МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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**

Направление подготовки: «Программная инженерия»

**ОТЧЕТ**

по лабораторной работе №3

на тему

**«Инструменты разработки мобильных приложений»**

**Выполнил:** студент группы 381808-2 Евсеев А.Д.

**Преподаватель:**

Карчков Д.А.

**Нижний Новгород**

**2021**

# Содержание

[**Содержание**](#_dyul5kgijpye) **1**

[**Постановка задачи**](#_l2z0b0z21xxd) **2**

[**Решение задачи**](#_uf01387ib17) **3**

[**Приложение**](#_4gwbr649kfxb) **5**

[FirstPage.qml](#_ns18652aya7z) 5

[SecondPage.qml](#_euvo2kwq9b9r) 8

# Постановка задачи

В рамках лабораторной работы предстоит применить типовые элементы интерфейса Sailfish OS .

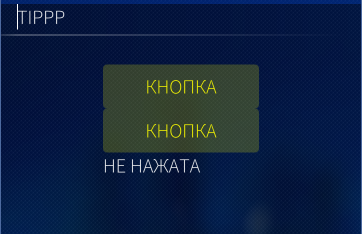
Работа состоит из следующих шагов:

* Создать текстовое поле для ввода числа с заголовком и подсказкой.
* Создать кнопку, которая будет сохранять визуально нажатое состояние, после того, как пользователь нажал на неё один раз.
* Создать кнопку и поле с текстом. Поле с текстом должно отображать нажата ли кнопка или нет выводом текста “Нажата” или “Отпущена”.
* Создать кнопку со значением, которая будет отображать количество нажатий на нее.
* Создать селектор даты, который будет отображать выбранную дату в консоли.
* Создать селектор времени, который будет выводить выбранное время в консоли.
* Создать поле с выпадающим списком, позволяющее выбрать строку из списка. Результат выбора отобразить в консоли.
* Создать переключатель с текстом, в тексте отобразить состояние переключателя “Включен” или “Выключен”.
* Создать ползунок и поле с текстом. Поле с текстом должно отображать текущее значение ползунка.

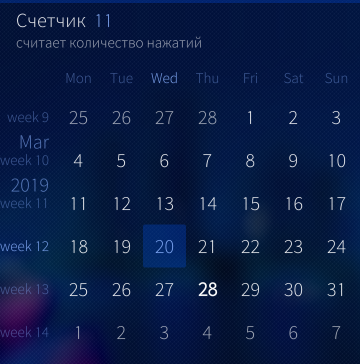
# 

# 

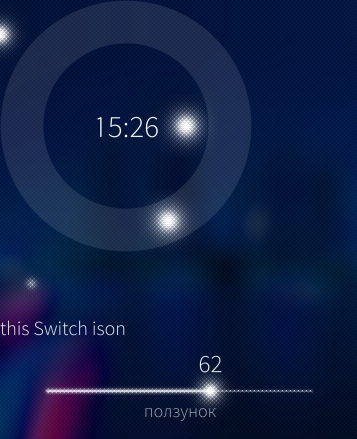
# Решение задачи



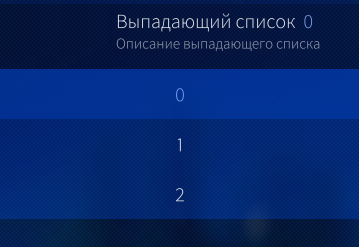
*Шаг :.*



*Шаг :.*



*Шаг :.*



*Шаг :.*

# Приложение

## FirstPage.qml

**import** QtQuick 2.0

**import** Sailfish.Silica 1.0

Page {

**id: page**

**allowedOrientations:** Orientation.All

SilicaFlickable {

**anchors.fill:** parent

PullDownMenu {

MenuItem {

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

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

}

}

ComboBox{

**visible:** **true**

**x:** 200

**label:**"Выпадающий список"

**description:** "Описание выпадающего списка"

**menu:** ContextMenu{

**id: contextMenu**

MenuItem { **text:** "0" }

MenuItem { **text:** "1" }

MenuItem { **text:** "2" }

}

**onCurrentIndexChanged:** console.log(contextMenu.data)

}

Column{

**width:** page.width

TextField {

**label:** "TIPPP"

**placeholderText:** label

**width:** parent.width

**color:** "white"

**placeholderColor:** "white"

}

Button {

**anchors.horizontalCenter:** parent.horizontalCenter

**color:** "yellow"

**highlightColor:** "red"

**highlightBackgroundColor:** "green"

**text:**"КНОПКА"

**onClicked:**{ color = "white"}

}

Button {

**id: button2**

property **int** **i:** 0;

Label { **anchors.top:** button2.bottom; **id: label1**; **text:**"НЕ НАЖАТА" }

**anchors.horizontalCenter:** parent.horizontalCenter

**color:** "yellow"

**highlightColor:** "red"

**highlightBackgroundColor:** "green"

**text:**"КНОПКА"

**onClicked:** {

**if**(i == 0){ label1.text = "НАЖАТА"; color = "white"; i = 1 }

**else** { label1.text="НЕ НАЖАТА"; i = 0; color = "green" }

}

}

ValueButton {

property **int** **i:** 0

**label:** "Счетчик"

**description:** "считает количество нажатий"

**value:** "0"

**onClicked:** value = i++

}

DatePicker {

**daysVisible:** **true**

**weeksVisible:** **true**

**on\_ChangingDateChanged:** console.log(date)

}

}

}

}

## SecondPage.qml

**import** QtQuick 2.0

**import** Sailfish.Silica 1.0

Page {

**id: page**

**allowedOrientations:** Orientation.All

SilicaFlickable {

**id: listView**

Column {

Item {

**id:tp**

**height:** timePicker.height

**width:** parent.width

TimePicker {

**id:timePicker**

**hour:** 15

**minute:** 26

**onMinuteChanged:** console.log(time)

**onHourChanged:** console.log(time)

}

Label {

**anchors.centerIn:** timePicker

**text:** timePicker.timeText

**font.pixelSize:** Theme.fontSizeExtraLarge

}

}

Column {

Switch { **id: mute** }

Label { **text:** "this Switch is" + (mute.checked ? "off" : "on") }

}

Slider {

**width:** page.width

**label:** "ползунок"

**maximumValue:** 100

**minimumValue:** 0

**value:** 50

**stepSize:** 1

**valueText:** value

**onValueChanged:** console.log(value)

}

}

}

}