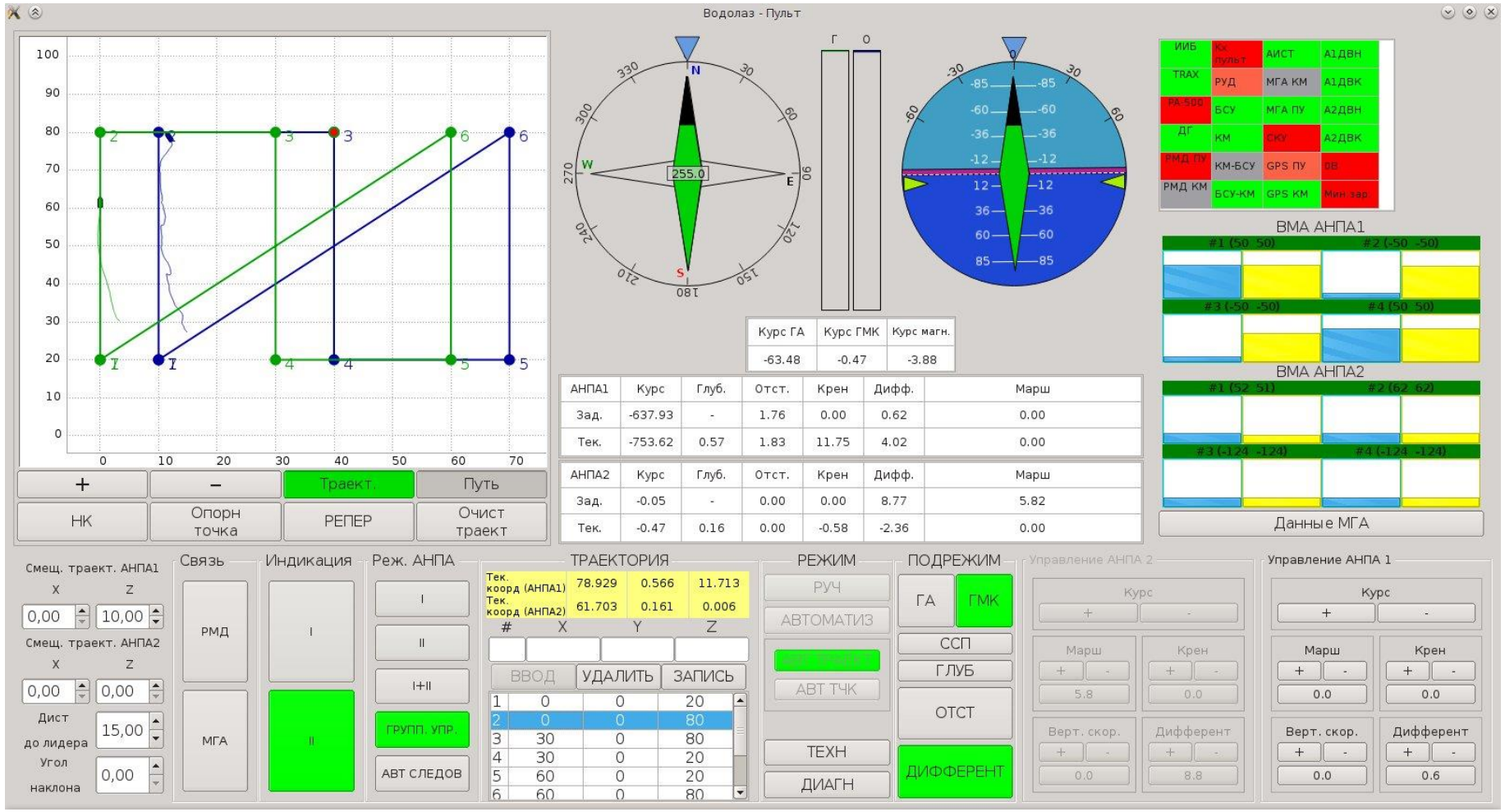
СЕМИНАР 3

Сигналы и слоты

Сигналы и слоты. Введение.



Особенности Qt. Механизм «сигнал-слот».

- Механизм «сигнал-слот» используется для коммуникации между объектами.

