

Николай Прохоренко

**PRO**

**ПРОФЕССИОНАЛЬНОЕ  
ПРОГРАММИРОВАНИЕ**

# Qt 6

## Разработка оконных приложений на C++

Управление окном приложения

Обработка сигналов и событий

Размещение компонентов внутри окна

Основные компоненты

Списки и таблицы

Работа с графикой и изображениями

Графическая сцена

Диалоговые окна

SDI- и MDI-приложения

Редактор Qt Creator



Материалы  
на [www.bhv.ru](http://www.bhv.ru)



УДК 004.738.5+004.43

ББК 32.973.26-018.2

П84

**Прохоренок Н. А.**

П84 Qt 6. Разработка оконных приложений на C++. — СПб.: БХВ-Петербург, 2022. — 512 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-1180-3

Описываются базовые возможности библиотеки Qt, позволяющей создавать приложения с графическим интерфейсом на языке C++. Книга ориентирована на тех, кто уже знаком с языком программирования C++ и хотел бы научиться разрабатывать оконные приложения. Рассматриваются способы обработки событий, управление свойствами окна, создание формы, а также все основные компоненты (кнопки, текстовые поля, списки, таблицы, меню и др.) и варианты их размещения внутри окна. Книга содержит большое количество практических примеров, помогающих начать разрабатывать приложения с графическим интерфейсом самостоятельно. Она будет полезна в качестве самоучителя для начинающих разработчиков. Читатели, уже имеющие опыт, могут использовать ее как удобный справочник.

*Для программистов*

УДК 004.738.5+004.43

ББК 32.973.26-018.2

**Группа подготовки издания:**

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Людмила Гауль</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Зои Канторович</i>

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

ISBN 978-5-9775-1180-3

© ООО "БХВ", 2022

© Оформление. ООО "БХВ-Петербург", 2022

# Оглавление

<b>Введение</b> .....	<b>9</b>
<b>Глава 1. Первые шаги</b> .....	<b>11</b>
1.1. Создание структуры каталогов .....	12
1.2. Добавление пути в переменную <i>PATH</i> .....	12
1.3. Работа с командной строкой .....	14
1.4. Установка Qt 6 .....	14
1.5. Настройка редактора Qt Creator .....	23
1.6. Первая программа на Qt .....	27
1.7. Структура программы .....	32
1.8. Запуск приложения двойным щелчком на значке файла .....	35
1.9. ООП-стиль создания окна .....	36
1.10. Создание проекта с формой .....	39
1.11. Доступ к документации .....	45
<b>Глава 2. Работа с символами и строками</b> .....	<b>47</b>
2.1. Псевдонимы для элементарных типов .....	47
2.2. Класс <i>QChar</i> : символ в кодировке Unicode .....	48
2.2.1. Создание объекта .....	48
2.2.2. Изменение регистра символа .....	50
2.2.3. Проверка типа содержимого символа .....	51
2.3. Класс <i>QString</i> : строка в кодировке Unicode .....	53
2.3.1. Создание объекта .....	53
2.3.2. Преобразование объекта в другой тип данных .....	54
2.3.3. Получение и изменение размера строки .....	55
2.3.4. Доступ к отдельным символам .....	58
2.3.5. Перебор символов строки .....	59
2.3.6. Итераторы .....	60
2.3.7. Конкатенация строк .....	62
2.3.8. Добавление и вставка символов .....	63
2.3.9. Удаление символов .....	64
2.3.10. Изменение регистра символов .....	68
2.3.11. Получение фрагмента строки .....	69

2.3.12. Поиск в строке .....	70
2.3.13. Замена в строке .....	73
2.3.14. Сравнение строк.....	75
2.3.15. Преобразование строки в число .....	77
2.3.16. Преобразование числа в строку .....	79
2.3.17. Форматирование строки.....	80
2.3.18. Разделение строки на подстроки по разделителю.....	81
2.4. Класс <i>QStringList</i> : список строк .....	83
2.4.1. Создание объекта.....	83
2.4.2. Вставка элементов .....	84
2.4.3. Определение и изменение количества элементов.....	87
2.4.4. Удаление элементов .....	88
2.4.5. Доступ к элементам .....	91
2.4.6. Итераторы.....	93
2.4.7. Перебор элементов .....	95
2.4.8. Сортировка списка.....	95
2.4.9. Получение фрагмента списка .....	95
2.4.10. Поиск элементов .....	96
2.4.11. Замена элементов.....	98
2.4.12. Фильтрация списка .....	99
2.4.13. Преобразование списка в строку .....	100

### **Глава 3. Управление окном приложения .....** 101

3.1. Создание и отображение окна .....	101
3.2. Указание типа окна.....	103
3.3. Изменение и получение размеров окна .....	105
3.4. Местоположение окна на экране.....	109
3.4.1. Получение информации о размере экрана.....	110
3.5. Указание координат и размеров .....	113
3.5.1. Класс <i>QPoint</i> : координаты точки .....	113
3.5.2. Класс <i>QSize</i> : размеры прямоугольной области .....	115
3.5.3. Класс <i>QRect</i> : координаты и размеры прямоугольной области .....	118
3.6. Разворачивание и сворачивание окна .....	125
3.7. Управление прозрачностью окна .....	127
3.8. Модальные окна.....	128
3.9. Смена значка в заголовке окна .....	130
3.10. Изменение цвета фона окна .....	132
3.11. Использование изображения в качестве фона.....	134
3.12. Создание окна произвольной формы .....	136
3.13. Всплывающие подсказки .....	137
3.14. Закрытие окна из программы .....	139

### **Глава 4. Обработка сигналов и событий.....** 141

4.1. Назначение обработчиков сигналов.....	141
4.2. Блокировка и удаление обработчика .....	147
4.3. Генерация сигнала из программы .....	151
4.4. Использование таймеров.....	153
4.5. Класс <i>QTimer</i> : таймер.....	156
4.6. Перехват всех событий.....	160

