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**

**Отчет по практическому заданию для лекции №5**

**Выполнила:**

студентка группы 382006-2

Кулёва Анна Андреевна

**Проверил:**

Карчков Денис Александрович

Рецензент:

Нижний Новгород  
2023

# Содержание

[Содержание 2](#_Toc148143156)

[1. Цель практического занятия 3](#_Toc148143157)

[2. Постановка задачи 4](#_Toc148143158)

[3. Руководство пользователя 7](#_Toc148143159)

[4. Руководство программиста 13](#_Toc148143160)

[Заключение 15](#_Toc148143161)

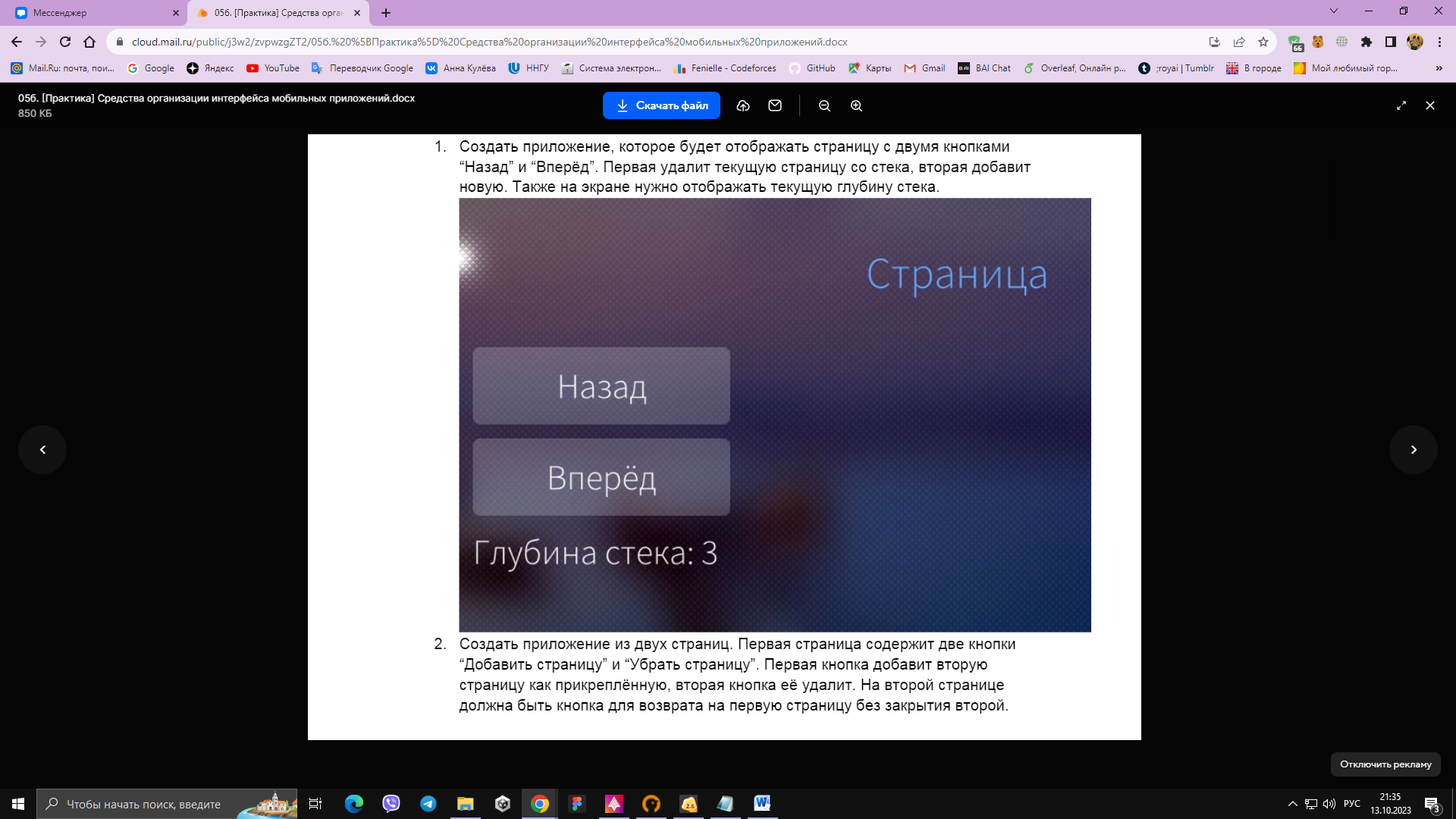
[Приложение 16](#_Toc148143162)

