Копирование значений элементов.....	94
Подсчет значений	95
Строки.....	95
Регулярные выражения	97
Произвольный тип <i>QVariant</i>	100
Модель общего использования данных	101
Резюме	102

ЧАСТЬ II. ЭЛЕМЕНТЫ УПРАВЛЕНИЯ	103
Глава 5. С чего начинаются элементы управления?	104
Класс <i>QWidget</i>	104
Размеры и координаты виджета	107
Механизм закулисного хранения.....	108
Установка фона виджета	108
Изменение указателя мыши	109
Стек виджетов	112
Рамки	112
Виджет видовой прокрутки	113
Резюме	115
Глава 6. Управление автоматическим размещением элементов	116
Менеджеры компоновки (layout managers).....	116
Горизонтальное и вертикальное размещение	118
Класс <i>QBoxLayout</i>	118
Горизонтальное размещение <i>QHBoxLayout</i>	120
Вертикальное размещение <i>QVBoxLayout</i>	121
Вложенные размещения	122
Табличное размещение <i>QGridLayout</i>	123
Порядок следования табулятора.....	129
Разделители <i>QSplitter</i>	129
Резюме	130
Глава 7. Элементы отображения.....	132
Надписи	132
Индикатор выполнения.....	136
Электронный индикатор	139
Резюме	141
Глава 8. Кнопки, флажки и переключатели	142
С чего начинаются кнопки? Класс <i>QAbstractButton</i>	142
Установка текста и изображения.....	142
Взаимодействие с пользователем	142
Опрос состояния	143
Кнопки	143
Флажки	145
Переключатели	147
Группировка кнопок.....	148
Резюме	151
Глава 9. Элементы настройки	152
Класс <i>QAbstractSlider</i>	152
Изменение положения	152
Установка диапазона	152
Установка шага	153
Установка и получение значений	153

Ползунок.....	153
Полоса прокрутки.....	155
Установщик.....	156
Резюме.....	158
Глава 10. Элементы ввода.....	159
Однорочное текстовое поле.....	159
Редактор текста.....	161
Запись в файл.....	164
Расцветка синтаксиса (syntax highlighting).....	165
С чего начинаются виджеты счетчиков?.....	171
Счетчик.....	171
Элемент ввода даты и времени.....	172
Проверка ввода.....	173
Резюме.....	175
Глава 11. Элементы выбора.....	176
Простой список.....	176
Вставка элементов.....	176
Выбор элементов пользователем.....	178
Изменение элементов пользователем.....	178
Режим пиктограмм.....	178
Сортировка элементов.....	179
Иерархические списки.....	180
Сортировка элементов.....	183
Таблицы.....	183
Выпадающий список.....	185
Вкладки.....	186
Виджет панели инструментов.....	187
Резюме.....	188
Глава 12. Интервью, или модель-представление.....	189
Концепция.....	190
Модель.....	190
Представление.....	192
Выделение элемента.....	193
Делегат.....	195
Индексы модели.....	197
Иерархические данные.....	198
Роли элементов.....	201
Создание собственных моделей данных.....	203
Промежуточная модель данных (Proxу model).....	211
Модель элементарно-ориентированных классов.....	213
Резюме.....	215
Глава 13. Цветовая палитра элементов управления.....	217
Резюме.....	220

ЧАСТЬ III. СОБЫТИЯ И ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ 221**Глава 14. События 222**

Переопределение специализированных методов обработки событий.....	224
События клавиатуры.....	225
Класс <i>QKeyEvent</i>	225
Класс <i>QFocusEvent</i>	227
Событие обновления контекста рисования. Класс <i>QPaintEvent</i>	227
События мыши.....	228
Класс <i>QMouseEvent</i>	228
Класс <i>QWheelEvent</i>	232
Методы <i>enterEvent()</i> и <i>leaveEvent()</i>	233
Событие таймера. Класс <i>QTimerEvent</i>	233
События перетаскивания (drag & drop).....	233
Класс <i>QDragEnterEvent</i>	233
Класс <i>QDragLeaveEvent</i>	233
Класс <i>QDragMoveEvent</i>	233
Класс <i>QDropEvent</i>	234
Остальные классы событий.....	234
Класс <i>QChildEvent</i>	234
Класс <i>QCloseEvent</i>	234
Класс <i>QHideEvent</i>	234
Класс <i>QMoveEvent</i>	234
Класс <i>QShowEvent</i>	234
Класс <i>QResizeEvent</i>	234
Реализация собственных классов событий.....	236
Переопределение метода <i>event()</i>	236
Мультиязыч 239	239
Сохранение работоспособности приложения.....	244
Резюме.....	244

Глава 15. Фильтры событий 246

Реализация фильтров событий.....	246
Резюме.....	249

Глава 16. Искусственное создание событий..... 250

Резюме.....	253
-------------	-----

ЧАСТЬ IV. ГРАФИКА И ЗВУК 255**Глава 17. Введение в компьютерную графику..... 256**

Классы геометрии.....	256
Точка.....	256
Двумерный размер.....	257
Прямоугольник.....	259
Прямая линия.....	259
Многоугольник.....	260
Цвет.....	260
Класс <i>QColor</i>	260

Цветовая модель RGB	261
Цветовая модель HSV	262
Цветовая модель CMYK.....	263
Палитра.....	264
Предопределенные цвета	265
Резюме	266
Глава 18. Легенда о короле Артуре и контекст рисования.....	267
Класс <i>QPainter</i>	268
Перья и кисти.....	270
Перо	270
Кисть.....	271
Градиенты.....	272
Техника сглаживания (Anti-aliasing)	274
Рисование	275
Рисование точек	275
Рисование линий	275
Рисование сплошных прямоугольников	277
Рисование заполненных фигур	277
Запись команд рисования.....	281
Трансформация систем координат.....	281
Перемещение.....	282
Масштабирование.....	283
Поворот.....	283
Скос.....	283
Трансформационные матрицы	283
Графическая траектория (painter path).....	284
Отсечения	285
Режим совмещения (composition mode).....	286
Графические эффекты	289
Резюме	291
Глава 19. Растровые изображения.....	293
Форматы графических файлов	293
Формат BMP	293
Формат GIF.....	294
Формат PNG.....	294
Формат JPEG.....	294
Формат XPM	294
Контекстно-независимое представление	296
Класс <i>QImage</i>	296
Класс <i>QImage</i> как контекст рисования	303
Контекстно-зависимое представление	304
Класс <i>QPixmap</i>	305
Класс <i>QPixmapCache</i>	306
Класс <i>QBitmap</i>	306
Создание нестандартного окна виджета	307
Резюме	308

Глава 20. Работа со шрифтами	310
Отображение строки.....	312
Резюме	315
Глава 21. Графическое представление.....	316
Сцена.....	317
Представление.....	318
Элемент	319
События	321
Виджеты в графическом представлении	326
Резюме	328
Глава 22. Анимация	330
Класс <i>QMovie</i>	330
SVG-графика	332
Анимационный движок и машина состояний	333
Смягчающие линии.....	336
Машина состояний и переходы	341
Резюме	343
Глава 23. Работа с OpenGL.....	345
Основные положения OpenGL	345
Реализация OpenGL-программы	347
Разворачивание OpenGL-программ во весь экран	350
Графические примитивы OpenGL	350
Трехмерная графика	354
Резюме	358
Глава 24. Вывод на печать	359
Класс <i>QPrinter</i>	359
Резюме	364
Глава 25. Разработка собственных элементов управления	365
Примеры создания виджетов	365
Резюме	370
Глава 26. Элементы со стилем.....	371
Встроенные стили	373
Создание собственных стилей.....	377
Метод рисования простых элементов управления.....	378
Метод рисования элементов управления.....	378
Метод рисования составных элементов управления	378
Реализация стиля простого элемента управления.....	379
Использование каскадных стилей документа	382
Основные положения	383
Изменение подэлементов	385
Управление состояниями	386
Пример.....	387
Резюме	391

Глава 27. Мультимедиа.....	392
Звук	392
Воспроизведение WAV-файлов: класс <i>QSound</i>	393
Более продвинутое возможности воспроизведения звуковых файлов: класс <i>QMediaPlayer</i>	394
Видео и класс <i>QMediaPlayer</i>	401
Резюме	403
ЧАСТЬ V. СОЗДАНИЕ ПРИЛОЖЕНИЙ.....	405
Глава 28. Сохранение настроек приложения.....	406
Резюме	413
Глава 29. Буфер обмена и перетаскивание.....	414
Буфер обмена	414
Перетаскивание.....	415
Реализация drag	417
Реализация drop.....	419
Создание собственных типов перетаскивания	421
Резюме	426
Глава 30. Интернационализация приложения	428
Подготовка приложения к интернационализации	428
Утилита <i>lupdate</i>	430
Программа Qt Linguist.....	431
Утилита <i>lrelease</i> . Пример программы, использующей перевод.....	433
Смена перевода в процессе работы программы	435
Завершающие размышления.....	437
Резюме	438
Глава 31. Создание меню	439
«Анатомия» меню	439
Контекстные меню	443
Резюме	444
Глава 32. Диалоговые окна	445
Правила создания диалоговых окон.....	445
Класс <i>QDialog</i>	446
Модальные диалоговые окна	446
Немодальные диалоговые окна	447
Создание собственного диалогового окна.....	447
Стандартные диалоговые окна	451
Диалоговое окно выбора файлов.....	451
Диалоговое окно настройки принтера	454
Диалоговое окно выбора цвета.....	455
Диалоговое окно выбора шрифта.....	457
Диалоговое окно ввода.....	458
Диалоговое окно процесса	459
Диалоговые окна мастера.....	460

Диалоговые окна сообщений.....	462
Окно информационного сообщения.....	464
Окно предупреждающего сообщения.....	464
Окно критического сообщения.....	465
Окно сообщения о программе.....	466
Окно сообщения <i>About Qt</i>	466
Окно сообщения об ошибке.....	467
Резюме.....	467
Глава 33. Предоставление помощи.....	469
Всплывающая подсказка.....	469
Система помощи (Online Help).....	471
Резюме.....	473
Глава 34. Главное окно, создание SDI- и MDI-приложений.....	475
Класс главного окна <i>QMainWindow</i>	475
Класс действия <i>QAction</i>	476
Панель инструментов.....	477
Доки.....	479
Строка состояния.....	480
Окно заставки.....	482
SDI- и MDI-приложения.....	484
SDI-приложение.....	484
MDI-приложение.....	488
Резюме.....	496
Глава 35. Рабочий стол (Desktop).....	497
Область уведомлений.....	497
Виджет экрана.....	502
Класс сервиса рабочего стола.....	506
Резюме.....	506
ЧАСТЬ VI. ОСОБЫЕ ВОЗМОЖНОСТИ QT.....	507
Глава 36. Работа с файлами, каталогами и потоками ввода/вывода.....	508
Ввод/вывод. Класс <i>QIODevice</i>	508
Работа с файлами. Класс <i>QFile</i>	510
Класс <i>QBuffer</i>	512
Класс <i>QTemporaryFile</i>	512
Работа с каталогами. Класс <i>QDir</i>	512
Просмотр содержимого каталога.....	513
Информация о файлах. Класс <i>QFileInfo</i>	516
Файл или каталог?.....	516
Путь и имя файла.....	517
Информация о дате и времени.....	517
Получение атрибутов файла.....	517
Определение размера файла.....	517
Наблюдение за файлами и каталогами.....	518

Потоки ввода/вывода.....	520
Класс <i>QTextStream</i>	521
Класс <i>QDataStream</i>	522
Резюме	523
Глава 37. Дата, время и таймер.....	524
Дата и время.....	524
Класс даты <i>QDate</i>	524
Класс времени <i>QTime</i>	526
Класс даты и времени <i>QDateTime</i>	527
Таймер	527
Событие таймера.....	528
Класс <i>QTimer</i>	530
Класс <i>QBasicTimer</i>	532
Резюме	532
Глава 38. Процессы и потоки.....	533
Процессы	533
Потоки	536
Приоритеты	538
Обмен сообщениями.....	539
Сигнально-слотовые соединения	540
Отправка событий.....	544
Синхронизация.....	547
Мьютексы.....	547
Семафоры	549
Ожидание условий.....	550
Блокировка чтения/записи	550
Возникновение тупиковых ситуаций	551
Фреймворк <i>QtConcurrent</i>	551
Резюме	553
Глава 39. Программирование поддержки сети	555
Сокетное соединение.....	555
Модель «клиент-сервер»	556
Реализация TCP-сервера	557
Реализация TCP-клиента.....	562
Реализация UDP-сервера и UDP-клиента	566
Управление доступом к сети	570
Блокирующий подход.....	577
Режим прокси.....	580
Информация о хосте.....	580
Есть ли соединение с Интернетом?.....	581
Резюме	581
Глава 40. Работа с XML	582
Основные понятия и структура XML-документа.....	582
XML и Qt.....	584
Работа с DOM	584
Чтение XML-документа	585
Создание и запись XML-документа	587