4.7. События окна .....	164
4.7.1. Изменение состояния окна.....	164
4.7.2. Изменение положения окна и его размеров .....	167
4.7.3. Перерисовка окна или его части.....	169
4.7.4. Предотвращение закрытия окна .....	170
4.8. События клавиатуры .....	171
4.8.1. Установка фокуса ввода.....	172
4.8.2. Назначение клавиш быстрого доступа.....	177
4.8.3. Нажатие и отпускание клавиши клавиатуры .....	181
4.9. События мыши.....	183
4.9.1. Нажатие и отпускание кнопки мыши.....	183
4.9.2. Перемещение указателя .....	185
4.9.3. Наведение и выведение указателя .....	186
4.9.4. Прокрутка колесика мыши.....	186
4.9.5. Изменение внешнего вида указателя мыши .....	187
4.10. Технология drag & drop.....	190
4.10.1. Запуск перетаскивания .....	190
4.10.2. Класс <i>QMimeData</i> .....	192
4.10.3. Обработка сброса.....	195
4.11. Работа с буфером обмена.....	198
<b>Глава 5. Размещение нескольких компонентов в окне .....</b>	<b>199</b>
5.1. Абсолютное позиционирование .....	199
5.2. Горизонтальное и вертикальное выравнивание .....	200
5.3. Выравнивание по сетке .....	204
5.4. Выравнивание компонентов формы .....	206
5.5. Классы <i>QStackedLayout</i> и <i>QStackedWidget</i> .....	209
5.6. Класс <i>QSizePolicy</i> .....	210
5.7. Объединение компонентов в группу .....	212
5.8. Панель с рамкой.....	214
5.9. Панель с вкладками .....	216
5.10. Компонент «аккордеон» .....	221
5.11. Панели с изменяемым размером .....	224
5.12. Область с полосами прокрутки .....	226
<b>Глава 6. Основные компоненты .....</b>	<b>229</b>
6.1. Надпись.....	229
6.2. Командная кнопка .....	234
6.3. Переключатель.....	237
6.4. Флажок .....	238
6.5. Однострочное текстовое поле .....	239
6.5.1. Основные методы и сигналы .....	239
6.5.2. Ввод данных по маске .....	245
6.5.3. Контроль ввода .....	246
6.6. Многострочное текстовое поле .....	247
6.6.1. Основные методы и сигналы .....	247
6.6.2. Изменение настроек поля.....	251
6.6.3. Изменение характеристик текста и фона.....	254
6.6.4. Класс <i>QTextDocument</i> .....	256
6.6.5. Класс <i>QTextCursor</i> .....	260

6.7. Текстовый браузер.....	265
6.8. Поля для ввода целых и вещественных чисел.....	267
6.9. Поля для ввода даты и времени.....	270
6.10. Календарь .....	273
6.11. Электронный индикатор .....	276
6.12. Индикатор процесса .....	277
6.13. Шкала с ползунком.....	279
6.14. Класс <i>QDial</i> .....	281
6.15. Полоса прокрутки.....	282

## **Глава 7. Списки и таблицы..... 283**

7.1. Раскрывающийся список.....	283
7.1.1. Добавление, изменение и удаление элементов .....	283
7.1.2. Изменение настроек.....	285
7.1.3. Поиск элемента внутри списка .....	287
7.1.4. Сигналы .....	288
7.2. Список для выбора шрифта .....	288
7.3. Роли элементов .....	289
7.4. Модели.....	290
7.4.1. Доступ к данным внутри модели.....	290
7.4.2. Класс <i>QStringListModel</i> .....	292
7.4.3. Класс <i>QStandardItemModel</i> .....	293
7.4.4. Класс <i>QStandardItem</i> .....	297
7.5. Представления .....	302
7.5.1. Класс <i>QAbstractItemView</i> .....	302
7.5.2. Простой список .....	307
7.5.3. Таблица.....	309
7.5.4. Иерархический список .....	312
7.5.5. Управление заголовками строк и столбцов.....	315
7.6. Управление выделением элементов.....	319
7.7. Промежуточные модели.....	322

## **Глава 8. Работа с графикой ..... 325**

8.1. Вспомогательные классы .....	325
8.1.1. Класс <i>QColor</i> : цвет .....	326
8.1.2. Класс <i>QPen</i> : перо .....	332
8.1.3. Класс <i>QBrush</i> : кисть .....	333
8.1.4. Класс <i>QLine</i> : линия.....	334
8.1.5. Класс <i>QPolygon</i> : многоугольник .....	335
8.1.6. Класс <i>QFont</i> : шрифт.....	337
8.2. Класс <i>QPainter</i> .....	340
8.2.1. Рисование линий и фигур.....	341
8.2.2. Вывод текста .....	344
8.2.3. Вывод изображения.....	346
8.2.4. Преобразование систем координат .....	348
8.2.5. Сохранение команд рисования в файл.....	349
8.3. Работа с изображениями .....	350
8.3.1. Класс <i>QPixmap</i> .....	350
8.3.2. Класс <i>QBitmap</i> .....	354

8.3.3. Класс <i>QImage</i> .....	355
8.3.4. Класс <i>QIcon</i> .....	360
<b>Глава 9. Графическая сцена.....</b>	<b>363</b>
9.1. Класс <i>QGraphicsScene</i> : сцена .....	363
9.1.1. Настройка параметров сцены .....	364
9.1.2. Добавление и удаление графических объектов.....	365
9.1.3. Добавление компонентов на сцену.....	366
9.1.4. Поиск объектов .....	367
9.1.5. Управление фокусом ввода.....	368
9.1.6. Управление выделением объектов .....	369
9.1.7. Прочие методы и сигналы.....	370
9.2. Класс <i>QGraphicsView</i> : представление.....	371
9.2.1. Настройка параметров представления .....	372
9.2.2. Преобразования между координатами представления и сцены .....	373
9.2.3. Поиск объектов .....	374
9.2.4. Трансформация систем координат.....	375
9.2.5. Прочие методы.....	375
9.3. Класс <i>QGraphicsItem</i> : базовый класс для графических объектов .....	376
9.3.1. Настройка параметров объекта.....	377
9.3.2. Трансформация объекта.....	381
9.3.3. Прочие методы.....	381
9.4. Графические объекты.....	382
9.4.1. Линия .....	383
9.4.2. Класс <i>QAbstractGraphicsShapeItem</i> .....	383
9.4.3. Прямоугольник .....	384
9.4.4. Многоугольник .....	384
9.4.5. Эллипс.....	384
9.4.6. Изображение .....	385
9.4.7. Простой текст.....	386
9.4.8. Форматированный текст .....	387
9.5. Группировка объектов.....	389
9.6. Эффекты .....	389
9.6.1. Класс <i>QGraphicsEffect</i> .....	390
9.6.2. Тень.....	390
9.6.3. Размытие.....	392
9.6.4. Изменение цвета .....	392
9.6.5. Изменение прозрачности .....	393
9.7. Обработка событий.....	394
9.7.1. События клавиатуры.....	394
9.7.2. События мыши.....	395
9.7.3. Обработка перетаскивания и сброса .....	399
9.7.4. Фильтрация событий .....	401
9.7.5. Обработка изменения состояния объекта .....	401
<b>Глава 10. Диалоговые окна .....</b>	<b>405</b>
10.1. Пользовательские диалоговые окна .....	405
10.2. Класс <i>QDialogButtonBox</i> .....	408