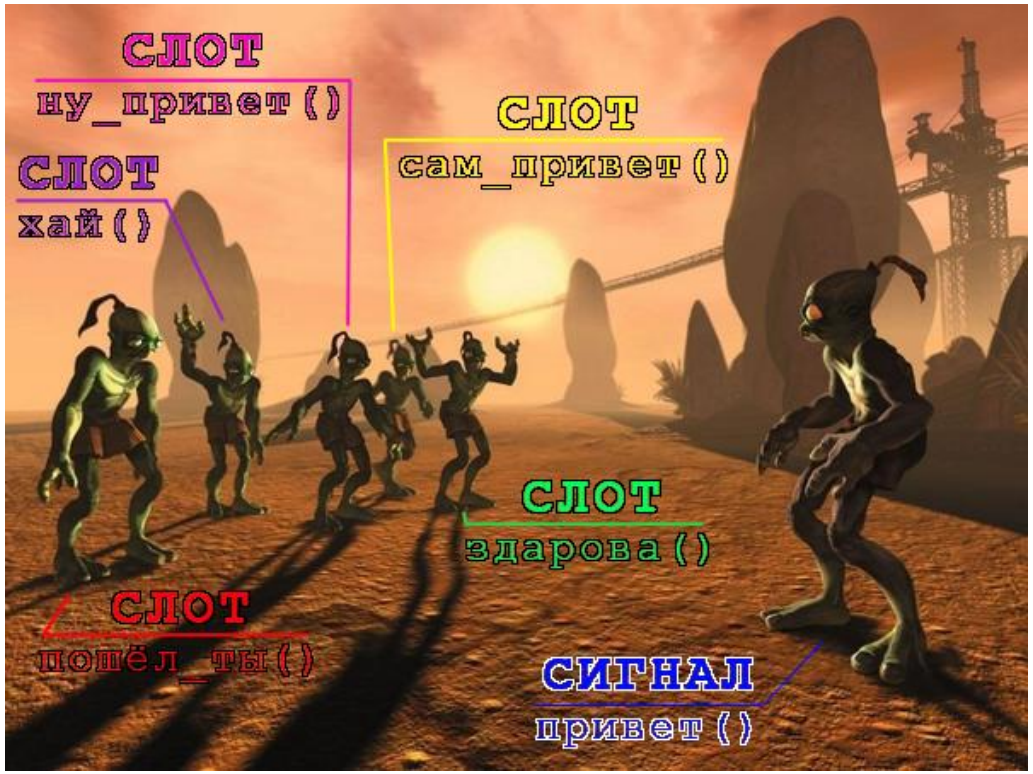


Рис. Механизм «сигнал-слот» в исполнении мудакенов (c) wiki ИУ5

Сигнал – метод, который в состоянии осуществить пересылку сообщений.

Слот – метод, который присоединяется к сигналам (вызывается в ответ на определенный сигнал).

Особенности Qt. Механизм «сигнал-слот».

Преимущества, которые дает программисту механизм «сигнал-слот»:

1. Соединяемые сигналы и слоты абсолютно независимы и реализованы отдельно друг от друга:
    1. Упрощается декомпозиция большого проекта и параллельная разработка;
    2. Предоставлен гибкий механизм для проектирования.
2. Соединение сигналов и слотов можно производить в любой точке приложения.

Особенности Qt. Механизм «сигнал-слот».

Преимущества, которые дает программисту механизм «сигнал-слот»:

3. Соединение сигнала и слота можно осуществлять даже между объектами, которые находятся в различных потоках.
4. При уничтожении объекта происходит автоматическое разъединение всех сигнально-слотовых связей.
5. Механизм сигналов и слотов типобезопасный

Особенности Qt. Механизм «сигнал-слот».

Одному сигналу может соответствовать много слотов.

Одному слоту может соответствовать много сигналов.

Особенности Qt. Механизм «сигнал-слот».

- Можно соединять сигнал и сигнал!

Особенности Qt. Механизм «сигнал-слот».

Недостатки, связанные с применением механизма «сигнал-слот»:

- Сигналы и слоты не являются частью языка C++, поэтому требуется запуск MOC перед компиляцией программы.
- Данный механизм реализован в классе QObject.
  - Для использования сигналов и слотов, класс должен быть унаследован от Qobject.
  - Класс – наследник QObject не может быть шаблонным.
  - В случае множественного наследования Qobject должен быть первым.
- Для реализации механизма MOC требует записи макроса Q_OBJECT сразу при объявлении класса.
  - Нельзя использовать макрос Q_OBJECT с шаблонными классами.

Особенности Qt. Механизм «сигнал-слот».