Работа с SAX.....	589
Чтение XML-документа	589
Класс <i>QXmlStreamReader</i> для чтения XML	592
Использование XQuery.....	594
Резюме	597
Глава 41. Программирование баз данных.....	599
Основные положения SQL	599
Создание таблицы.....	600
Операция вставки.....	600
Чтение данных	600
Изменение данных.....	601
Удаление.....	601
Использование языка SQL в библиотеке Qt	601
Соединение с базой данных (второй уровень)	603
Исполнение команд SQL (второй уровень).....	604
Классы SQL-моделей для интервью (третий уровень)	607
Модель запроса	607
Табличная модель	608
Реляционная модель	610
Резюме	611
Глава 42. Динамические библиотеки и система расширений.....	613
Динамические библиотеки.....	613
Динамическая загрузка и выгрузка библиотеки.....	614
Расширения (plug-ins).....	617
Расширения для Qt.....	617
Поддержка собственных расширений в приложениях	619
Создание расширения для приложения	623
Резюме	625
Глава 43. Совместное использование Qt с платформозависимыми API	627
Совместное использование с Windows API.....	629
Совместное использование с Linux.....	632
Совместное использование с Mac OS X	632
Системная информация.....	637
Резюме	639
Глава 44. Qt Designer. Быстрая разработка прототипов.....	640
Создание новой формы в Qt Designer	640
Добавление виджетов	643
Компоновка (layout).....	644
Порядок следования табулятора.....	645
Сигналы и слоты	646
Использование в формах собственных виджетов	648
Использование форм в проектах	648
Компиляция	651
Динамическая загрузка формы.....	651
Резюме	654

Глава 45. Проведение тестов.....	655
Создание тестов	656
Тесты с передачей данных	659
Создание тестов графического интерфейса	661
Параметры для запуска тестов.....	663
Резюме	664
Глава 46. Qt WebEngine	665
А зачем?.....	666
Быстрый старт.....	667
Создание простого веб-браузера	669
Ввод адресов	669
Управление историей	669
Загрузка страниц и ресурсов.....	670
Пишем веб-браузер: попытка номер два.....	670
Резюме	674
Глава 47. Интегрированная среда разработки Qt Creator.....	675
Первый запуск.....	676
Создаем проект «Hello Qt Creator».....	677
Пользовательский интерфейс Qt Creator	681
Окна вывода	682
Окно проектного обозревателя.....	682
Секция компилирования и запуска.....	682
Редактирование текста	685
Как подсвечен ваш синтаксис?	685
Скрытие и отображение кода.....	686
Автоматическое дополнение кода	686
Поиск и замена.....	687
Комбинации клавиш для ускорения работы.....	691
Вертикальное выделение текста.....	691
Автоматическое форматирование текста	691
Комментирование блоков	692
Просмотр кода методов класса, их определений и атрибутов.....	692
Помощь, которая всегда рядом	693
Интерактивный отладчик и программный экзорцизм	694
Синтаксические ошибки.....	695
Ошибки компоновки.....	696
Ошибки времени исполнения	696
Логические ошибки	697
Трассировка.....	697
Команда <i>Step Over</i>	698
Команда <i>Step Into</i>	698
Команда <i>Step Out</i>	699
Контрольные точки.....	699
Окно переменных (Local and Watches)	700
Окно цепочки вызовов (Call Stack)	701
Резюме	701

Глава 48. Рекомендации по миграции программ из Qt 4 в Qt 5	703
Основные отличия Qt 5 от Qt 4.....	703
Подробности перевода на Qt 5	703
Виджеты	704
Контейнерные классы.....	704
Функция <i>qFindChildren<T>()</i>	705
Сетевые классы	705
WebKit	705
Платформозависимый код	705
Система расширений Plug-ins.....	705
Принтер <i>QPrinter</i>	706
Мультимедиа	706
Модульное тестирование	706
Реализация обратной совместимости Qt 5 с Qt 4.....	706
Резюме	709

ЧАСТЬ VII. ЯЗЫК СЦЕНАРИЕВ JAVASCRIPT

Глава 49. Основы поддержки сценариев JavaScript	712
Принцип взаимодействия с языком сценариев	713
Первый шаг использования сценария.....	716
Привет, сценарий.....	717
Резюме	718
Глава 50. Синтаксис языка сценариев	720
Зарезервированные ключевые слова.....	720
Комментарии.....	721
Переменные.....	721
Предопределенные типы данных	722
Целый тип.....	722
Вещественный тип.....	722
Строковый тип	723
Логический тип	723
Преобразование типов.....	723
Операции	725
Операторы присваивания.....	725
Арифметические операции	725
Поразрядные операции.....	726
Операции сравнения	726
Приоритет выполнения операций	727
Управляющие структуры	728
Условные операторы	728
Оператор <i>if... else</i>	728
Оператор <i>switch</i>	729
Оператор условного выражения.....	730
Циклы.....	730
Операторы <i>break</i> и <i>continue</i>	730
Цикл <i>for</i>	730

Цикл <i>while</i>	731
Цикл <i>do...while</i>	731
Оператор <i>with</i>	732
Исключительные ситуации.....	732
Оператор <i>try...catch</i>	732
Оператор <i>throw</i>	733
Функции.....	733
Встроенные функции.....	735
Объектная ориентация.....	735
Статические классы.....	738
Наследование.....	738
Перегрузка методов.....	741
Сказание о «джейсоне».....	742
Резюме.....	743
Глава 51. Встроенные объекты JavaScript.....	744
Объект <i>Global</i>	744
Объект <i>Number</i>	744
Объект <i>Boolean</i>	744
Объект <i>String</i>	745
Замена.....	745
Получение символов.....	745
Получение подстроки.....	745
Объект <i>RegExp</i>	745
Проверка строки.....	746
Поиск позиции совпадений.....	746
Найденное совпадение.....	746
Объект <i>Array</i>	746
Дополнение массива элементами.....	747
Адресация элементов.....	747
Изменение порядка элементов массива.....	747
Преобразование массива в строку.....	748
Объединение массивов.....	748
Упорядочивание элементов.....	748
Многомерные массивы.....	748
Объект <i>Date</i>	749
Объект <i>Math</i>	750
Модуль числа.....	750
Округление.....	751
Определение максимума и минимума.....	751
Возведение в степень.....	751
Вычисление квадратного корня.....	751
Генератор случайных чисел.....	752
Тригонометрические методы.....	752
Вычисление натурального логарифма.....	752
Объект <i>Function</i>	753
Резюме.....	753

Глава 52. Классы поддержки JavaScript и практические примеры	754
Класс <i>QJSValue</i>	754
Класс <i>QJSEngine</i>	754
Практические примеры	755
«Черепашья» графика	755
Сигналы, слоты и функции	762
Полезные дополнительные функции.....	765
Резюме	769
ЧАСТЬ VIII. ТЕХНОЛОГИЯ QT QUICK.....	771
Глава 53. Знакомство с Qt Quick.....	772
А зачем?.....	772
Введение в QML	774
Быстрый старт.....	776
Использование JavaScript в QML	782
Резюме	783
Глава 54. Элементы	785
Визуальные элементы	785
Свойства элементов.....	788
Собственные свойства.....	790
Создание собственных элементов	793
Создание собственных модулей	795
Динамическое создание элементов	795
Элемент <i>Flickable</i>	796
Готовые элементы пользовательского интерфейса	797
Диалоговые окна.....	802
Резюме	805
Глава 55. Управление размещением элементов	806
Фиксаторы	806
Традиционные размещения	813
Размещение в виде потока	817
Резюме	819
Глава 56. Элементы графики.....	820
Цвета.....	820
Растровые изображения	821
Элемент <i>Image</i>	821
Элемент <i>BorderImage</i>	825
Градиенты.....	826
Шрифты	828
Рисование на элементах холста	828
Шейдеры и эффекты.....	833
Резюме	837
Глава 57. Пользовательский ввод	838
Область мыши.....	838
Сигналы	841

Ввод с клавиатуры	845
Фокус	846
«Сырой» ввод	848
Мультитач	850
Резюме	852
Глава 58. Анимация	853
Анимация при изменении свойств	853
Анимация для изменения числовых значений	855
Анимация с изменением цвета	856
Анимация с поворотом	857
Анимации поведения	858
Параллельные и последовательные анимации	860
Состояния и переходы	863
Состояния	863
Переходы	866
Модуль частиц	868
Резюме	872
Глава 59. Модель/Представление	873
Модели	873
Модель списка	873
XML-модель	874
JSON-модель	876
Представление данных моделей	877
Элемент <i>ListView</i>	877
Элемент <i>GridView</i>	880
Элемент <i>PathView</i>	882
Визуальная модель данных	884
Резюме	886
Глава 60. Qt Quick и C++	888
Использование языка QML в C++	888
Взаимодействие из C++ со свойствами QML-элементов и вызов их функций	889
Соединение QML-сигналов со слотами C++	891
Использование компонентов языка C++ в QML	894
Экспорт объектов и виджетов из C++ в QML	895
Использование зарегистрированных объектов C++, их свойств и методов в QML	897
Реализация визуальных элементов QML на C++	901
Класс <i>QQuickImageProvider</i>	904
Резюме	909
Глава 61. 3D-графика Qt 3D	910
Основы	910
Свет	911
Камера	912
3D-объекты	913
Материалы	916
Трансформация	919

Анимация.....	921
Qt 3D Studio.....	923
Резюме	924

ЧАСТЬ IX. МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И QT 927

Глава 62. Введение в мир мобильных приложений	928
Смартфоны меняют все.....	928
Виртуальные магазины приложений	930
Распространение приложений вне виртуального магазина	932
Qt и разработка мобильных приложений	932
Резюме	934

Глава 63. Подготовка к работе над мобильными приложениями	935
Подготовка среды для разработки iOS-приложений	935
Подготовка среды для разработки Android приложений	938
Запуск приложений на реальном устройстве	945
Резюме	946

Глава 64. Особенности разработки приложений для мобильных устройств	948
Анатомия файлов свойств для iOS- и Android-приложений	949
Файл свойств iOS-приложений.....	949
Файл свойств Android-приложений.....	951
Полноэкранный режим.....	955
iOS-реализация.....	956
Android-реализация.....	956
Автоматический поворот	956
Конфигурирование приложений для поддержки поворота.....	957
iOS-реализация.....	957
Android-реализация.....	958
Обработка поворота в приложениях	958
Сенсоры	960
Пользовательский ввод при помощи пальцев	964
Положение рук.....	967
Резюме	967

Глава 65. Пример разработки мобильного приложения	969
Обдумывание и планирование приложения	969
Название приложения.....	970
Значок приложения.....	970
Что будет в первой версии?	971
Пишем код.....	973
Добавление к приложению значков и стартовых экранов	980
iOS-реализация.....	980
Android-реализация.....	982
Резюме	983

Глава 66. Публикация в магазине мобильных приложений	984
Этапы работы для App Store	984
Регистрация	984

Настройки для запуска приложений на реальных устройствах	985
Создание электронной подписи.....	991
Создание страницы приложения	992
Загрузка и публикация приложения	998
Этапы работы для Google Play	1000
Регистрация	1001
Создание страницы приложения	1001
Создание электронной подписи.....	1005
Загрузка и публикация приложения	1007
Резюме	1009
ПРИЛОЖЕНИЯ	1011
Приложение 1. Настройка среды для работы над Qt-приложениями	1012
Настройка среды для Mac OS X	1012
Настройка среды для Windows	1013
Настройка среды для Ubuntu Linux	1015
Приложение 2. Таблица простых чисел.....	1018
Приложение 3. Таблицы семибитной кодировки ASCII	1021
Приложение 4. Описание архива с примерами	1024
Предметный указатель	1034

Любая достаточно передовая технология неотличима от магии.