10.3. Класс <i>QMessageBox</i> .....	411
10.3.1. Основные методы и сигналы .....	413
10.3.2. Окно для вывода обычного сообщения .....	415
10.3.3. Окно запроса подтверждения .....	415
10.3.4. Окно для вывода предупреждающего сообщения .....	416
10.3.5. Окно для вывода критического сообщения .....	417
10.3.6. Окно «О программе» .....	418
10.3.7. Окно «About Qt» .....	418
10.4. Класс <i>QInputDialog</i> .....	418
10.4.1. Основные методы и сигналы .....	419
10.4.2. Окна для ввода строки .....	422
10.4.3. Окно для ввода целого числа .....	423
10.4.4. Окно для ввода вещественного числа .....	424
10.4.5. Окно для выбора пункта из списка .....	425
10.5. Класс <i>QFileDialog</i> .....	426
10.5.1. Основные методы и сигналы .....	426
10.5.2. Окно для выбора каталога .....	429
10.5.3. Окно для открытия файла .....	430
10.5.4. Окно для сохранения файла .....	431
10.6. Окно для выбора цвета .....	432
10.7. Окно для выбора шрифта .....	433
10.8. Окно для вывода сообщения об ошибке .....	434
10.9. Окно с индикатором хода процесса .....	435
10.10. Создание многостраничного мастера .....	437
10.10.1. Класс <i>QWizard</i> .....	437
10.10.2. Класс <i>QWizardPage</i> .....	442
<b>Глава 11. Создание SDI- и MDI-приложений</b> .....	<b>447</b>
11.1. Создание главного окна приложения .....	447
11.2. Меню .....	454
11.2.1. Класс <i>QMenuBar</i> .....	454
11.2.2. Класс <i>QMenu</i> .....	455
11.2.3. Контекстное меню .....	459
11.2.4. Класс <i>QAction</i> .....	460
11.2.5. Объединение переключателей в группу .....	464
11.3. Панели инструментов .....	466
11.3.1. Класс <i>QToolBar</i> .....	466
11.3.2. Класс <i>QToolButton</i> .....	469
11.4. Прикрепляемые панели .....	470
11.5. Управление строкой состояния .....	472
11.6. MDI-приложения .....	474
11.6.1. Класс <i>QMdiArea</i> .....	474
11.6.2. Класс <i>QMdiSubWindow</i> .....	478
11.7. Добавление значка приложения в область уведомлений .....	479
<b>Заключение</b> .....	<b>483</b>
<b>Приложение. Описание электронного архива</b> .....	<b>485</b>
<b>Предметный указатель</b> .....	<b>487</b>





# ГЛАВА 1

## Первые шаги

Прежде всего необходимо сделать два замечания:

1. Имя пользователя компьютера должно состоять только из латинских букв. Никаких русских букв и пробелов, т. к. многие программы сохраняют различные настройки и временные файлы в каталоге `C:\Users\<Имя пользователя>`. Если имя пользователя содержит русские буквы, то они могут быть искажены до неузнаваемости из-за неправильного преобразования кодировок и программа не сможет сохранить настройки. Помните, что в разных кодировках русские буквы могут иметь разный код. Разработчики программ в основном работают с английским языком и ничего не знают о проблемах с кодировками, т. к. во всех однобайтовых кодировках и в кодировке UTF-8 коды латинских букв одинаковы. Так что, если хотите без проблем заниматься программированием, от использования русских букв в имени пользователя лучше отказаться.
2. Имена каталогов и файлов в пути не должны содержать русских букв и пробелов. Допустимы только латинские буквы, цифры, тире, подчеркивание и некоторые другие символы. С русскими буквами та же проблема, что описана в предыдущем пункте. При наличии пробелов в пути обычно требуется дополнительно заключать путь в кавычки. Если этого не сделать, то путь будет обрезан до первого встретившегося пробела. Такая проблема будет возникать при сборке и компиляции программ из командной строки.

Соблюдение этих двух простых правил позволит избежать множества проблем в дальнейшем при сборке и компиляции программ сторонних разработчиков.

### **ПРИМЕЧАНИЕ**

Листинги для всех глав книги доступны отдельно. Во-первых, большие листинги в книге не смотрятся, а во-вторых, это позволило уменьшить объем всей книги, т. к. объем листингов превышает объем книги. Информация о способе получения дополнительных листингов доступна на обложке книги и в *приложении*.

## 1.1. Создание структуры каталогов

Перед установкой программ создадим следующую структуру каталогов:

```
book
cpp
    projectsQt
    lib
```

Каталоги `book` и `cpp` лучше разместить в корне какого-либо диска. В моем случае это будет диск `C:`, следовательно, пути к содержимому каталогов — `C:\book` и `C:\cpp`. Можно создать каталоги и в любом другом месте, но в пути не должно быть русских букв и пробелов — только латинские буквы, цифры, тире и подчеркивание. Остальных символов лучше избегать, если не хотите проблем с компиляцией и запуском программ.

В каталоге `C:\book` мы станем размещать наши тестовые программы. Внутри каталога `C:\cpp` у нас созданы два вложенных каталога:

- `projectsQt` — в этом каталоге мы станем сохранять проекты из редактора Qt Creator;
- `lib` — путь к этому каталогу мы добавим в системную переменную `PATH` и будем размещать в нем различные библиотеки, которые потребуются для наших программ.

## 1.2. Добавление пути в переменную `PATH`

Когда мы в командной строке вводим название программы без предварительного указания пути к ней, то сначала поиск программы выполняется в текущем рабочем каталоге (обычно это каталог, из которого запускается программа), а затем в путях, указанных в системной переменной `PATH`. Аналогично выполняется поиск библиотек динамической компоновки при запуске программы с помощью двойного щелчка на значке файла, но системные каталоги имеют более высокий приоритет, чем каталоги, указанные в переменной `PATH`. Пути в системной переменной `PATH` просматриваются слева направо до первого нахождения искомого объекта. Так что, если в путях расположено несколько объектов с одинаковыми именами, мы получим только первый найденный объект. Поэтому если вдруг запустилась другая программа, то следует либо удалить путь, ведущий к другой программе, либо переместить новый путь в самое начало системной переменной `PATH`.

Давайте добавим путь к каталогу `C:\cpp\lib` в переменную `PATH`. Чтобы изменить системную переменную в Windows, переходим в **Параметры | Панель управления | Система и безопасность | Система | Дополнительные параметры системы**. В результате откроется окно **Свойства системы** (рис. 1.1). На вкладке **Дополнительно** нажимаем кнопку **Переменные среды**. В открывшемся окне (рис. 1.2) в списке **Системные переменные** выделяем строку с переменной `Path` и нажимаем кнопку **Изменить**. Добавляем путь к каталогу `C:\cpp\lib` и сохраняем изменения.