Недостатки, связанные с применением механизма «сигнал-слот»:

- Сигналы и слоты медленнее чем механизм callback-ов;
- При некоторых способах формирования сигналов и слотов в процессе компиляции не производится никаких проверок: имеется ли сигнал или слот в соответствующих классах или нет; совместимы ли сигнал и слот друг с другом и могут ли они быть соединены вместе. (хотя в Qt5 реализован новый механизм, который частично устраняет этот недостаток).

Сигналы и слоты. Что такое сигнал?

Сигнал - метод, который в состоянии осуществить пересылку сообщений.

Сигнал объявляется однажды и на этом всё, ему не нужна реализация! (Реализацию сигнала осуществляет MOC за вас)

То, что пишете вы в своих исходниках (controlframe.h):

controlframe.h | marshValueChanged(): void

```
1  #ifndef CONTROLFRAME_H
2  #define CONTROLFRAME_H
3
4  #include <QFrame>
5  #include "ui_frame.h"
6
7  namespace PULT {
8
9  class ControlFrame : public QFrame, public Ui::Frame {
10     Q_OBJECT
11 public:
12     ControlFrame( QWidget * pwgt=0);
13     virtual ~ControlFrame();
14     //Объявляются сигналы после ключевого слова signals:
15 signals:
16     //методы сигналов не возвращают никаких значений, поэтому перед
17     //именем метода должен стоять возвращаемый параметр типа void
18     void marshValueChanged();
19
20 };
21 } //namespace PULT
22 #endif // CONTROLFRAME_H
23
```

Сигналы и слоты. Что такое сигнал?

То, что дописывает MOC в (moc_controlframe.cpp) за вас:

moc_controlframe.cpp | PULT::ControlFrame::qt_metacall(QMetaObject::Call, int, void **): int

```
124
125 // SIGNAL 0
126 void PULT::ControlFrame::marshValueChanged()
127 {
128     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
129 }
130 QT_END_MOC_NAMESPACE
131
```

Сигналы и слоты. Как выслать сигнал?

Для вызова сигнала используется ключевое слово `emit`.

```
void ControlFrame::sendMarshSignal(){

    emit marshValueChanged();
    //данная конструкция приводит к обычному вызову метода marshValueChanged()
    //реализация которого дописана MOC (метаобъектным компилятором)
}
```

Сигналы могут высылаться только объектами классов, которые их содержат.

*В нашем примере только объектами класса `ControlFrame`.

Кроме того, сигналы могут высылать информацию, передаваемую в параметре.

Сигналы и слоты. Что такое слот?

Слоты – методы, которые присоединяется к сигналам.

По сути обычные методы, но отличие в том, что могут принимать сигналы!

Объявляются в классе после ключевых слов

`private slots:`

`protected slots:`

`public slots:`

controlframe.h:

controlframe.h* | deltaMarsh: float

```
1  #ifndef CONTROLFRAME_H
2  #define CONTROLFRAME_H
3
4  #include <QFrame>
5  #include "ui_frame.h"
6
7  namespace PULT {
8
9  class ControlFrame : public QFrame, public Ui::Frame {
10     Q_OBJECT
11 public:
12     ControlFrame( QWidget * pwgt=0);
13     virtual ~ControlFrame();
14     void sendMarshSignal();
15 signals:
16     void marshValueChanged();
17
18 public slots:
19     void addMarshValue();
20
21 private:
22     float marshValue;
23     float marshValueMax; //максимальная скорость по маршу
24     float deltaMarsh; //приращение марша при нажатии кнопки
25 };
26 } //namespace PULT
27 #endif // CONTROLFRAME_H
```

Сигналы и слоты. Что такое слот?

Реализация функции слота в controlframe.cpp:

```
void ControlFrame::addMarshValue(){
    //при нажатии на кнопку "+" скорость по маршу должна увеличиться на заданную величину
    marshValue+=deltaMarsh;
    //проверка того, что набранная скорость не превышает ограничений
    if (marshValue>marshValueMax) marshValue = marshValueMax;
    //выведем текущее значение скорости на кнопке сброса
    btnMarshReset->setText(QString::number(marshValue));
}
```

Реализация слота ничем не отличается от обычной функции C++. За исключением нескольких особенностей;

1). В слотах нельзя использовать параметры по умолчанию (например `SlotMethod(int n = 0)`);

2). Слоты нельзя определять как `static`.