Артур Кларк

Предисловие Маттиаса Эттриха

Let's start with a fictional story. Imagine ten years ago, someone came to me and asked: «Is it possible to write a feature-rich graphical application, and then compile and run this application natively on all different major operating systems? On Linux, on UNIX, on Windows, and on the Macintosh?» Back then — as a young computer scientist — I would probably have answered, «No, that's not possible. And if it was, the system would be very difficult to use, and limited by the weakest platform. Better choose one platform, or write your code several times.»

A few years later I discovered Qt — and how wrong I was!

Qt makes true cross-platform programming a reality, without limiting your choices and creativity. It gives users what users want: fast, native applications that look and feel just right. It gives developers what developers want: a framework that lets us write less code, and create more. A framework that makes programming fun again, no matter whether we do commercial work or contribute to Open Source projects.

Too good to be true? You don't believe me? Well, the proof is easy. I'll pass the word on to Max, who will tell you exactly how it's done. Max, your turn.

Before I leave, let me wish you good luck with your first Qt-steps. But be careful, it may very well turn into a lifetime addiction. Either way, I hope you will have as much fun using Qt as we have creating it for you.

Matthias Ettrich

October 1st, 2004, Oslo

Давайте пофантазируем. Представьте себе, будто бы 10 лет назад кто-то подошел ко мне и спросил: «Возможно ли создать многофункциональное приложение с графическим интерфейсом пользователя, а затем откомпилировать его и пользоваться на всех распространенных операционных системах? На Linux, UNIX, Windows, Macintosh?» В то время я был молодым программистом, и я бы, наверное, ответил: «Нет, это невозможно. А если это и было бы возможным, то такая система была бы очень трудна в обращении и ограничена возможностями самой слабой платформы. Лучше выбрать одну операционную систему или переписать свою программу несколько раз».

Несколько лет спустя я открыл для себя Qt — и понял, как я был не прав!

Qt делает платформонезависимое программирование действительно, не ограничивая ваш выбор и творческие возможности. Qt предоставляет пользователям то, чего они хотят: быстрые программы, которые выглядят и работают должным образом. Qt предоставляет разработчикам программ то, чего они желают: среду, позволяющую писать меньше кода, создавая при этом больше. Благодаря этому программирование становится интереснее, и при этом неважно, является оно коммерческим или проектом с открытым исходным кодом (Open Source).

Слишком хорошо, чтобы быть правдой? Вы мне не верите? Ну что же, доказать это просто. Я передаю слово Макс, который расскажет вам подробно, как это делается. Макс, теперь твоя очередь.

Прежде чем я попрошаюсь, позвольте пожелать вам удачи в ваших первых шагах с Qt. Но осторожно, Qt может вызвать у вас зависимость на всю жизнь. В любом случае, я надеюсь, что вам будет так же интересно работать с Qt, как нам было интересно создавать ее для вас.

*Маттиас Эттрих
1 октября 2004, Осло*

Благодарности

Автор выражает глубокую признательность своей первой наставнице в области информатики — Татьяне Дмитриевне Оболенцевой — преподавателю Новосибирского филиала Московского технологического университета легкой промышленности, разбудившей в нем творческий потенциал. А также профессору, доктору Ульриху Айзэнекеру (Ulrich W. Eisenpecker), который помог ему определиться в многообразном мире информатики.

Большую помощь в создании этой книги оказали самые близкие автору люди: Алена Шлее, родители Евгений и Галина Шлее, сестра Натали Гоуз.

Глубокую признательность и уважение испытывает автор ко всему коллективу издательства «БХВ-Петербург», а в особенности к Игорю Владимировичу Шишигину, Юрию Викторовичу Рожко, Андрею Геннадиевичу Смышляеву, Юрию Владимировичу Якубовичу, Евгению Евгеньевичу Рыбакову и Григорию Лазаревичу Добину за их поддержку и сотрудничество.

Особая благодарность Маттиасу Эттриху (Matthias Ettrich) — сотруднику фирмы Nokia и основателю KDE — за проявленный интерес и поддержку, оказанную при подготовке книги. Автор благодарит Кента Ханзена (Kent Hansen) и Андреаса Ардаль Ханссена (Andreas Aardal Hanssen) за проверку примеров книги, а также остальных сотрудников фирмы Nokia за замечательную библиотеку, которая вдохновила его на написание этой книги.

Я также выражаю благодарность моим читателям, присылавшим свои отклики, замечания и предложения: Виталию Улыбину, Александру Климову, Артуру Акоюну, Ирине Романенко, Вячеславу Гурковскому, Николаю Прокушину, Юрию Зинченко, Людмиле Брагиной, Алексею Старченко, Дмитрию Оленченко, Антону Матросову, Михаилу Кипа, Денису Песоцкому, Павлу Плотникову, Ярославу Васильеву, Михаилу Ермоленко, Виталию Венделю, Александру Басову и Николаю Прохоренку, Александру Матвееву, Стасу Койнову, Андрею Донцову, Ивану Ензхаеву, Александру Гилевичу, Александру Миргородскому, Максиму Вальтеру, Эрну Белинину, Максиму Дзамбаеву, Денису Тену, Александру Марченко, Никите Липовичу, Артему Спиридонову, Ярославу Баранову, Семену Пейтонову, Олегу Белекову, Всеволоду Лукьянину, Александру Кузнецову, Ивану Ежову, Вадиму Сорочану, Алексею Пуц и Евгению Карныгину.

Предисловие автора

Занимайтесь любимым делом и тогда в вашей жизни не будет ни одного рабочего дня...

Здравствуйте, дорогие читатели! Рад сообщить вам, что наконец-то задуманное свершилось, и воплощенное в жизнь новое издание этой книги вышло в свет! Перед вами не просто знакомая вам всем по предыдущим изданиям книга о Qt, которая, надеюсь, гарантированно обеспечит вас дополнительными знаниями о том, как разрабатывать приложения. Новое издание постепенно и доходчиво введет вас в курс того, как вы сможете *применять* свои знания. И это очень важно, ведь на самом деле, как говорится: «Не в знаниях сила, а в их умелом применении!»

В этом издании значительно расширены темы по технологии Quick и добавлена новая глава о программировании трехмерной графики в QML: «3D-графика Qt 3D». Остальные главы предыдущего издания также тщательно обновлены и дополнены. Все главы книги содержат новую информацию и приведены в соответствие с самой актуальной на данный момент версией библиотеки Qt — 5.10.

Завершая подготовку этого издания книги, я счел необходимым добавить в нее также совершенно новую часть с пятью главами, посвященными разработке мобильных приложений для iOS и Android, после изучения которых у вас появятся возможности и шансы применить и эти новые знания себе во благо. Без разницы: будут это ваши собственные хобби-проекты, которым вы уделите свободное от работы время и которые помогут вам заработать дополнительную копейку, или же вы, овладев применением знаний, решите стать полностью независимым от работодателя и уйти от него, всецело посвятив себя только своим, личным проектам. И сделать их источником своего основного достатка. Как это в свое время сделал я.

Помните, теперь, с этой книгой, все задуманное вами станет возможным! Явным! Поэтому для воплощения своей мечты настоятельно рекомендую — внимательно читайте каждую страницу этой книги и не забывайте святую истину: «Пища к размышлению кормит только людей сообразительных!». Ради того, чтобы предоставить вам такую возможность, я провел многие бессонные ночи в бесконечных часах работы. Фактически прописался в университетской библиотеке, которая на время подготовки этой книги, можно так сказать, стала моим вторым домом. Сделано это было с одной целью — чтобы донести до вас всю необходимую информацию.

Итак, книга у вас в руках! Поздравляю! Но это не значит, что моя работа над ней закончилась. Я постарался сделать эту книгу более открытой и интересной для общения с вами и поэтому создал для нее домен: www.qt-book.com. И внедрил в каждую главу книги индивидуальные ссылки на соответствующие им веб-страницы, по которым вы, дорогие читатели,

можете оставлять свои отзывы о том, что вам понравилось и что нет, а также ваши предложения и пожелания о том, чтобы вам хотелось узнать дополнительно. Словом, пишите обо всем, что заинтересовало вас после прочтения глав книги. Система обратной связи хорошо зарекомендовала себя еще со времени выхода моей первой книги — она отлично работает и помогает мне выявить дополнительные моменты, которые интересны вам, а также с вашей помощью исправить некоторые неточности. Поэтому спасибо вам за ваши отклики и большой интерес, проявленный вами к моим предыдущим книгам! За все это я вам искренне благодарен.

Пожалуйста, помните, что, несмотря на внушительный объем книги, основной ее задачей является ознакомить вас с большим спектром возможностей библиотеки Qt 5 и подтолкнуть к тому, чтобы в дальнейшем вы могли «копать глубже» и находить нужную вам информацию самостоятельно.

Мне же остается в очередной раз пожелать вам счастливого путешествия по главам моей книги. И, конечно же, счастливых открытий в познании нашей любимой библиотеки Qt.

Структура книги

Книга состоит из девяти частей. Хочу сразу обратить ваше внимание вот на что: если вы уже имели опыт программирования с предыдущей версией Qt 4, то полезнее всего начать ознакомление с книгой с материала *главы 48*, которая описывает отличия Qt 5 от Qt 4 и содержит рекомендации по внесению изменений в код для переноса существующего кода на новую версию.

Часть I. Основы Qt

Основная задача этой части — описать новый подход при программировании с использованием Qt.

- ◆ **Глава 1. Обзор иерархии классов Qt.** Глава эта вводная, она знакомит с модульной архитектурой и классами Qt, а также с реализацией первой программы, созданной с помощью Qt.
- ◆ **Глава 2. Философия объектной модели.** В эту главу входит подробное описание механизма сигналов и слотов, организация объектов в иерархии, свойства объектов.
- ◆ **Глава 3. Работа с Qt.** Эта глава описывает процесс создания проектных файлов, которые можно переработать на любой платформе в соответствующие make-файлы, методы и средства отладки приложений.
- ◆ **Глава 4. Библиотека контейнеров.** Глава содержит описание классов, которые в состоянии хранить в себе элементы различных типов данных и манипулировать ими. Здесь описываются также различные категории итераторов. Контейнерные классы в Qt являются составной частью основного модуля, и знания о них необходимы на протяжении всей книги. Эта глава содержит также описание механизма «общих данных», дающего возможность экономично и эффективно использовать ресурсы. Все контейнерные классы: списки, словари, хэш-таблицы и др. — описаны в отдельности, особое внимание уделено классу строк `QString` и мощному механизму для анализа строк, именуемому «регулярное выражение». Здесь также осуществляется знакомство с классом `QVariant`, объекты которого способны содержать в себе данные разного типа.

Часть II. Элементы управления

Задача второй части — описание элементов, из которых строятся пользовательские интерфейсы. Эта часть дает навыки грамотного и обоснованного применения таких элементов.

- ◆ **Глава 5. С чего начинаются элементы управления?** Глава вводит понятие виджета как синонима элемента управления. Описываются три класса, от которых наследуются все элементы управления, и самые важные методы этих классов, такие как изменение размера, местоположения, цвета и др. Рассказывается, как управлять из виджета изменением изображения указателя мыши. Говорится и о классе `QStackedWidget`, который способен показывать в отдельно взятый момент времени только лишь один из содержащихся в нем виджетов.
- ◆ **Глава 6. Управление автоматическим размещением элементов.** Эта глава описывает классы для размещений (`Layouts`), позволяющие управлять различными вариантами размещения виджетов на поверхности другого виджета, знакомит с классом разделителя `QSplitter`. В качестве примера разрабатывается программа калькулятора.
- ◆ **Глава 7. Элементы отображения.** Глава описывает элементы управления, не принимающие непосредственного участия в действиях пользователя и служащие только для отображения информации. В группу таких элементов входят надписи, индикатор выполнения и электронный индикатор. Подробно рассматриваются основные особенности этих виджетов.
- ◆ **Глава 8. Кнопки, флажки и переключатели.** В этой главе после описания основных возможностей базового класса кнопок рассматриваются следующие типы интерфейсных элементов: обычные кнопки, флажки и переключатели. Делается акцент на особенностях их применения. Описывается возможность группировки таких интерфейсных элементов.
- ◆ **Глава 9. Элементы настройки.** Глава описывает группу виджетов, позволяющих выполнять не требующие большой точности настройки: ползунков, полос прокрутки, установщиков.
- ◆ **Глава 10. Элементы ввода.** В этой главе описывается группа виджетов, представляющих собой фундамент для ввода пользовательских данных. Детально рассматривается каждый виджет этой группы: однострочные и многострочные текстовые поля, счетчик, элемент ввода даты и времени. Описано использование класса `QValidator` для предотвращения неправильного ввода пользователя.
- ◆ **Глава 11. Элементы выбора.** Глава знакомит с группой виджетов, в которую входят списки, таблицы, вкладки, инструменты и др.
- ◆ **Глава 12. Интервью, или модель-представление.** Эта глава знакомит с подходом «модель-представление» и преимуществами, связанными с его использованием.
- ◆ **Глава 13. Цветовая палитра элементов управления.** Глава описывает процесс изменения цветов как для каждого виджета в отдельности, так и для всех виджетов приложения.

