

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

Направление подготовки: «Фундаментальная информатика и информационные
технологии»

Отчет по практическому заданию №6:

«Инструменты разработки мобильных приложений»

Выполнила:

студентка группы 381906-2

Яшина Дарья Степановна

Нижний Новгород

2022

Постановка задачи

Цель: научиться использовать различные модели для отображения данных в прокручиваемых списках, взаимодействовать с базой данных и управлять настройками приложения.

Шаги:

1. 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.

2. 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.

3. Выполнить задание 1 с использованием javascript-модели.

4. Получить и отобразить курсы валют из ресурса ЦБ РФ по адресу http://www.cbr.ru/scripts/XML_daily.asp.

5. Выполнить задание 4 с использованием XMLHttpRequest.

6. 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.

7. Создать приложение с текстовым полем и полем с флажком, значение которых сохраняется в настройках приложения с помощью ConfigurationValue.

8. Выполнить задание 7 с помощью ConfigurationGroup.

Руководство программиста

Для реализации данной лабораторной работы нам потребовались следующие инструменты :

- **ListModel**

id: *listModel* – идентификатор модели

ListElement { text: "Черный"; bg_color: "black"; text_color: "white"} – элемент списка модели

- **SilicaListView**

model: listModel – модель, входные данные

delegate - предоставляет шаблон, определяющий каждый элемент

header: PageHeader { } - заголовок

- **XmlListModel** - модель только для чтения из данных XML

XmlRole - объекты определяют атрибуты элемента модели

Component.onCompleted: loadRate() – обработчик сигнала

- **ConfigurationValue** - предоставляет доступ к отдельному значению конфигурации.

id: *setting* – идентификатор

key: " " – ключ

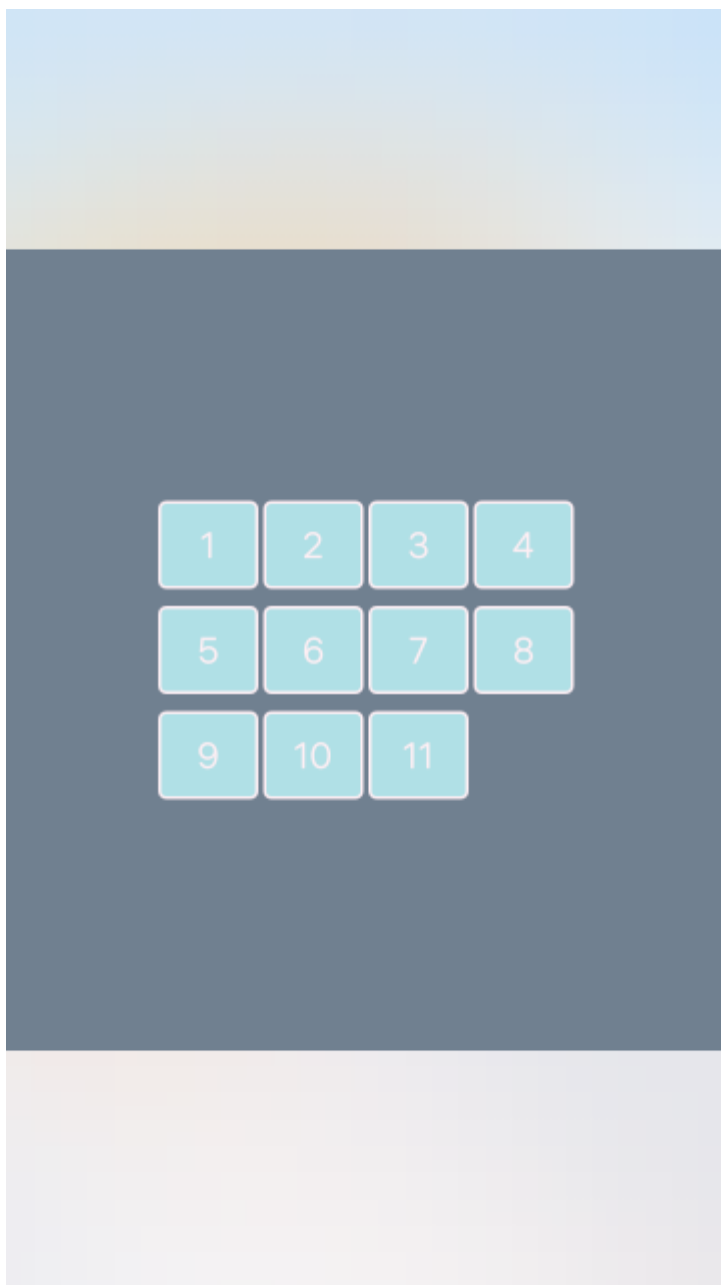
defaultValue: "Setting" - это свойство содержит значение по умолчанию для свойства значения_ если ключ не существует.

