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 №4

на тему

**«Средства организации интерфейса мобильных приложений»**

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

**Преподаватель:**

Карчков Д.А.

**Нижний Новгород**

**2021**

Содержание

[Постановка задачи 2](#_Toc4876657)

[Результаты выполнения задач 3](#_Toc4876658)

[Приложение 10](#_Toc4876659)

[FirstPage.qml 10](#_Toc4876660)

[SecondPage.qml 13](#_Toc4876661)

[ThirdPage.qml 14](#_Toc4876662)

[FourthPage.qml 15](#_Toc4876663)

[FifthPage.qml 16](#_Toc4876664)

[SixthPage.qml 17](#_Toc4876665)

[CoverPage.qml 18](#_Toc4876666)

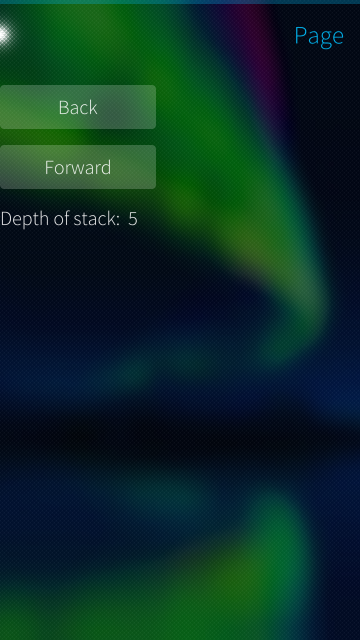
# Постановка задачи

В рамках лабораторной работы научиться организовывать многостраничное приложение, использовать контейнеры Silica, вытягиваемые меню и обложку приложения.

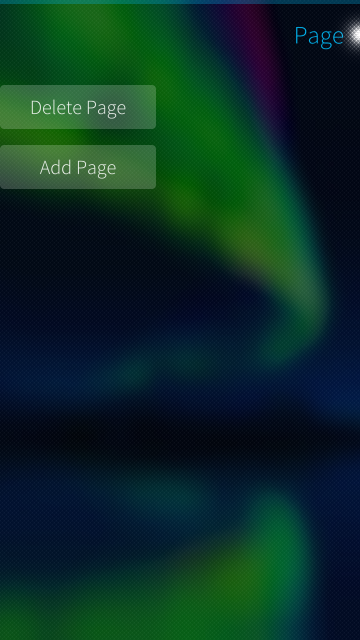
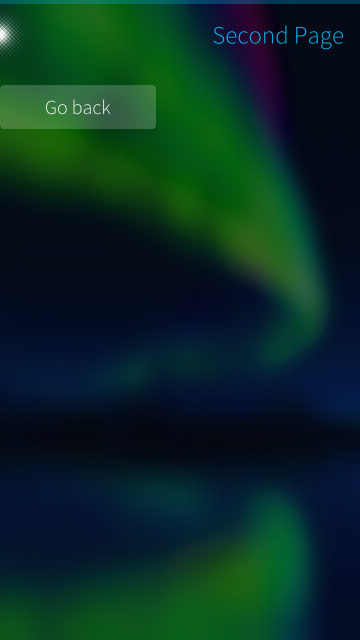
Работа состоит из 11 подзадач:

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.
2. 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.
3. Создать приложение с одной кнопкой и текстовом полем. После нажатия на кнопку отображается диалог для ввода текста. После согласия с результатом введённый текст отображается в текстовое поле.
4. 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.
5. 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.
6. Создать приложение со списком SilicaListView, из задач на неделю. Задачи должны содержать дату и описание. В списке задачи группировать по датам.
7. Создать приложение с SilicaWebView для доступа к вашему любимому сайту.
8. Использовать SlideshowView для просмотра и перелистывания задач на неделю. На одном слайде – одна задача.
9. Создать приложение с вытягиваемыми меню сверху и снизу и текстовым полем. После выбора какого-либо элемента меню, его название отобразить в текстовом поле.
10. Создать приложение со списком и контекстным меню. После выбора элемента контекстного меню отобразить в консоли название выбранного элемента меню и индекс элемента списка.
11. Создать приложение с обложной-счётчиком. На обложке отобразить текущий счёт и две кнопки для добавления единицы к счёту и для сброса счётчика.