# Цель практического занятия

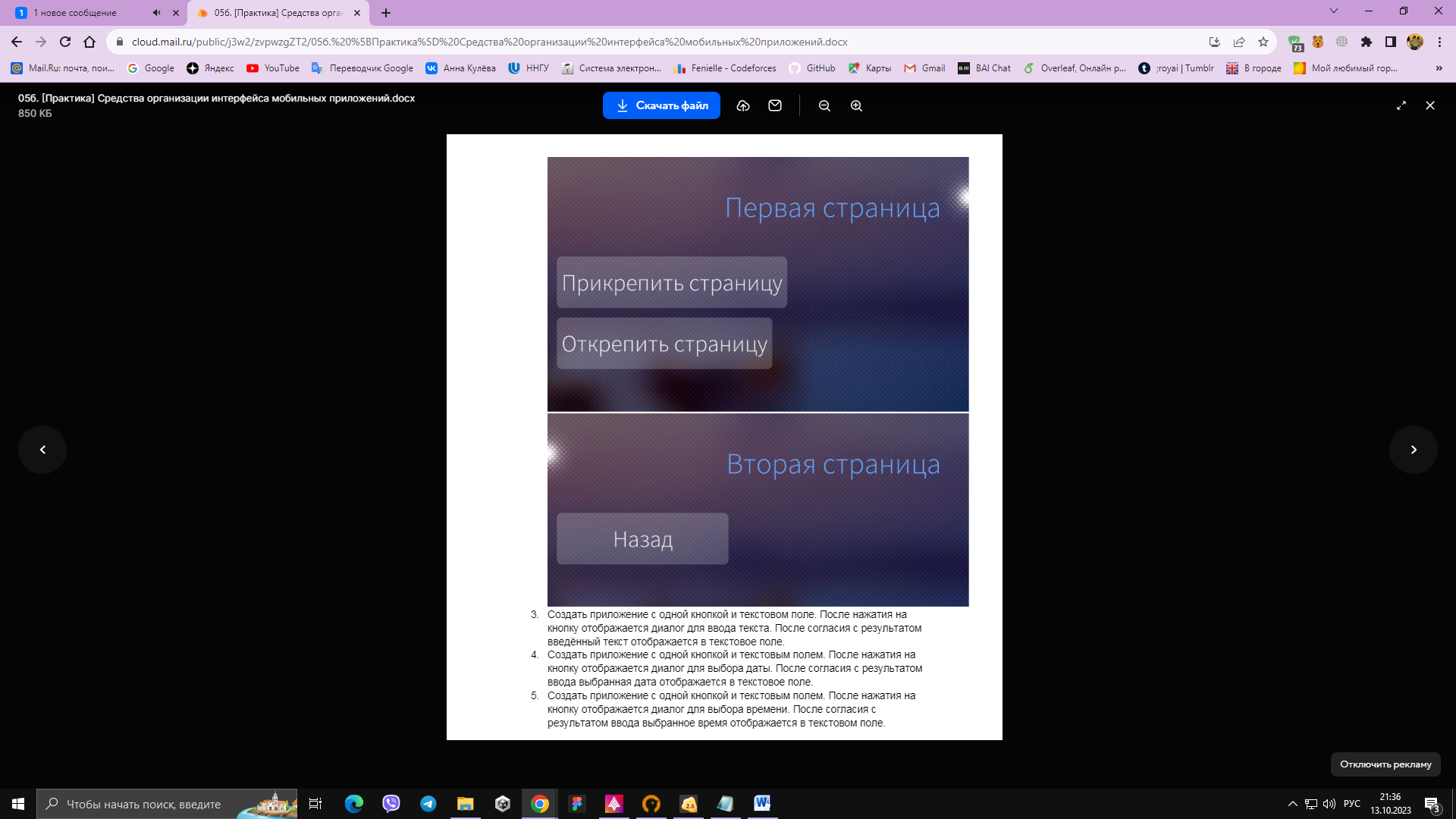
Цель данного практического занятия состоит в том, чтобы научиться организовывать многостраничное приложение, использовать контейнеры Silica, вытягиваемые меню и обложку приложения.

