



ПРОСТОТА

Быстрота



МОЩНОСТЬ

Кросс-платформенная библиотека QT



- разработана финской компанией Trolltech, ныне принадлежит корпорации Nokia.
- Qt реализована на языке программирования C++
- Возможности C++ в Qt значительно расширены с помощью макросов и МОС (Meta Object Compiler): добавлены сигналы/слоты, появилось возможность использовать в собственных классах свойства (property).
- Для программистов Python, Ruby, Php и Perl также реализованы интерфейсы взаимодействия, которые, как правило, используются для построения графического интерфейса пользователя.



QT



- Кросс-платформенная среда для разработки приложений с графическим интерфейсом на C++
- Подразумевает коммерческое и свободное лицензирование
- Среда для разработки мобильных приложений систем IOS, Android, Windows Phone, Windows RT, BlackBerry
- Полностью объектно-ориентированная библиотека

QT – полный инструментарий для программирования



- Поддержка 2-х и 3-хмерной графики
- Возможность интернационализации
- Использование форматов JSON, XML
- STL-совместимая библиотека контейнеров
- Поддержка стандартных протоколов ввода-вывода
- Классы для работы с сетью
- Поддержка программирования баз данных Oracle, Microsoft SQL Server, MySQL, SQLite, PostgreSQL, IBM DB2, Sybase ,...
- И многое другое...

Используют QT



- QT используют более 4000 компаний:
- Adobe, Amazon, Tesla
- Canon, Bosch, Sun, NASA,
- Google, Oracle, Intel, IBM,
- Panasonic, Xerox, Sony,
- Siemens, Yamaha, Nokia,
- Rakuten, HP, Disney, BMW, Mercedes, Samsung, Volvo, Skype, Dream Works,

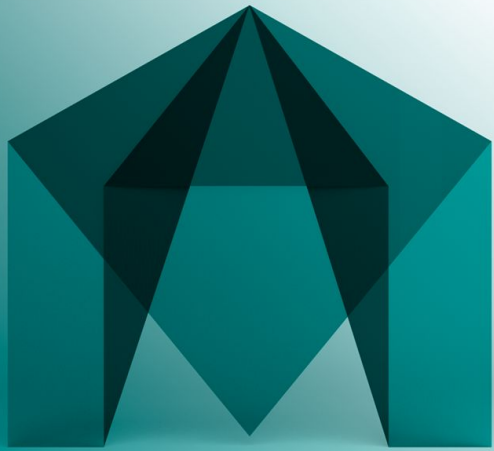


Написаны на QT



- Рабочий стол KDE, используемый в Linux
- Редактор трехмерной графики Autodesk Maya
- Viber
- Telegram
- Google Earth (карта мира)
- Virtual Box (virtualbox.org)
- VLC Media player (свободный проигрыватель) (videolan.org)
- Kindle от Amazon (загрузка, чтение книг, газет, журналов) (amazon.com)
- Bitcoin (bitcoin.org)
- Adobe Photoshop Album (программа для обработки





MAYA



Wolfram
Mathematica®9



VirtualBox

Google™ earth



Модули QT для других языков программирования



- PyQt
- Perl Qt
- Qt#
- PHP
- Ruby
-



Установка Qt



- https://download.qt.io/official_releases/qt/
- Установка Qt предполагает регистрацию в Qt и создание Qt Account, данное действие необходимо для проверки лицензии (коммерческая/некоммерческая).
- Выбираем каталог для установки Qt *.* Обратите внимание, адрес каталога указывается латинскими буквами (без кириллицы) и без пробелов!
- Выбираем компоненты, которые хотим установить (обязательно MinGW 32bit или 64bit ! QtCreator)



business goals plus access to the official Qt Support Helpdesk and a close strategic relationship with The Qt Company.

Pricing and packaging

Buy Qt now

framework, tools for desktop and embedded development, Qt Design Studio, plus other enterprise add-ons.

Download Qt

What's in Qt?

Existing customers

Got Qt already and want to upgrade your license or try out the latest Qt 6 release? Sign into the customer portal to download a trial or contact your Customer Success Manager to discuss your options.