- **ConfigurationGroup** - предоставляет доступ к группе значений конфигурации.

Руководство пользователя

После запуска программы, открывается окно эмулятора, в котором отображается страница с кнопками. Каждая кнопка соответствует определенному номеру задания.

- Главная страница



- Задания по порядку

Белый

Чёрный

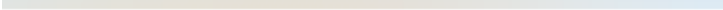
Синий



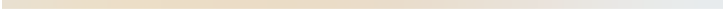


Добавить

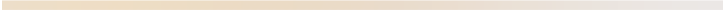
Элемент 1



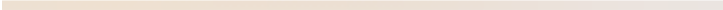
Элемент 2



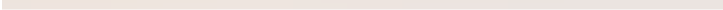
Элемент 3



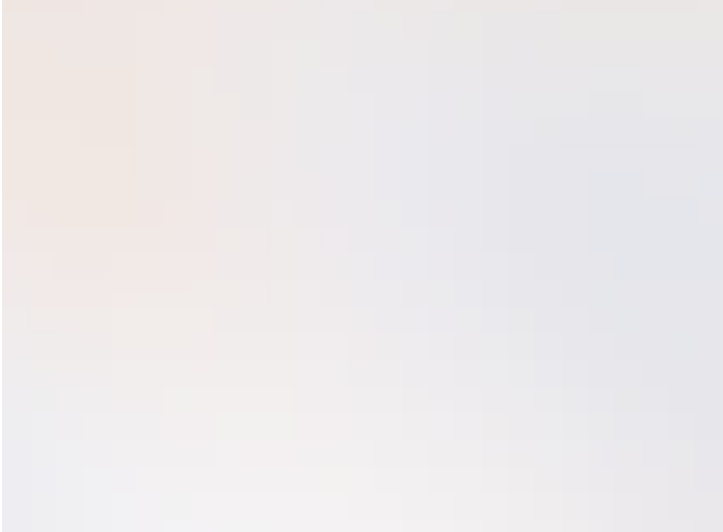
Элемент 4

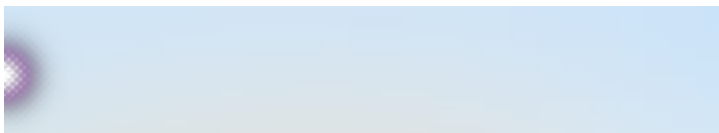


Элемент 5

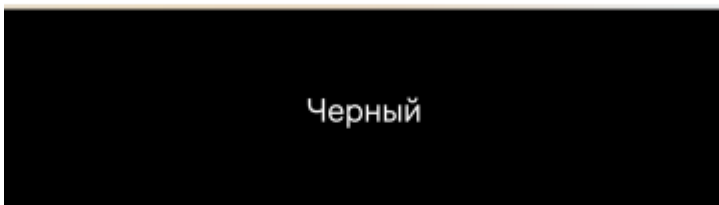


Элемент 6

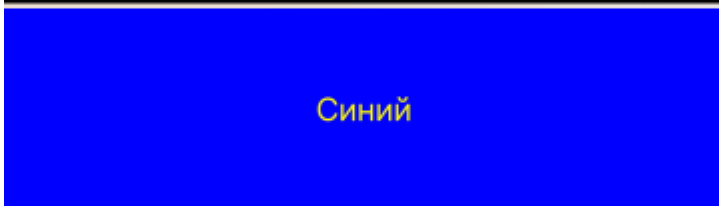




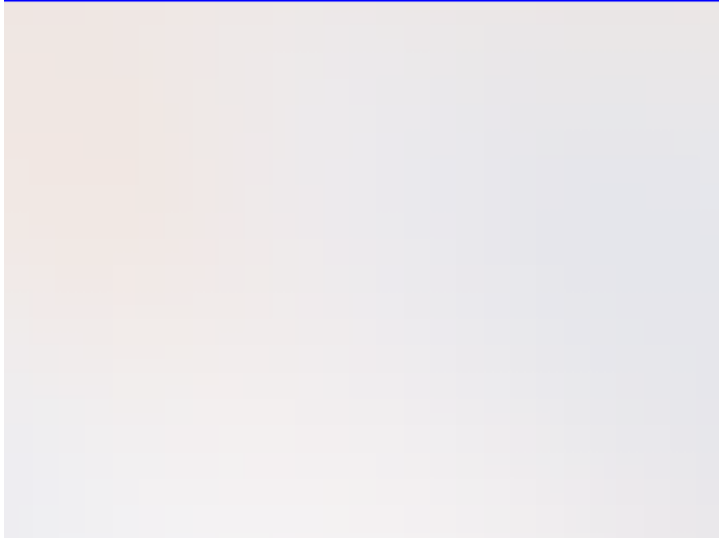
Белый



Черный



Синий



Валюты

Австралийский доллар

42,0235

Азербайджанский манат

36,8072

Фунт стерлингов Соединенного королевств

76,3256

Армянских драмов

15,8350