# Постановка задачи

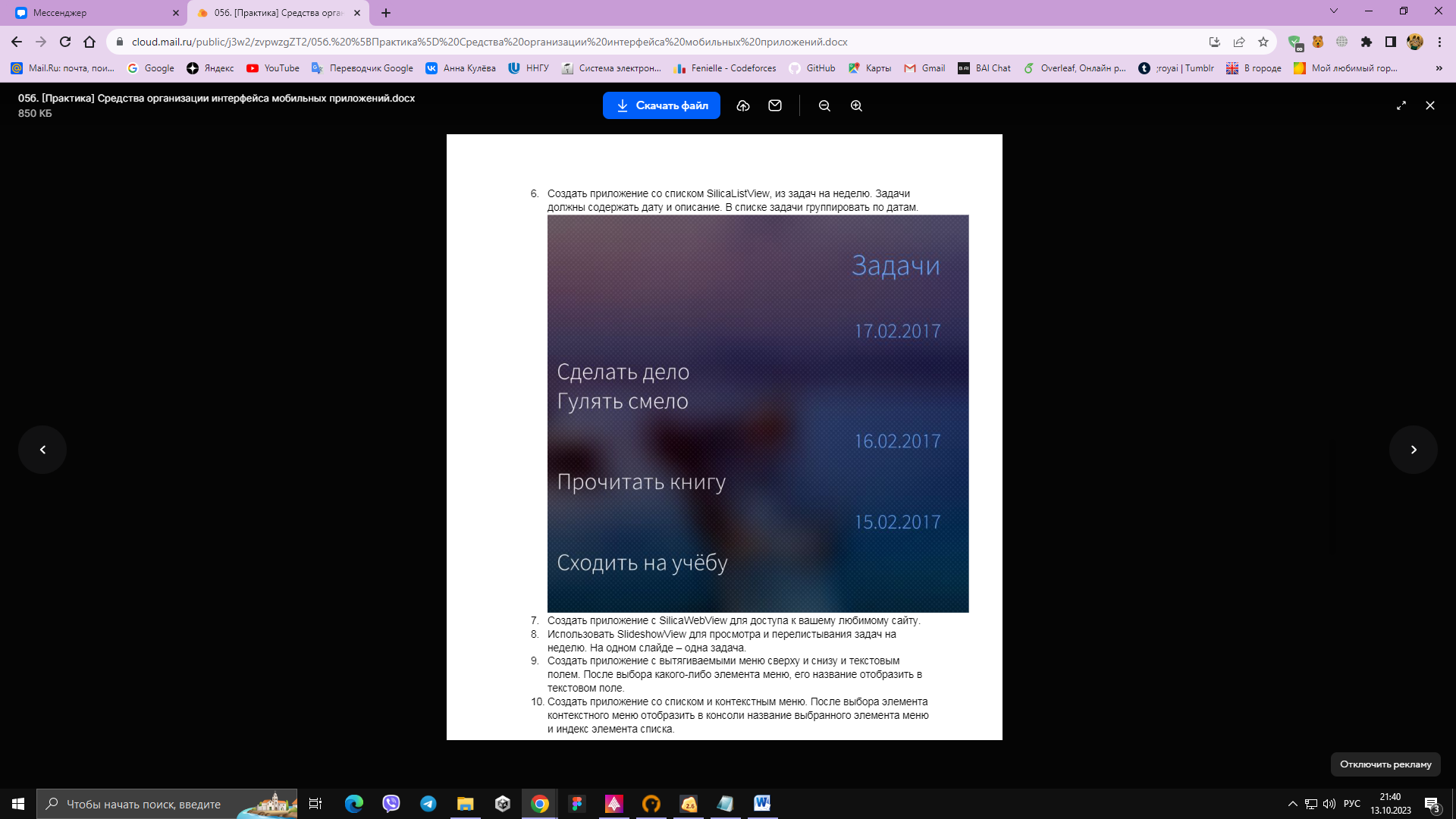
1. Создать приложение, которое будет отображать страницу с двумя кнопками “Назад” и “Вперёд”. Первая удалит текущую страницу со стека, вторая добавит новую. Также на экране нужно отображать текущую глубину стека.