Log in

Contact us about a Qt 6 license

Downloads for open source users

Find out how you can use Qt under the (L)GPL and contribute to the Qt project.

Go open source

View Qt product map

Contact Us

Qt Downloads

[Qt Home](#) [Bug Tracker](#) [Code Review](#) [Planet Qt](#) [Get Qt Extensions](#)

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
6.2/	29-Sep-2021 12:06	-	
6.1/	01-Sep-2021 10:42	-	
6.0/	04-May-2021 07:38	-	
5.15/	19-Nov-2020 13:11	-	
5.12/	25-May-2021 07:09	-	

For Qt Downloads, please visit qt.io/download

Qt and the Qt logo is a registered trade mark of The Qt Company Ltd and is used pursuant to a license from The Qt Company Ltd.
All other trademarks are property of their respective owners.

The Qt Company Ltd. Bertel Junnikin aukio 03A, 02600 Espoo, Finland. Org. Nr. 2637805-2

b29585ca4474fe84....jpg

qt-unified-window....exe

5cc34a5c14090.gif

ИбрагимоваЭ_Ф....docx

Показать все

20:03
25.10.2021



Find them in the Qt Online Installer. It will steer you to the right download version and help you install tools and add-on components that are available for your open source license.

Download the Qt Online Installer

Contact Us

Frequently Asked Questions

Why is Qt licensed also under an open source license?



Thank We are happy

Your download

Note to Con

Please account

Qt Setup

Welcome to the Qt online installer



Welcome

Setup - Qt

Installation Folder

Select Components

License Agreement

Start Menu shortcuts

Ready to Install

Installing

Finished

Please log in to Qt Account

varika@list.ru

.....

Forgot password?

Don't have Qt account? [Sign up](#)

The Qt Account will give you access to Qt downloads, exclusive services, bug reports, code review, and forums & wiki.



Next

Cancel

Download. Try.

sources



Показать все

Выбор компонентов

Пожалуйста, выберите компоненты, которые вы хотите установить.

По умолчанию

- Qt
 - Qt 5.13.0
 - Developer and Designer Tools

Internationalization, sign and other tools to accelerate development.

Этот компонент займёт приблизительно 496.13 МБ на жестком диске.

< Назад

Далее >

Отмена

Выбор компонентов

Пожалуйста, выберите компоненты, которые вы хотите установить.

По умолчанию Qt

- Qt
 - Qt 5.13.0
 - Desktop gcc 64-bit
 - Android x86
 - Android x86_64
 - Android ARM64-v8a
 - Android ARMv7
 - Sources
 - Qt Charts
 - Qt Data Visualization
 - Qt Lottie Animation (Technology Preview)
 - Qt Purchasing
 - Qt Virtual Keyboard
 - Qt WebEngine
 - Qt Network Authorization
 - Qt WebGL Streaming Plugin
 - Qt Script (Deprecated)
 - Developer and Designer Tools
 - Qt Creator 4.9.1

Этот компонент займёт приблизительно 496.13 МБ на жестком диске.

< Назад

Далее >

Отмена

- Начало
- Редактор
- Дизайн
- Отладка
- Проекты
- Справка

Проекты

Примеры

Учебники

Впервые с Qt?

Узнайте, как разрабатывать собственные приложения, и освоите Qt Creator.