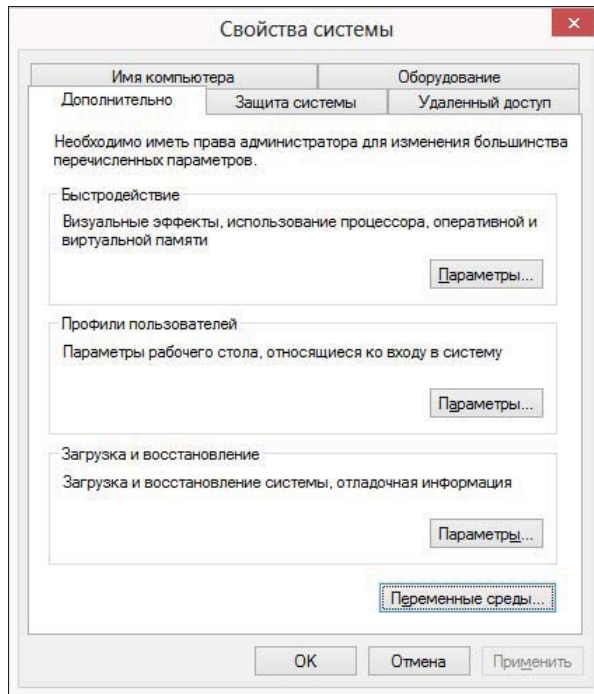


Рис. 1.1. Окно Свойства системы

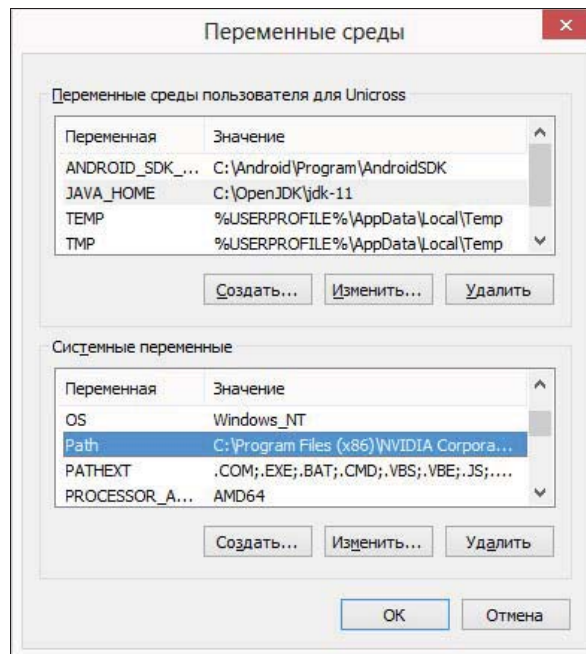


Рис. 1.2. Окно Переменные среды

Добавлять пути в переменную `PATH` мы будем несколько раз, поэтому способ изменения значения этой системной переменной нужно знать наизусть.

### **ВНИМАНИЕ!**

Случайно не удалите существующее значение переменной `PATH`, иначе другие приложения перестанут запускаться.

## 1.3. Работа с командной строкой

При изучении материала мы часто будем пользоваться приложением Командная строка. Вполне возможно, что вы никогда не пользовались командной строкой и не знаете, как запустить это приложение. Давайте рассмотрим некоторые способы его запуска в Windows.

- Через поиск находим приложение Командная строка.
- Нажимаем комбинацию клавиш `<Windows>+<R>`. В открывшемся окне вводим `cmd` и нажимаем кнопку **ОК**.
- Находим файл `cmd.exe` в каталоге `C:\Windows\System32`.
- В Проводнике щелкаем правой кнопкой мыши на свободном месте списка файлов, удерживая при этом нажатой клавишу `<Shift>`, и из контекстного меню выбираем пункт **Открыть окно команд**.
- В Проводнике в адресной строке вводим `cmd` и нажимаем клавишу `<Enter>`.

В некоторых случаях для выполнения различных команд могут потребоваться права администратора. Чтобы запустить командную строку с правами администратора, через поиск находим приложение Командная строка, щелкаем на значке правой кнопкой мыши и затем выбираем пункт **Запуск от имени администратора**.

Заучите способы запуска командной строки наизусть. В дальнейшем мы просто будем говорить «запустите командную строку» без уточнения, как это сделать.

## 1.4. Установка Qt 6

Для загрузки библиотеки Qt переходим на сайт <https://www.qt.io/> и нажимаем кнопку **Download**. Далее нажимаем кнопку **Go open source** из раздела **Downloads for open source users**. На следующей странице нажимаем кнопку **Download the Qt Online Installer**, а затем кнопку **Download**. Скачиваем файл `qt-unified-windows-x86-4.1.0-online.exe` и запускаем программу установки. Обратите внимание: для установки библиотеки потребуется активное подключение к сети Интернет.

### **СИСТЕМНЫЕ ТРЕБОВАНИЯ**

Для корректной работы Qt 6 в Windows требуется операционная система Windows 10 64-bit. Полный список поддерживаемых платформ можно посмотреть на странице <https://doc.qt.io/qt-6/supported-platforms.html>.

После запуска программы установки отобразится окно **Qt Setup** (Установка Qt), показанное на рис. 1.3. Вводим регистрационные данные или создаем новый аккаунт. Нажимаем кнопку **Next** (Далее). Если создавался новый аккаунт, то на адрес электронной почты придет письмо со ссылкой, позволяющей подтвердить адрес. Переходим по ссылке, а затем возвращаемся к программе установки и нажимаем кнопку **Next**.

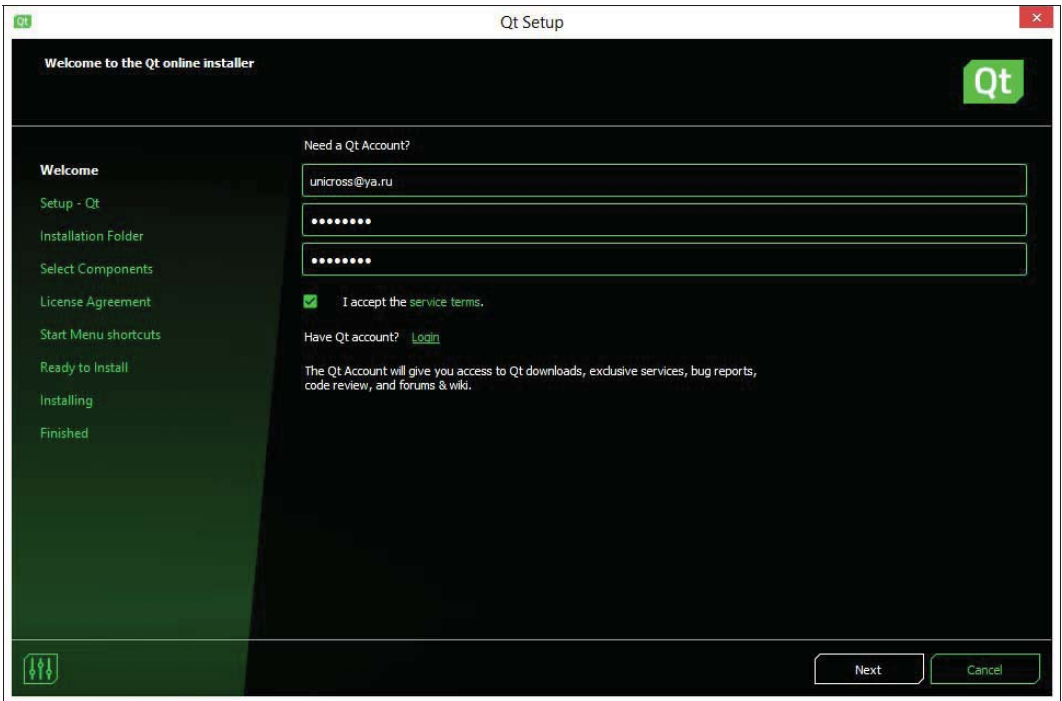


Рис. 1.3. Установка Qt. Шаг 1

На следующем шаге (рис. 1.4) устанавливаем флажки и нажимаем кнопку **Next**. В следующем окне (рис. 1.5) нажимаем кнопку **Next**. Далее нужно выбрать один из переключателей (рис. 1.6). Выбираем и нажимаем кнопку **Next**. На следующем шаге (рис. 1.7) должен быть указан путь `C:\Qt` и установлен флажок **Custom installation**. Нажимаем кнопку **Next**.