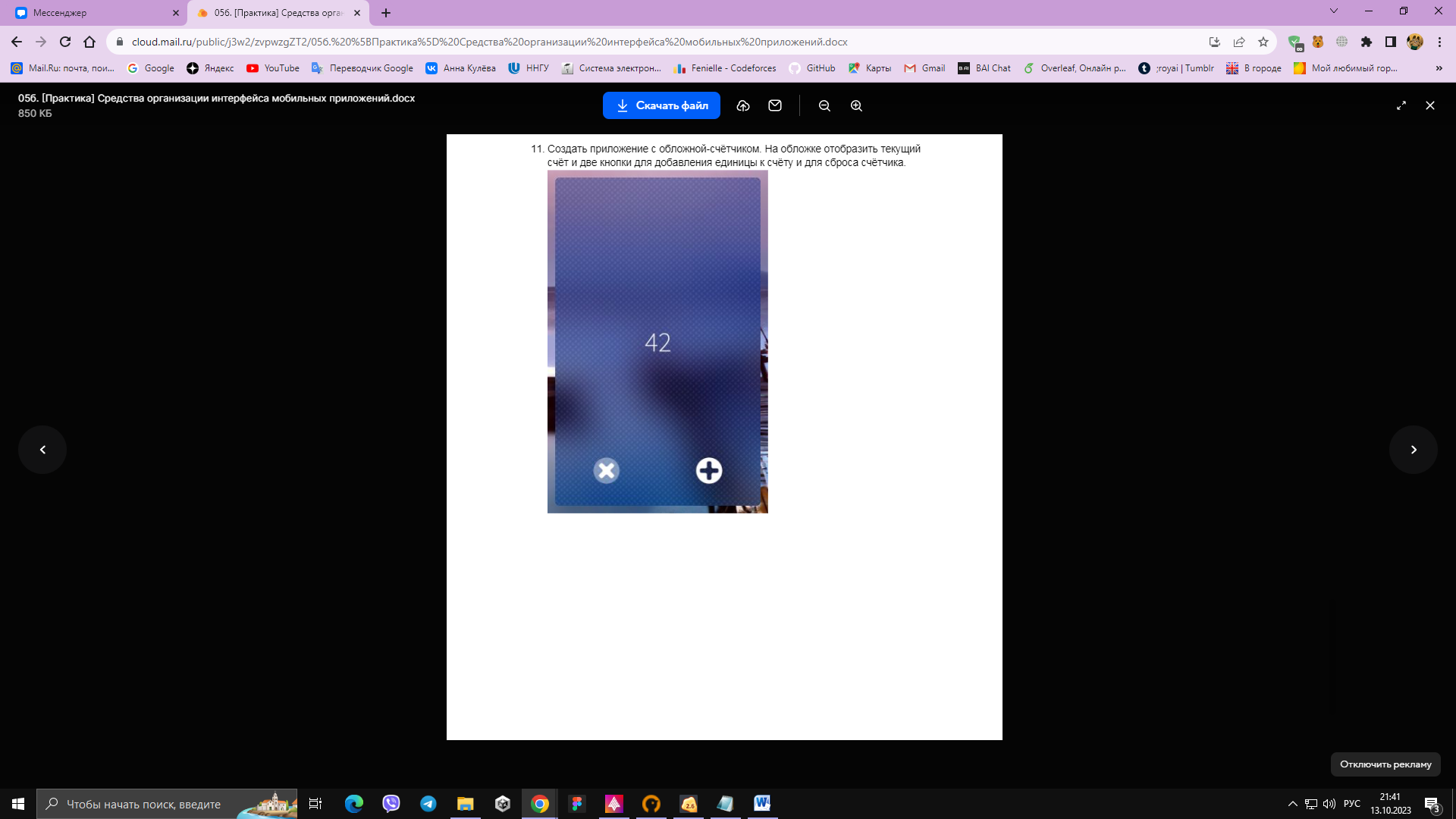
1. Создать приложение из двух страниц. Первая страница содержит две кнопки “Добавить страницу” и “Убрать страницу”. Первая кнопка добавит вторую страницу как прикреплённую, вторая кнопка её удалит. На второй странице должна быть кнопка для возврата на первую страницу без закрытия второй.



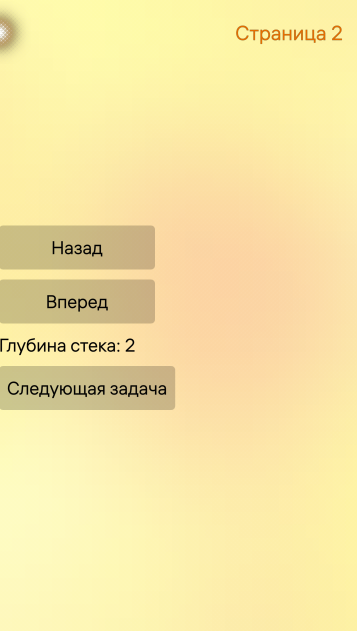
1. Создать приложение с одной кнопкой и текстовым поле. После нажатия на кнопку отображается диалог для ввода текста. После согласия с результатом введённый текст отображается в текстовое поле.
2. Создать приложение с одной кнопкой и текстовым полем. После нажатия на кнопку отображается диалог для выбора даты. После согласия с результатом ввода выбранная дата отображается в текстовое поле.
3. Создать приложение с одной кнопкой и текстовым полем. После нажатия на кнопку отображается диалог для выбора времени. После согласия с результатом ввода выбранное время отображается в текстовом поле.
4. Создать приложение со списком SilicaListView, из задач на неделю. Задачи должны содержать дату и описание. В списке задачи группировать по датам.



1. Создать приложение с SilicaWebView для доступа к вашему любимому сайту.
2. Использовать SlideshowView для просмотра и перелистывания задач на неделю. На одном слайде – одна задача.
3. Создать приложение с вытягиваемыми меню сверху и снизу и текстовым полем. После выбора какого-либо элемента меню, его название отобразить в текстовом поле.
4. Создать приложение со списком и контекстным меню. После выбора элемента контекстного меню отобразить в консоли название выбранного элемента меню и индекс элемента списка.