Часть III. События и взаимодействие с пользователем

Цель третьей части — подробно ознакомить с тонкостями применения событий при программировании с использованием библиотеки Qt.

- ◆ **Глава 14. События.** В этой главе разъясняется необходимость сосуществования двух механизмов, связанных с оповещением: сигналов и слотов и событий. После этого сле-

дует описание целого ряда классов событий для мыши, клавиатуры, таймера и др. Отдельно рассматривается каждый метод, который предназначен для получения и обработки этих событий. Рассмотрены механизмы работы с событиями множественных касаний «мультикас».

- ◆ **Глава 15. Фильтры событий.** Глава знакомит с очень мощным механизмом, дающим возможность объекту фильтра осуществлять перехват управлением событиями. Это позволяет объектам классов, унаследованных от класса `QObject`, реализовывать, например, один класс фильтра и устанавливать его в нужные объекты, что значительно экономит время на разработку, так как отпадает необходимость наследования или изменения класса, если при этом преследуется цель только переопределить методы для обработки событий.
- ◆ **Глава 16. Искусственное создание событий.** Здесь рассказывается о способах создания события искусственным образом, что может оказаться очень полезным, например, для имитации ввода пользователя.

Часть IV. Графика и звук

Задача четвертой части — познакомить с разнообразием возможностей, связанных с программированием компьютерной графики. Затрагивается также тема реализации приложений со звуком и мультимедиаприложений.

- ◆ **Глава 17. Введение в компьютерную графику.** Глава описывает основные классы геометрии, необходимые, прежде всего, для рисования. Дается понятие цвета и палитры.
- ◆ **Глава 18. Легенда о короле Артуре и контекст рисования.** Эта глава описывает перья и кисти, отсечения, градиентные заливки и многое другое. В ней содержатся примеры рисования различных графических примитивов — от точек до полигонов, рассказывается о записи команд рисования при помощи класса `QPicture`, о трансформации систем координат и о других аспектах, связанных с рисованием.
- ◆ **Глава 19. Растровые изображения.** Глава содержит подробное описание двух классов для растровых изображений: `QPixmap` и `QImage`. Рассматриваются преимущества подобного разделения растровых изображений на два класса и функциональные возможности этих классов. Вводится понятие «прозрачность» и перечисляются поддерживаемые графические форматы.
- ◆ **Глава 20. Работа со шрифтами.** В этой главе рассматривается использование шрифтов.
- ◆ **Глава 21. Графическое представление.** Глава описывает иерархию классов `QGraphicsScene`, предоставляющих интерфейс рисования высокого уровня. Эти классы можно применять там, где необходимо дать пользователю возможность манипулировать большим количеством графических изображений, — например, в качестве спрайтов для компьютерных игр.
- ◆ **Глава 22. Анимация.** Эта глава содержит описание класса `QMovie`, предназначенного для отображения анимированных изображений в GIF- и MNG-форматах, а также описывает возможности анимационного движка, машины состояний и использование масштабируемой графики и анимации в формате SVG.
- ◆ **Глава 23. Работа с OpenGL.** Глава описывает использование библиотеки OpenGL в Qt, где OpenGL привлекается в качестве дополнительного средства для вывода трехмерной графики. Подробно рассматриваются классы Qt, созданные для поддержки OpenGL. Для полноты проводится краткое знакомство с возможностями самого OpenGL: примитива-

ми, проекциями, пикселями и изображениями, трехмерной графикой и дисплейными списками.

- ◆ **Глава 24. Вывод на печать.** Глава рассказывает о возможностях, связанных с выводом на печатающее устройство: об использовании принтера в качестве контекста рисования, о настройке параметров печати и о многом другом.
- ◆ **Глава 25. Разработка собственных элементов управления.** Глава описывает факторы, которые необходимо учитывать при создании собственных виджетов. Например, обсуждается, какой из классов взять в качестве базового, и какие методы нуждаются в переопределении.
- ◆ **Глава 26. Элементы со стилем.** Эта глава рассказывает о механизме Look & Feel, позволяющем изменять внешний вид приложения и его поведение. Глава знакомит со встроенными стилями, демонстрирует их применение, а также описывает механизм создания своих собственных стилей и использование CSS (Cascading Style Sheets) для этой цели.
- ◆ **Глава 27. Мультимедиа.** Глава знакомит с возможностями воспроизведения звука и видео, предоставляемыми модулем `QtMultimedia`.

Часть V. Создание приложений

В пятой части описываются все необходимые составляющие для реализации профессиональных приложений.

- ◆ **Глава 28. Сохранение настроек приложения.** Здесь объясняется механизм сохранения измененных пользователем настроек и их восстановления при дальнейших загрузках приложения.
- ◆ **Глава 29. Буфер обмена и перетаскивание.** В этой главе демонстрируются возможности обмена данными между разными приложениями посредством буфера обмена и перетаскивания (`drag & drop`).
- ◆ **Глава 30. Интернационализация приложения.** Хорошее приложение предоставляет многоязыковую поддержку, обеспечивающую его комфортное использование в различных языковых средах. *Глава 30* описывает технику, связанную с интернационализацией и локализацией создаваемых приложений.
- ◆ **Глава 31. Создание меню.** Меню — это неотъемлемая часть каждого приложения для настольных и переносных компьютеров. *Глава 31* описывает процесс создания меню разных типов: строк меню, выпадающих меню, контекстных меню, а также ускорителей, предназначенных для быстрого доступа к отдельным пунктам меню.
- ◆ **Глава 32. Диалоговые окна.** В этой главе вводятся понятия модальных и немодальных диалоговых окон. Описываются стандартные диалоговые окна для выбора файлов, шрифтов, цвета и др. Объясняется, как применять простые диалоговые окна для выдачи сообщений и как создавать собственные диалоговые окна.
- ◆ **Глава 33. Предоставление помощи.** Предоставление подсказок в программах необходимо для облегчения работы пользователя. В *главе 33* рассмотрены различные варианты помощи и методы их реализации.
- ◆ **Глава 34. Главное окно, создание SDI- и MDI-приложений.** Эта глава описывает технику создания панелей инструментов для приложений и использования строк состояния, знакомит с анатомией главного окна приложения и возможностями класса для главного

окна приложения `QMainWindow`. Приведен пример создания полноценного текстового редактора: сначала как приложения SDI (Single Document Interface), а затем MDI (Multiple Document Interface).

- ◆ **Глава 35. Рабочий стол (Desktop).** Здесь рассматриваются основные приемы использования классов работы с рабочим столом операционной системы. Операционные системы предоставляют возможности размещения значков в области уведомлений (в Windows эта область находится в нижнем правом углу) и взаимодействия пользователя с приложением из этой области. Рассматривается класс `QSystemTrayIcon`, в котором реализованы механизмы работы с областью уведомлений. Кроме того, рассматривается класс `QDesktopWidget`, предоставляющий доступ к графической области рабочего стола.

Часть VI. Особые возможности Qt

Задача шестой части — подробно ознакомить с теми возможностями Qt, которые не обязательно связаны с программированием графики и пользовательского интерфейса, но очень важны, поскольку предоставляют программисту набор функциональных возможностей практически на все случаи жизни и тем самым позволяют добиться полной платформенезависимости.

- ◆ **Глава 36. Работа с файлами, каталогами и потоками ввода/вывода.** Здесь описываются возможности, предоставляемые Qt для чтения и записи файлов, а также для просмотра каталогов и получения подробной информации о файлах. Завершается глава примером реализации программы, осуществляющей поиск файлов в заданном каталоге (папке).
- ◆ **Глава 37. Дата, время и таймер.** Эта глава описывает область назначения и применения таймеров, а также знакомит с классами, предоставляющими информацию о текущей дате и времени и методы для работы с ними.
- ◆ **Глава 38. Процессы и потоки.** Глава рассказывает о назначении процессов, описывает использование многопоточности для параллельного выполнения задач, необходимые для этого классы и методы. Рассматриваются совместное использование данных и сложности, связанные с этим. Вводятся понятия мьютекса (mutex) и задач синхронизации, а также семафора, как обобщения мьютексов. Описываются высокоуровневые классы Qt, которые упрощают реализацию многопоточных приложений.
- ◆ **Глава 39. Программирование поддержки сети.** Глава знакомит с классами, позволяющими реализовывать как TCP/UDP-клиенты, так и серверы. После этого рассматривается специализированный класс для работы с сетью на более высоком уровне: `QNetworkAccessManager`.
- ◆ **Глава 40. Работа с XML.** Эта глава содержит краткий вводный курс в очень популярный формат для описания, хранения и обмена данными XML (eXtensible Markup Language). Анализируются преимущества и недостатки различных способов представления данных XML-документа. После небольшого введения в DOM (Document Object Model) объясняется, как можно осуществлять чтение и проводить операции с узлами DOM-представления XML-документа. Говорится также о чтении при помощи SAX (Simple API for XML) и о записи XML-документов.
- ◆ **Глава 41. Программирование баз данных.** Глава содержит краткий вводный курс в базы данных. Описываются процессы соединения с базой данных и ее открытия. Подробно говорится о классе `QSqlQuery` и исполнении SQL-команд (Structured Query Lan-

guage), получении, удалении и добавлении данных. Рассматривается возможность использования классов, базирующихся на технологии «Интервью».

- ◆ **Глава 42. Динамические библиотеки.** Эта глава рассказывает, как объединить используемый различными приложениями или их частями код в отдельные динамические библиотеки. Описывается процесс создания и загрузки динамических библиотек. Кроме того, рассказывается о системе расширений (plug-ins).
- ◆ **Глава 43. Совместное использование Qt с платформозависимыми API.** Глава описывает включение платформозависимых функций ОС Windows, Mac OS X и Linux в программы, базирующиеся на библиотеке Qt.
- ◆ **Глава 44. Qt Designer. Быстрая разработка прототипов.** После небольшого описания возможностей Qt Designer производится разработка приложения средствами, предоставляемыми этой средой.
- ◆ **Глава 45. Проведение тестов.** Тестирование — это залог правильной разработки программного обеспечения. Глава 45 знакомит с возможностями, предоставляемыми Qt для проведения модульного тестирования.
- ◆ **Глава 46. Qt WebEngine.** Глава описывает модуль QtWebEngine, который предоставляет инструментарий для получения и отображения информации из Всемирной паутины (WWW, World Wide Web, или просто Web). Прочитав эту главу, вы научитесь быстро создавать веб-браузеры и другие веб-клиенты.
- ◆ **Глава 47. Интегрированная среда разработки Qt Creator.** Здесь вы познакомитесь с новой интегрированной средой разработки, узнаете, какие преимущества дает ее использование, и из каких компонентов она состоит. Каждый компонент отдельно описан, особое внимание уделяется встроенному интерактивному отладчику.
- ◆ **Глава 48. Рекомендации по миграции программ из Qt 4 в Qt 5.** Эта глава призвана познакомить читателя с основными изменениями, сделанными в Qt 5, и дать рекомендации для миграции программ на Qt 5. Описываются решения, которые способны обеспечить обратную совместимость с Qt 4.

Часть VII. Язык сценариев JavaScript

Задача седьмой части — ознакомить с языком сценариев JavaScript, который базируется на стандарте ECMA Script 4.0. С предоставлением поддержки этого языка в своих программах перед разработчиком открываются расширенные возможности.

- ◆ **Глава 49. Основы поддержки сценариев JavaScript.** Эта глава объясняет принцип работы и принцип взаимодействия языка JavaScript с Qt-программами и описывает, в каких случаях и какие преимущества дает использование языка сценариев.
- ◆ **Глава 50. Синтаксис языка сценариев.** Здесь описываются ключевые слова языка сценариев и приводятся примеры их использования: объявление переменных, операции присвоения, логические операции, циклы, определение функций, определение классов и многое другое.
- ◆ **Глава 51. Встроенные объекты JavaScript.** Эта глава описывает встроенные в JavaScript объекты. К ним относятся: Object, Math, String, Boolean, RegExp, Number, Date и т. д.
- ◆ **Глава 52. Классы поддержки JavaScript и практические примеры.** Здесь описаны классы, необходимые разработчику программного обеспечения на языке C++ для предоставления поддержки языка сценариев в своих программах, и приводится несколько примеров, подытоживающих материал предшествующих глав.

