В случае использования Qt

1. Создание DLL с помощью Qt.

2. Создание элемента управления с помощью Qt.

3. Подключение элемента управления в проект Qt.

4. Добавление свойств элементам управления.

5. Создание новых сигналов в Qt.

6. Приведение типов элементов управления.

7. Получение имени класса виджета средствами Qt .

8. Использование иерархии виджетов Qt (методы, связанные с понятиями child и parent).

9. Работа с событиями.

10. Отличия виджетов от элементов QML

1. Чтобы создать DLL в Qt, вам нужно настроить проектный файл .pro для сборки как библиотеки (TEMPLATE = lib), а также указать тип библиотеки (CONFIG += dll). Вам также потребуется экспортировать классы и функции с использованием макросов Q\_DECL\_EXPORT и Q\_DECL\_IMPORT.
2. Элементы управления в Qt создаются путем наследования от базовых классов виджетов, таких как QWidget или QMainWindow. Вы определяете свои виджеты, добавляя к ним свойства, методы и события.
3. Подключение пользовательского элемента управления осуществляется путем его инстанцирования и добавления в макет (layout) или другой виджет. Это можно сделать как программно, так и через Qt Designer.
4. Свойства в Qt добавляются с помощью макроса Q\_PROPERTY. Эти свойства могут быть затем использованы в QML или в C++ для настройки поведения виджета.
5. Новые сигналы создаются в классе с помощью макроса signals. Сигналы определяются как методы, но без реализации. Они используются для уведомления о событиях внутри класса.
6. В Qt приведение типов виджетов обычно осуществляется с помощью qobject\_cast, что позволяет безопасно преобразовать типы между различными виджетами и классами, основанными на QObject.
7. Имя класса виджета можно получить с помощью метода metaObject()->className().
8. В Qt каждый виджет может иметь дочерние виджеты (children) и родительский виджет (parent). Это используется для управления жизненным циклом виджетов и организации макета интерфейса.
9. В Qt события обрабатываются путем переопределения методов, таких как mousePressEvent, keyPressEvent и т.д. в пользовательских классах.
10. Виджеты (используемые в C++) предоставляют более традиционный подход к созданию GUI, в то время как QML предлагает более декларативный и динамичный способ разработки интерфейсов. QML также облегчает разработку анимаций и более интерактивных интерфейсов.