В следующем окне (рис. 1.8) устанавливаем флажок **Latest releases** и нажимаем кнопку **Filter** (Отфильтровать). Затем в разделе **Qt** раскрываем список **Qt 6.1.0** и устанавливаем флажок **MinGW 8.1.0 64-bit**. В разделе **Developer and Designer Tools** (рис. 1.9) устанавливаем флажки: **Qt Creator 4.15.0 CDB Debugger Support**, **Debugging Tools for Windows** и **MinGW 8.1.0 64-bit**. Нажимаем кнопку **Next**. В открывшемся окне (рис. 1.10) принимаем лицензионное соглашение и нажимаем кнопку **Next**. Затем (рис. 1.11) опять нажимаем кнопку **Next**. Нажимаем кнопку **Install** (Установить) для запуска установки (рис. 1.12). После завершения установки нажимаем кнопку **Finish** (Завершить) (рис. 1.13).

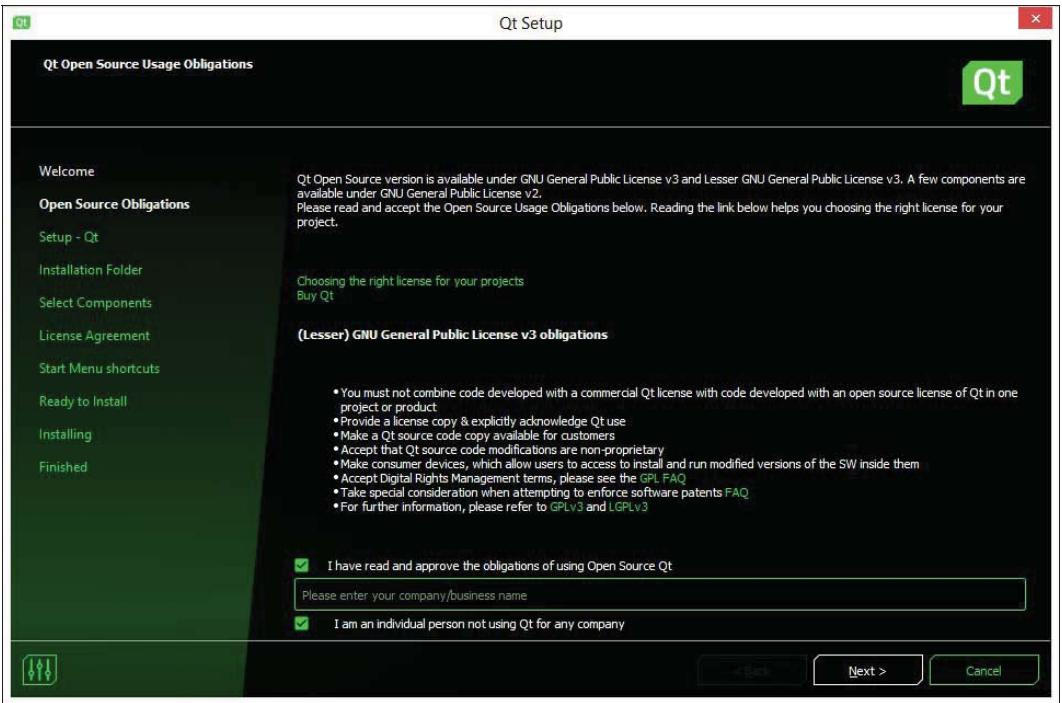


Рис. 1.4. Установка Qt. Шаг 2

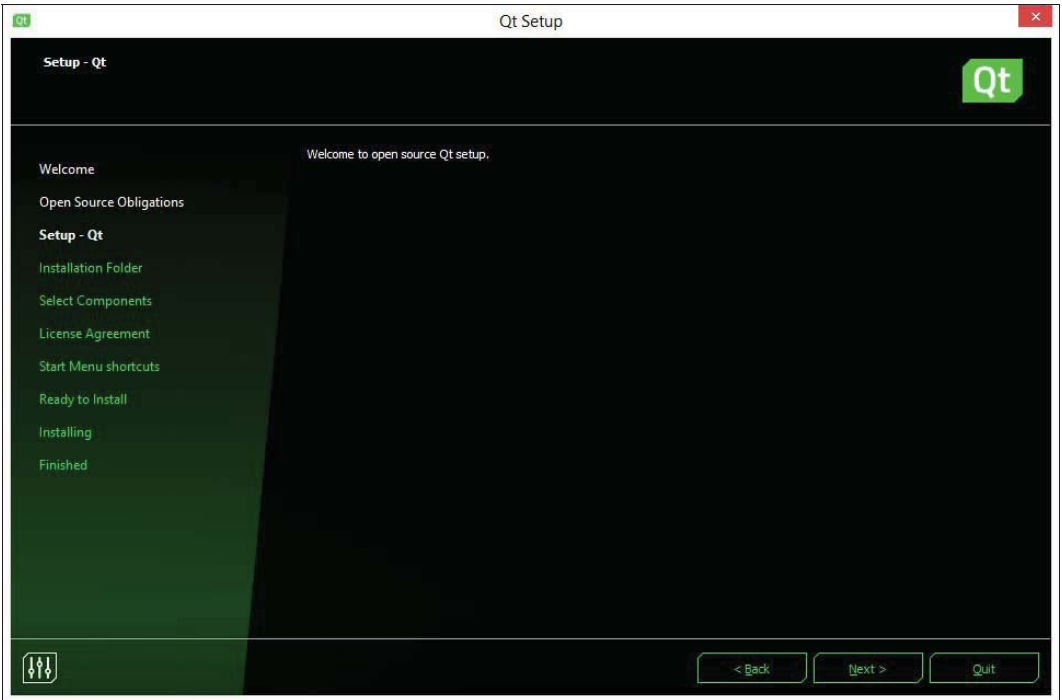


Рис. 1.5. Установка Qt. Шаг 3

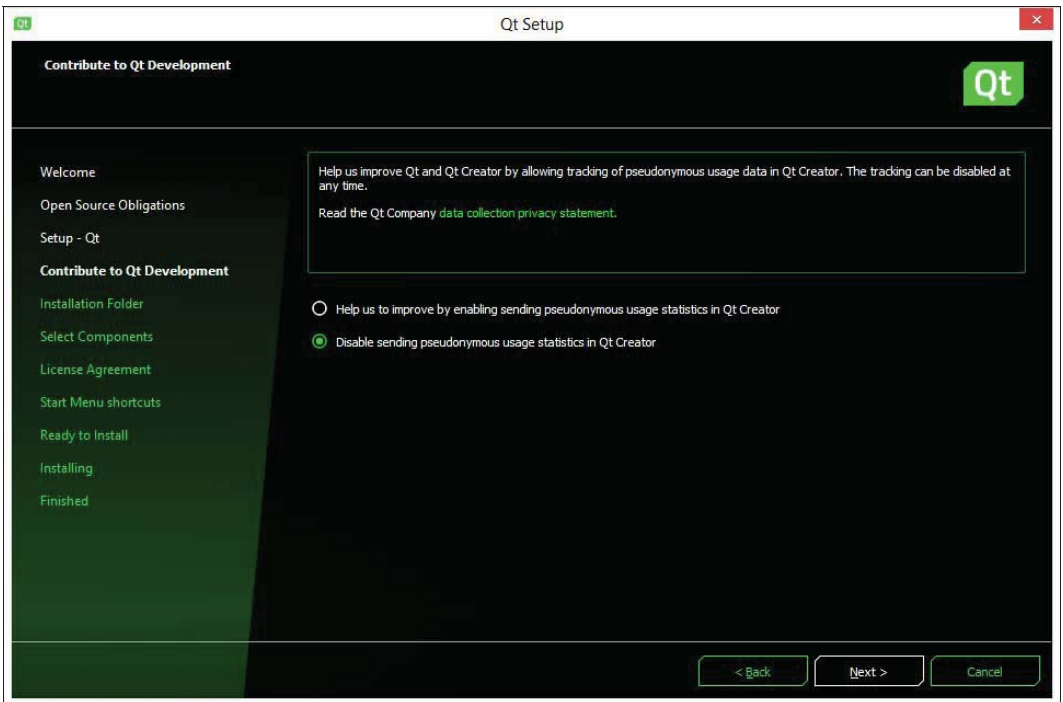


Рис. 1.6. Установка Qt. Шаг 4

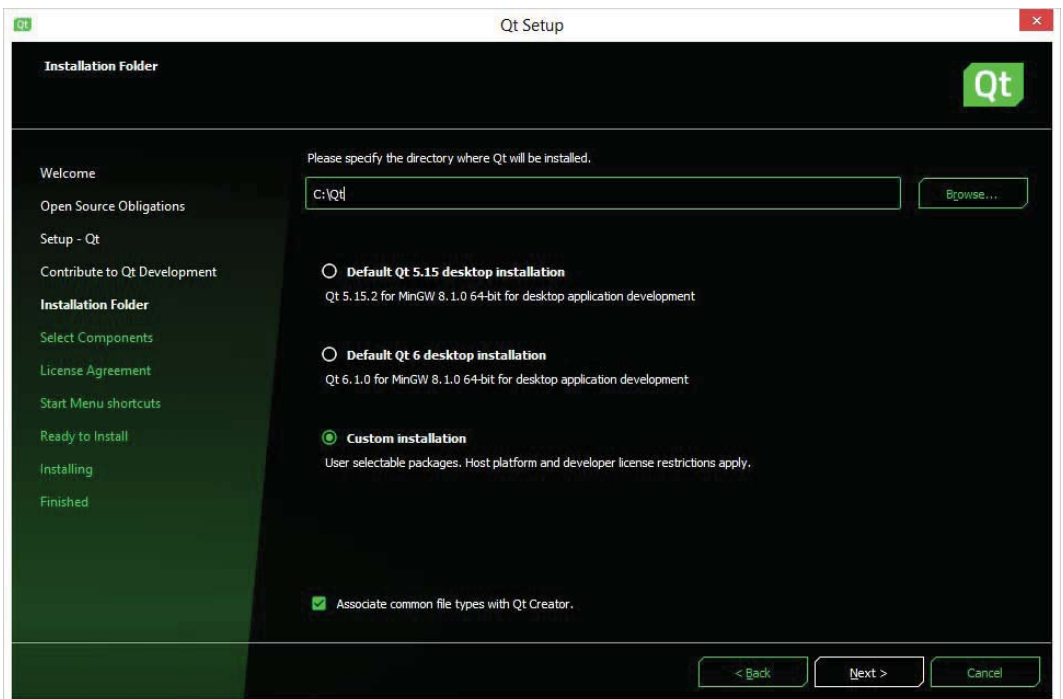


Рис. 1.7. Установка Qt. Шаг 5



Рис. 1.8. Установка Qt. Шаг 6. Раздел Qt

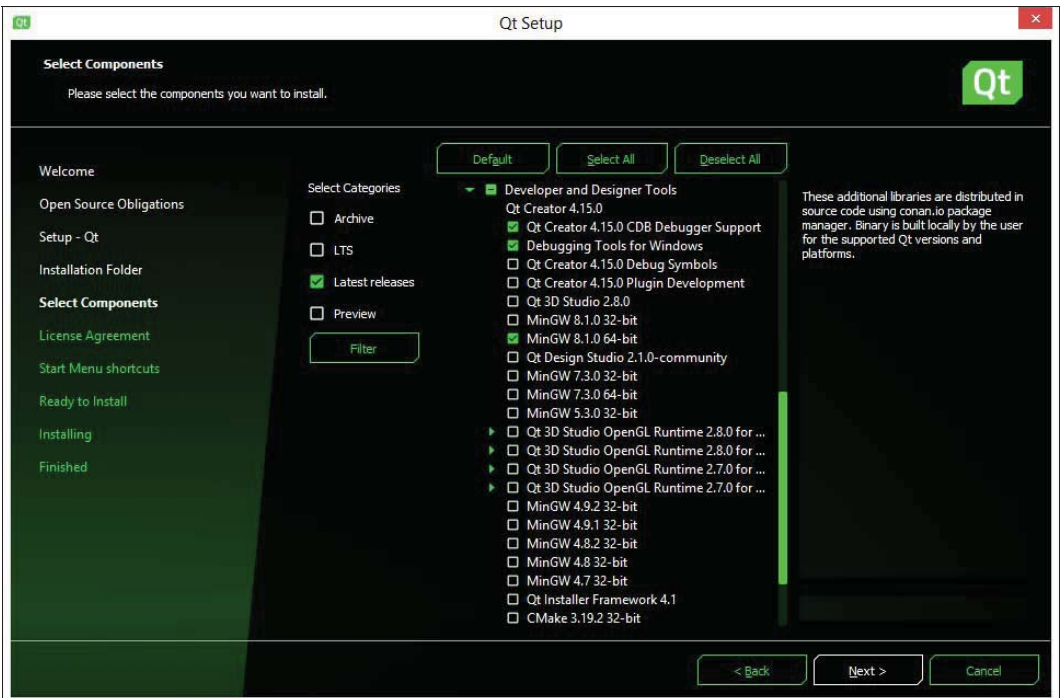


Рис. 1.9. Установка Qt. Шаг 6. Раздел Developer and Designer Tools



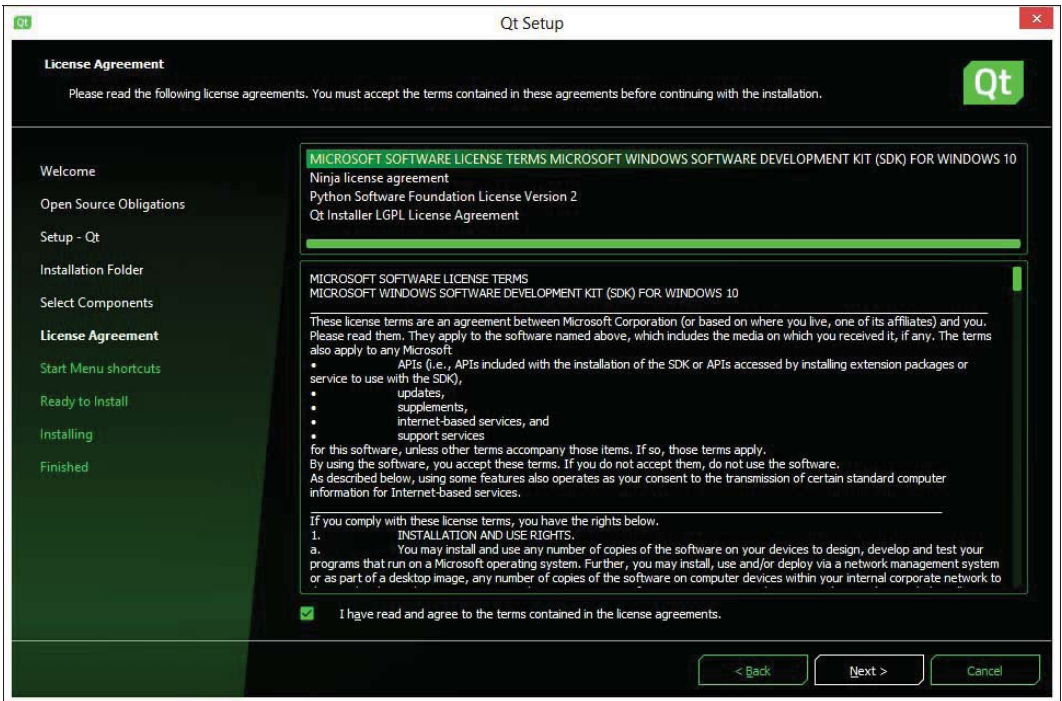


Рис. 1.10. Установка Qt. Шаг 7

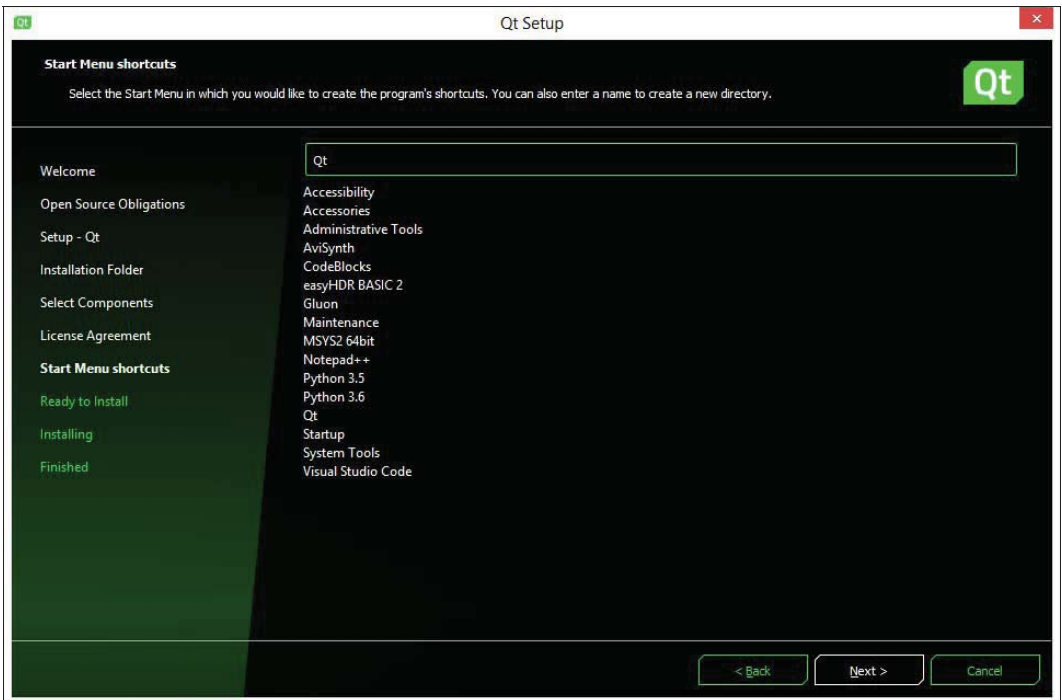


Рис. 1.11. Установка Qt. Шаг 8

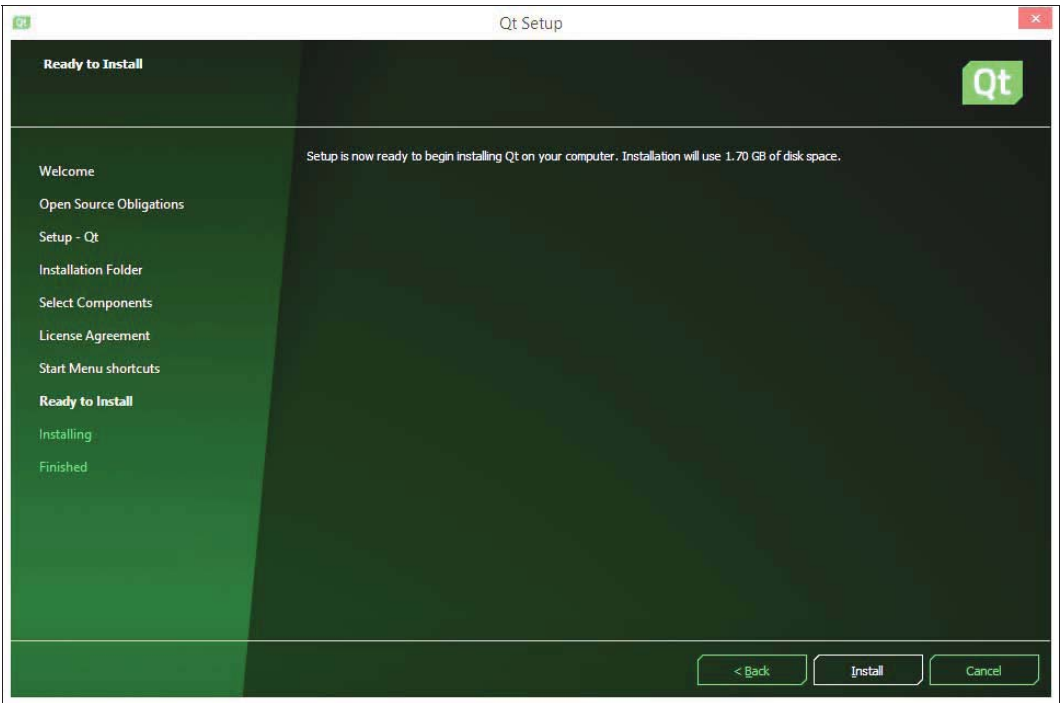


Рис. 1.12. Установка Qt. Шаг 9

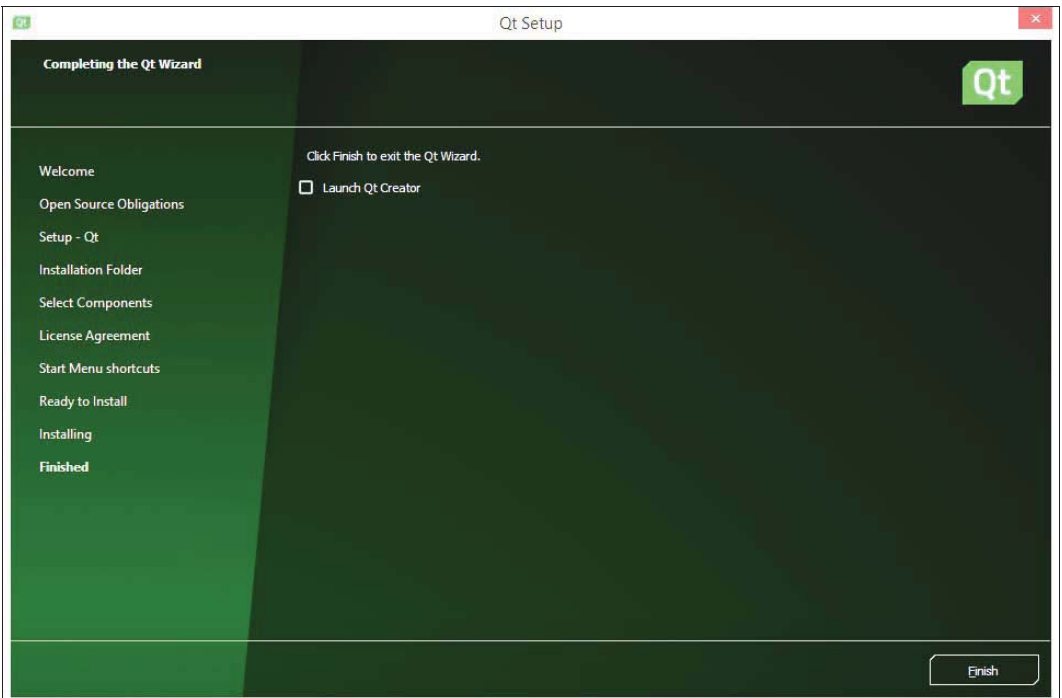