Часть VIII. Технология Qt Quick

Задача восьмой части — познакомить с новой технологией Qt Quick, которая предоставляет язык QML для создания графического пользовательского интерфейса.

- ◆ **Глава 53. Знакомство с Qt Quick.** Глава знакомит с набором инструментов, формирующих технологию Qt Quick, раскрываются преимущества, связанные с применением этой технологии. Завершается глава созданием первого проекта, выполненного с помощью Qt Quick и с использованием языка JavaScript.
- ◆ **Глава 54. Элементы.** Эта глава описывает анатомию, типы элементов и возможности взаимодействия элементов друг с другом. А также рассматриваются возможности создания собственных модулей, элементов и использование модулей с уже готовыми элементами.
- ◆ **Глава 55. Управление размещением элементов.** Здесь объясняется отличие привычных методов размещения элементов в библиотеке Qt при помощи классов размещения (layout) от нового подхода фиксации, используемого в языке QML. Кроме того, показаны различные методы использования техники фиксации на примерах.
- ◆ **Глава 56. Элементы графики.** В этой главе приведено описание возможности использования элементов растровых изображений, градиентов, шрифтов, цвета, рисования на элементах холста и использования эффектов шейдеров.
- ◆ **Глава 57. Пользовательский ввод.** Глава раскрывает возможности элементов, предназначенных для работы пользователя с клавиатурой и мышью, описывает механизмы использования сигналов, свойств их обработки, а также возможность использования механизмов обработки множественных касаний «мультитач».
- ◆ **Глава 58. Анимация.** В главе рассматриваются основные типы анимаций, свойств, поворота, поведения и т. п. Кроме того, рассмотрены последовательные и параллельные анимации, применение смягчающих линий, использование состояний и переходов и система частиц для создания анимаций с участием большого количества элементов.
- ◆ **Глава 59. Модель/Представление.** Здесь рассматриваются элементы отображения различных моделей данных, реализация делегатов и самих моделей данных.
- ◆ **Глава 60. Qt Quick и C++.** Эта глава посвящена внедрению компонентов, разработанных на языке C++, в язык QML (в технологию Qt Quick) и в противоположном направлении.
- ◆ **Глава 61. 3D-графика Qt 3D.** Глава представляет собой введение в создание трехмерной графики с помощью модулей группы Qt 3D. Рассматриваются источники света, камеры, анатомия 3D-объектов, их стандартные формы, а также использование материалов, проведение трансформаций и анимации 3D-объектов.

Часть IX. Мобильные приложения и Qt

Задача девятой части — снабдить вас всей необходимой информацией для создания мобильных приложений под платформы iOS и Android. Прочитав и овладев навыками этой части, вы научитесь воплощать свои идеи в продукты, которые могут быть доступны для огромной аудитории пользователей мобильных устройств.

- ◆ **Глава 62. Введение в мир мобильных приложений.** Эта глава проводит небольшой экскурс в историю появления такого явления, как смартфоны. Знакомит с самыми распространенными для платформ iOS и Android магазинами приложений и их возможно-

стями. А так же содержит информацию об особенностях использования Qt в разработке мобильных приложений.

- ◆ **Глава 63. Подготовка к работе над мобильными приложениями.** В этой главе мы установим и настроим на компьютере все необходимое, чтобы начать разрабатывать приложения для iOS и Android.
- ◆ **Глава 64. Особенности разработки приложений для мобильных устройств.** Подходы к разработке приложений для мобильных устройств отличаются от подходов к разработке приложений для настольных и переносных компьютеров. Эта глава делает акцент на существующей разнице и дает информацию о том, что необходимо учитывать при разработке мобильных приложений.
- ◆ **Глава 65. Пример разработки мобильного приложения.** В этой главе мы займемся разработкой примера реального приложения, в котором будут реализованы некоторые из описанных ранее подходов и технологий, присущих мобильным устройствам.
- ◆ **Глава 66. Публикация в магазине мобильных приложений.** В завершающей главе книги показано, что необходимо сделать, чтобы опубликовать приложение из главы 65 в магазинах App Store и Google Play.

Приложения

Книга включает четыре приложения. В *приложении 1* описаны шаги, которые необходимо проделать для установки Qt на ОС Windows, Mac OS X и Linux (Ubuntu), в *приложении 2* приведена таблица простых чисел, в *приложении 3* содержится таблица ASCII-кодировки, а в *приложении 4* — описание электронного архива с примерами.

Электронный архив

Электронный архив с примерами к этой книге можно скачать с FTP-сервера издательства «БХВ-Петербург» по ссылке <ftp://ftp.bhv.ru/9785977536783.zip> или со страницы книги на сайте www.bhv.ru (см. *приложение 4*).

Введение

Путешествие в тысячу миль начинается с первого шага.

Древняя китайская мудрость

Сегодня практически невозможно представить себе приложение, не обладающее интерфейсом пользователя. Понятия *Software* (программный продукт), *Apps* (приложения) и *GUI* (Graphical User Interface, графический интерфейс пользователя) неразрывно связаны друг с другом.

Хотя каждая из операционных систем обладает всем необходимым для создания графического интерфейса пользователя, использование этих доступных «инструментов» требует больших затрат времени и практического опыта. Даже библиотеки, призванные облегчить процесс написания программ, не дают процессу создания программ и приложений той простоты и легкости, какой хотелось бы. Поэтому и сегодня разработчики по-прежнему тратят массу времени на реализацию интерфейса пользователя. Но самый большой недостаток, связанный с применением таких библиотек, — это платформозависимость.

В самом деле, если вы программируете только для какой-то одной конкретной операционной системы, например ОС Windows, то у вас, наверняка, возникнет вопрос — зачем мне испытывать что-то новое? И одна из причин, почему это все же стоит сделать, — реализация платформонезависимых приложений. Платформонезависимая реализация приложений — это будущее программной индустрии. С каждым днем она будет приобретать все более возрастающее значение. Только задумайтесь — зачем оставлять без внимания пользователей Mac OS X или мобильных устройств, базирующихся на Android, только лишь потому, что вы являетесь программистом для ОС Windows? Позволив своему приложению работать под разными ОС, вы заметно увеличите количество пользователей (клиентов). Выигрыш же от реализации платформонезависимых приложений налицо: значительно сокращается время разработки, поскольку вам не приходится писать код многократно — под каждую платформу, и, что не менее важно, отпадает необходимость знать нюансы каждой из платформ, для которой пишется программа. Не понадобится также во время разработки продукта формировать специальные подкоманды разработчиков для каждой платформы реализации — все это может значительно сократить не только время разработки, но и себестоимость вашего продукта. Вы сможете использовать самые передовые инструменты для отладки и совершенствования, улучшения кода программ — например, абсолютно бесплатную интегрированную среду разработки XCode для Mac OS X.

И вместе с тем заметно улучшится и качество ваших приложений, так как они будут тестироваться на нескольких платформах, а ошибки исправляться в одном и том же исходном коде программы.

Qt — это луч надежды для программистов, пишущих на языке C++, которые вынуждены сейчас выполнять тройную работу по реализации своих приложений для ОС Windows, Linux и Mac OS X. Выбор в пользу Qt избавит вас от этих проблем. Qt предоставляет поддержку большого числа операционных систем: Microsoft Windows, Mac OS X, Linux, FreeBSD и других клонов UNIX с X11, а также и для мобильных операционных систем iOS, Android, Windows Phone, Windows RT и BlackBerry. Более того, благодаря встраиваемому пакету Qt Embedded все возможности Qt доступны также и в интегрированных системах (Embedded Systems). Qt использует интерфейс API низкого уровня, что позволяет кросс-платформенным приложениям работать столь же эффективно, как и приложениям, разработанным специально для конкретной платформы.

Несмотря на то, что предоставляемая платформенезависимость является одной из самых заманчивых возможностей библиотеки, многие разработчики используют Qt и для создания приложений, работающих только на одной платформе. Делают они это из тех соображений, что им нравится инструментарий и идейный подход самой библиотеки, который предоставляет им дополнительную гибкость и скорость разработки. А учитывая, что требования к программному продукту с течением времени постоянно подвергаются изменениям, не составит большой сложности при появлении необходимости предоставить продукт и для какой-либо иной платформы.

Использование в разработке разных компиляторов C++ еще больше повышает правильность и надежность кода ваших программ, поскольку предупреждающие сообщения и сообщения об ошибках вы станете получать от разных компиляторов, что будет делать код вашей программы с каждым разом все более совершенным.

Для ускорения и упрощения создания пользовательских интерфейсов Qt предоставляет программу Qt Designer, позволяющую делать это в интерактивном режиме. Очень сильно повысить скорость создания пользовательских интерфейсов можно также и при помощи технологии Qt Quick с описательным языком QML, модули и инструменты которой являются неотъемлемой частью Qt.

На сегодняшний день Qt — это продукт, широко используемый разработчиками всего мира. Компаний, ориентированных на эту библиотеку, более четырех тысяч. В число активных пользователей Qt входят такие компании, как: Adobe, Amazon, AMD, Bosch, BMW, Blackberry, Canon, Cisco Systems, Disney, Intel, IBM, Panasonic, Parallels, Pioneer, Philips, Oracle, HP, Goober, Google, Mercedes, NASA, NEC, Neonway, Nokia, Rakuten, Samsung, Siemens, Sony, SUN, Tesla, Xerox, Xilinx, Yamaha и др.

Используя сегодня ту или иную программу, вы, возможно, даже и не догадываетесь, что при ее написании была задействована библиотека Qt. Приведу лишь несколько, на мой взгляд, самых ярких примеров:

- ◆ рабочий стол KDE Software Compilation 4 (www.kde.org), используемый в Linux и FreeBSD (рис. В.1);
- ◆ редактор трехмерной графики Autodesk Maya (www.autodesk.com) (рис. В.2);
- ◆ приложение Viber (www.viber.com) компании Rakuten, предназначенное для голосовой связи VoIP (Voice Over IP) — звонков на обычные телефоны через Интернет (рис. В.3);
- ◆ мессенджер Telegram (www.telegram.org) от Telegram LLP, предназначенный для обмена сообщениями и файлами через Интернет, а также и для голосовой связи VoIP (рис. В.4);
- ◆ программа Adobe Photoshop Album (www.adobe.com) для обработки растровых изображений (рис. В.5);
- ◆ сетевая карта мира Google Earth (earth.google.com), которая позволяет рассматривать интересующие нас участки поверхности нашей планеты с высоты до 200 м (рис. В.6);

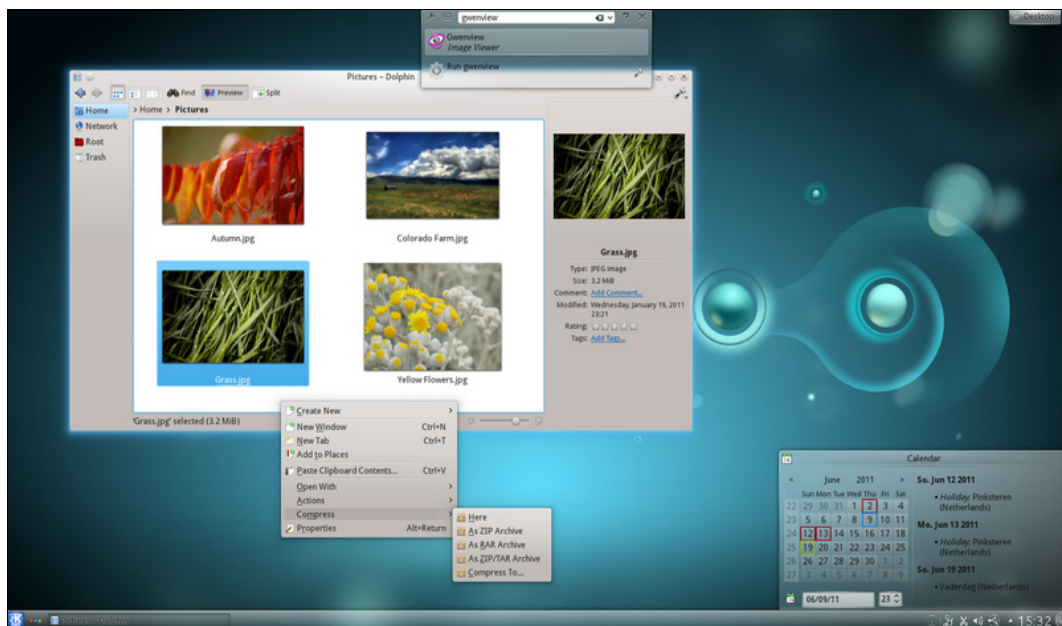


Рис. В.1. KDE Software Compilation 4 (взято с сайта www.wikipedia.org)