# Руководство пользователя

При запуске программы пользователь увидит главную страницу с первым заданием: две кнопки «Вперёд» и «Назад» и отображение длины стека (рисунок 1-2). Переключаться между заданиями можно с помощью кнопок «Следующая задача» и «Назад», нажав на которые пользователь перейдёт на следующую страницу или вернётся к предыдущей.

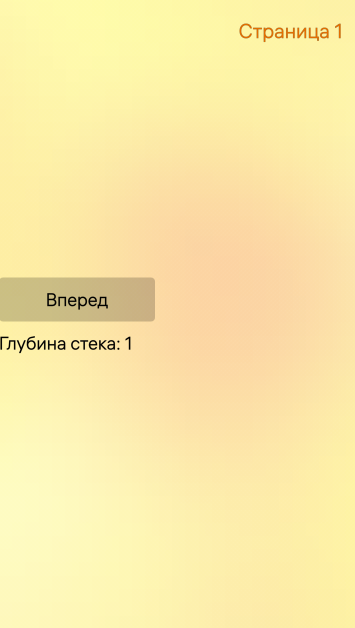
Когда вы нажимаете на кнопку "прикрепить" во втором задании, появляется индикатор, указывающий, что вы переходите на следующую страницу (рисунок 3-4).

Рисунок 1-2. Глубина стека на первой и на второй страницах

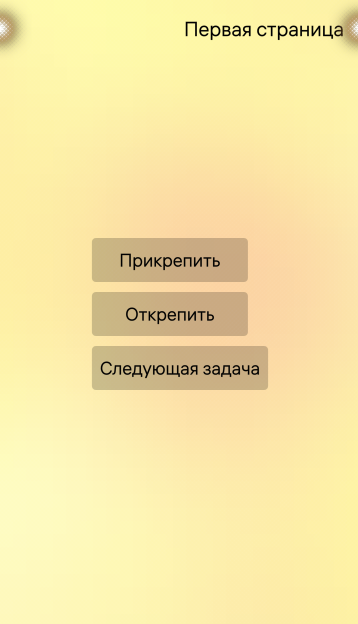
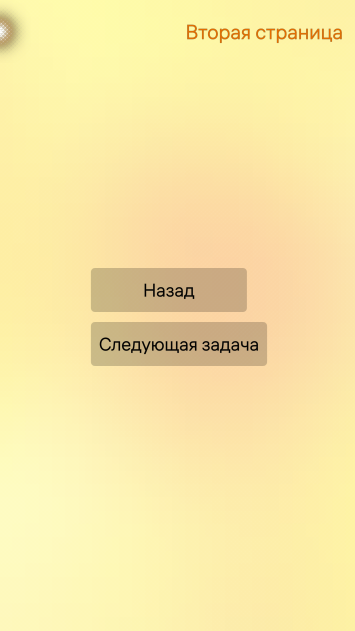
В третьем задании осуществляется ввод в текстовое поле (рисунок 5).

Рисунок 3-4. Прикрепление страницы

В четвертом задании ввод в текстовом поле управляется диалогом выбора даты (рисунок 6).

В пятом задании ввод в текстовом поле управляется диалогом выбора времени (рисунок 7).

В шестом задании реализовано приложение со списком SilicaListView, из задач на неделю, задачи группируются по датам (рисунок 8).

В седьмом задании на странице отображается браузер (рисунок 9).

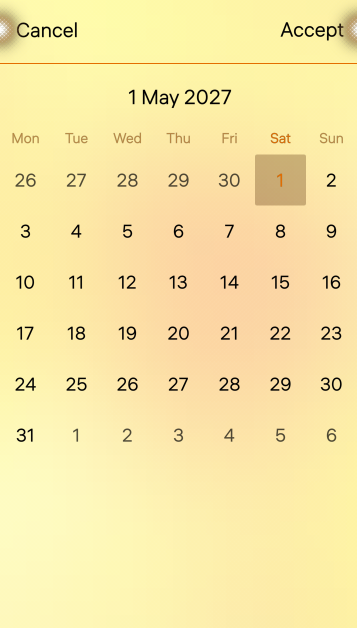
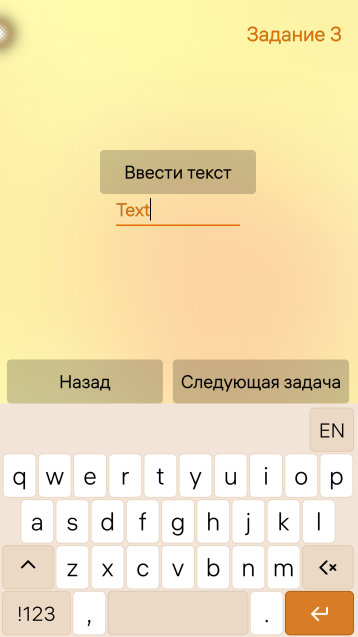