Рис. 1.13. Установка Qt. Шаг 10

В результате библиотека Qt будет установлена в каталог `C:\Qt\6.1.0\mingw81_64`. В каталоге `C:\Qt\6.1.0\mingw81_64\bin` можно найти библиотеки, необходимые для работы оконных приложений. Библиотека MinGW будет установлена в каталог `C:\Qt\Tools\mingw810_64`. Программу `g++.exe`, предназначенную для компиляции программ, написанных на языке C++, можно найти в каталоге `C:\Qt\Tools\mingw810_64\bin`.

В каталог `C:\Qt\Tools\QtCreator\bin` был установлен редактор Qt Creator, который мы будем использовать для набора кода и сборки проектов. Запустить редактор можно с помощью файла `qtcreator.exe`. Если на последнем шаге мастера был установлен флажок **Launch Qt Creator** (см. рис. 1.13), то редактор запустится автоматически.

Пути к каталогам `C:\Qt\Tools\mingw810_64\bin` и `C:\Qt\6.1.0\mingw81_64\bin` можно добавить в системную переменную `PATH`. Однако мы этого делать не станем, чтобы можно было использовать сразу несколько компиляторов. Вместо изменения переменной `PATH` на постоянной основе мы будем выполнять изменение в командной строке только для текущего сеанса. Продемонстрируем это на примере, а заодно проверим работоспособность компилятора. Запускаем командную строку и выполняем следующие команды:

```
C:\Users\Unicross>cd C:\
```

```
C:\>set Path=C:\Qt\Tools\mingw810_64\bin;C:\Qt\6.1.0\mingw81_64\bin;%Path%
```

```
C:\>gcc --version
```

```
