Лабораторная работа 2.
Qt. Механизм сигналов и слотов

Во время своей работы, приложение реагирует на различные события. Для того чтобы реализовать необходимую “реакцию” на определенное событие могут быть использованы различные подходы. В библиотеке Qt таким подходом является механизм сигналов и слотов. Сигналами в Qt являются методы, которые предназначены для пересылки сообщений. Слоты – это методы, которые описывают “реакцию” приложения на сигналы. Для использования механизма сигналов и слотов в описании класса, сразу на следующей строке после ключевого слова class, должен находится макрос `Q_OBJECT`, а сам класс должен быть унаследован от `QObject`. После макроса `Q_OBJECT` не должно стоять точки с запятой.

Сигналы описываются как пустые методы класса после ключевого слова “signals”. Например,

```
signals:
class MySignalClass : public QObject {
    Q_OBJECT
    signals:
       void mySignal();
};
```

Для подачи сигнала используется ключевое слово **`emit`**. Например, `emit mySignal()`.

Слоты в отличие от сигналов могут быть определены как `public`, `private` или `protected`. Соответственно, перед каждой группой слотов необходимо указать: `private slots:`, `protected slots:` или `public slots:`. Например,

```
class MySlotClass : public QObject {
 Q_OBJECT
 … public
 slots:
 void mySlot(){
      …
 }};
```

Каждый сигнал может быть связан с произвольным количеством слотов, а каждый слот – с произвольным количеством сигналов. Для связи сигнала со слотом используется метод `connect()` класса `QObject`.

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

`sender` – указатель на объект, отправляющий сигнал;

`signal` – это сигнал, с которым осуществляется соединение. Прототип (имя и аргументы) метода сигнала должен быть заключен в специальный макрос `SIGNAL(method());`

`receiver` – указатель на объект, который имеет слот для обработки сигнала;

`slot` – слот, который вызывается при получении сигнала. Прототип слота должен быть заключен в специальном макросе `SLOT(method());type` – управляет режимом обработки.

Пример:

```
QObject::connect(MySignalClass,
SIGNAL(mySignal()), MySlotClass, SLOT(mySlot()));
```

Если вызов метода `connect()` происходит из потомка класса `QObject`, тогда `QObject::` можно опустить. Если сигнал или слот находится в классе, из которого происходит вызов `connect()`, то можно опустить первый или третий параметр соответственно. Например, если указанное выше соединение осуществляется в классе `MySlotClass`, то вызов `connect()` можно записать в следующем виде: `connect(MySignalClass, SIGNAL(mySignal()), SLOT(mySlot()))`.

В некоторых ситуациях возникает необходимость в разъединение сигналов и слотов, для этих целей может быть использован статический метод `disconnect()` класса `QObject`, набор параметров которого аналогичен параметрам метода `connect()`.

Задание:

1. Изучить главу 2 [1]. Выполнить разработку примера Программы счетчика.
2. Выполнить разработку примера из урока https://evileg.com/ru/post/87/