Начать сейчас

- Учётная запись Qt
- Онлайн сообщество
- Блоги
- Справка

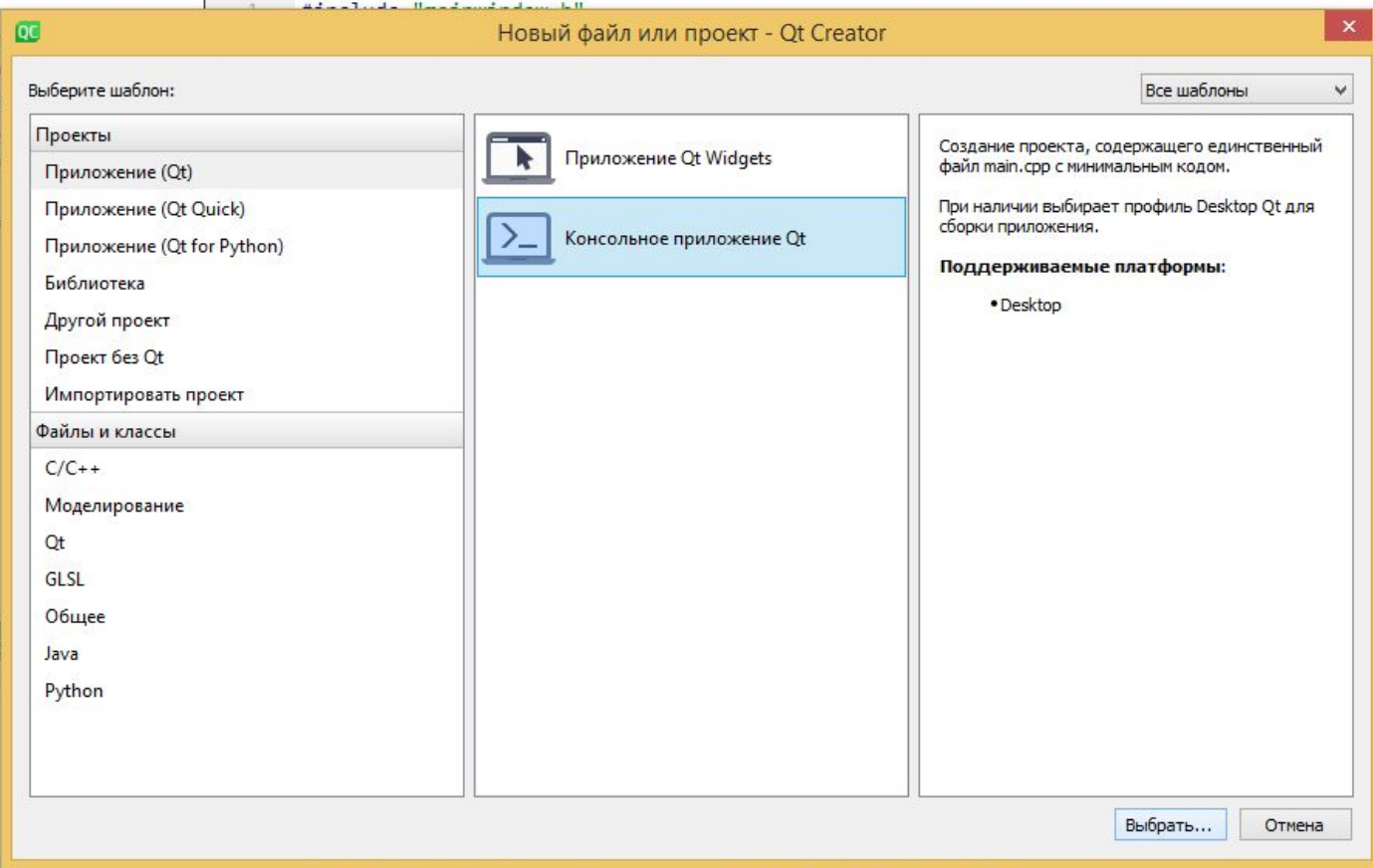
+ Новый проект

Открыть проект

Сессии

1 default

Последние проекты




```
1 #include "mainwindow.h"
```

Консольное приложение Qt

Размещение проекта

This wizard creates a simple Qt-based console application.

- Размещение
- Система сборки
- Перевод
- Комплекты
- Итог

Название:

Создать в: Обзор...