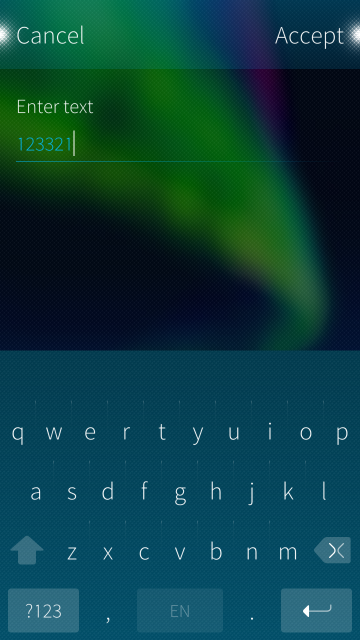
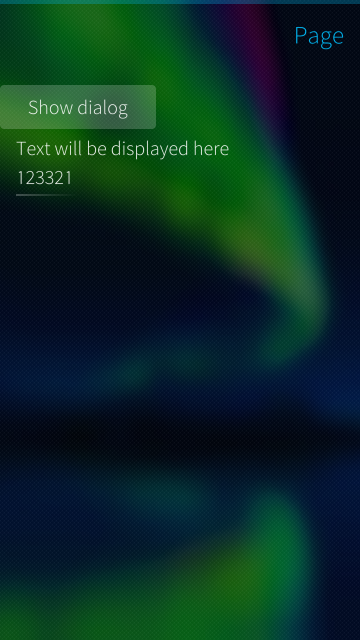
# Результаты выполнения задач



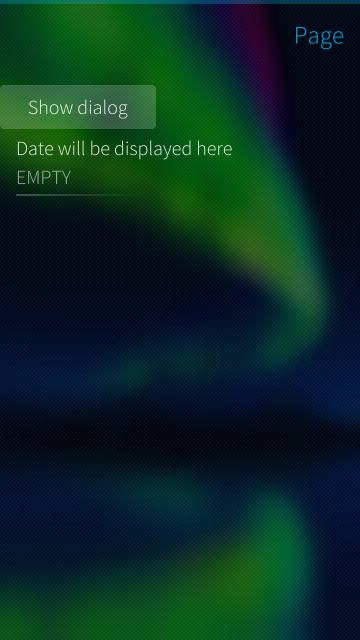
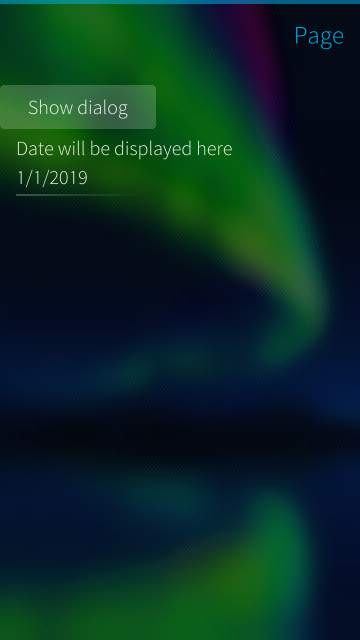
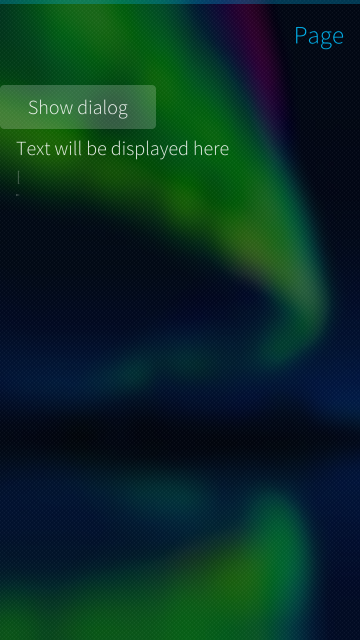
*Подзадача 1.*