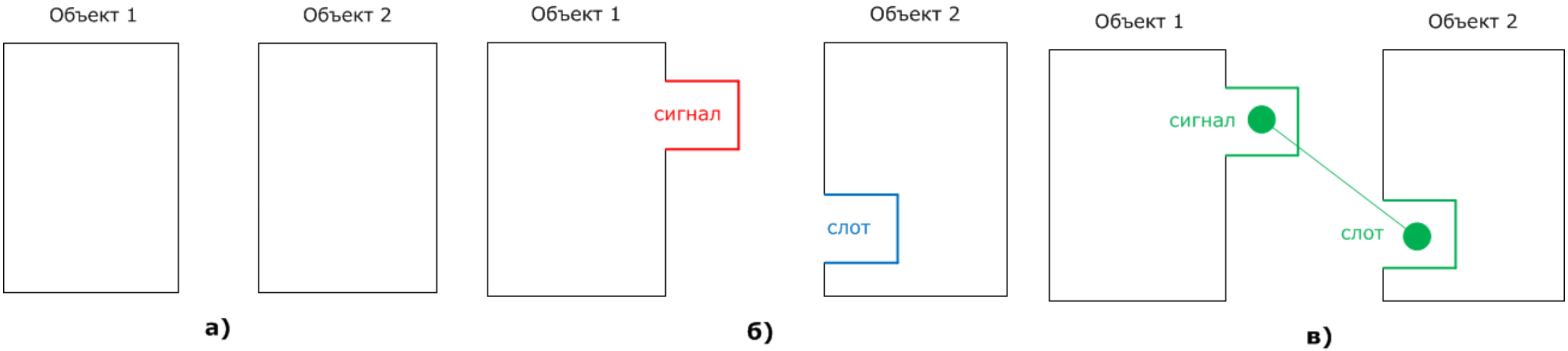
*классы Qt содержат множество уже реализованных слотов.

Сигналы и слоты. Как соединить сигнал и слот?

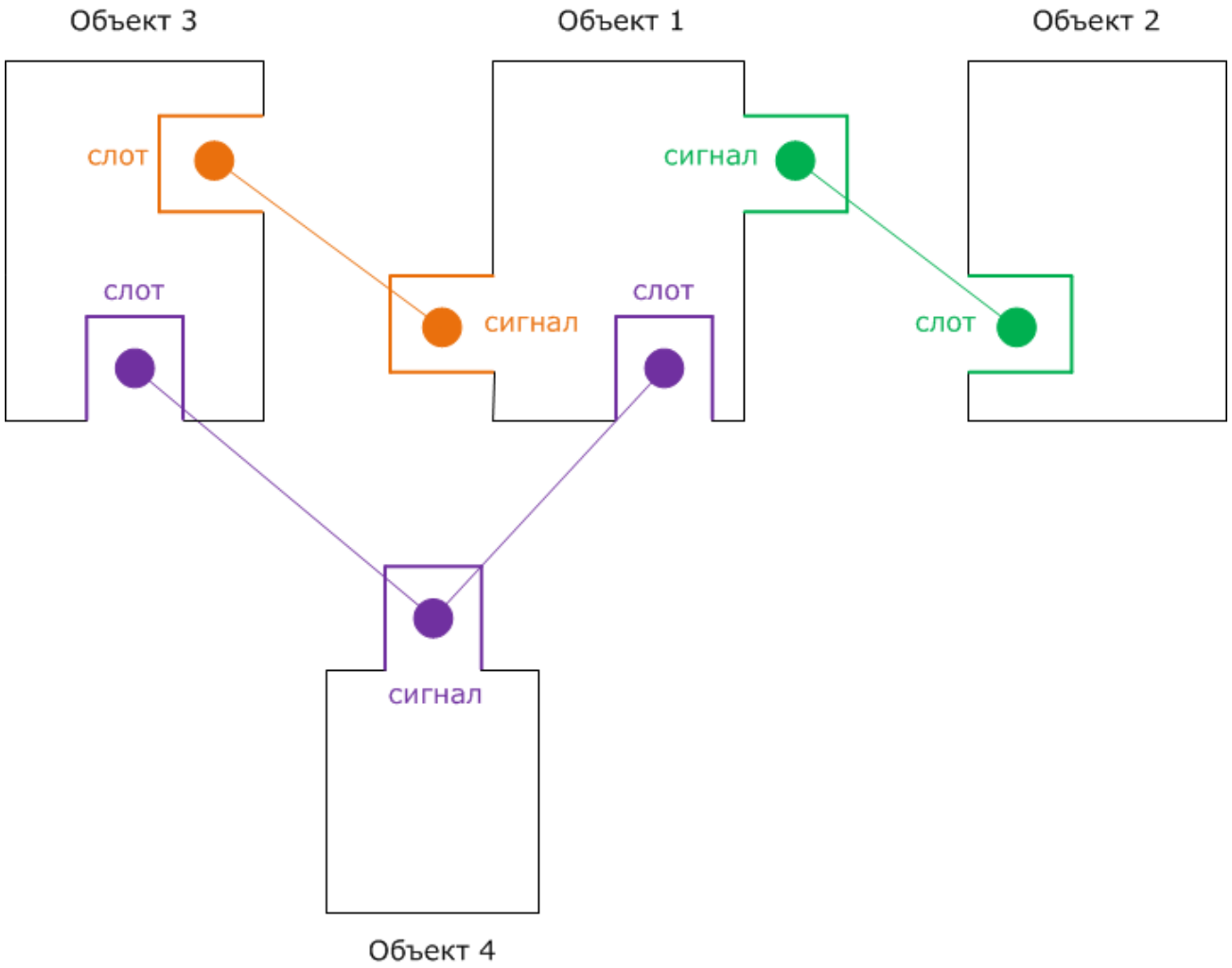
Связь между объектами устанавливается следующим образом: у одного объекта должен быть сигнал, а у второго – слот.

