## Введение в QML

**Определение QML**

QML (Qt Meta-Object Language) - это декларативный язык разметки, разработанный компанией Qt для создания пользовательских интерфейсов.

Основной целью QML является упрощение процесса создания интерфейсов и легкая интеграция с кодом на C++.

**QML и HTML**

Чтобы лучше разобраться в том, что такое QML, давайте сравним этот язык с другим известным языком разметки — HTML.

QML и HTML представляют собой два разных языка разметки, но они имеют некоторые общие концепции. Вот как можно объяснить разницу между QML и HTML:

Контекст использования:

QML применяется в основном для создания настольных и встроенных приложений, использующих фреймворк Qt.

HTML широко используется для создания веб-страниц, доступных через веб-браузеры на различных устройствах.

Поддержка функциональности:

QML позволяет создавать богатые графические пользовательские интерфейсы с анимациями, эффектами и даже векторной графикой.

HTML позволяет создавать структурированный текст, изображения и интерактивные элементы, но для более сложных графических эффектов часто используются CSS и JavaScript.

**Структура QML-файла**

QML-файл представляет собой текстовый файл с расширением .qml. Файл состоит из элементов интерфейса, описанных согласно декларативному синтаксису QML.

Основные элементы интерфейса в QML:

Item: Основной строительный блок интерфейса, используется для группировки и структурирования элементов.

Rectangle: Элемент для создания прямоугольных областей, которые могут содержать другие элементы.

Text: Используется для отображения текстовой информации.

Image: Позволяет отображать изображения на интерфейсе.

Button: Элемент для создания интерактивных кнопок.

ListView: Используется для отображения списков элементов с возможностью прокрутки.

MouseArea: Элемент для обработки событий мыши.

К каждому элементу применимы свойства. Они являются способом настройки внешнего вида и поведения элементов.

Пример: width, height, color, text, font, border.

## Основы тестирования

**Для чего нужно тестирование ПО**

Представьте, что мы создали новую игру или программу. Чтобы узнать, что она работает правильно и не вызывает ошибок, мы используем тестирование.

Тестирование помогает найти ошибки в программе, которые могут сделать её ненадежной или неработоспособной. Также тестирование позволяет убедиться, что программа делает всё то, что обещала делать, и не делает лишних или неправильных вещей.

**Основные типы тестирования**

1. Функциональное тестирование — это проверка, что программа выполняет свои задачи так, как должна. Например, кнопка "Старт" на игре действительно начинает игру.
2. Регрессионное тестирование — это когда мы проверяем, что новые изменения в программе не сломали то, что раньше уже работало. Например, после обновления игры проверяем, что старые уровни всё ещё возможно пройти без багов.
3. Пользовательское тестирование — самое массовое и естественное тестирование на обычных пользователей. Когда пользователи в отзывах говорят, что им нравится, а что нет, это помогает сделать программу удобнее.

**План тестирования**

План тестирования (Test Plan) — это документ, который описывает весь объем работ по тестированию и включает: описание объекта, задачи и цели тестирования, описание необходимого в процессе работы оборудования, сценарий тестирования и др.

Объект тестирования — полное описание того, что мы будем тестировать. Например, тестовыми объектами могут быть ПО для персональных компьютеров, веб-сайт или мобильные приложения.

Основная цель тестирования обычно заключается в том, чтобы подтвердить, что продукт соответствует спецификации, то есть он делает то, что указано в спецификации, и не делает того, чего в документе нет.

Основной задачей тестирования ПО является получение информации о статусе готовности заявленной функциональности системы или приложения.

Данные цель и задачи могут дополняться или корректироваться при планировании конкретных исследований.

Окружение — это конфигурация ПО, виртуального контейнера или сервера в котором планируется проводить тестирование, т.е. песочница для тестирования. Исходя из необходимого окружения, подбирается соответствующий тестовый стенд. Обычно так называют устройство (ноутбук, сервер, смартфон и планшет), на котором будут тестировать работу различных функций тестируемого ПО.

Тестовый сценарий (Test Case) — это раздел плана тестирования, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

**Баг-репорт**

По результатам работы специалист по тестированию составляет баг-репорт (от английского "bug report"). Это документ, в котором специалист сообщает о возникновении ошибки или дефекта в программе. Баг-репорт содержит не просто описание проблемы, а помогает разработчикам воспроизвести ошибку, чтобы они быстрее распознали ее корень.

Документ может отличаться в зависимости от проекта, но есть обязательные поля\*, которые везде примерно одинаковые.

| **Поле** | **Что содержит** |
| --- | --- |
| Заголовок\* | Суть проблемы. Должен быть ёмким и понятным. |
| Описание | Обычно дублирует заголовок, поэтому не всегда обязательно. |
| Окружение\* | Где нашли баг. Например, ОС, браузер или тестовая среда. |
| Шаги воспроизведения\* | Описание действий, которые нужно совершить, чтобы дойти до бага. |
| Фактический результат\* | Что видим после того, как воспроизвели баг. |
| Ожидаемый результат\* | Что на самом деле хотели увидеть, как это должно работать по ТЗ. |
| Вложения | Логи, скриншоты |
| Дополнительная информация | Например, пояснения от тестировщика: какие способы он ещё пробовал, чтобы воспроизвести баг, и что получилось. |

## Работа с запросами из консоли

Консоль помогает быстро и точно выполнять команды, не используя графический интерфейс. Чтобы запустить консоль в ОС Windows, нажмите на клавиши Win+R, в появившемся окне введите “cmd”, после чего нажмите Enter.

Вот некоторые команды, которые часто используют при работе из консоли:

cd — перейти в другую папку;

ls (или dir в Windows) — показать список файлов и папок;

mkdir — создать новую папку;

cp — скопировать файлы или папки;

mv — переместить файлы или папки;

rm — удалить файлы или папки.

Но мы рассмотрим и опробуем на практике команду curl. С ее помощью можно быстро скачивать файлы в нужную папку из интернета по ссылке.  
  
Попробуйте ввести команду в консоль:  
curl -O https://…