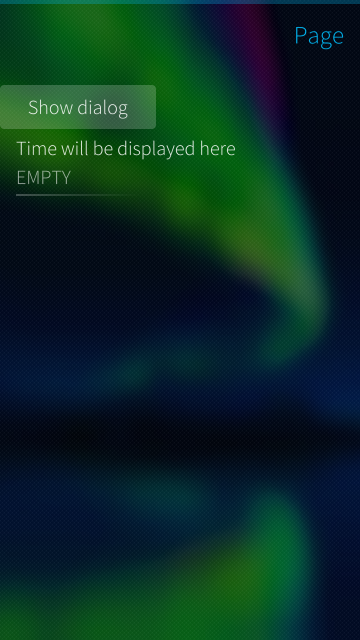
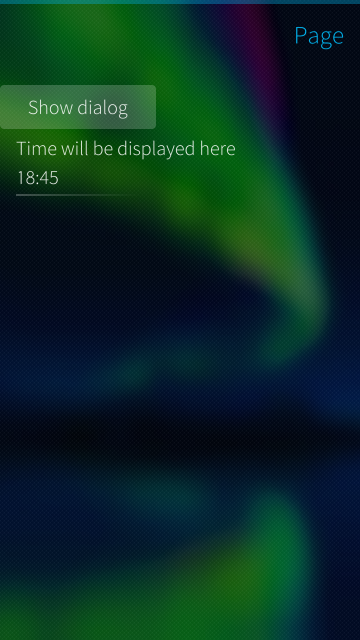
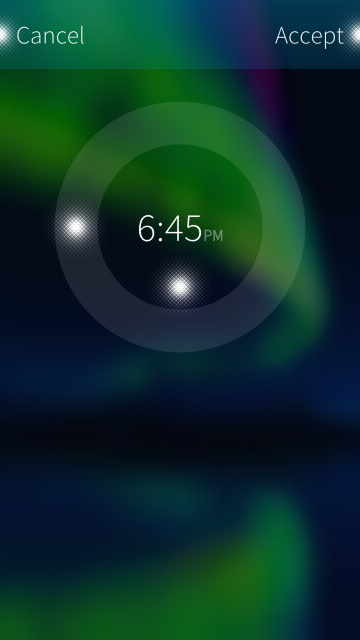
*Подзадача 2.*



*Подзадача 3.*

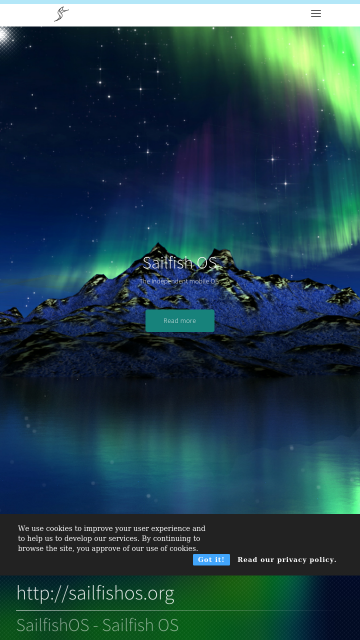
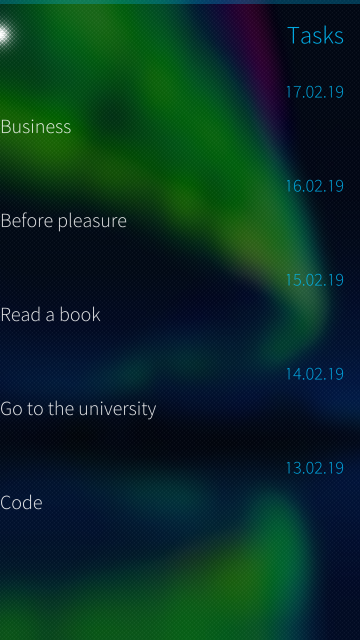


*Подзадача 4.*



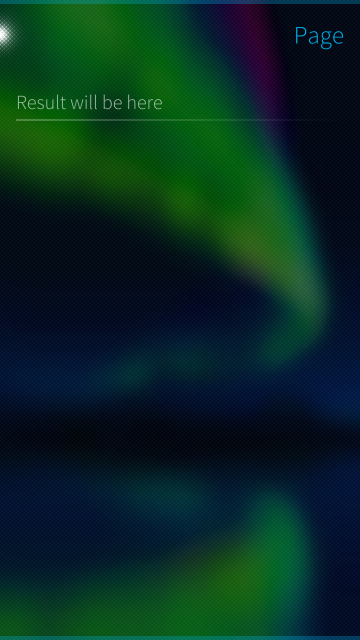
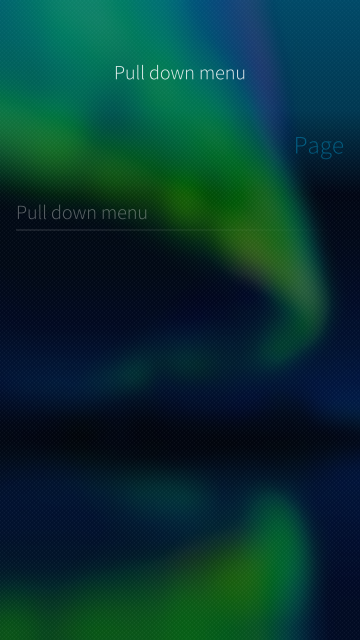
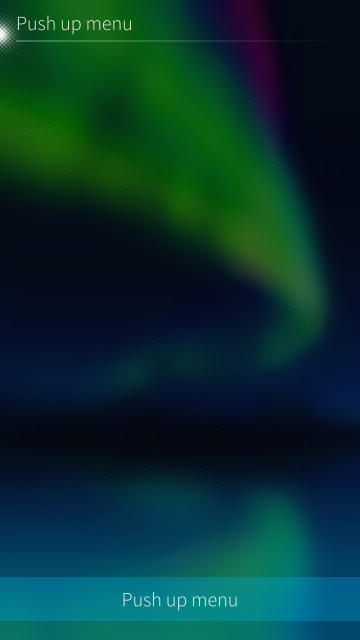
*Подзадача 5.*

