МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

##### Федеральное государственное автономное образовательное учреждение высшего образования

#### «Национальный исследовательский

#### Нижегородский государственный университет им. Н.И. Лобачевского» (ННГУ)

#### Институт информационных технологий, математики и механики Кафедра Математического обеспечения и суперкомпьютерных технологий

##### Направление подготовки: «Фундаментальная информатика и информационные технологии»

##### Профиль подготовки: «Инженерия программного обеспечения»

## Реферат

## Connections

##### по дисциплине «Технология программирования мобильных систем»

**Выполнил(а):** студент группы 381908-4

##### Кургузиков Кирилл Александрович

#### Руководитель:

##### доцент кафедры программной инженерии

##### Борисов Николай Анатольевич

##### Нижний Новгород 2022

# Система сигналов и обработчиков событий

Компоненты приложения и пользовательского интерфейса должны взаимодействовать друг с другом. Например, кнопка должна знать, что пользователь нажал на нее. Кнопка может менять цвет, чтобы показать свое состояние или выполнить некоторую логику. Кроме того, приложение должно знать, нажимает ли пользователь кнопку. Приложению может потребоваться передать это событие щелчка другим приложениям.

В QML есть механизм сигналов и обработчиков, где сигнал — это событие, а ответ на сигнал поступает через обработчик сигнала . Когда сигнал испускается, вызывается соответствующий обработчик сигнала. Размещение логики, такой как сценарий или другие операции, в обработчике позволяет компоненту реагировать на событие.

# Описание connections

Объект Connections создает соединение с сигналом QML.

При подключении к сигналам в QML обычным способом является создание обработчика on<Signal>, который реагирует на получение сигнала, например:

**MouseArea {**

**onClicked: (mouse)=> { foo(mouse) }**

**}**

Однако в некоторых случаях невозможно подключиться к сигналу таким образом, например, когда:

* Требуется несколько подключений к одному и тому же сигналу
* Создание соединений вне области действия отправителя сигнала
* Подключение к целям, не определенным в QML

Когда какой-либо из них необходим, вместо него можно использовать тип Connections.

Например, приведенный выше код можно изменить для использования объекта Connections, например:

MouseArea { Connections

{ function onClicked (мышь)

{ foo ( мышь )}}}

## Описание свойств

**ignoreUnknownSignals** - Обычно подключение к несуществующему сигналу приводит к ошибкам времени выполнения. Если для этого свойства задано значение true, такие ошибки игнорируются. Это полезно, если вы собираетесь подключаться к разным типам объектов, обрабатывая разные наборы сигналов для каждого объекта.

**target** - Это свойство содержит объект, который отправляет сигнал. Если это свойство не установлено, целью по умолчанию является родительский объект Connection. Если установлено значение null, соединение не устанавливается и любые обработчики сигналов игнорируются до тех пор, пока цель не станет нулевой.

## Получение сигналов с помощью обработчиков сигналов

Чтобы получить уведомление, когда определенный сигнал испускается для определенного объекта, определение объекта должно объявить обработчик сигнала с именем on<Signal> , где <Signal> — это имя сигнала с заглавной первой буквой. Обработчик сигнала должен содержать код JavaScript, который будет выполняться при вызове обработчика сигнала.

### Обработчики сигналов изменения свойства

Сигнал генерируется автоматически при изменении значения свойства QML. Этот тип сигнала является сигналом изменения свойства, и обработчики сигналов для этих сигналов записываются в форме on<Property>Changed , где <Property> — это имя свойства с заглавной первой буквой.

### Использование типа соединения

В некоторых случаях может быть желательно получить доступ к сигналу за пределами объекта, который его излучает. Для этих целей в QtQuickмодуле предусмотрен тип [Connections](https://doc.qt.io/qt-5/qml-qtqml-connections.html) для подключения к сигналам произвольных объектов. Объект [Connections](https://doc.qt.io/qt-5/qml-qtqml-connections.html) может получать любой сигнал от указанной [цели](https://doc.qt.io/qt-5/qml-qtqml-connections.html#target-prop) .

### Прикрепленные обработчики сигналов

Присоединенный [обработчик сигнала](https://doc.qt.io/qt-5/qtqml-syntax-objectattributes.html#attached-properties-and-attached-signal-handlers) получает сигнал от присоединяемого типа , а не от объекта, в котором объявлен обработчик.

Например, [Component.onCompleted](https://doc.qt.io/qt-5/qml-qtqml-component.html" \l "completed-signal) — это прикрепленный обработчик сигнала. Он часто используется для выполнения некоторого кода JavaScript, когда процесс его создания завершен.

Обработчик onCompletedне отвечает на completedсигнал от типа [Rectangle](https://doc.qt.io/qt-5/qml-qtquick-rectangle.html) . Вместо этого объект типа Component присоединения с completedсигналом был автоматически присоединен к объекту [Rectangle](https://doc.qt.io/qt-5/qml-qtquick-rectangle.html) механизмом QML. Движок выдает этот сигнал при создании объекта Rectangle, тем самым запуская Component.onCompletedобработчик сигнала.

Присоединенные обработчики сигналов позволяют объектам уведомляться о конкретных сигналах, которые важны для каждого отдельного объекта. Если бы не было Component.onCompletedприкрепленного обработчика сигнала, например, объект не мог бы получить это уведомление без регистрации на какой-то специальный сигнал от какого-то специального объекта. Присоединенный механизм обработчика сигналов позволяет объектам получать определенные сигналы без дополнительного кода.

## Добавление сигналов к пользовательским типам QML

Сигналы можно добавлять в пользовательские типы QML с помощью signalключевого слова.

Синтаксис для определения нового сигнала:

signal <name>[([<type> <parameter name>[, ...]])]

Сигнал испускается путем вызова сигнала как метода.

## Подключение сигналов к методам и сигналам

Объекты сигналов имеют connect()метод для подключения сигнала либо к методу, либо к другому сигналу. Когда сигнал подключается к методу, этот метод автоматически вызывается всякий раз, когда генерируется сигнал. Этот механизм позволяет получать сигнал методом, а не обработчиком сигнала.

MessageReceivedсигнал подключается к трем методам с помощью connect()метода:

Во многих случаях достаточно получать сигналы через обработчики сигналов, а не использовать функцию connect(). Однако использование этого connectметода позволяет получать сигнал несколькими методами, как показано ранее, что было бы невозможно с обработчиками сигналов, поскольку они должны иметь уникальные имена. Также connectметод полезен при подключении сигналов к [динамически создаваемым объектам](https://doc.qt.io/qt-5/qtqml-javascript-dynamicobjectcreation.html) .

Существует соответствующий disconnect()метод удаления подключенных сигналов

Сигнал для подключения сигнала

Соединяя сигналы с другими сигналами, connect()метод может формировать разные цепочки сигналов. Всякий раз, когда сигнал [TapHandler](https://doc.qt.io/qt-5/qml-qtquick-taphandler.html)tapped излучается, сигнал sendтакже будет излучаться автоматически.