Рисунок 6. Диалог ввода даты

Рисунок 5. Ввод в текстовое поле

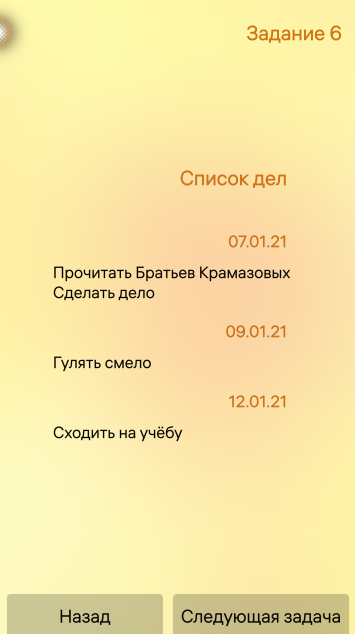




Рисунок 8. Список SilicaListView

Рисунок 7. Диалог ввода времени



Рисунок 9. Браузер

В восьмом задании список дел отображается как слайд-шоу (рисунок 10).

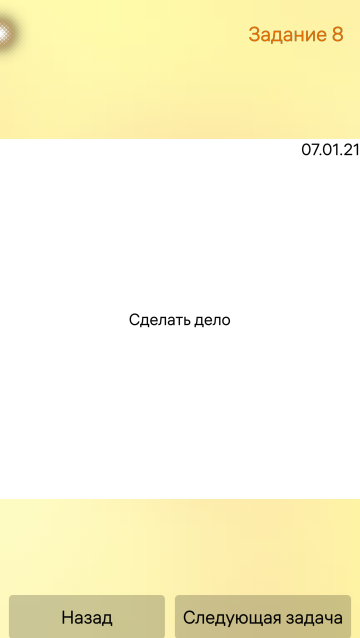


Рисунок 10. Слайд-шоу

В девятом задании есть два вытягиваемых сверху и снизу меню (рисунок 11).

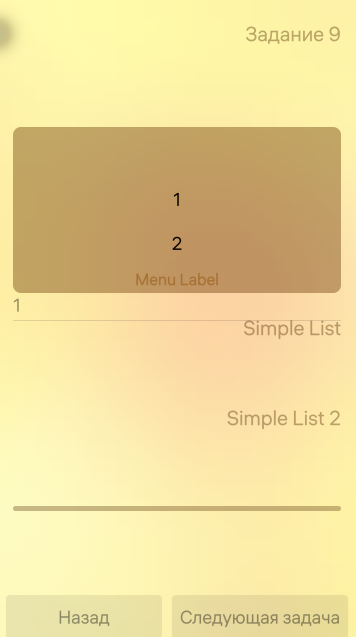


Рисунок 11. Вытягиваемые сверху и снизу меню

В десятом задании на элементах списка открывается контекстное меню (рисунок 12).

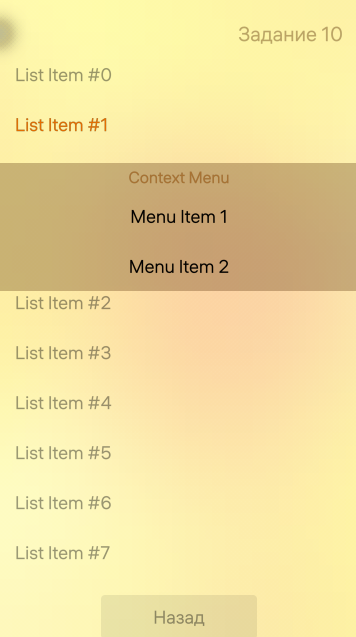


Рисунок 12. Контекстное меню

На обложке приложения отображается счётчик (рисунок 13). При нажатии на «+» счётчик увеличивается, при нажатии на «-» - уменьшается.

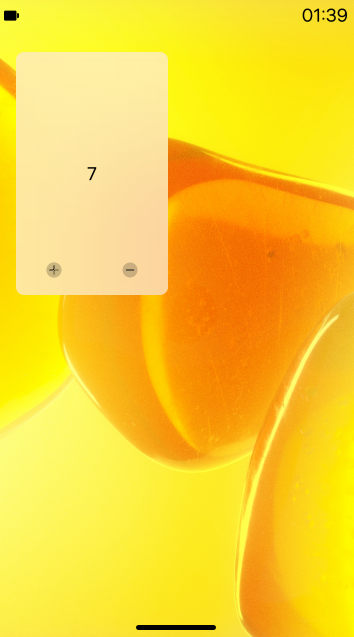


Рисунок 13. Счётчик на обложке

# Руководство программиста

Программа реализована на языке программирования QML.