Белорусский рубль

25,2531

Болгарский лев

33,6844

Бразильский реал

11,9859

Венгерских форинтов

15,8435

Гонконгских долларов

80,5201

Датских крон

88,5727

Доллар США

62,5722

Валюты

Австралийский доллар	42,0235
Азербайджанский манат	36,8072
Фунт стерлингов Соединенного королевств	76,3256
Армянских драмов	15,8350
Белорусский рубль	25,2531
Болгарский лев	33,6844
Бразильский реал	11,9859
Венгерских форинтов	15,8435
Гонконгских долларов	80,5201
Датских крон	88,5727
Доллар США	62,5722
Евро	65.6762

Rou

Добавить

hello

hooo

wow

wow

EN

q w e r t y u i o p

a s d f g h j k l

^ z x c v b n m <x

!123 , . ↵

Текст|поле



Active

Переключа
тель

RU

й ц у к е н г ш щ з х ъ

ф ы в а п р о л д ж э

^ я ч с м и т ь б ю <*

!123 - , . ↵

Текстовое пол



Inactive

Переключ
атель(2)

Армянских драмов

15,8350

Белорусский рубль

25,2531

Болгарский лев

33,6844

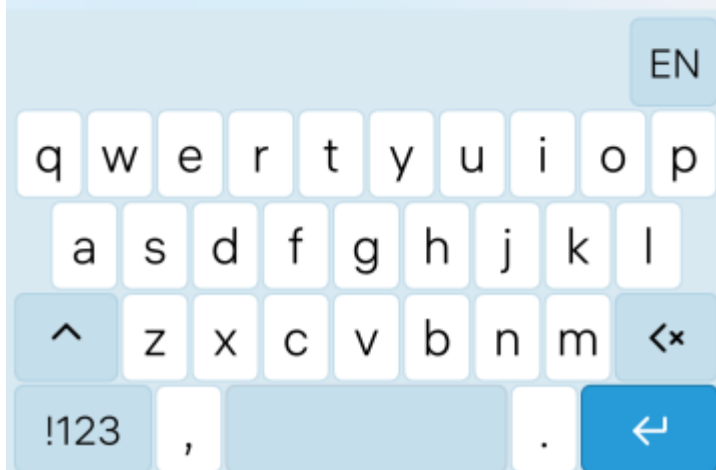
Бразильский реал

11,9859

Венгерских форинтов

15,8435

5 BGN = 168.42 руб.



Доп.задание:

Приложение

Task_1

```
import QtQuick 2.0
import Sailfish.Silica 1.0

Page {
    id: page

    ListModel {
        id: listModel
        ListElement {
            text: "Белый";
            bg_color: "white";
            text_color: "black" }
        ListElement {
            text: "Чёрный";
            bg_color: "black";
            text_color: "white"}
        ListElement {
            text: "Синий";
            bg_color: "blue";
            text_color: "white"}
    }

    SilicaListView {

        spacing: 10
        anchors.fill: parent
        model: listModel
    }
}
```

```

delegate: Rectangle {

    width: parent.width
    height: 150
    color: model.bg_color

    Text {

        anchors.centerIn: parent
        text: model.text
        color: model.text_color
    }
}
}
}

```

Task_2

```
import QtQuick 2.0
```

```
import Sailfish.Silica 1.0
```

```
Page {
```

```
    id: page
```

```
    allowedOrientations: Orientation.All
```

```
ListModel {
```

```
    id: list
```

```
}
```

```
Column {
```

```
    id: col
```

```
    property int count: 0
```

```
    width: page.width
```

```
    spacing: 10
```

```
Button {  
  
    id: myButton  
  
    anchors.horizontalCenter: parent.horizontalCenter  
  
    text: "Добавить"  
  
    onClicked: list.append({myText: "Элемент " + (++col.count)})  
}
```

```
SilicaListView {  
  
    height: parent.height  
  
    width: parent.width  
  
    model: list  
  
    spacing: 10  
  
    delegate: Rectangle {  
  
        width: parent.width  
  
        height: 100  
  
        color: "white"  
  
        Text {  
            anchors.centerIn: parent  
  
            text: model.myText  
        }  
  
        MouseArea{
```



```

        anchors.fill: parent

        onClicked: {list.remove(model.index,1)}
    }
}
}
}
}
}

```

Task_3

```
import QtQuick 2.0
```

```
import Sailfish.Silica 1.0
```

```
import "func.js" as Func
```

```
Page {
```

```
    objectName: "Task_3"
```

```
    allowedOrientations: Orientation.All
```

```
    Item {
```

```
        id: container
```

```
        anchors {
```

```
            left: parent.left; right: parent.right;
```

```
            verticalCenter: parent.verticalCenter;
```

```
        }
```

```
        height: parent.height * 0.8
```

```
        property var rectanglesModel: [
```

```
{ idx: 1, name: "Белый", bgcolor: "#ffffff" },  
{ idx: 2, name: "Черный", bgcolor: "#000000" },  
{ idx: 3, name: "Синий", bgcolor: "#0000ff" },  
  
]
```

```
SilicaListView {  
    anchors.fill: parent  
    model: container.rectanglesModel  
    delegate: Rectangle {  
        color: modelData.bgcolor  
        width: parent.width  
        height: 200  
        Text {  
            text: modelData.name  
            anchors.centerIn: parent  
            color: Func.invertColor(modelData.bgcolor, 0)  
        }  
    }  
    spacing: 5  
}  
}
```

