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»

**ОТЧЕТ**

**По лабораторной работе №1**

по дисциплине «Технология программирования мобильных систем»

**Выполнила:** студентка группы 381906-3

Сорокина Екатерина Вячеславовна

**Руководитель:**

Доцент, кандидат технических наук

Борисов Николай Анатольевич

Нижний Новгород  
2022

**Содержание**

[1. Цель 3](#_Toc85441990)

[2. Постановка задачи 4](#_Toc85441991)

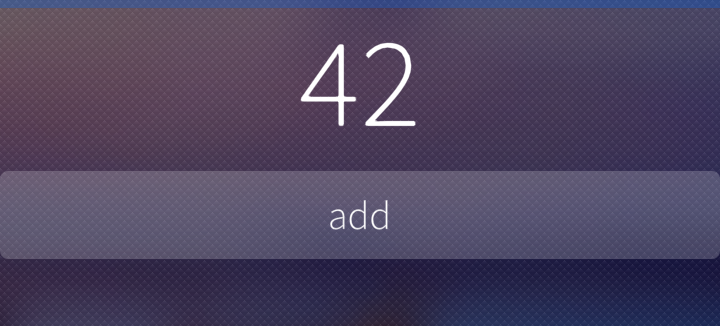
[3. Решение поставленной задачи 5](#_Toc85441992)

[4. Руководство пользователя 6](#_Toc85441993)

[Приложение 7](#_Toc85441994)

1. Цель

Целью данной работы является: освоить процесс создания нового проекта, изучить его структуру, научиться запускать приложение. Посмотреть возможности среды Qt Creator и эмулятора.

1. Постановка задачи
2. Создать новый проект со стандартной заготовкой приложения.
3. Посмотреть содержимое вкладок Qt creator (Welcome,Edit, Debug, Projects, Sailfish OS, Help). Выяснить назначение каждой из них.
4. Изучить структуру созданного проекта (каталоги, расположение файлов). Выяснить соглашения по размещению файлов для проектов Qt для Sailfish OS.
5. 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>
6. 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).
7. 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).
8. Собрать и запустить заготовку приложения на эмуляторе
9. 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.  
   
10. Собрать и запустить приложение на эмуляторе. Убедиться в правильности его работы.
11. Решение поставленной задачи

Создав проект со стандартной заготовкой приложения, мы переходим в файл FirstPage.qml, который подходит для решения задачи.

Внутри файла уже есть контейнер Column, поэтому создавать другой для размещения элементов нам не нужно.

Так как по условию задачи нужно только одно текстовое поле, то элемент PageHeader можно удалить.

Внутри Column есть элемент Label, который подойдёт для размещения текстового поля.

Чтобы создать кнопку, увеличивающую значение счетчика на 1, мы в дереве страницы (внутри Column, но за пределами Label) создаём элемент Button. Ему присваиваем такие свойства, как id (чтобы к Button и его свойствам можно было обратиться и за пределами элемента), text (для отображения текста поверх кнопки). Пусть уникальным id нашего Button будет id: counter. Текстом, отображаемым поверх кнопки, станет text: qsTr("Click to add"). Далее понадобится переменная для запоминания числа счетчика. Создадим новое свойство с ключевым словом property укажем его тип как var (тип var — это общий тип свойства, который может ссылаться на любой тип данных). Выберем переменной имя и приравняем её значение к нулю property var clickCounter: 0.

Чтобы значение счетчика изменялось на 1, создадим свойство onClicked для изменения переменной, внутри фигурных скобок которого пишем clickCounter++; Это позволит увеличивать счетчик на 1 каждый раз, при нажатии на кнопку.

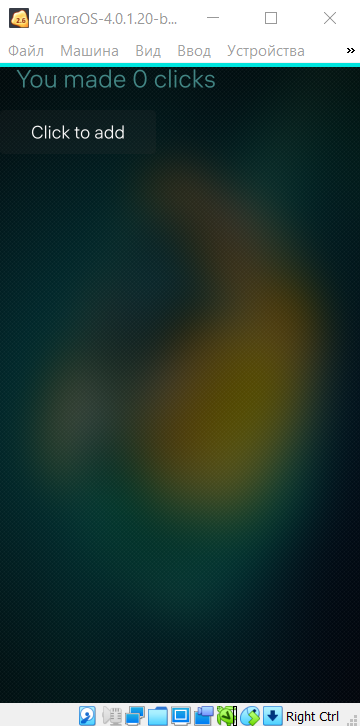
Чтобы значение счетчика выводилось в текстовом поле, изменим свойство text элемента Label и прибавим туда *counter*.clickCounter (через id обращаемся к переменной clickCounter).

Осталось только сохранить все изменения и запустить эмулятор.

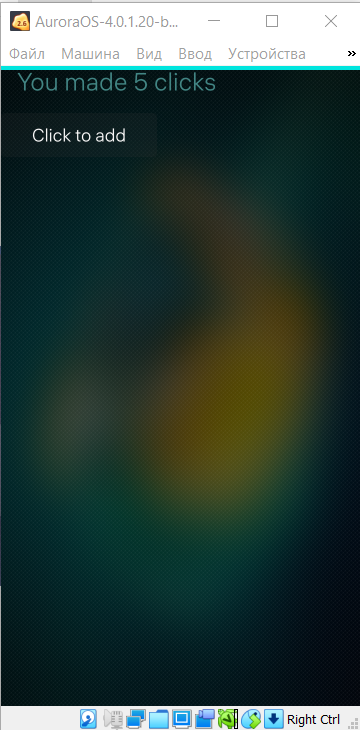
Таким образом, при каждом нажатии на кнопку, счетчик будет увеличиваться на единицу.

1. Руководство пользователя

При запуске приложения пользователь увидит кнопку и текстовое поле с начальным значением счетчика (по умолчанию это 0).



При нажатии на созданную нами кнопку «Click to add» значение счетчика в текстовом поле изменится. Каждое нажатие увеличивает значение поля на 1.



Приложение

// FirstPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

// The effective value will be restricted by ApplicationWindow.allowedOrientations

allowedOrientations: Orientation.All

// To enable PullDownMenu, place our content in a SilicaFlickable

SilicaFlickable {

anchors.fill: *parent*

// PullDownMenu and PushUpMenu must be declared in SilicaFlickable, SilicaListView or SilicaGridView

PullDownMenu {

MenuItem {

text: *qsTr*("Show Page 2")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("SecondPage.qml"))

}

}

// Tell SilicaFlickable the height of its content.

contentHeight: *column*.height

// Place our content in a Column. The PageHeader is always placed at the top

// of the page, followed by our content.

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

Label {

x: Theme.horizontalPageMargin

text: *qsTr*("You made ") + *counter*.clickCounter + *qsTr*(" clicks")

color: Theme.secondaryHighlightColor

font.pixelSize: Theme.fontSizeExtraLarge

}

Button {

id: *counter*

text: *qsTr*("Click to add")

property var clickCounter: 0

onClicked: {

*clickCounter*++;

}

}

}

}

}