

Немного про Qt и C++

Звонарев Никита

СПбГУ, СтатМод

3 ноября 2017 г.

Что такое Qt?

Qt (читается как “cute”, можно и “кью-ти”) — фреймворк для разработки приложений на языке C++.

Почему “фреймворк”, а не библиотека?

- Не только работа с GUI, но и с мультимедиа/сетью/вебом/БД и т.п.
- Код на Qt предварительно проходит стадию метакомпиляции. Поэтому код на Qt — не чистый C++, а его надмножество.
- Кроссплатформенность: различные ОС и даже средства вывода. Linux (X11, Wayland, Android), Apple (OS X, iOS), Windows (+WinRT), прочие платформы.

Не принято писать Qt как QT (=QuickTime, плеер от Apple)

moc (meta object compiler) — утилита в составе Qt. Представляет из себя генератор кода, который на вход принимает файл на Qt, на выходе — код на C++ (условно).

Проекты на Qt собираются с помощью утилиты qmake, которая сама генерирует вызовы moc для нужных файлов перед запуском компилятора.

Вопрос: зачем всё это нужно?

Пользователь произвёл с интерфейсом некоторое действие (например, ткнул мышкой в кнопку). Далее, нажатие кнопки должно привести к вызову процедуры (метода). Как это сделать?

- Классический подход — *коллбеки* (*callback*). Внутри кнопки лежит указатель на функцию (=callback), который вы указываете при построении. Когда нажимаете на кнопку, процедура-обработчик вызывает коллбек. Неинтуитивно, проблемы с совместимостью параметров и прочее.

Подходы к реализации GUI: сигналы и слоты

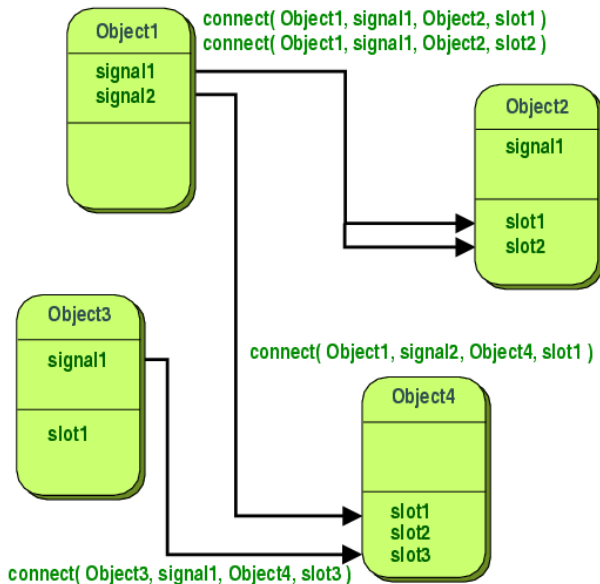
Пользователь произвёл с интерфейсом некоторое действие (например, ткнул мышкой в кнопку). Далее, нажатие кнопки должно привести к вызову процедуры (метода). Как это сделать?

- *Сигналы и слоты* — механизм Qt. У каждого объекта есть свой набор слотов (slots) и сигналов (signals), которые вы можете связывать, как угодно (с точностью до совместимости параметров).

Происходит событие с неким объектом $A \rightarrow$ объект A генерирует сигнал (по-сути, набор аргументов с именем сигнала) \rightarrow механизм слотов и сигналов вызывает связанный слот у объекта B (по-сути, метод) с параметрами, переданными сигналу.

Набор параметров абсолютно произвольный. Объект, испускающий сигнал, и объект со слотом могут не состоять ни в каких отношениях.

Сигналы и слоты: схема



Простейший пример

```
class Counter
{
    public:
        Counter() { m_value = 0; }

        int value() const { return m_value; }
        void setValue(int value);

    private:
        int m_value;
};
```

Простейший пример, теперь с Qt

```
#include <QObject>

class Counter : public QObject {
    Q_OBJECT

    int m_value;

public:
    Counter() { m_value = 0; }
    int value() const { return m_value; }

public slots:
    void setValue(int value);

signals:
    void valueChanged(int newValue);
};
```


Простейший пример, реализация setValue

Любой класс, участвующий в системе сигналов и слотов, должен быть унаследован от QObject, и должен содержать макрос Q_OBJECT.

```
void Counter::setValue(int value)
{
    if (value != m_value) {
        m_value = value;
        emit valueChanged(value);
    }
}
```

emit — сгенерировать сигнал.

Простейший пример: связывание

```
Counter a, b;  
QObject::connect(&a, &Counter::valueChanged,  
                &b, &Counter::setValue);  
  
a.setValue(12);      // a.value() == 12, b.value() == 12  
b.setValue(48);      // a.value() == 12, b.value() == 48
```

Вызов `a.setValue(12)` генерирует сигнал `valueChanged(12)`, который принимает объект `b` в слот `b.setValue(12)`. Вызов же `b.setValue(48)` тоже генерирует сигнал, который игнорируется.

Для двух связей будет сгенерировано два сигнала. Попытка сделать *connect* уже существующей связи приведёт к отказу и возврату *false*. Отвязать сигнал от слота можно с помощью вызова *disconnect()*.

Что такое QtCreator?

QtCreator — среда разработки на Qt. Можно использовать и просто как среду отладки/разработки обычных C++ приложений.

Нас интересует разработка простейшего приложения на `QWidget`.
Есть другая (более современная) альтернатива — `QtQuick`.

Давайте попробуем в ней что-нибудь открыть и посмотреть.

И вы откройте и посмотрите, если она у вас уже установилась.

Где узнать больше про Qt?

Как скачать? Что это? <https://www1.qt.io/download-open-source/>

Сигналы и слоты: <http://doc.qt.io/qt-5/signalsandslots.html>

Система мета-объектов: <http://doc.qt.io/qt-5/metaobjects.html>

Написание простейшей программы используя QtCreator:

<http://doc.qt.io/qtcreator/creator-writing-program.html>

Другой пример: <http://doc.qt.io/qt-5/gettingstartedqt.html>

Документация: <http://doc.qt.io/> + она уже установлена у вас локально (поищите)

Простенький пример, написанный мной на третьем курсе (без использования форм): <https://github.com/furiousbean/figurez/>

Promoting a custom widget

<https://stackoverflow.com/questions/25684189/how-to-promote-a-custom-widget-in-qt-creator>