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**

**ОТЧЕТ**

по лабораторной работе №5

на тему

**«Управление данными и использование шаблона MVC»**

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

**Преподаватель**

Карчков Д.А.

**Нижний Новгород**

**2021**

# Содержание

[**Содержание**](#_m1wcvuc53js8) **1**

[**Постановка задачи**](#_78j2pfsccmus) **2**

[**Результаты выполнения задачи**](#_9la3jfeurv38) **3**

[**Приложение**](#_d8zkhqa31wzd) **6**

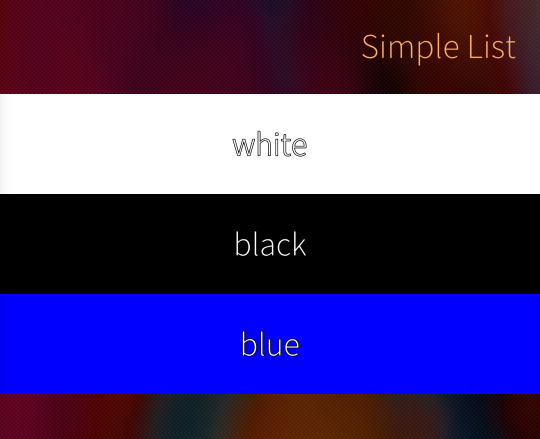
# Постановка задачи

В рамках лабораторной работы предстоит научиться использовать различные модели для отображения данных в прокручиваемых списках, взаимодействовать с базой данных и управлять настройками приложения.

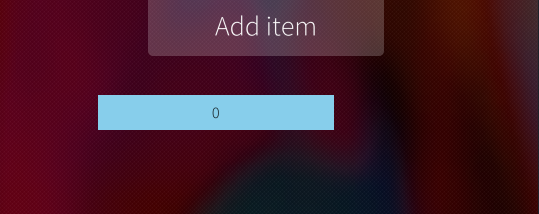
Работа состоит из 8 подзадач:

* Создать приложение, которое позволяет отображать список из прямоугольников с использованием ListModel. В модели должны настраиваться цвет фона и текста внутри прямоугольника. Текст содержит название цвета фона прямоугольника.
* Создать приложение, которое позволяет отображать список из прямоугольников. Нажатие на кнопку над списком добавит новый элемент. Нажатие на элемент в списке удалит его из списка. В прямоугольниках должен отображаться порядковый номер, присваиваемый при добавлении в список. При удалении элементов порядковые номера у добавленных прямоугольников остаются неизменными.
* Выполнить задание 1 с использованием javascript-модели.
* Получить и отобразить курсы валют из ресурса ЦБ РФ по адресу http://www.cbr.ru/scripts/XML\_daily.asp.
* Выполнить задание 4 с использованием XMLHttpRequest.
* Создать приложение, позволяющее добавлять и удалять заметки с использованием базы данных и отображать их в списке. Текстовое поле служит для ввода текста, кнопка для добавления заметки, нажатие на заметку удалит её.
* Создать приложение с текстовым полем и полем с флажком, значение которых сохраняется в настройках приложения с помощью ConfigurationValue.
* Выполнить задание 7 с помощью ConfigurationGroup.