Размещение проекта по умолчанию

Далее Отмена

```
20:32:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.  
20:33:16: Debugging C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe ...  
20:33:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.
```



Начало



Редактор



Дизайн



Отладка



Проекты



Справка

hello1



Отладка

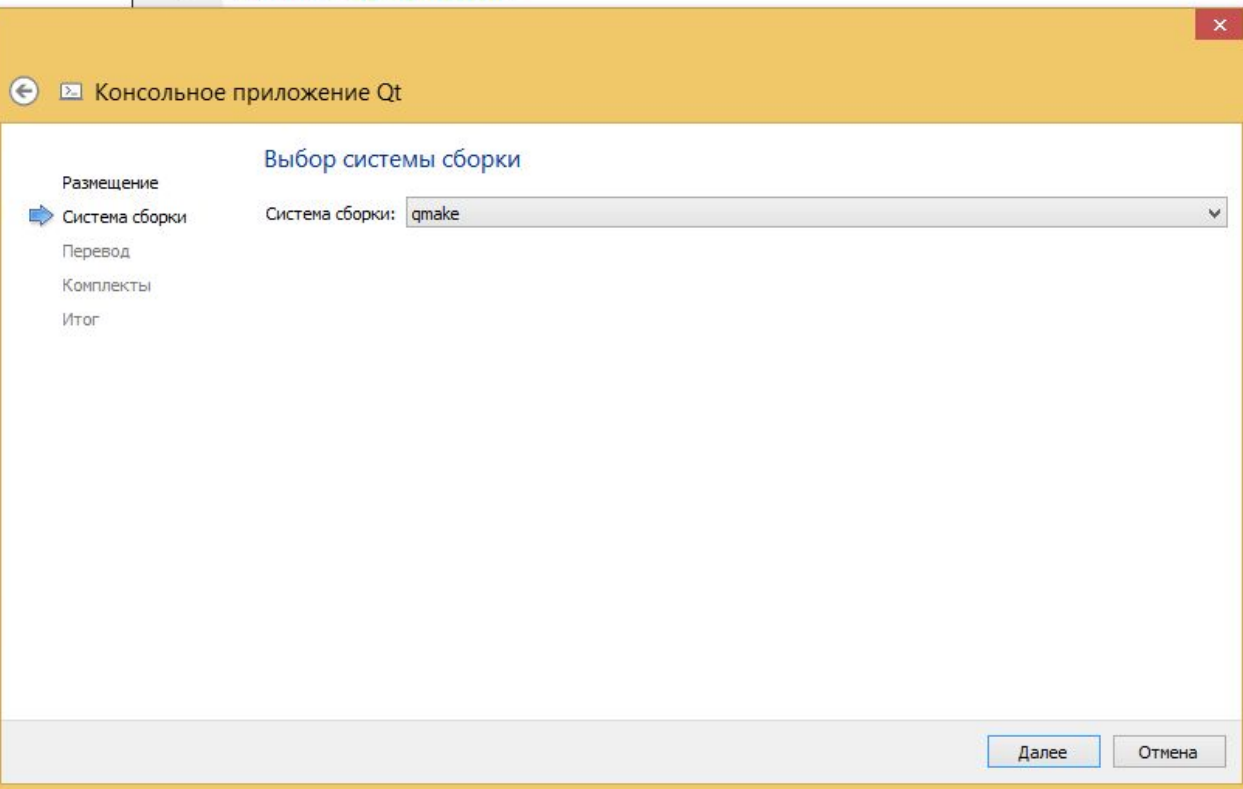
Открытые документы

main.cpp*

mainwindow.ui

main.cpp* <No Symbols> Windows (CRLF) Строка: 11, Столбец: 16

1 #include "mainwindow.h"



```
20:32:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.
20:33:16: Debugging C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe ...
20:33:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.
```

```
1 #include "mainwindow.h"
```

Консольное приложение Qt

Выбор комплекта

К проекту **hello3** применимы следующие комплекты:

Фильтр комплектов по имени...