*Подзадача 6.*



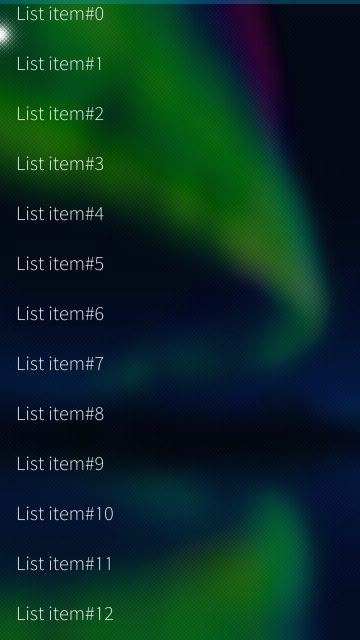
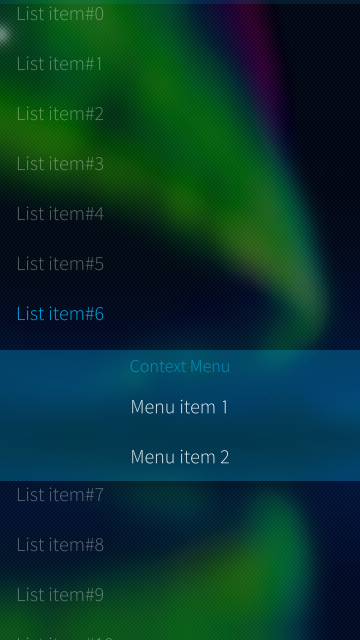
*Подзадача 7.*

*Подзадача 8.*



*Подзадача 9.*

*Подзадача 10.*



*Подзадача 11.*

# 

# 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 3 - tasks 6, 10")

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

}

MenuItem {

text: *qsTr*("Show Page 4 - task 7")

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

}

MenuItem {

text: *qsTr*("Show Page 5 - task 8")

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

}

MenuItem {

text: *qsTr*("Show Page 6 - task 9")

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

}

}

contentHeight: *column*.height

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Page")

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Back"

onClicked: {

*pageStack*.pop()

}

}

Button {

text: "Forward"

onClicked: {

*pageStack*.push(*Qt*.resolvedUrl("FirstPage.qml"))

}

}

Label {

text: "Depth of stack: " + *pageStack*.depth

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Delete Page"

onClicked: {

*pageStack*.popAttached()

}

}

Button {

text: "Add Page"

onClicked: {

*pageStack*.pushAttached(*Qt*.resolvedUrl("SecondPage.qml"))

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#3\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Show dialog"

onClicked: {

*pageStack*.push(*dialogComponent*)

}

Label {

y: 100

x: Theme.paddingLarge

text: "Text will be displayed here"

}

TextField {

y: 150

id: *textField1*

placeholderText: "EMPTY"

}

}

Component {

id: *dialogComponent*

Dialog {

Column {

width: *parent*.width

anchors.fill: *parent*

spacing: Theme.paddingMedium

DialogHeader {

id: *header*

}

Label {

x: Theme.paddingLarge

text: "Enter text"

}

TextField {

id: *textField2*

placeholderText: "Enter something here"

width: *parent*.width

}

}

onAccepted: {

*textField1*.text = *textField2*.text

}

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Show dialog"

onClicked: {

*pageStack*.push(*dialogComponent*)

}

Label {

y: 100

x: Theme.paddingLarge

text: "Date will be displayed here"

}

TextField {

y: 150

id: *textField1*

placeholderText: "EMPTY"

width: *parent*.width

}

}

Component {

id: *dialogComponent*

DatePickerDialog {

id: *tDialog*

onAccepted: {

*textField1*.text = *tDialog*.day + "/" + *tDialog*.month + "/" + *tDialog*.year

}

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#5\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Show dialog"

onClicked: {

*pageStack*.push(*dialogComponent*)