gcc (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
C:\>g++ --version
```

```
g++ (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Первая команда (`cd C:\`) делает текущим корневым каталогом диска `C:`. Вторая команда (`set Path=C:\Qt\Tools\mingw810_64\bin;C:\Qt\6.1.0\mingw81_64\bin;%Path%`) изменяет значение системной переменной `PATH` для текущего сеанса. Пути к каталогам `C:\Qt\Tools\mingw810_64\bin` и `C:\Qt\6.1.0\mingw81_64\bin` мы добавили в самое начало переменной `PATH`. Третья команда (`gcc --version`) выводит версию программы `gcc.exe`. Эту программу можно использовать для компиляции программ, написанных на языке C. Четвертая команда (`g++ --version`) выводит версию программы `g++.exe`. Эту программу мы будем использовать для компиляции программ, написанных на языке C++. Фрагменты перед командами означают приглашение для ввода команд. Текст после команд является результатом выполнения этих команд.

Вместо выполнения отдельных команд можно написать скрипт, который выполняет сразу несколько команд и отображает результат их выполнения в отдельном окне. Запускаться такой скрипт будет с помощью двойного щелчка левой кнопкой мыши на значке файла. Для создания файла потребуется текстовый редактор, позволяющий корректно работать с различными кодировками. Советую установить на компьютер редактор Notepad++. Скачать редактор можно абсолютно бесплатно со страницы <https://notepad-plus-plus.org/>. Из двух вариантов (архив и инсталлятор) советую выбрать именно инсталлятор, т. к. при установке можно будет указать язык интерфейса программы. Установка Notepad++ предельно проста и в комментариях не нуждается.

