

Основные модули Qt

Дмитрий Фёдоров
Начальник отдела разработки ПО



Проверка связи



Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



Поставьте в чат:

-  если меня видно и слышно
-  если нет

Дмитрий Фёдоров

О спикере:

- более 10 лет в разработке авиационных систем
- возглавляет отдел разработки ПО в НИЦ «ИРТ»



Вспоминаем прошрое занятие

Вопрос: Что такое Qt?



Вспоминаем прошрое занятие

Вопрос: Что такое Qt?

Ответ: Qt — это библиотека классов C++



Вспоминаем прошное занятие

Вопрос: Что такое комплект Qt?



Вспоминаем прошрое занятие

Вопрос: Что такое комплект Qt?

Ответ: Комплект состоящий из компилятора, отладчика, профиля Qt и системы сборки



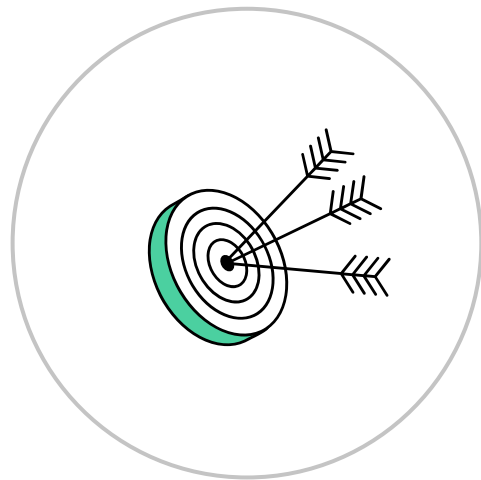
Вспоминаем прошрое занятие

Вопрос: У всех получилось установить Qt?



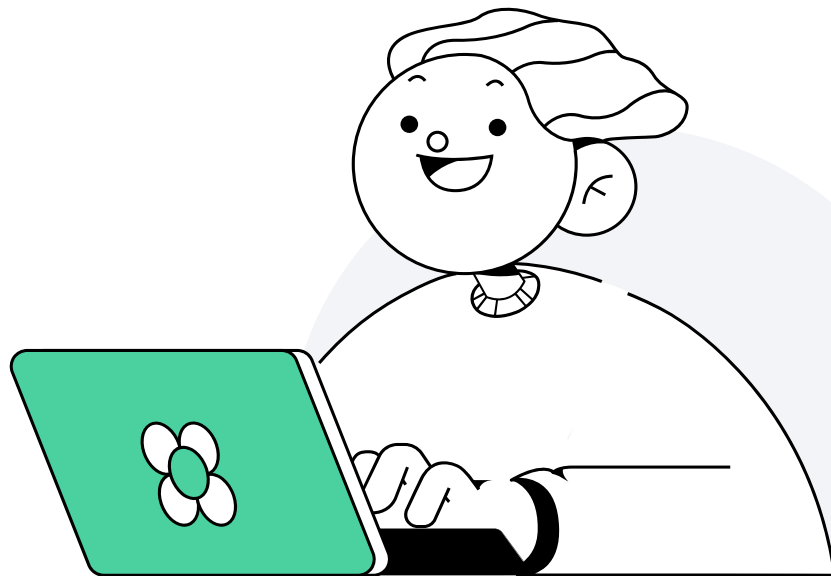
Цель занятия

- Изучим основные модули Qt



План занятия

- 1 Библиотеки и структура Qt
- 2 QtCore
- 3 QtGui
- 4 QtWidgets
- 5 QtNetwork
- 6 QtSql
- 7 Итоги
- 8 Домашнее задание



Библиотеки и структура Qt



1

Библиотека Qt

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.

Все классы Qt не являются чем-то единым, они разделены на независимые модули (наборы классов), которые могут быть подключены в проект по мере необходимости.