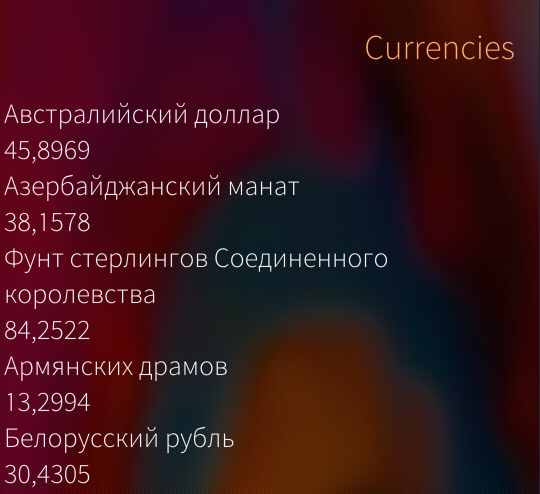
# Результаты выполнения задачи



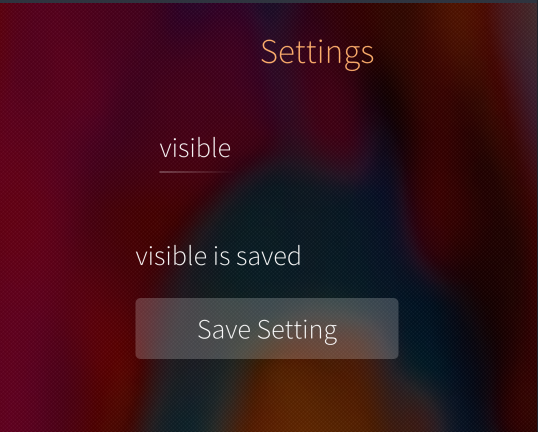
*Подзадача 1.*



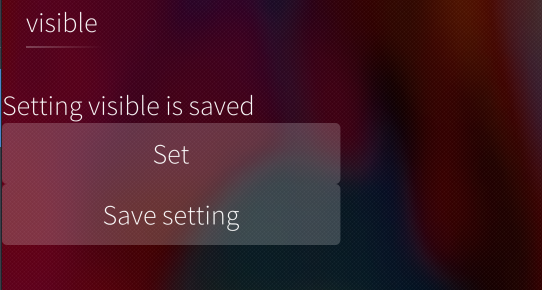
*Подзадача 2.*



*Подзадачи 4-5.*



*Подзадача 7.*



*Подзадача 8.*

# 

# 

# Приложение

**import** QtQuick 2.3

**import** Sailfish.Silica 1.0

**import** QtQuick.XmlListModel 2.0 *// Necessary for task 4*

**import** QtQuick.LocalStorage 2.0 *// Necessary for task 6*

**import** Nemo.Configuration 1.0 *// Necessary for task 7-8*

Page {

*// Task 1*

ListModel {

**id: dataModel**

ListElement { **color:** "white"; **text:** "white" }

ListElement { **color:** "black"; **text:** "black" }

ListElement { **color:** "blue"; **text:** "blue" }

}

SilicaListView {

**anchors.fill:** parent

**header:** PageHeader { **title:** "Simple List" }

**model:** dataModel

**delegate:** Rectangle {

**width:** parent.width

**height:** 200

**color:** model.color

Text {

**font.pointSize:** 50

**anchors.centerIn:** parent

**text:** model.text

**color:** "white"

**style:** Text.Outline

}

}

}

*// Task 2 (add only one element to list)*

ListModel { **id: dataModel** }

Column {

SilicaListView {

**x:** 200

**y:** 200

**width:** parent.width

**height:** parent.height - addButton.height;

**model:** dataModel

**delegate:** Rectangle {

**width:** parent.width; **height:** 70; **color:** "skyblue";

Text { **anchors.centerIn:** parent; **text:** model.text + model.index }

}

}

Button {

**id: addButton**

**x:** 300

**y:** 300

**text:** "Add item"

**onClicked:** { dataModel.append({ **text:** "" }) }

}

}

*// Task 4*

XmlListModel {

**id: xmlListModel**

**source:** "http://www.cbr.ru/scripts/XML\_daily.asp"

**query:** "/ValCurs/Valute" *// Get all currencies using XPath*

XmlRole { **name:** "Name"; **query:** "Name/string()"; } *// Currency name*

XmlRole { **name:** "Value"; **query:** "Value/string()"; } *// Currency value*

}

SilicaListView {

**anchors.fill:** parent

**header:** PageHeader { **title:** "Currencies" }

**model:** xmlListModel

**delegate:** Column {

**x:** 10

**width:** parent.width - 2 \* x

Label { **width:** parent.width; **wrapMode:** Text.WordWrap; **text:** Name }

Label {**width:** parent.width; **wrapMode:** Text.WordWrap; **text:** Value }

}

}

*// Task 5 (correctly parse data from site but encoding is invalid)*

*// JavaScript function to parse currencies from cbr.ru*

**function** loadNews() {

**var** contentType = "application/x-www-form-urlencoded; charset=win-1251";

**var** xhr = **new** XMLHttpRequest();

xhr.open('GET', 'http://www.cbr.ru/scripts/XML\_daily.asp', **true**);

xhr.setRequestHeader('Content-type', contentType);

xhr.onreadystatechange = **function**() {

**if** (xhr.readyState === XMLHttpRequest.DONE) {

xmlListModel.xml = xhr.responseText;

}

}

xhr.send();

}

XmlListModel {

**id: xmlListModel**

**query:** "/ValCurs/Valute"

XmlRole { **name:** "Name"; **query:** "Name/string()"; }

XmlRole { **name:** "Value"; **query:** "Value/string()"; }

}

SilicaListView {

**anchors.fill:** parent

**header:** PageHeader { **title:** "Currencies" }

**model:** xmlListModel

**delegate:** Column {

**x:** 10

**width:** parent.width - 2 \* x

Label { **width:** parent.width; **wrapMode:** Text.WordWrap; **text:** Name }

Label {**width:** parent.width; **wrapMode:** Text.WordWrap; **text:** Value }

}

}

**Component.onCompleted:** loadNews()

*// Task 7 (not sure if i get the task correctly)*

ConfigurationValue { **id: setting**; **key:** "/apps/app\_name/setting\_name"; **defaultValue:** "Setting" }

Column {

**anchors.horizontalCenter:** parent.horizontalCenter

**spacing:** 50

PageHeader { **title:** "Settings" }

TextField {**id: textField**; **text:** "Enter setting value"; **onClicked:** text="" }

Label { **text:** setting.value + " is saved" }

Button { **width:** parent.width; **text:** "Save Setting"; **onClicked:** setting.value = textField.text }

}

*// Task 8 (again not sure i understand the task)*

ConfigurationGroup { **id: settings**; **path:** "/apps/app\_name/settings"; property bool **set:** **false** }

Column {

TextField { **id: textField**; **text:** "Enter setting name"; **onClicked:** text="" }

Label { **id: label**; **text:** "" }

Button {

**width:** parent.width

**text:** "Set"

**onClicked:** {

settings.set = **true**

label.text = textField.text + " is set and ready to save"

}

}

Button {

**width:** parent.width

**text:** "Save setting"

**onClicked:** {

settings.setValue(settings.set, textField.text)

label.text = "Setting " + textField.text + " is saved"

}

}

}

}