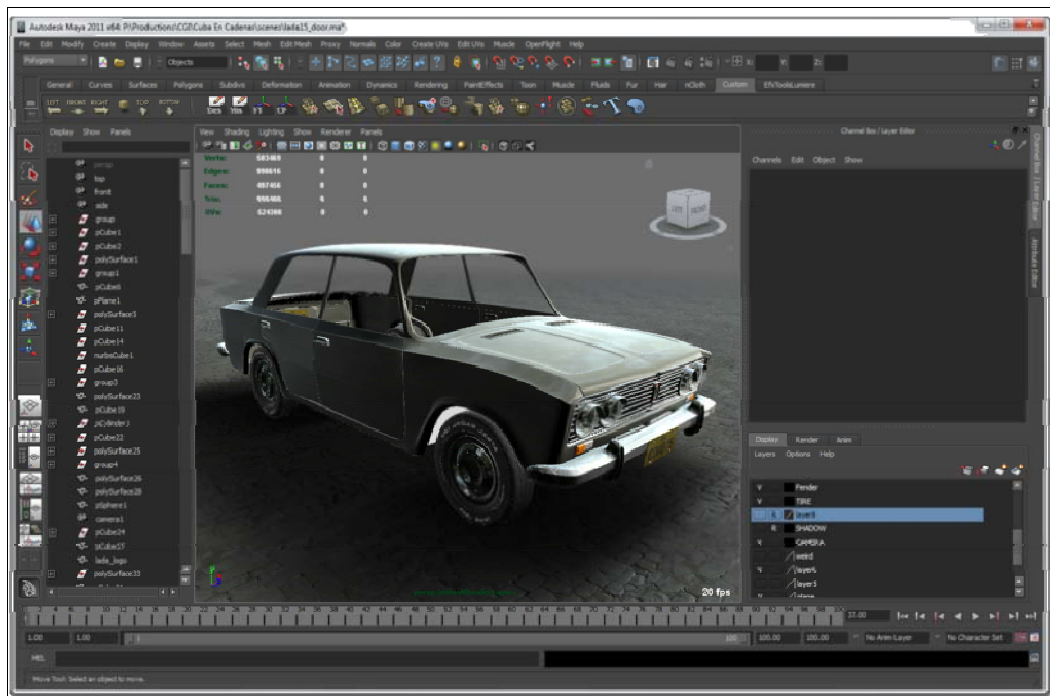


Рис. В.2. 3D-редактор Autodesk Maya (взято с сайта www.wikipedia.org)