Модули Qt

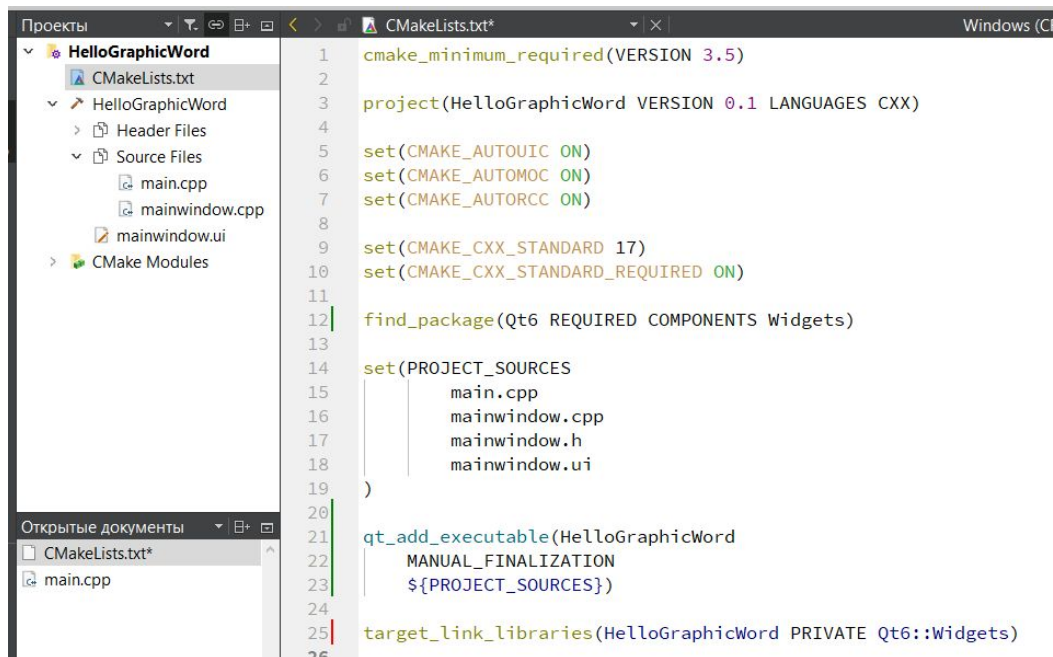
<u>Qt Core</u>	Основополагающий модуль, состоящий из классов, не связанных с графическим интерфейсом
<u>Qt GUI</u>	Модуль базовых классов для программирования графического интерфейса
<u>Qt Widgets</u>	Модуль, дополняющий QtGui «строительным материалом» для графического интерфейса в виде виджетов на C++
<u>Qt Test</u>	Модуль, содержащий классы для выполнения unit-тестирования.
<u>Qt QML</u>	Модуль, содержащий движок для языка QML и JavaScript
<u>Qt Network</u>	Модуль для программирования сети

Модули Qt

<u>Qt SQL</u>	Модуль для программирования баз данных
<u>Qt Charts</u>	Модуль для визуализации данных.
<u>Qt Print Support</u>	Модуль для работы с принтером
<u>Qt Concurrent</u>	Модуль для разработки многопоточных приложений без использования примитивов синхронизации
<u>Qt PDF</u>	Модуль для рендеринга PDF документов
<u>Qt Serial Port</u>	Модуль для работы с виртуальными и “железными” последовательными портами.

Модули Qt

Любая Qt-программа так или иначе должна использовать хотя бы один из модулей модулей: QtCore, QtGui и QtWidgets, они присутствуют во всех программах с графическим интерфейсом и поэтому определены в программе создания make-файлов по умолчанию.



Где читать про модули

- Какой способностью должен обладать лучший программист?
- Умением гуглить

Вся информация по интересующим модулям или классам доступна в [официальной](#) документации.

На этом курсе вам самостоятельно предстоит много работы с документацией.

Пространство имен Qt

Пространство имен Qt содержит различные типы и перечисления, которые наиболее часто встречаются при разработке приложений.

Header:	<code>#include <Qt></code>
CMake:	<code>find_package(Qt6 REQUIRED COMPONENTS Core)</code> <code>target_link_libraries(mytarget PRIVATE Qt6::Core)</code>
qmake:	<code>QT += core</code>

enum Qt::Axis

This enum type defines three values to represent the three axes in the cartesian coordinate system.

Constant	Value	Description
<code>Qt::XAxis</code>	0	The X axis.
<code>Qt::YAxis</code>	1	The Y axis.
<code>Qt::ZAxis</code>	2	The Z axis.