Task_4

```
import QtQuick 2.0  
import QtQuick.XmlListModel 2.0  
import Sailfish.Silica 1.0
```

```

Page {
    id: page
    allowedOrientations: Orientation.All

    XmlListModel {

        id: model
        source: "http://www.cbr.ru/scripts/XML_daily.asp"
        query: "/ValCurs/Valute"
        XmlRole {
            name: "Name";
            query: "Name/string()"
        }
        XmlRole {
            name: "Value";
            query: "Value/string()"
        }
    }

    SilicaListView {

        anchors.fill: parent
        header: PageHeader {
            description: "Валюты"
        }
        model: model
        spacing: 10

        delegate: Column {
            spacing: 5
            Label {
                text: Name
            }
            Label {
                text: Value
            }
        }
    }
}

```

Task_5

```

import QtQuick 2.0
import QtQuick.XmlListModel 2.0
import Sailfish.Silica 1.0

```

```
Page {
```

```
    id: page
```

```
    allowedOrientations: Orientation.All
```

```
    XmlListModel {
```

```
        id: model
```

```
        query: "/ValCurs/Valute"
```

```
        XmlRole {
```

```
            name: "Name";
```

```
            query: "Name/string()"
```

```
        }
```

```
        XmlRole {
```

```
            name: "Value";
```

```
            query: "Value/string()"
```

```
        }
```

```
    }
```

```
    SilicaListView {
```

```
        anchors.fill: parent
```

```
        header: PageHeader {
```

```
            description: "Валюты"
```

```
        }
```

```
        model: model
```

```
        delegate: Column {
```

```
            spacing: 5
```

```
            Label {
```

```
                text: Name
```

```
            }
```

```
            Label {
```

```
                text: Value
```

```
            }
```

```
        }
```

```
    }
```

```
    Component.onCompleted: loadRate()
```

```
    function loadRate(){
```

```
        var x = new XMLHttpRequest();
```

```
        x.open('GET', 'https://www.cbr-xml-daily.ru/daily_utf8.xml', true);
```

```
        x.onreadystatechange = function() {
```

```
            if (x.readyState === XMLHttpRequest.DONE){
```

```
                model.xml = x.responseText;
```

```
            }
```

```
        }
```

```
        x.send();
```

```
    }
```

```
}
```

```

Task_6
import QtQuick 2.0
import Sailfish.Silica 1.0
import QtQuick.LocalStorage 2.0
Page {
    objectName: "Task_6"
    allowedOrientations: Orientation.All

    property var db: LocalStorage.openDatabaseSync("QDeclarativeExampleDB", "1.0", "The
Example QML SQL!", 1000000)

    Column {
        y: 100
        width: parent.width

        TextField {
            id: txtfield
            placeholderText: "Текст заметка"
        }

        Button {
            text: "Добавить"
            onClicked: {
                db.transaction(function(tx) {
                    tx.executeSql("INSERT INTO notes (note_text) VALUES(?);", [txtfield.text]);

                    // Show all added greetings
                    var rs = tx.executeSql('SELECT * FROM notes');

                    var r = []
                    for (var i = 0; i < rs.rows.length; i++) {
                        r.push(rs.rows.item(i))
                    }
                    console.log(r)
                    container.notesModel = r
                });
            }
        }
    }

    Item {
        id: container
        anchors {
            left: parent.left; right: parent.right;
            verticalCenter: parent.verticalCenter;
        }
    }
}

```

```
height: parent.height * 0.5
```

```
property var notesModel: []
```

```
SilicaListView {  
    anchors.fill: parent  
    model: container.notesModel  
    delegate: Label {  
        width: parent.width  
        height: 100  
        Text {  
            text: modelData.note_text  
            anchors.centerIn: parent  
        }  
    }  
    spacing: 5  
}
```

```
function findGreetings() {  
    db.transaction(  
        function(tx) {  
            // Create the database if it doesn't already exist  
            tx.executeSql('CREATE TABLE IF NOT EXISTS notes(note_text TEXT)');  
  
            // Add (another) greeting row  
            tx.executeSql('INSERT INTO notes VALUES(?)', [ 'hello' ]);  
  
            // Show all added greetings  
            var rs = tx.executeSql('SELECT * FROM notes');  
  
            var r = []  
            for (var i = 0; i < rs.rows.length; i++) {  
                r.push(rs.rows.item(i))  
            }  
            console.log(r)  
            container.notesModel = r  
        }  
    )  
}  
Component.onCompleted: findGreetings()  
}  
  
}
```