1. Выполнение шага №1.

Возможность добавления страниц и отображение глубины стека реализуется с помощью использования pageStack.depth для получения значения и отображения его в Label. Чтобы добавить новую страницу, используется pageStack.push(Qt.resolvedUrl("Task1\_2.qml")), а для удаления страницы из стека - pageStack.pop().

1. Выполнение шага №2.

Возможность прикрепления страницы реализована с использованием pageStack.pushAttached(Qt.resolvedUrl("Task2\_2.qml")), а открепления страницы - pageStack.popAttached().

1. Выполнение шага №3.

Для изменения ввода в TextField в диалоге используется onAccepted: field.text=firstTextFieldValue.

1. Выполнение шага №4.

Для диалога с календарем в TextField вносятся изменения с использованием DatePickerDialog с onAccepted: field.text = date.toDateString().

1. Выполнение шага №5.

Для диалога с часами используется TimePickerDialog, где при onAccepted будет присваиваться значение времени полю field.text.

1. Выполнение шага №6.

Для списка задач используется SilicaListView с ListModel.

1. Выполнение шага №7.

Браузер реализуется с помощью SilicaWebView, где устанавливается свойство url.

1. Выполнение шага №8.

Для слайд-шоу используется SlideshowView с ListModel.

1. Выполнение шага №9.

Вытягиваемые меню реализуются с помощью SilicaListView с PullDownMenu.

1. Выполнение шага №10.

Для списка с контекстным меню используется SilicaListView с ContextMenu.

1. Выполнение шага №11.

Для обложки приложения используется CoverActionList с CoverAction.

# Заключение

В данной лабораторной работе я научилась организовывать многостраничное приложение, использовать контейнеры Silica, вытягиваемые меню и обложку приложения. Также были выполнены все шаги практического задания.

# Приложение

**Task1\_1.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Страница 1")

}

Column {

anchors.verticalCenter: *parent*.verticalCenter

anchors.left: *parent*.left

spacing: 20

Button {

text: "Вперед"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task1\_%1.qml").arg((pageStack.depth + 1) % 3)))

}

Label {

anchors.left: *parent*.left

text: *qsTr*("Глубина стека: %1").arg(pageStack.depth)

}

}

}

**Task1\_2.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Страница 2")

}

Column {

anchors.verticalCenter: *parent*.verticalCenter

anchors.left: *parent*.left

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Вперед"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task1\_2.qml").arg((pageStack.depth + 1) )))

}

Label {

anchors.left: *parent*.left

text: *qsTr*("Глубина стека: %1").arg(pageStack.depth)

}

Button {

text: "Следующая задача"

onClicked: pageStack.replace(*Qt*.resolvedUrl(*qsTr*("Task2\_1.qml")))

}

}

}

}

**Task2\_1.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Первая страница")

}

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.centerIn: *parent*

spacing: 20

Button {

text: "Прикрепить"

onClicked: pageStack.pushAttached(*Qt*.resolvedUrl(*qsTr*("Task2\_2.qml")))

}

Button {

text: "Открепить"

onClicked: pageStack.popAttached()

}

Button {

text: "Следующая задача"

onClicked: pageStack.replace(*Qt*.resolvedUrl(*qsTr*("Task3.qml")))

}

}

}

**Task2\_2.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Вторая страница")

}

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.centerIn: *parent*

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.navigateBack(1)

}

Button {

text: "Следующая задача"

onClicked: pageStack.replace(*Qt*.resolvedUrl(*qsTr*("Task3.qml")))

}

}

}

**Task3.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 3")

}

Column {

anchors.centerIn: *parent*

Button {

text: "Ввести текст"

onClicked: *dialog*.open()

}

TextField {

id: *field*

text: "123"

}

}

Dialog {

id: *dialog*

*property* *alias* firstTextFieldValue: *firstTextField*.text

Column {

width: *parent*.width

spacing: Theme.paddingMedium

DialogHeader { }

TextField {

id: *firstTextField*

width: *parent*.width

}

}

onAccepted: *field*.text = *firstTextFieldValue*

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.replace(*Qt*.resolvedUrl(*qsTr*("Task4.qml")))

}

}

}

**Task4.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 4")

}

Column {

anchors.centerIn: *parent*

Button {

text: "Ввести дату"

onClicked: *dialog*.open()

}

TextField {

id: *field*

text: "123"

}

}

DatePickerDialog {

id: *dialog*

onAccepted: *field*.text = *date*.toDateString()

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task5.qml")))

}

}

}

**Task5.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 5")

}

Column {

anchors.centerIn: *parent*

Button {

text: "Ввести время"

onClicked: *dialog*.open()

}

TextField {

id: *field*

text: "123"