Чтобы соединить два объекта, нужно:

- создать у одного сигнал, а у второго слот;
- соединить сигнал первого и слот второго.



Сигналы и слоты. Как соединить сигнал и слот?



Сигналы и слоты. Как соединить сигнал и слот?

1. Метод connect():
    1. С использованием макросов SIGNAL() и SLOT();
    2. С использованием указателей на функции;
    3. Соединение сигналов с лямбдами.
2. Автоматическое соединение сигналов и слотов.

Практическая часть 1.

Соединить сигналы и слоты таким образом, чтобы:

1. При нажатии на кнопку «+» заданная скорость, отображаемая на кнопке сброса увеличивалась;
2. При нажатии на кнопку «-» заданная скорость – уменьшалась;
3. При нажатии на кнопку сброса заданная скорость – обнулялась.

- При этом использовать все методы соединения сигналов и слотов.

Сигналы и слоты. Как соединить сигнал и слот?

Соединение объектов осуществляется при помощи статического метода connect():

```
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` – сигнал, с которым осуществляется соединение, причем имя сигнала заключается в специальный макрос `SIGNAL(method());`

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

`slot` – слот, который вызывается при получении сигнала. Прототип слота заключается в специальный макрос `SLOT(method());`

`type` – управляет режимом обработки*.

Сигналы и слоты. Как соединить сигнал и слот?

Пример соединения сигнала и слота:

controlframe.cpp* | ControlFrame::ControlFrame(QWidget *) | # Line

```
4
5  ControlFrame::ControlFrame(QWidget *pwgt): QFrame (pwgt) {
6      setupUi(this);
7      marshValueMax=10;
8      deltaMarsh=0.5;
9
10     //соединим сигнал clicked() о нажатии кнопки увеличения скорости по маршу (btnMarshPlus)
11     //со слотом addMarshValue(), который увеличит значение внутренней переменной
12     //в которой хранится значение заданной скорости
13     //в качестве указателя на объект, которому принадлежит слот используем ключевое слово C++ "this"
14     //так как слот содержится в том классе, в котором мы проводим соединение объектов
15     connect (btnMarshPlus, SIGNAL (clicked()),
16              this, SLOT(addMarshValue()));
17     //кстати говоря, сигнал clicked() является сигналом, содержащемся в классе кнопки QPushButton
18     //(и многих дргуих классах), который вызывается когда вы кликаете по кнопке (мышкой,
19     //переходом по "горячей клавише"или любым иным доступным способом))
20     //т.е. мы сами этот сигнал не создавали, а использовали уже существующий...
21
22 }
```

Сигналы и слоты. Как соединить сигнал и слот?

Альтернативный метод connect():

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

Отличие от предыдущего метода в том, что сигнал и метод передаются через указатели на методы сигналов и слотов классов напрямую и без использования макросов `SIGNAL(method())` и `SLOT(method())`.

Сигналы и слоты. Как соединить сигнал и слот?

Пример соединения сигнала и слота альтернативным способом:

controlframe.cpp* | ControlFrame::ControlFrame(QWidget *)

