

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования
Национальный исследовательский Нижегородский государственный университет им. Н.И.
Лобачевского

Институт информационных технологий, математики и механики

Отчет по практическому заданию №2
«Инструменты разработки мобильных приложений»

Выполнил:

студент группы 381908-4

Рябцев М. В.

Нижний Новгород
2022

Оглавление

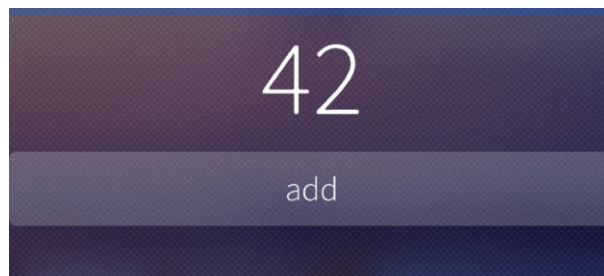
Цель задачи	3
Постановка задачи	4
Описание программной реализации	5
Руководство пользователя	6
Заключение	7
Приложение	8

Цель задачи

В данном лабораторной работе требуется освоить процесс создания нового проекта, изучить его структуру, научиться запускать приложение. Посмотреть возможности среды Qt Creator и эмулятора.

Постановка задачи

1. Создать новый проект со стандартной заготовкой приложения.
2. Посмотреть содержимое вкладок Qt creator (Welcome, Edit, Debug, Projects, Sailfish OS, Help). Выяснить назначение каждой из них.
3. Изучить структуру созданного проекта (каталоги, расположение файлов). Выяснить соглашения по размещению файлов для проектов Qt для Sailfish OS.
4. Изучить содержимое *.pro файла проекта. Выяснить назначение разделов файла. Документация по файлам проекта доступна по адресу <http://doc.qt.io/qt-5/qmake-project-files.html>
5. Изучить содержимое *.qml файлов. Выяснить назначение элементов, используемых в файле, с помощью интерактивной справки (нажать на элемент в файле, затем вызвать справку нажатием на кнопку F1 на клавиатуре).
6. Запустить эмулятор, освоить принципы навигации в Sailfish OS, посмотреть возможности и настройки эмулятора. Научиться осуществлять навигацию на устройстве Sailfish OS, узнать возможности настроек устройства (приложение Settings).
7. Собрать и запустить заготовку приложения на эмуляторе
8. Используя материал слайдов 35 и 39 из лекции, изменить приложение таким образом, чтобы оно содержало одно текстовое поле со счётчиком и одну кнопку, позволяющую увеличивать значение счётчика на 1. Размещению элементов на экране внимания можно не уделять.



Описание программной реализации

- Режим **Редактор** используется для редактирования проекта и файлов исходного кода.
- Режим **Дизайн** используется для проектирования и создания интерфейсов пользователя приложения. Этот режим доступен для файлов интерфейса пользователя.
- Режим **Отладка** используется для просмотра состояния нашей программы во время отладки.
- Режим **Проекты** используется для настройки сборки и запуска проекта. Этот режим доступен только если открыт проект.
- Режим **Справка** используется для просмотра документации Qt.

Для реализации счетчика воспользуемся элементами **Label** и **Button**:

Элемент Label

- *property int count: 0* – пользовательское свойство внутри элемента.
- *id: counter* – идентификатор элемента.
- *text: count* – текст надписи.

Элемент Button

- *text: "Кнопка"* – текст кнопки.
- *onClicked: counter.count++* – функция кнопки по нажатию. Увеличивает count на 1.

Руководство пользователя

После запуска программы пользователь видит страницу с счетчиком и кнопкой, при нажатии на которую счетчик увеличивается.

