

Junior C++ developer

Qt Лекция 3

- Слоты и сигналы

СЛОТЫ И СИГНАЛЫ

Средства, позволяющие эффективно производить обмен информацией о событиях между объектами.

Класс унаследованный от **QObject**
может иметь сколько угодно слотов и
сигналов

Сообщения, передаваемые через сигналы, могут иметь сколько угодно аргументов любого типа

Сигнал может соединяться с различным количеством слотов

Слот может принимать множество
сигналов от множества объектов

При уничтожении объекта все связи слот-сигнал этого объекта уничтожаются

Недостатки

- сигналы и слоты не являются частью C++, поэтому перед компиляцией требуется запуск дополнительного препроцессора
- отправка сигналов происходит медленнее, чем вызов обычной функции
- необходимость наследоваться от класса QObject
- некоторые ошибки выявляются только во время выполнения

Сигнал

- Сигнал - метод, осуществляющий пересылку сообщений.
- Сигналы определяются в классе как методы, только без реализации.
- Сигнал не обязательно соединять со слотом.
- Существуют готовые сигналы, также можно реализовывать свои.

СЛОТ

- Метод, присоединяющийся к сигналу.
- В слотах нельзя использовать значения по умолчанию.
- Нельзя определять их как `static`.

Виртуальные слоты

Соединение сигнала с виртуальным слотом примерно в **десять раз медленнее**, чем с не- виртуальным. Поэтому не стоит делать слоты виртуальными, если нет особой необходимости.

Ограничения слотов

- нельзя использовать значение по умолчанию
- нельзя делать слоты `static`

Соединение. Qt4 style

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

- **sender** - объект, отправляющий сигнал
- **signal** - сигнал, с которым устанавливается соединение
- **context** - указатель на объект, имеющий слот для обработки сигнала
- **slot** - функция, вызываемая при получении сигнала
- **type** - режим обработки

Режим обработки

type - управляет режимом обработки. Имеются три возможных значения:

- **Qt: :DirectConnection** сигнал обрабатывается сразу вызовом соответствующего метода слота;
- **Qt: :QueuedConnection**- сигнал преобразуется в событие и ставится в общую очередь для обработки;
- **Qt: :AutoConnection** - это автоматический режим, который действует следующим образом: если отсылающий сигнал объект находится в одном потоке с принимающим его объектом, то устанавливается режим **Qt: :DirectConnection**, в противном случае - режим **Qt: :QueuedConnection**.

Qt: :AutoConnection определен в методе `connection ()` по умолчанию.

Соединение. Qt4 style

!!!! Выявление ошибок только на этапе выполнения !!!!

Соединение. Qt5 style

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

signal - указатель на метод сигнала

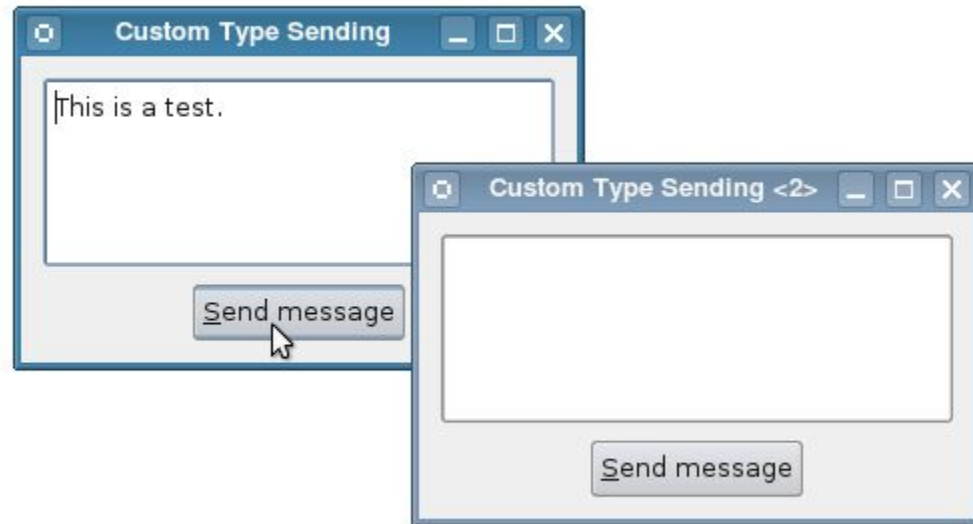
slot - указатель на метод слота

Соединение. Qt5 style

Ошибки на этапе компиляции.

Практика

Реализовать обмен сообщениями между окнами.



У нас имеется **QLabel**, **QPushButton** и **VLayout**. По нажатию кнопки **"Add button"** в **VLayout** будут добавляться другие динамические кнопки, по нажатию на которые в **QLabel** будет отображаться текст с номером кнопки в следующем виде: **"Button 2 pushed"**.

Кнопки **"Dynamic created button ..."** создаются динамически, по нажатию кнопки **"Add button"**, во время работы программы. Мы не знаем заранее сколько их будет, создавать их в дизайнера не нужно.