```
20      //т.е. мы сами этот сигнал не создавали, а использовали уже существующий...
21
22      //Соединим сигнал о нажатии кнопки уменьшения скорости
23      // со слотом, уменьшающим текущую заданную скорость ПА
24      //альтернативным способом соединения сигналов
25      connect (btnMarshMinus, &QPushButton::clicked,
26               this, &ControlFrame::decMarshValue);
27
28  }
29
```

Особенность метода в том, что если вы ошибетесь с названием слотов и сигналов, ваша ошибка будет выявлена сразу в процессе компиляции программы.
Недостаток – то, что необходимо явно указывать имена классов для сигнала и слота и следить за совпадением их параметров.

Сигналы и слоты. Соединение с лямбдами.

1. [*capture*] (*params*) mutable exception attribute -> ret { body }
2. [*capture*] (*params*) -> ret { body }
3. [***capture***] (***params***) { body }
4. [***capture***] { body }

capture - определяет, какие символы, видимые в области объявления функции, будут видны внутри тела функции.

params - Список параметров, как в объявлении функции

ret - Возвращаемый тип. Если нет, то он выводится из возвращаемого значения (или void, если функция не возвращает никакого значения)

Сигналы и слоты. Соединение с лямбдами.

[*capture*] (*params*) { body }

capture - определяет, какие символы, видимые в области объявления функции, будут видны внутри тела функции.
Список символов может быть передан следующим образом:
[a,&b] где *a* захвачена по значению, а *b* захвачена по ссылке.
[this] захватывает указатель **this** по значению.
[&] захват всех символов по ссылке
[=] захват всех символов по значению
[] ничего не захватывает
params - Список параметров, как в объявлении функции
ret - Возвращаемый тип. Если нет, то он выводится из возвращаемого значения (или void, если функция не возвращает никакого значения)

Сигналы и слоты. Соединение с лямбдами.

```
    //по нажатию кнопки "+" заданная скорость увеличивается на 1
    connect(btnPlus, &QPushButton::clicked, this,
            [&](){
        speedValue++;
        btnreset->setText(QString::number(speedValue));
    });
    //по нажатию кнопки "-" заданная скорость уменьшается на 1
    connect(btnMinus,&QPushButton::clicked,this,
            [=](){
        speedValue--;
        btnreset->setText(QString::number(speedValue));
    });
    //по нажатию кнопки "сброс" заданная скорость = 0
    connect(btnreset,&QPushButton::clicked,
            [this](){
        speedValue=0;
        btnreset->setText("0");
    });
```

Сигналы и слоты. Автоматическое соединение.

void on<object name>_<signal_name>(<signal parameters>)

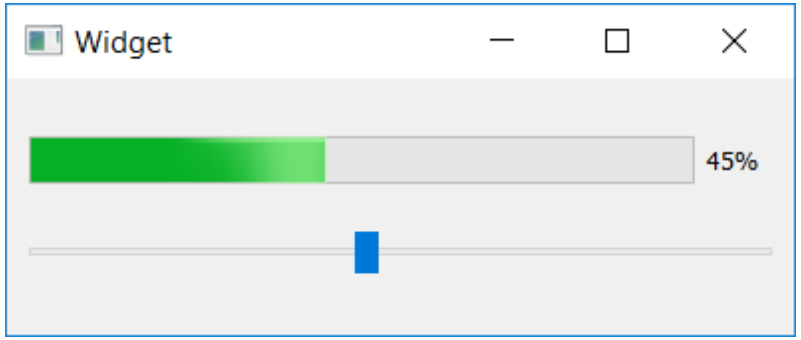
```
void Widget::on_btnPlus_clicked() {