Выбрать все комплекты

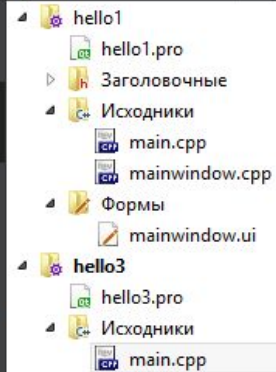
<input type="checkbox"/> Desktop	Подробнее ▾
<input checked="" type="checkbox"/> Desktop Qt 6.2.0 MinGW 64-bit	Подробнее ▾
<input type="checkbox"/> Замена для «Desktop Qt 5.12.2 MSVC2015 64bit»	Подробнее ▾
<input type="checkbox"/> Замена для «Desktop Qt 5.12.2 MinGW 32-bit»	Подробнее ▾
<input type="checkbox"/> Замена для «Desktop Qt 5.12.2 MinGW 64-bit»	Подробнее ▾

Далее Отмена

Открытые документы

main.cpp*
mainwindow.ui

```
20:32:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.
20:33:16: Debugging C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe ...
20:33:27: Debugging of C:\Users\elena\Documents\build-hello1-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello1.exe has finished with exit code 0.
```



```
1 #include <QCoreApplication>
2
3 int main(int argc, char *argv[])
4 {
5     QCoreApplication a(argc, argv);
6
7     return a.exec();
8 }
9
```

- hello1/main.cpp*
- hello3/main.cpp
- mainwindow.ui

Код консольной программы



- `#include <QCoreApplication>`
- `#include<iostream>`
- `int main(int argc, char *argv[])`
- `{`
- `QCoreApplication a(argc, argv);`
- `std::cout<<"Qt version:"<< qVersion()<< std:: endl;`
-
- `return a.exec();`
- `}`



Проекты

- hello1
 - hello1.pro
 - Заголовочные
 - Исходники
 - main.cpp
 - mainwindow.cpp
 - Формы
 - mainwindow.ui
- hello3
 - hello3.pro
 - Исходники
 - main.cpp

```
1 #include <QCoreApplication>
2 #include<iostream>
3 int main(int argc, char *argv[])
4 {
5     QCoreApplication a(argc, argv);
6     std::cout<<"Qt version:"<< qVersion()<< std::endl;
7
8     return a.exec();
9 }
10
```

Строка: 6, Столбец: 51

Открытые документы

- hello1/main.cpp
- hello3/main.cpp
- mainwindow.ui

Вывод приложения

hello3

```
20:41:11: Запускается C:\Users\elena\Documents\build-hello3-Desktop_Qt_6_2_0_MinGW_64_bit-Debug\debug\hello3.exe ...
Qt version:6.2.0
```

Быстрый поиск (Ctrl+K)

1 Проблемы 2 Результаты поиска 3 Вывод приложения 4 Вывод сборки 5 Консоль отладчика QML 6 Основные сообщения 8 Результаты тестирования

Код оконной программы



- `#include "mainwindow.h"`
- `#include <QApplication>`
- `#include<QLabel>`
- `int main(int argc, char *argv[])`
- `{`
- `QApplication a(argc, argv);`
- `//MainWindow w;`
- `QLabel lbl("Hello, world!");`
- `// w.show();`

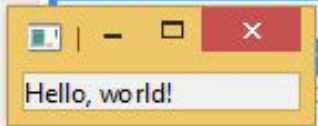


Проекты

- hello1
 - hello1.pro
 - Заголовочные
 - Исходники
 - main.cpp
 - mainwindow.cpp
 - Формы
 - mainwindow.ui

```

1  #include "mainwindow.h"
2
3  int main(int argc, char *argv[])
4  {
5      QApplication a(argc, argv);
6      MainWindow w;
7      QLabel lbl("Hello, world!");
8      // w.show();
9      lbl.show();
10     return a.exec();
11 }
12
13
14
15
    
```



Отладчик Преднастройка отладчика Начать отладку запускающего проекта

Режимы отладки

Перспектива Отлаживаемое приложение

Преднастройк... -

Открытые документы

main.cpp

Вывод приложения

hello1



Фильтр



Модули Qt



- QtCore
- QtGui
- QtWidgets
- QtQuick
- QtQML
- QtNetwork
- QtSql
- QtSvg
- QtXml
- QtXmlPattern
- QtMultimedia
- QtMultimediaWidgets
- QtPrintSupport
- QtTest