Task_7

```
import QtQuick 2.0
import Sailfish.Silica 1.0
import Nemo.Configuration 1.0
```

```
Page {
    objectName: "Task_7"
    allowedOrientations: Orientation.All
```

```
    ConfigurationValue {
        id: setting_1
        key: "/apps/app_name/setting_1"
        defaultValue: "Menu Default"
    }
```

```
    ConfigurationValue {
        id: setting_2
        key: "/apps/app_name/setting_2"
        defaultValue: false
    }
```

```
    Column {
        y: 200
        TextField {
            width: 300
            text: "Текстовое поле"
            onTextChanged: {
                setting_1.value = text
                console.log(setting_1.value)
            }
        }
    }
```

```
    TextSwitch {
        text: checked ? qsTr("Active") : qsTr("Inactive")
        description: qsTr("Переключатель")
        onCheckedChanged: {
            setting_2.value = checked
            console.log(setting_2.value)
        }
    }
}
```

```
}
```

Task_8

```
import QtQuick 2.0
import Sailfish.Silica 1.0
import Nemo.Configuration 1.0

Page {
    objectName: "Task_8"
    allowedOrientations: Orientation.All

    ConfigurationGroup {
        id: settings
        path: "/apps/app_name/settings"
        property var tf: "empty"
        property bool sw: false
    }

    Column {
        y: 200
        TextField {
            width: 300
            text: "Текстовое поле(2)"
            onTextChanged: {
                settings.tf = text
                console.log(settings.tf)
            }
        }

        TextSwitch {
            text: checked ? qsTr("Active") : qsTr("Inactive")
            description: qsTr("Переключатель(2)")
            onCheckedChanged: {
                settings.sw = checked
                console.log(settings.sw)
            }
        }
    }
}
```

Task_9

```
import QtQuick 2.0
import QtQuick.XmlListModel 2.0
import Sailfish.Silica 1.0
```



```
Page {
  id: page
  allowedOrientations: Orientation.All
```

```
XmlListModel {

  id: model
  source: "http://www.cbr.ru/scripts/XML_daily.asp"
  query: "/ValCurs/Valute"
  XmlRole {
    name: "Name";
    query: "Name/string()"
  }
  XmlRole {
    name: "Value";
    query: "Value/string()"
  }
  XmlRole {
    name: "CharCode";
    query: "CharCode/string()"
  }
}
```

```
SilicaListView {
  id: list
  anchors.top: parent.top
  anchors.left: parent.left
  anchors.right: parent.right
  anchors.bottom: parent.bottom
  anchors.bottomMargin: 250
  header: PageHeader {
    description: "Валюты"
  }
  model: model
  spacing: 10
```

```
delegate: Rectangle {
  width: parent.width
  height: 100
  color: "transparent"
  Column {
    spacing: 5
    Label {
      text: Name
    }
    Label {
      text: Value
    }
  }
}
```

```

    }
    MouseArea {
        anchors.fill: parent
        onClicked: {
            info.charCode = CharCode
            info.value = Value
            list.currentIndex = index
        }
    }
}
highlight: Rectangle {
    color: Theme.highlightColor
}
}

TextField {
    id: input
    anchors.bottom: parent.bottom
    width: 150
    text: "1"
}

Label {
    id: info
    anchors.left: input.right
    anchors.top: input.top
    anchors.topMargin: 15
    property string charCode
    property string value
    property var convertVal:
parseFloat(parseInt(input.text)*parseFloat(value.replace(",","."))).toFixed(2)
    text: charCode + " = " + convertVal + " pyб."
}

}

```