Запускаем Notepad++ и создаем новый документ. Консоль в Windows по умолчанию работает с кодировкой windows-866, поэтому мы должны и файл сохранить в этой кодировке, иначе русские буквы будут искажены. В меню **Кодировки** выбираем пункт **Кодировки | Кириллица | OEM 866**. Вводим текст скрипта (листинг 1.1) и сохраняем под названием script.bat в каталоге C:\book. Запускаем скрипт с помощью двойного щелчка левой кнопкой мыши на значке файла script.bat. Результат выполнения показан на рис. 1.14. Для закрытия окна консоли достаточно нажать любую клавишу.

#### Листинг 1.1. Содержимое файла script.bat

```
@echo off
title Заголовок окна
cd C:\
echo Текущий каталог: %CD%
@echo.
set Path=C:\Qt\Tools\mingw810_64\bin;%Path%
rem Вывод версии gcc.exe
echo gcc --version
@echo.
gcc --version
rem Вывод версии g++.exe
echo g++ --version
@echo.
g++ --version
pause
```

Рассмотрим инструкции из этого скрипта:

- title — позволяет вывести текст в заголовок окна консоли;
- echo — выводит текст в окно консоли;
- @echo. — выводит пустую строку в окно консоли;
- %CD% — переменная, содержащая путь к текущему рабочему каталогу;
- %Path% — переменная, содержащая значение системной переменной PATH;
- rem — вставляет комментарий, поясняющий фрагмент кода;