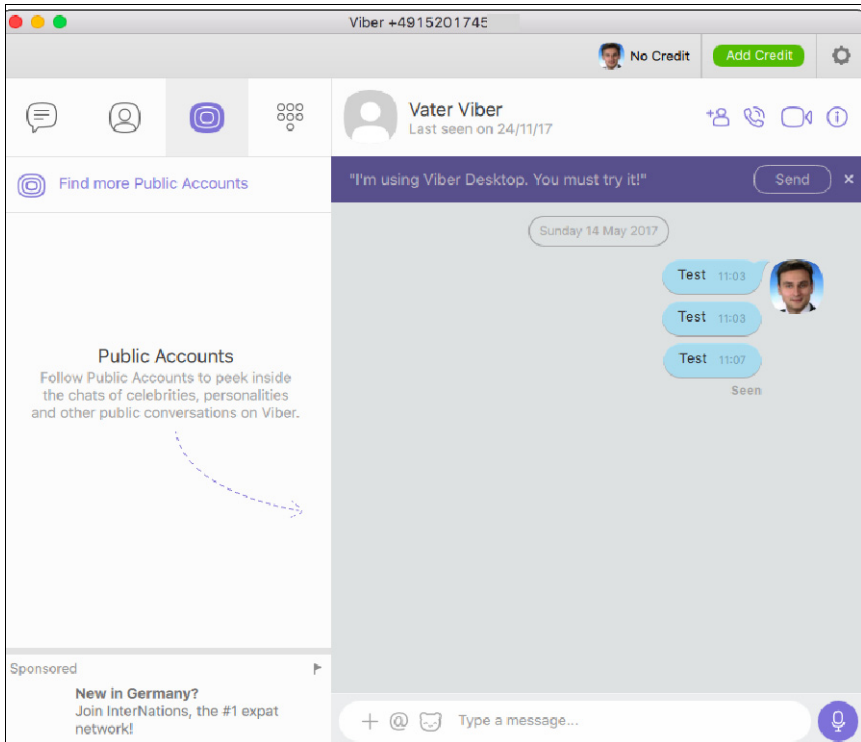


Рис. В.3. Приложение Viber от компании Rakuten

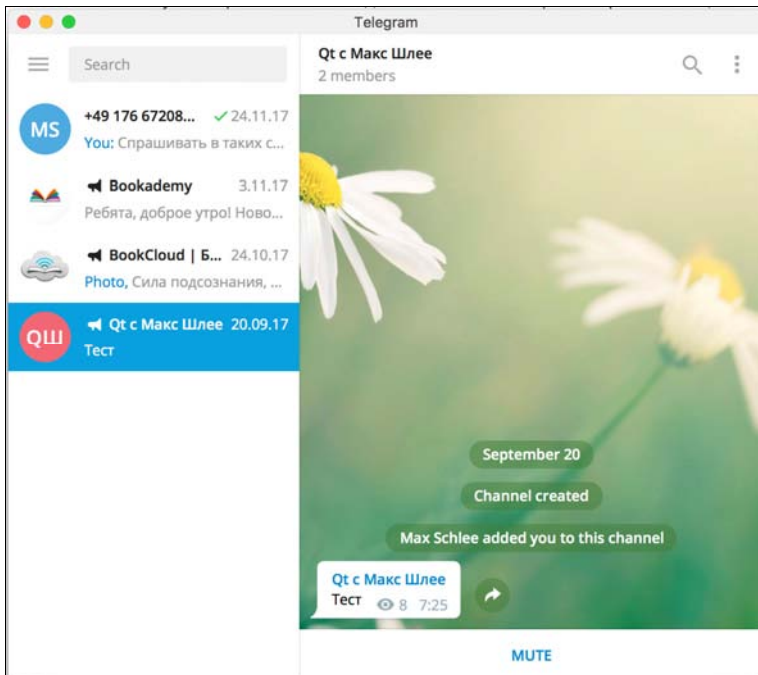


Рис. В.4. Мессенджер Telegram от Telegram LLC

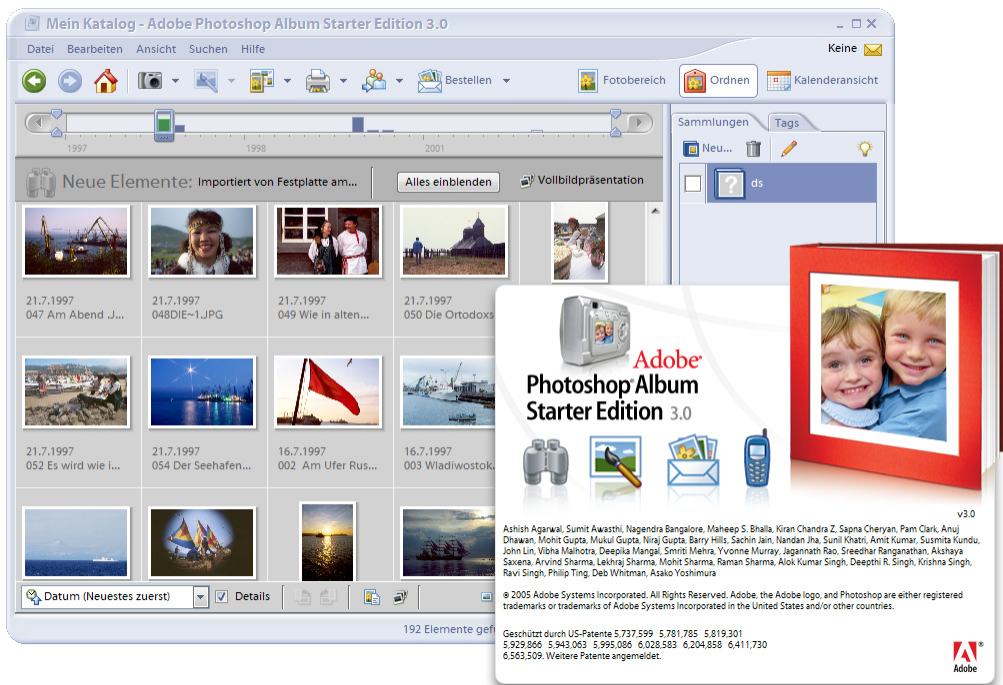


Рис. В.5. Программа обработки растровых изображений Adobe Photoshop Album

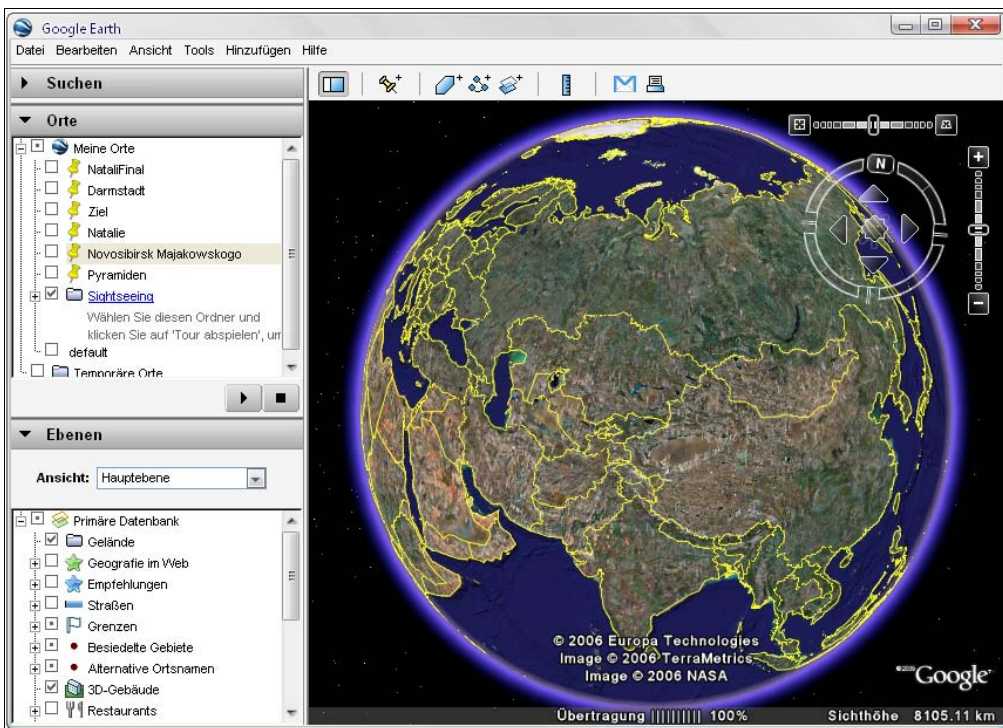


Рис. В.6. Сетевая карта мира Google Earth

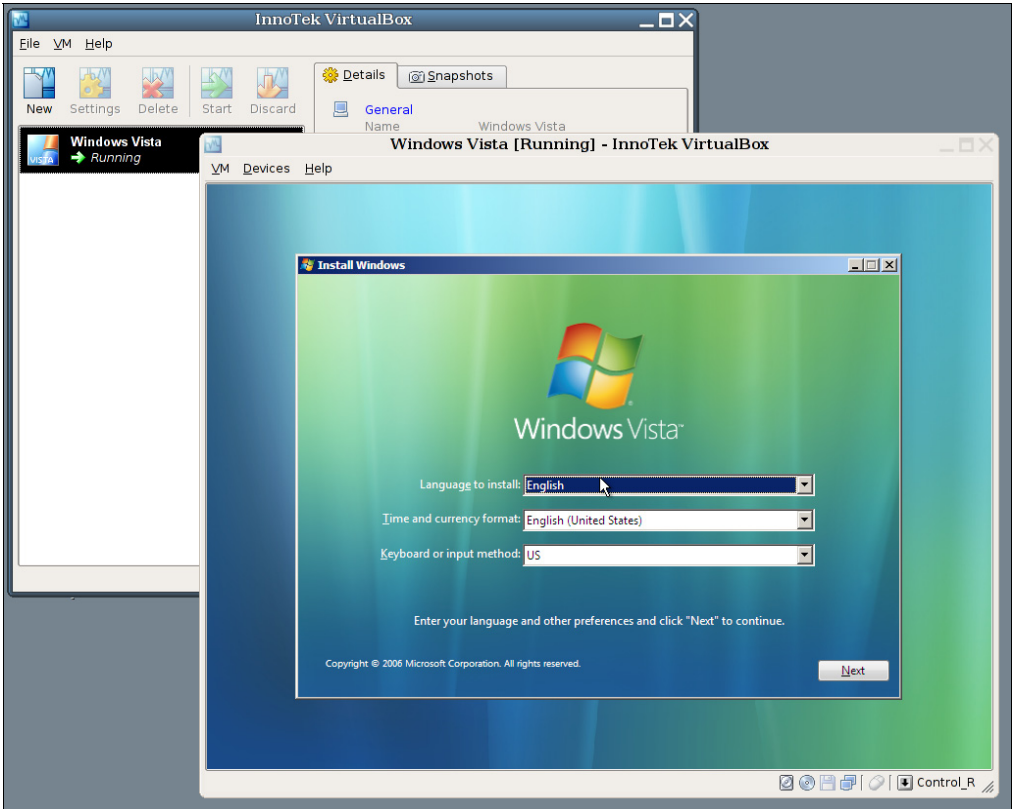


Рис. В.7. Эмулятор VirtualBox (взято с сайта www.virtualbox.org)

- ◆ программа для виртуализации операционных систем VirtualBox (www.virtualbox.org) от Sun Microsystems (рис. В.7);
- ◆ свободный проигрыватель VLC media player (www.videolan.org/vlc/), начиная с версии 0.9 (рис. В.8);
- ◆ программа для виртуализации операционных систем Parallels (www.parallels.com) от компании Parallels (рис. В.9);
- ◆ программа Kindle (рис. В.10) от компании Amazon (www.amazon.com), разработанная для загрузки, просмотра и чтения электронных книг, газет и журналов, купленных в магазине Kindle-Shop;
- ◆ программы официальных клиентов виртуальных валют Bitcoin (www.bitcoin.org) (рис. В.11) и Litecoin (www.litecoin.org).

Я не только пишу о библиотеке Qt, но и весьма интенсивно использую ее сам. За последние годы с моим личным участием было реализовано на Qt уже более 50 действующих проектов приложений, которые можно найти на странице моей компании и других компаний, в которых я когда-то работал, а также на Apple App Store, Google Play, Amazon Appstore и BlackBerry World. В связи с этим упомяну некоторые из таких Qt-проектов:

- ◆ программа PhotoGUN (www.neonway.ru/photogun/), предназначенная для пакетной обработки фотографий (рис. В.12);

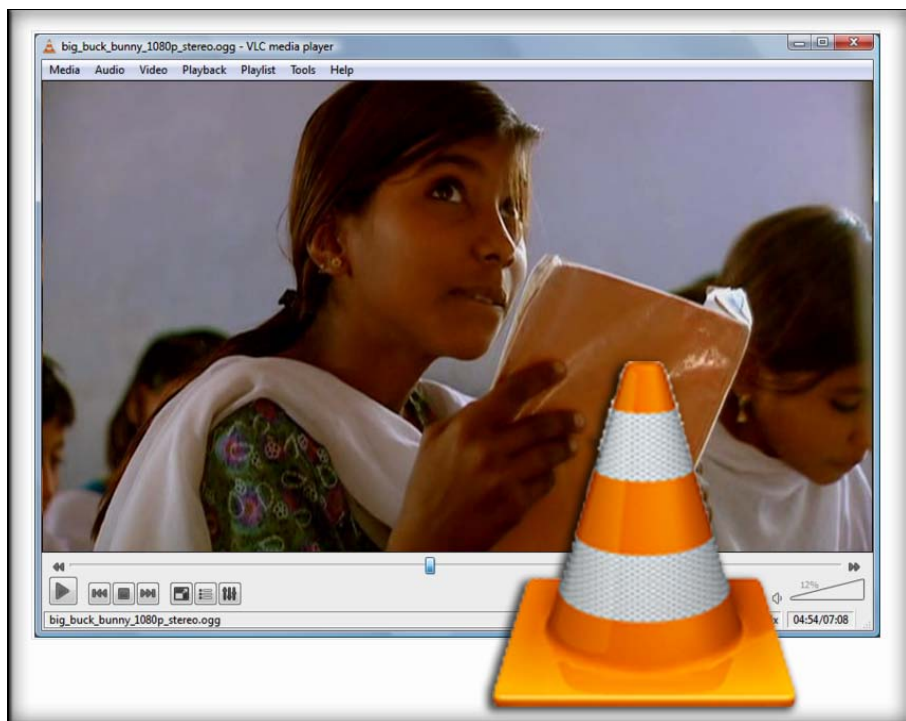


Рис. В.8. Проигрыватель VLC media player



Рис. В.9. Эмулятор Parallels

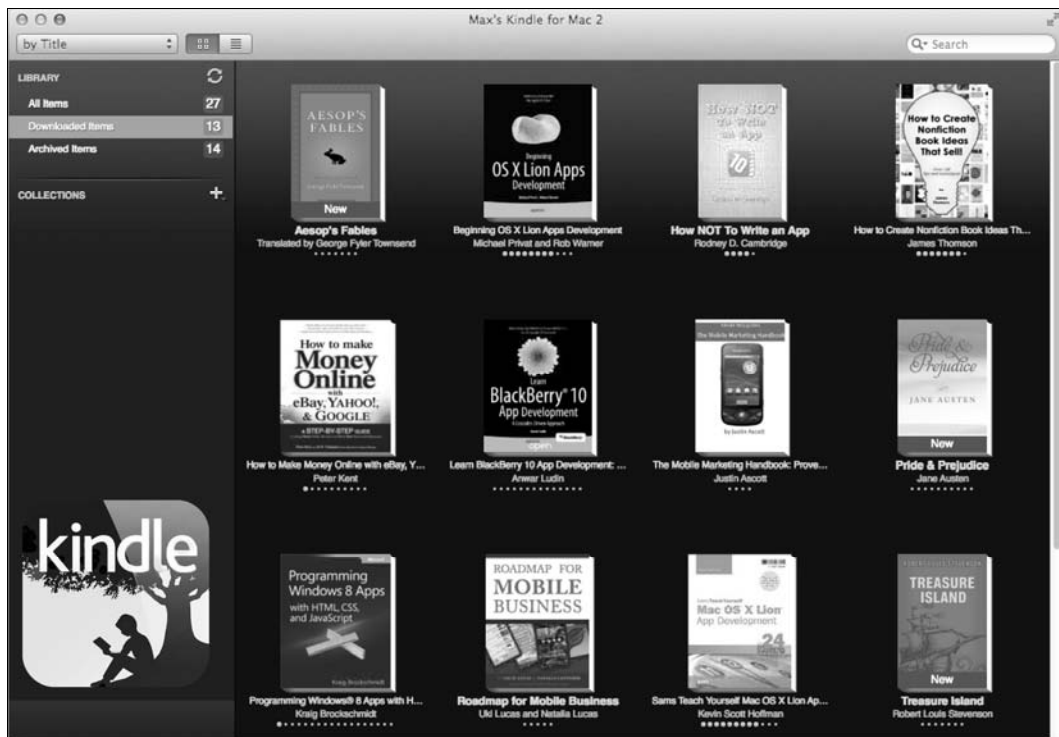


Рис. В.10. Программа Amazon Kindle

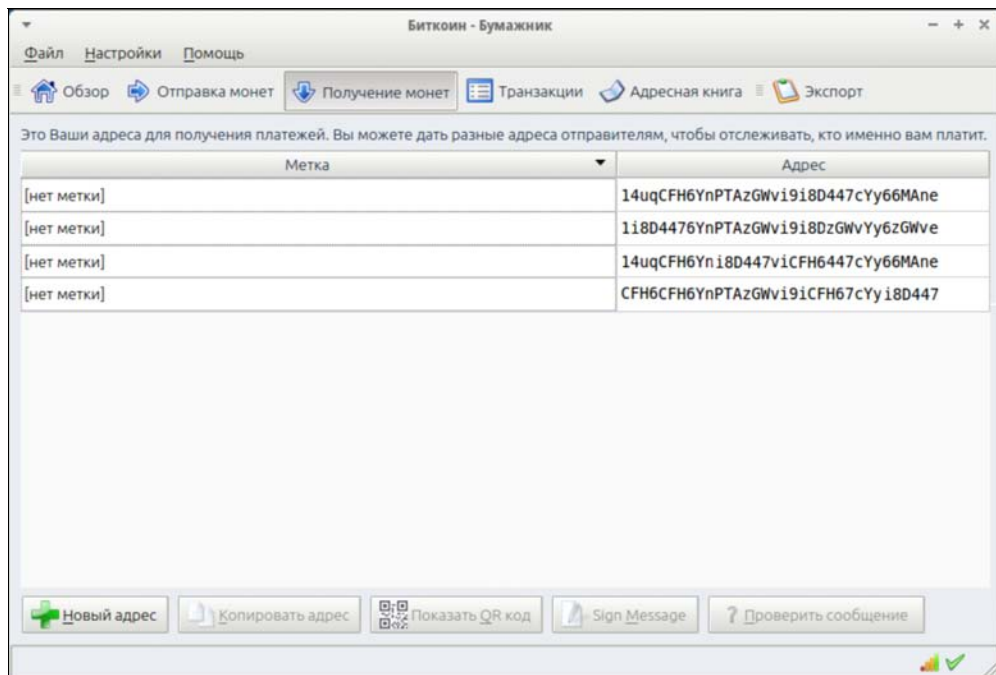


Рис. В.11. Клиент для виртуальной валюты Bitcoin



Рис. В.12. Программа PhotoGUN от Neonway



Рис. В.13. Программа QuickIcons от Neonway

- ♦ программа QuickIcons (www.neonway.com/apps/quickicons/) — для упрощения и ускорения процесса создания значков и стартовых экранов приложений на iOS, Android, Mac OS X, Windows Phone и других платформ (рис. В.13);
- ♦ ChordsMaestro (рис. В.14) — программа для обучения игре на семи музыкальных инструментах (www.neonway.ru/chordsmaestro/);
- ♦ программа FLEXXITY (www.dft-film.com/archive/flexxity_archive.php) от DigitalFilmTechnology Weiterstadt, предназначенная для редактирования, применения эффектов и реставрации видеоматериала (рис. В.15);
- ♦ программа ChatCube от Goober Networks, Inc. — для обеспечения текстовой, голосовой и видеосвязи через Интернет между компьютерами и смартфонами (рис. В.16).

В настоящее время Qt все чаще задействуется в разработке графических интерфейсов пользователя для автомобилей. Его использовали такие знаменитые фирмы, как Tesla, BMW, Mercedes и др. На рис. В.17 показана модель «Мерседес», представленная на конференции Qt World Summit 2017.

Многие привыкли считать, что Qt — лишь средство для создания только интерфейса пользователя. Это не так — Qt представляет собой полный инструментарий для программирования, который состоит из отдельных модулей и предоставляет:

- ♦ поддержку двух- и трехмерной графики (фактически, являясь стандартом для платформонезависимого программирования на OpenGL), а также имеет свою собственную альтернативную реализацию и собственный модуль Qt 3D;



Рис. В.14. Программа ChordsMaestro на iPad от Neonway



Рис. В.15. Программа FLEXXITY от DigitalFilmTechnology Weiterstadt



Рис. В.16. Программа ChatCube от Goober Networks, Inc.