Пространство имен Qt



- `Qt::red`
- `using namespace Qt;`



QtCore



- Контейнерные классы:
- QList, QVector, QMap,
- QVariant, QString
- Классы ввода-вывода
- QIODevice, QTextStream, QFile
- Классы процессов и многопоточности:
- QThread, QWaitCondition, QMutex
- Классы для работы с таймером:
- QTimer, QBasicTimer
- Классы для работы с датой и временем:
- QDateTime
- Класс QObject
- Класс событий: QEvent
- Класс настроек приложения :



QtCore



- Класс приложения:
QCoreApplication
- Классы поддержки анимации:
 - QAbstractAnimation ,
QVariantAnimation
- Классы для машины состояний:
 - QStateMachine, QState
- Классы моделей интервью:
 - QAbstractModelItem,
 - QStringListModel,
 - QAbstractProxyModel

QCoreApplication



- Должен создаваться в приложение только один раз
-
- Управляет событиями между приложением и операционной системой
- Передает аргументы командной строке
- Этот класс можно наследовать, чтобы перезаписать некоторые методы

Модуль QtGui



- Предоставляет классы интеграции с оконной системой, с OpenGL
- Класс QWindow
- Класс QApplication
- Содержит механизм цикла событий
- Получает доступ к буферу обмена
- Управляет формой курсора мыши
- Инициализирует необходимые настройки приложения

Модуль QtWidgets



- QWidget
- QVBoxLayout, QHBoxLayout
- QLabel, QLCDNumber
- QPushButton, QCheckBox, QRadioButton
- QSlider, QScrollBar
- QLineEdit, QSpinBox
- QComboBox, QToolBox
- QMainWindow, QMenu
- QMessageBox, QDialog
- QPainter, QBrush, QPen, QColor
- QImage, QPixmap
- Классы стилей
- QApplication

QApplication



- QCoreApplication, QtGuiApplication
- Используется для :
- Получения событий клавиатуры, таймера, мыши и др.
- Обеспечивает правильное завершение работы приложения при завершении работы ОС
- Можно в нем:
- Устанавливать стиль приложения
- Получать указатель на объект рабочего стола
- Управлять глобальными манипуляциями с мышью и регистрировать движения мыши в окне и за его пределами

Дополнительные модули Qt



- QtWebEngineCore
- QtWebEngineWidgets
- Qt 3D
- QtBluetooth
- QtLocation
- QtSensors
- QtCharts
- QtDataVisualization
- QtVirtualKeyboard
- QtRemoteObjects

Философия объектной модели



- QObject – основной базовый класс. Все классы, имеющие сигналы и слоты, должны быть от него унаследованы
- `Class MyClass : public QObject, public AnotherClass`
- `{`
- `...`
- `};`

Философия объектной модели



- QObject содержит в себе поддержку:
 - сигналов и слотов,
 - таймера,
 - Механизма объединения объектов в иерархии
 - Событий и механизма фильтрации
 - Метаобъектной информации
 - Приведения типов
 - СВОЙСТВ



Механизм сигналов и слотов



- Функции обратного вызова
- callback functions
- Механизм сигналов и слотов

Метаобъектный компилятор (МОС)



- Анализирует классы на наличие в их определении
- макроса Q_ОБЪЕСТ
- и внедряет в отдельный файл необходимую дополнительную информацию

Преимущества механизма сигналов и слотов



- Каждый класс, унаследованный от QObject, может иметь любое количество сигналов и слотов
- Сообщения, посылаемые посредством сигналов, могут иметь множество аргументов любого типа
- Сигнал можно соединять с любым количеством слотов. Отправляемый сигнал поступит ко всем соединенным слотам.
- Соединение сигналов и слотов можно производить в любой точке приложения



- При уничтожении объекта происходит автоматическое разъединение всех сигнально-слотовых связей

Недостатки сигналов-слотов



- Сигналы и слоты не являются частью C++, перед компиляцией программы требуется запуск дополнительного компилятора
- Отправка сигнала происходит медленнее, чем вызов функции при механизме обратного вызова
- Существует необходимость наследования класса QObject
- В процессе компиляции не производится никаких проверок: имеется ли сигнал или слот в соответствующих классах, совместимы ли они.