}

Label {

y: 100

x: Theme.paddingLarge

text: "Time will be displayed here"

}

TextField {

y: 150

id: *textField1*

placeholderText: "EMPTY"

width: *parent*.width

}

}

Component {

id: *dialogComponent*

TimePickerDialog {

id: *tDialog*

onAccepted: {

*textField1*.text = *tDialog*.hour + ":" + *tDialog*.minute

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

}

}

}

SecondPage.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 1")

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

}

}

contentHeight: *column*.height

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Second Page")

}

*//* *\*\*\*\*\*\*\*\*\*\*\*TASK* *#2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

Button {

text: "Go back"

onClicked: {

pageStack.\_navigateBack()

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

}

}

}

ThirdPage.qml

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaListView {

PullDownMenu {

MenuItem {

text: *qsTr*("Show Page 1")

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

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#6\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

id: *listView*

anchors.fill: *parent*

header: PageHeader {

title: *qsTr*("Tasks")

}

width: 480

height: 800

delegate: Item {

width: ListView.view.width

height: Theme.itemSizeSmall

Label {

text: task

}

}

section {

property: 'time'

delegate: SectionHeader {

text: section

}

}

model: ListModel {

ListElement {

task: "Business"

time: "17.02.19"

}

ListElement {

task: "Before pleasure"

time: "16.02.19"

}

ListElement {

task: "Read a book"

time: "15.02.19"

}

ListElement {

task: "Go to the university"

time: "14.02.19"

}

ListElement {

task: "Code"

time: "13.02.19"

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK* *#10\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

id: listView

anchors.fill: *parent*

model: 20

delegate: ListItem {

id: *delegate*

Label {

x: Theme.paddingLarge

text: "List item#" + index

color: *delegate*.highlighted ? Theme.highlightColor : Theme.primaryColor

}

menu: ContextMenu {

MenuLabel {

text: "Context Menu"

}

MenuItem {

text: "Menu item 1"

onClicked: {

*console*.log("Menu item 1 was clicked on list item #" + index)

}

}

MenuItem {

text: "Menu item 2"

onClicked: {

*console*.log("Menu item 2 was clicked on list item #" + index)

}

}

}

}

*//* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

}

}

FourthPage.qml

*import* QtQuick 2.2

*import* Sailfish.Silica 1.0

Page {

SilicaWebView {

PullDownMenu {

MenuItem {

text: *qsTr*("Show Page 1")

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

}

}

id: *webView*

anchors {

top: *parent*.top

left: *parent*.left

right: *parent*.right

bottom: *urlField*.top

}

url: "http://sailfishos.org"

}

TextField {

id: *urlField*

anchors {

left: *parent*.left

right: *parent*.right

bottom: *parent*.bottom

}

inputMethodHints: *Qt*.ImhUrlCharactersOnly

text: "http://sailfishos.org"

label: *webView*.title

EnterKey.onClicked: {

*webView*.url = *text*

*parent*.focus = true

}

}

}

FifthPage.qml

*import* QtQuick 2.2

*import* Sailfish.Silica 1.0

SlideshowView {

PullDownMenu {

MenuItem {

text: *qsTr*("Show Page 1")

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

}

}

id: *view*

anchors.centerIn: *parent*

height: *width*

itemWidth: *width*

itemHeight: *width*

model: 5

delegate: Rectangle {

width: *view*.itemWidth

height: *view*.height

border.width: 1

Text {

anchors.centerIn: *parent*

text: "Business" + " - " + (index + 1) + ".02.19"

}

}

}

SixthPage.qml

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

PullDownMenu {

MenuItem {

text: *qsTr*("Pull down menu")

onClicked: *textField*.text = "Pull down menu"

}

}

PushUpMenu {

MenuItem {

text: *qsTr*("Push up menu")

onClicked: *textField*.text = "Push up menu"

}

}

contentHeight: *column*.height

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Page")

}

TextField {

id: *textField*

placeholderText: "Result will be here"

width: *parent*.width

}

}

}

}

CoverPage.qml

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

CoverBackground {

Label {

id: *label*

anchors.centerIn: *parent*

text: *count*

*property* *int* count: 0

}

CoverActionList {

id: *coverAction*

CoverAction {

iconSource: "image://theme/icon-m-dismiss"

onTriggered: *label*.count = 0

}

CoverAction {

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

onTriggered: *label*.count--

}

CoverAction {

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

onTriggered: *label*.count++

}

}

}