}
```

Практическая часть 2.

Для тех, кто подготовил панель управления ПА или первые прототипы своих пультов:

- соединить задатчики движения ПА с виджетами, выводящими заданные значения каждым из рассмотренных способов соединения сигналов и слотов.

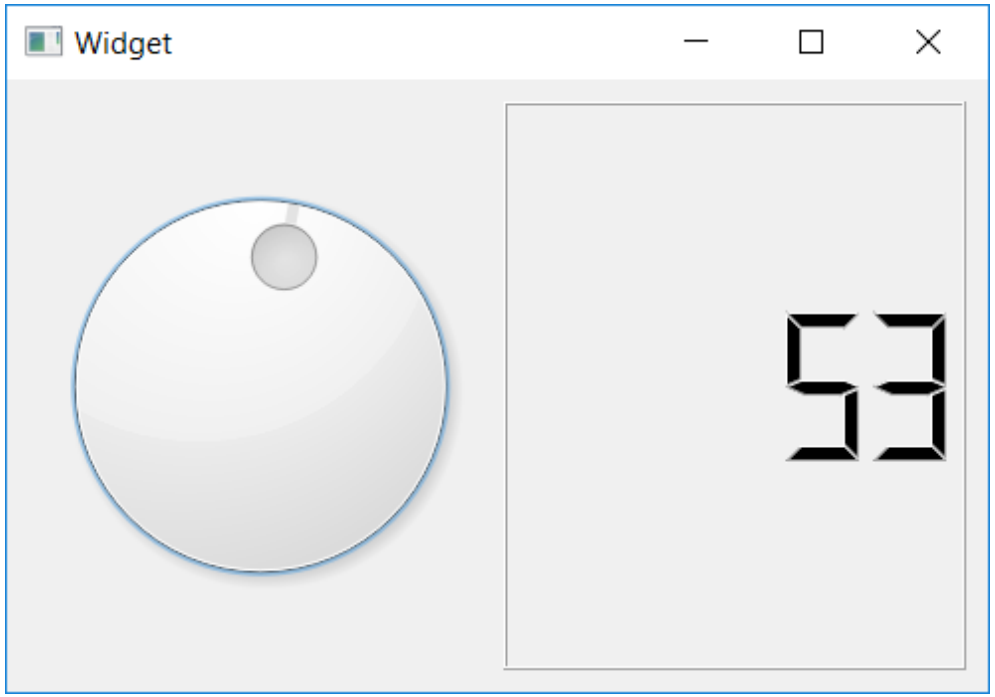
Задание 1



1. При изменении положения слайдера, должно соответственно измениться заполнение индикатора процесса
2. Подсказки:

- Сигнал изменения положения задатчика dial QSlider::valueChanged(int)
- Слот, который может изменить заполнение индикатора процесса – QProgressBar:: setValue(int)

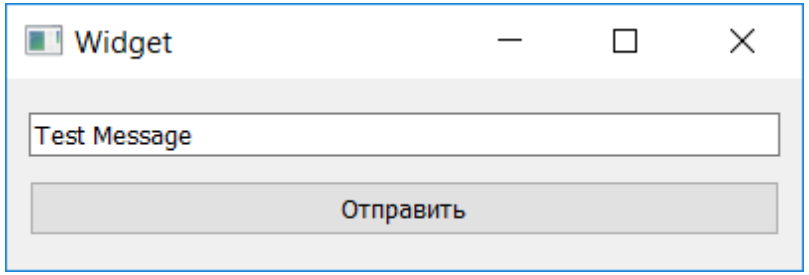
Задание 2

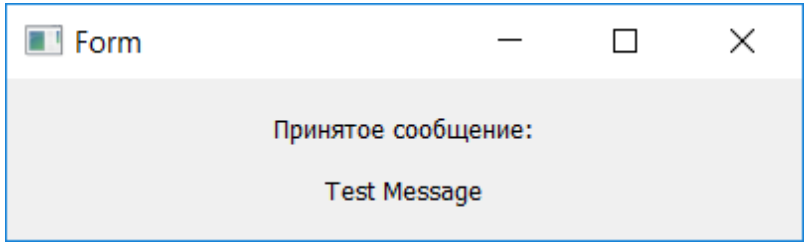


1. При изменении положения элемента dial, значение положения задающего элемента, должно отобразиться на цифровом индикаторе lcdNumber
2. Подсказки:

- Сигнал изменения положения задатчика dial QDial::valueChanged(int)
- Слот, который может вывести значение на цифровом индикаторе - QLCDNumber::display(int)- Вывести текст на метке можно с помощью

Задание 3





1. При нажатии кнопки «Отправить» сообщение, содержащееся в строке ввода окна «Widget» должно быть отображено в окне Form.
2. Подсказки:

- Сигнал нажатия кнопки QPushButton::clicked()
- Получить текст из строки ввода можно с помощью метода QLineEdit::text()
- Вывести текст на метке можно с помощью метода Qlabel::setText(QString)