Сигнал



- Это методы, которые могут выполнять пересылку сообщений
- Причина появления сигнала – изменение состояния управляющего элемента
- Присоединенный объект, получив сигнал, **МОЖЕТ** на него отреагировать.
- Соединяемые объекты могут быть независимы !

Сигнал



- Сигнал определяется в классе как метод, только без реализации
- Вся работа по реализации кода для такого метода берет на себя МОС
- Методы сигналов ничего не возвращают и имеют тип void
- Сигнал не обязан соединяться со слотом
- Библиотека предоставляет большой набор уже готовых сигналов для существующих элементов управления

Определение сигнала в своем классе



- `class Mysignal`
- `{ Q_OBJECT`
- `...`
- `signals:`
- `void dolt();`
- `...`
- `};`

Определение сигнала в своем классе



- `void MySignal:: doIt()`
- `{`
- `QMetaObject:: activate(this, &staticMetaObject, 0,0);`
- `}`

Реализация сигнала



- emit dolt();
- class MySignal
- { Q_OBJECT
- public:
- void sendSignal ()
- { emit dolt();}
- signals:
- void dolt();
- };

-
- class MySignal
- { Q_OBJECT
- public:
- void sendSignal()
- { emit dolt(«Hello»);}
- signals:
- void dolt(const QString);
- };

Слоты



- Это методы, которые присоединяются к сигналам
- Слоты определяются в классе как `public`, `private`, `protected`
- В объявлении группы слотов должно стоять
- `private slots`, `protected slots`, `public slots`
- В библиотеке есть целый ряд уже реализованных слотов
- Слоты могут быть виртуальными
- В слотах нельзя использовать параметры по умолчанию
- Слоты нельзя определять как `static`

Создание слота для своего класса



- class MySlot
 - { Q_OBJECT
 - public: MySlot();
 - public slots:
 - void slot()
 - {
 - qDebug () « I am a slot«;
 - }
 - };
- void slot()
 - {
 - qDebug<< sender()
→objectName();
 - }



Соединение объектов



- `QObject :: connect (const QObject * sender,`
- `const char* signal,`
- `const QObject* receiver,`
- `const char* slot,`
- `Qt:: ConnectionType type= Qt:: AutoConnection);`
-
- `SIGNAL(method())`
- `SLOT(method())`

Соединение объектов



- type:
- Qt:: DirectConnection
- Qt:: QueuedConnection
- Qt::AutoConnection



Соединение объектов



- `QObject :: connect (const QObject * sender,`
- `const QMetaMethod& signal,`
- `const QObject* receiver,`
- `const QMetaMethod& slot,`
- `Qt:: ConnectionType type= Qt:: AutoConnection);`
-

Пример соединения объектов



- void main()
- {
-
- QObject::connect(pSender, SIGNAL(signalMethod()),
pReceiver, SLOT(slotMethod()));
-
- }
- QObject::connect(pSender, &SenderClass:: signalMethod(),
pReceiver, &ReceiverClass::slotMethod());



- MyClass :: MyClass ():QObject ()
- {
- ...
- connect(pSender, SIGNAL(signalMethod()), pReceiver,
SLOT(slotMethod()));
- ...
- }



- `MyClass :: MyClass ():QObject ()`
- `{`
- `...`
- `connect(pSender, SIGNAL(signalMethod()),`
`SLOT(slot()));`
- `...`
- `}`
- `void MyClass:: slot()`
- `{ qDebug() <<»I am a slot»; }`

Отслеживание ошибки соединения



- `bool bOk =true;`
- `bOk &= connect(pcmd1, SIGNAL(clicked()), pObjReceiver1, SLOT(slotButtonClicked()));`
- `Q_ASSERT(bOk);`
-
-

Передача сигнала без обработки



- `MyClass :: MyClass ():QObject ()`
- `{`
- `...`
- `connect(pSender, SIGNAL(signalMethod()),`
`SIGNAL(mySignal()));`
- `...`
- `}`

Временная блокировка отправки сигналов



- `blockSignal(true);`
- `blockSignal(false);`
- `signalsBlocked();`