}

}

TimePickerDialog {

id: *dialog*

onAccepted: *field*.text = *time*.toTimeString()

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task6.qml")))

}

}

}

**Task6.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 6")

}

ListModel {

id: *tasksModel*

ListElement { desc: "Прочитать Братьев Крамазовых"; date: "07.01.21"; }

ListElement { desc: "Сделать дело"; date: "07.01.21"; }

ListElement { desc: "Гулять смело"; date: "09.01.21"; }

ListElement { desc: "Сходить на учёбу"; date: "12.01.21"; }

}

Item {

anchors.centerIn: *parent*

width: 500

height: 700

SilicaListView {

anchors.fill: *parent*

model: *tasksModel*

header: PageHeader { title: "Список дел" }

section {

property: 'date'

delegate: SectionHeader { text: section }

}

delegate: Text { text: desc }

}

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task7.qml")))

}

}

}

**Task7.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 7")

}

Item {

anchors {

left: *parent*.left; right: *parent*.right;

verticalCenter: *parent*.verticalCenter;

}

height: *parent*.height \* 0.8

SilicaWebView {

id: *webView*

anchors.fill: *parent*

url: "http://sailfishos.org"

}

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task8.qml")))

}

}

}

**Task8.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 8")

}

ListModel {

id: *tasksModel*

ListElement { desc: "Сделать дело"; date: "07.01.21"; }

ListElement { desc: "Гулять смело"; date: "07.01.21"; }

ListElement { desc: "Почитать лекции по МО"; date: "09.01.21"; }

ListElement { desc: "Сходить на лекции по МО"; date: "12.01.21"; }

}

Item {

anchors {

left: *parent*.left; right: *parent*.right;

verticalCenter: *parent*.verticalCenter;

}

height: *parent*.height \* 0.8

SlideshowView {

id: *view*

anchors.centerIn: *parent*

height: *width*

itemHeight: *width*; itemWidth: *width*;

model: *tasksModel*

delegate: Rectangle {

width: *view*.itemWidth;

height: *view*.itemHeight;

Text {

anchors {

top: *parent*.top;

right: *parent*.right;

}

text: date

}

Text {

anchors.centerIn: *parent*

text: desc

}

}

}

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task9.qml")))

}

}

}

**Task9.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 9")

}

SilicaListView {

PullDownMenu {

MenuItem {

text: "1"

onClicked: *field*.text = *text*

}

MenuItem {

text: "2"

onClicked: *field*.text = *text*

}

MenuLabel { text: *qsTr*("Menu Label") }

}

anchors {

left: *parent*.left; right: *parent*.right;

}

height: *parent*.height \* 0.2

y: *parent*.height \* 0.2

header: PageHeader { title: "Simple List" }

}

TextField {

id: *field*

text: "123"

anchors.centerIn: *parent*

}

SilicaListView {

PushUpMenu {

MenuItem {

text: "3"

onClicked: *field*.text = *text*

}

MenuItem {

text: "4"

onClicked: *field*.text = *text*

}

MenuLabel { text: *qsTr*("Menu Label") }

}

anchors {

left: *parent*.left; right: *parent*.right;

}

height: *parent*.height \* 0.2

y: *parent*.height \* 0.6

header: PageHeader { title: "Simple List 2" }

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

Button {

text: "Следующая задача"

onClicked: pageStack.push(*Qt*.resolvedUrl(*qsTr*("Task10.qml")))

}

}

}

**Task10.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Задание 10")

}

SilicaListView {

anchors {

left: *parent*.left; right: *parent*.right;

verticalCenter: *parent*.verticalCenter

}

height: *parent*.height \* 0.8

model: 20

delegate: ListItem {

id: *delegate*

Label {

x: Theme.paddingLarge

text: "List Item #" + index

*//* *color:* *delegate.highlighted* *?* *Theme.highlightColor* *:* *Theme.primaryColo*

}

menu: ContextMenu {

MenuLabel { text: "Context Menu" }

MenuItem {

text: "Menu Item 1"

onClicked: *console*.log("Menu Item 1 is clicked", index)

}

MenuItem {

text: "Menu Item 2"

onClicked: *console*.log("Menu Item 2 is clicked", index)

}

}

}

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.bottom: *parent*.bottom

spacing: 20

Button {

text: "Назад"

onClicked: pageStack.pop()

}

}

}

**DefaultCoverPage.qml**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

CoverBackground {

objectName: "defaultCover"

*property* *int* counter: 0

Label {

id: *label123*

anchors.centerIn: *parent*

text: *counter*

}

CoverActionList {

CoverAction {

iconSource: "image://theme/icon-splus-add"

onTriggered: *counter*++

}

CoverAction {

iconSource: "image://theme/icon-splus-remove"

onTriggered: *counter*--

}

}

}