Рис. В.17. Модель «Мерседес» на конференции Qt World Summit 2017

- ◆ возможность интернационализации, которая позволяет значительно расширить рынок сбыта ваших программ;
- ◆ использование форматов JSON (JavaScript Object Notation) и XML (eXtensible Markup Language);
- ◆ STL-совместимую библиотеку контейнеров;
- ◆ поддержку стандартных протоколов ввода/вывода;
- ◆ классы для работы с сетью;
- ◆ поддержку программирования баз данных, включая Oracle, Microsoft SQL Server, IBM DB2, MySQL, SQLite, Sybase, PostgreSQL;
- ◆ и многое другое.

Qt — полностью объектно-ориентированная библиотека. Новая концепция ведения меж-объектных коммуникаций, именуемая «сигналы и слоты», полностью заменяет применявшуюся ранее не вполне надежную модель обратных вызовов. Имеется также возможность обработки событий — например, нажатия клавиш клавиатуры, перемещения мыши и т. д.

Предоставляемая система расширений (plug-ins) позволяет создавать модули, расширяющие функциональные возможности создаваемых приложений. Эти расширения пользователи вашей программы могут получать не только от вас, но и от других разработчиков.

Несмотря на то, что библиотека Qt изначально создавалась для языка программирования C++, это вовсе не означает, что ее использование невозможно в других языках. Напротив, во многих языках программирования существуют модули для работы с этой библиотекой — например: Qt# в C#, PerlQt в Perl, PyQt в Python, PHP, Ruby и т. д.

Программы, реализованные с помощью Qt, могут использовать язык сценариев JavaScript. Эта технология позволяет пользователям вашего приложения расширить его возможности без изменения исходного кода и без перекомпилировки самого приложения изменить «поведение» приложения.

Qt прекрасно документирована, благодаря чему с помощью программы Qt Assistant вы всегда можете почерпнуть о ней любую интересующую вас информацию. А если и этого окажется недостаточно, то не забывайте, что Qt — библиотека с открытым исходным кодом (Open Source), и вы всегда можете взглянуть в него и детально разобраться в том, как работает та или иная часть этой библиотеки.

И если быть предельно кратким, то библиотеку Qt можно охарактеризовать в трех словах: Простота + Быстрота + Мощность.

Добро пожаловать в мир Qt 5!

*Макс Шлее
Дармштадт 9 декабря 2017 г.*



ЧАСТЬ I

ОСНОВЫ Qt

Вы не обязаны быть великим, чтобы начать, но обязаны начать, чтобы стать великим.

Джо Сабак

- Глава 1.** Обзор иерархии классов Qt
- Глава 2.** Философия объектной модели
- Глава 3.** Работа с Qt
- Глава 4.** Библиотека контейнеров



ГЛАВА 1

Обзор иерархии классов Qt

Если вы хотите знать территорию — нужно сначала изучить карту.

Тони Бьюзен

Первая программа на Qt

Для того чтобы написать и запустить первую программу на Qt, нужно обязательно установить саму библиотеку Qt и все необходимое для ее работы. И если вы этого еще не сделали, то о том, как это сделать, рассказано в *приложении 1* к этой книге.

Итак, теперь у вас для работы все готово! И это значит, что, как и заведено в самом начале знакомства, настало, наконец, время поздороваться, и, чтобы никого не оставить без внимания, мы обратимся, не больше и не меньше, а сразу ко всему миру. Давайте для этого напишем короткую программу «Hello, World» («Здравствуй, Мир»), результат выполнения которой показан на рис. 1.1.



Рис. 1.1. Окно программы «Hello, World»

Написание подобного рода программ стало уже традицией при знакомстве с новым языком или библиотекой. И хотя такой пример не в состоянии продемонстрировать весь потенциал и возможности самой библиотеки, он дает представление о базовых понятиях и позволяет оценить объем и сложность процесса реализации программ, использующих ту или иную библиотеку. Кроме того, на этом примере можно убедиться, что все необходимое для компиляции и компоновки установлено правильно.

Листинг 1.1. Программа «Hello, World» (файл hello.cpp)

```
#include <QtWidgets>
int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    QLabel lbl("Hello, World !");
    lbl.show();
    return app.exec();
}
```

ЭЛЕКТРОННЫЙ АРХИВ

Напомню, что электронный архив с примерами к этой книге можно скачать с FTP-сервера издательства «БХВ-Петербург» по ссылке <ftp://ftp.bhv.ru/9785977536783.zip> или со страницы книги на сайте www.bhv.ru (см. приложение 4). Файлы, упомянутые в названиях листингов, находятся в папках с номерами соответствующих глав.

В первой строке листинга 1.1 подключается заголовочный файл `QtWidgets`, представляющий собой модуль, включающий в себя заголовочные файлы для используемых в нашей программе классов: `QApplication` и `QLabel`. Конечно, мы могли бы обойтись и без модуля `QtWidgets`, а непосредственно подключить заголовочные файлы для поддержки классов `QApplication` и `QLabel`, но при большом количестве включаемых классов, задействованных в программе, читаемость самой программы заметно ухудшается. Кроме того, подключение заголовочного модуля `QtWidgets` ускоряет саму работу с кодом и, благодаря механизму предварительной компиляции заголовочных файлов (Precompiled Headers), не должно отразиться на скорости компиляции самой программы — конечно в том случае, если ваш компилятор этот механизм поддерживает.

Теперь давайте рассмотрим наш пример подробнее. В нем сначала создается объект класса `QApplication`, который и осуществляет управление приложением. Для его создания в конструктор этого класса необходимо передать два аргумента. Первый аргумент представляет собой информацию о количестве аргументов в командной строке, из которой происходит обращение к программе, а второй — это указатель на массив символьных строк, содержащих аргументы, по одному в строке. Любая использующая Qt-программа с графическим интерфейсом должна создавать только один объект этого класса, и он должен быть создан до использования операций, связанных с пользовательским интерфейсом.

Затем создается объект класса `QLabel`. После создания элементы управления Qt по умолчанию невидимы, и для их отображения необходимо вызвать метод `show()`. Объект класса `QLabel` является *основным управляющим элементом приложения*, что позволяет завершить работу приложения при закрытии окна элемента. Если вдруг окажется, что в созданном приложении имеется сразу несколько независимых друг от друга элементов управления, то при закрытии окна последнего такого элемента управления завершится и само приложение. Это правильно, иначе приложение осталось бы в памяти компьютера и расходовало бы его ресурсы.

Наконец, в последней строке программы приложение запускается вызовом `QApplication::exec()`. С его запуском приводится в действие цикл обработки событий, определенный в классе `QCoreApplication`, являющемся базовым для `QGuiApplication`, от которого унаследован класс `QApplication`. Этот цикл передает получаемые от системы события на обработку соответствующим объектам. Он продолжается до тех пор, пока либо не будет вызван статический метод `QCoreApplication::exit()`, либо не закроется окно последнего элемента управления. По завершении работы приложения метод `QApplication::exec()` возвращает значение целого типа, содержащее код, информирующий о его завершении.

Модули Qt

У программистов, начинающих изучение классов новой библиотеки, из-за большого объема информации, которую надо усвоить, зачастую создается ощущение перенасыщения. Но иерархия классов Qt имеет четкую внутреннюю структуру, которую важно сразу понять, чтобы потом уметь хорошо и интуитивно в этой библиотеке ориентироваться.

Библиотека Qt — это множество классов (около 1500), которые охватывают большую часть функциональных возможностей операционных систем, предоставляя разработчику мощные

механизмы, расширяющие и вместе с тем упрощающие разработку приложений. При этом не нарушается идеология операционной системы. Qt не является единым целым — она разбита на модули (табл. 1.1).

Таблица 1.1. Некоторые модули Qt

Библиотека	Обозначение в проектном файле	Назначение
QtCore	core	Основополагающий модуль, состоящий из классов, не связанных с графическим интерфейсом (см. <i>части I, IV</i>)
QtGui	gui	Модуль базовых классов для программирования графического интерфейса
QtWidgets	widgets	Модуль, дополняющий QtGui «строительным материалом» для графического интерфейса в виде виджетов на C++ (см. <i>части II, III, IV, V</i>)
QtQuick	quick	Модуль, содержащий описательный фреймворк для быстрого создания графического интерфейса (см. <i>часть VIII</i>)
QtQML	qml	Модуль, содержащий движок для языка QML и JavaScript (см. <i>часть VIII</i>)
QtNetwork	network	Модуль для программирования сети (см. <i>главу 39</i>)
QtSql	sql	Модуль для программирования баз данных (см. <i>главу 41</i>)
QtSvg	svg	Модуль для работы с SVG (Scalable Vector Graphics, масштабируемая векторная графика) (см. <i>главу 22</i>)
QtXml	xml	Модуль поддержки XML, классы, относящиеся к SAX и DOM (см. <i>главу 40</i>)
QtXmlPatterns	xmlpatterns	Модуль поддержки XPath, XQuery, XSLT и XmlSchemaValidator (см. <i>главу 40</i>)
QtMultimedia	multimedia	Модуль мультимедиа. Собрание классов для работы со звуком, видео, камерой и радио (см. <i>главу 27</i>)
QtMultimediaWidgets	multimediawidgets	Модуль с виджетами для модуля QtMultimedia (см. <i>главу 27</i>)
QPrintSupport	printsupport	Модуль для работы с принтером (см. <i>главу 24</i>)
QtTest	test	Модуль, содержащий классы для тестирования кода (см. <i>главу 45</i>)

Любая Qt-программа так или иначе должна использовать хотя бы один из модулей нашего примера из листинга 1.1 — это три модуля: QtCore, QtGui и QtWidgets, они присутствуют во всех программах с графическим интерфейсом и поэтому определены в программе создания make-файлов (см. *главу 3*) по умолчанию. Для использования других модулей в своих проектах необходимо перечислить их в проектном файле (см. *главу 3*). Например, чтобы добавить модуль, нужно написать:

```
QT += widgets network sql
```

А чтобы исключить модуль из проекта:

```
QT -= gui
```

Наиболее значимый из приведенных в табл. 1.1 модулей — это QtCore, так как он является базовым для всех остальных модулей. Далее идут модули, которые непосредственно зависят от QtCore, это: QtNetwork, QtGui, QtSql и QtXml.

Для каждого модуля Qt предоставляет отдельный заголовочный файл, содержащий заголовочные файлы всех классов этого модуля. Название такого заголовочного файла соответствует названию самого модуля. Например, для включения модуля QtWidgets нужно добавить в программу строку, как мы это уже сделали в листинге 1.1:

```
#include <QtWidgets>
```

Пространство имен Qt

Пространство имен Qt содержит ряд типов перечислений и констант, которые часто применяются при программировании. Если вам необходимо получить доступ к какой-либо константе этого пространства имен, то вы должны указать префикс Qt (например, не red, а Qt::red). Если вы все-таки хотите опускать префикс Qt, то необходимо в начале файла с исходным кодом добавить следующую директиву:

```
using namespace Qt;
```

Модуль QtCore

Как уже было сказано ранее, базовым является модуль QtCore. При этом он является базовым для приложений и не содержит классов, относящихся к интерфейсу пользователя. Если вы собираетесь реализовать консольное приложение, то, вполне возможно, можете ограничиться одним этим модулем. В модуль QtCore входят более 200 классов, вот некоторые из них:

- ◆ контейнерные классы: QList, QVector, QMap, QVariant, QString и т. д. (см. главу 4);
- ◆ классы для ввода и вывода: QIODevice, QTextStream, QFile (см. главу 36);
- ◆ классы процесса QProcess и для программирования многопоточности: QThread, QWaitCondition, QMutex (см. главу 38);
- ◆ классы для работы с таймером: QBasicTimer и QTimer (см. главу 37);
- ◆ классы для работы с датой и временем: QDate и QTime (см. главу 37);
- ◆ класс QObject, являющийся *краеугольным камнем* объектной модели Qt (см. главу 2);
- ◆ базовый класс событий QEvent (см. главу 14);
- ◆ класс для сохранения настроек приложения QSettings (см. главу 28);
- ◆ класс приложения QCoreApplication, из объекта которого, если требуется, можно запустить цикл событий;
- ◆ классы поддержки анимации: QAbstractAnimation, QVariantAnimation и т. д. (см. главу 22);
- ◆ классы для машины состояний: QStateMachine, QState и т. д. (см. главу 22);
- ◆ классы моделей интервью: QAbstractItemModel, QStringListModel, QAbstractProxyModel (см. главу 12).

Модуль содержит также механизмы поддержки файлов ресурсов (см. главу 3).