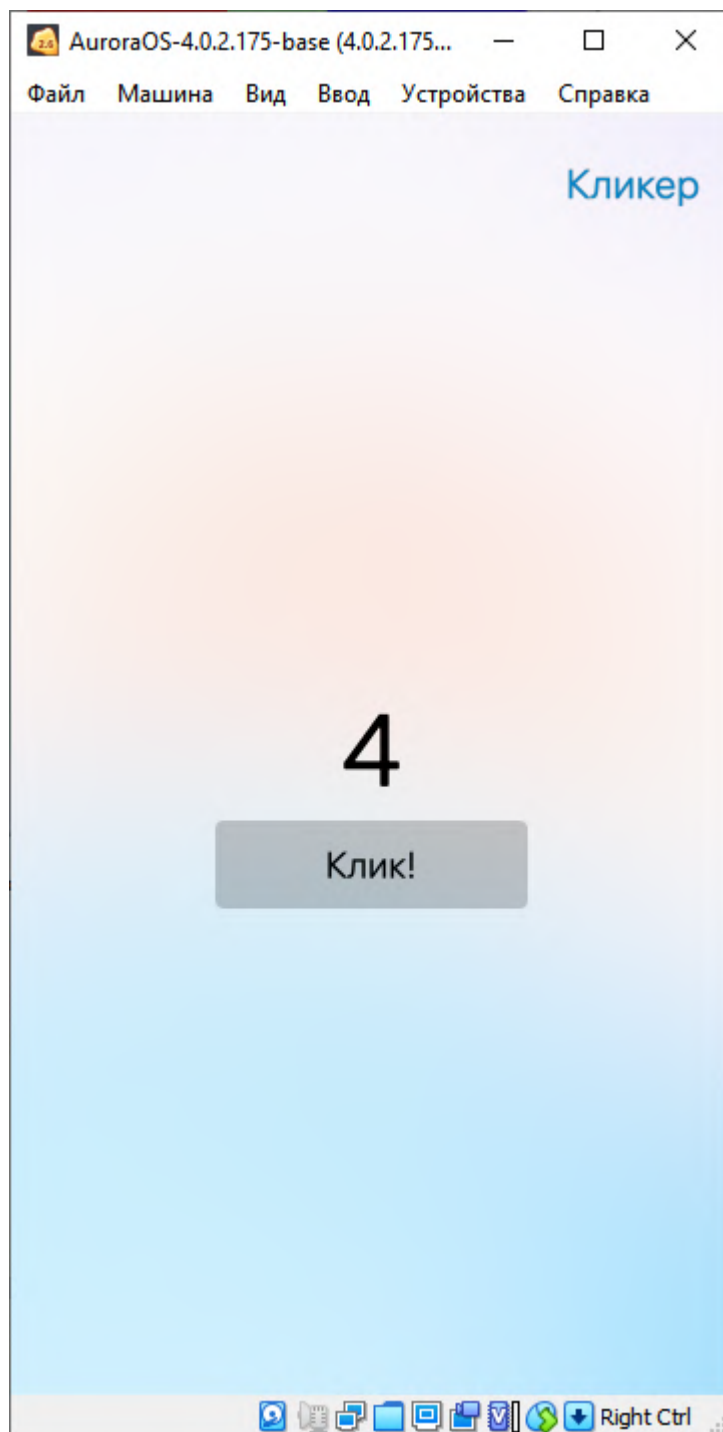


Рис. 1: Страница со счетчиком

Заключение

В процессе выполнения данной лабораторной работы мы освоили процесс создания нового проекта, изучили его структуру, научились запускать приложение, посмотрели возможности среды Qt Creator и эмулятора. А также выполнили практическое задание, в котором при нажатии на кнопку значение счетчика увеличивалось на 1.

Приложение

MainPage.qml

```
import QtQuick 2.0
import Sailfish.Silica 1.0

Page {
    objectName: "mainPage"
    allowedOrientations: Orientation.All

    PageHeader {
        objectName: "pageHeader"
        title: "Кликер"
    }

    Label {
        property int count: 0
        id: counter
        text: count
        anchors.centerIn: parent
        bottomPadding: 10
        font.pixelSize: 100
    }

    Button {
        text: "Клик 123123123!"
        anchors.horizontalCenter: counter.horizontalCenter
        anchors.top: counter.bottom
        onClicked: counter.count++
    }
}
```