QtCore



2

Модуль QtCore

Это самый базовый модуль, все остальные модули полагаются на него. Он является базовым для всех приложений разработанных при помощи Qt и не содержит классов относящихся к графическому интерфейсу.

```
find_package(Qt6 REQUIRED COMPONENTS Core)
target_link_libraries(mytarget PRIVATE Qt6::Core)
```

При использовании cmake этот модуль добавляется автоматически

Модуль QtCore

Модуль QtCore содержит в себе:

- контейнерные классы: QList, QVector, QMap, QVariant, QString и т. д;
- классы для ввода и вывода: QIODevice, QTextStream, QFile;
- классы процесса QProcess и для программирования многопоточности: QThread, QWaitCondition, QMutex;
- классы для работы с таймером: QBasicTimer и QTimer;
- классы для работы с датой и временем: QDate и QTime;
- класс QObject, являющийся краеугольным камнем объектной модели Qt;
- базовый класс событий QEvent;
- класс для сохранения настроек приложения QSettings;
- класс приложения QCoreApplication, из объекта которого, если требуется, можно запустить цикл событий;

Класс QApplication

- Создается один раз, срок жизни соответствует времени работы всего приложения.
- Доступен в любой момент времени работы приложения.
- Управляет событиями между ОС и приложением.
- Передает и предоставляет аргументы командной строки.

QtGui



3

Модуль QtGui

Модуль предоставляет классы интеграции оконной системы, с OpenGL и OpenGL ES. Он содержит класс QWindow, который является элементарной областью с возможностью получения событий пользовательского ввода. Т.е. окна приложения.

```
find_package(Qt6 REQUIRED COMPONENTS Gui)
target_link_libraries(mytarget PRIVATE Qt6::Gui)
```

При использовании cmake этот модуль добавляется автоматически

Класс QApplication

Основной класс модуля QtGui - QApplication. QApplication является наследником QApplication, содержит механизм цикла событий и обладает также возможностями:

- получения доступа к буферу обмена;
- инициализации необходимых настроек приложения;
- управления формой курсора мыши.

QtWidgets



4

Модуль QtWidgets

Этот модуль содержит около 300 классов виджетов, представляющих собой “строительный материал” для программирования графического интерфейса пользователя.

Виджет — элементарный элемент, который может быть размещен на пользовательском интерфейсе.

```
find_package(Qt6 REQUIRED COMPONENTS Widgets)
target_link_libraries(mytarget PRIVATE Qt6::Widgets)
```

```
QT += widgets
```

Модуль QtWidgets

Модуль QtWidgets включает в себя следующие основные классы:

- класс QWidget - это базовый класс для всех элементов управления библиотеки Qt. По своему внешнему виду он не что иное, как заполненный четырехугольник, но за этой внешней простотой скрывается большой потенциал непростых функциональных возможностей.
- классы для автоматического размещения элементов: QVBoxLayout, QHBoxLayout;
- классы элементов отображения: QLabel, LCDNumber;
- классы кнопок: QPushButton, QCheckBox, QRadioButton;
- классы элементов установок: QSlider, QScrollBar;
- классы элементов ввода: QLineEdit, QSpinBox;
- классы элементов выбора: QComboBox, QToolBox;
- классы меню: QMainWindow и QMenu;
- классы окон сообщений и диалоговых окон: QMessageBox, QDialog;
- классы для рисования: QPainter, QBrush, QPen, QColor;
- классы для растровых изображений: QImage, QPixmap;
- классы стилей - как отдельному элементу, так и всему приложению может быть присвоен определенный стиль, изменяющий их внешний облик;
- класс приложения QApplication, который предоставляет цикл событий.

Класс QApplication

Класс QApplication напрямую унаследован от QApplication и дополняет его следующими возможностями:

- установка стиля приложения. Таким способом можно устанавливать виды и поведения (Look & Feel) приложения, включая и свои собственные;
- управление глобальными манипуляциями с мышью (например, установка интервала двойного щелчка кнопкой мыши) и регистрация движения мыши в пределах и за пределами окна приложения;
- обеспечение правильного завершения работающего приложения при завершении работы операционной системы.

QtNetwork

5



Модуль QtNetwork

Модуль QtNetwork предлагает классы для взаимодействия по различным сетевым протоколам. Модуль предлагает два уровня классов:

- низкоуровневые для разработки клиент-серверных соединений на уровне TCP/IP. Такие как QTcpServer, QTcpSocket и QUdpSocket;

Модуль QtNetwork

Модуль QtNetwork предлагает классы для взаимодействия по различным сетевым протоколам. Модуль предлагает два уровня классов:

- низкоуровневые для разработки клиент-серверных соединений на уровне TCP/IP. Такие как QTcpServer, QTcpSocket и QUdpSocket;
- высокоуровневые такие как QNetworkRequest, QNetworkReply и QNetworkAccessManager, для выполнения сетевых операций(например HTTP запросов) с использованием общих протоколов.

Модуль QtNetwork

```
find_package(Qt6 REQUIRED COMPONENTS Network)  
target_link_libraries(mytarget PRIVATE Qt6::Network)
```

```
QT += network
```


QtSql

6

Модуль QSql

Модуль [QtSQL](#) обеспечивает полноценный API для взаимодействия разрабатываемого приложения с базами данных. Данный модуль можно условно разделить на три уровня:

- Низкий уровень, уровень драйвера, классы отвечают за установление соединения непосредственно с БД, аутентификацию и подключение к БД (QSqlDriver, QSqlDriverCreator).

Модуль QtSql

Модуль [QtSQL](#) обеспечивает полноценный API для взаимодействия разрабатываемого приложения с базами данных. Данный модуль можно условно разделить на три уровня:

- Низкий уровень, уровень драйвера, классы отвечают за установление соединения непосредственно с БД, аутентификацию и подключение к БД (QSqlDriver, QSqlDriverCreator).
- Уровень SQL API, классы этого уровня отвечают за взаимодействие с БД посредством SQL-запросов (QSqlQuery, QSqlQueryModel).

Модуль QtSql

Модуль [QtSQL](#) обеспечивает полноценный API для взаимодействия разрабатываемого приложения с базами данных. Данный модуль можно условно разделить на три уровня:

- Низкий уровень, уровень драйвера, классы отвечают за установление соединения непосредственно с БД, аутентификацию и подключение к БД (QSqlDriver, QSqlDriverCreator).
- Уровень SQL API, классы этого уровня отвечают за взаимодействие с БД посредством SQL-запросов (QSqlQuery, QSqlQueryModel).
- Уровень пользовательского интерфейса - классы этого уровня отвечают за отображение результатов полученных из БД (QSqlTableModel).

Модуль QtSql

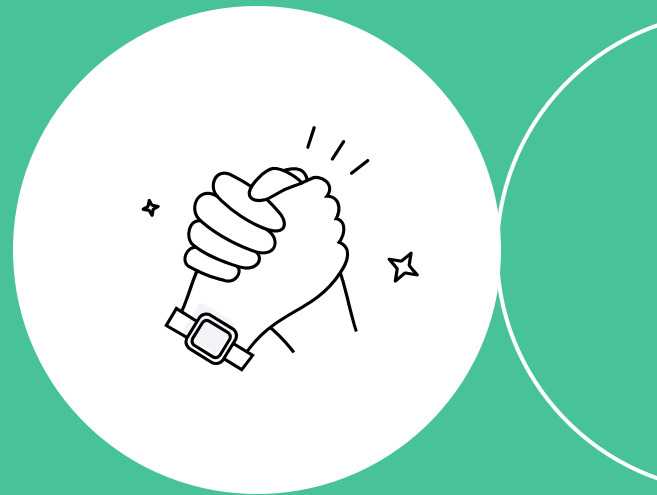
```
find_package(Qt6 REQUIRED COMPONENTS Sql)  
target_link_libraries(mytarget PRIVATE Qt6::Sql)
```

```
QT += sql
```

Оставшиеся модули

Помимо рассмотренных модулей в Qt имеется еще множество узконаправленных и специализированных модулей которые могут быть использованы при разработке приложений. Подробнее о модулях можно прочитать в [официальной документации](#).

Итоги



Итоги занятия

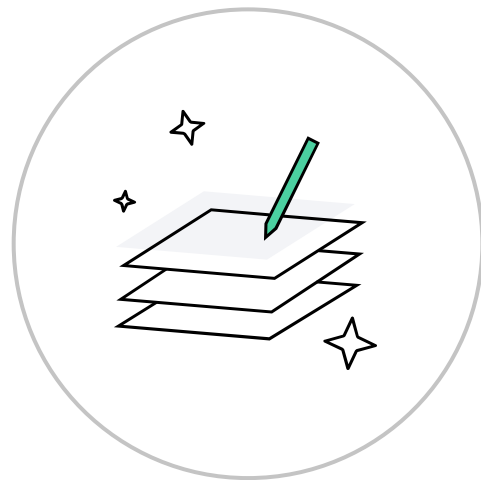
- 1 Узнали, что такое библиотека Qt.
- 2 Узнали где искать информацию о модулях и классах Qt
- 3 Узнали какие основные модули используются в Qt.



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Дополнительные материалы

- [Официальная документация Qt](#)



Задавайте вопросы и пишите отзыв о лекции

Дмитрий Фёдоров
Начальник отдела разработки ПО

