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

#### Кафедра МОСТ

##### Направление подготовки: «Программная инженерия»

##### Профиль подготовки: «Разработка программно-информационных систем»

## ОТЧЕТ

### По лабораторной работе №5

##### по дисциплине «Технология программирования мобильных систем»

**Выполнил:** студент группы 381908-4

Кургузиков Кирилл

#### Руководитель:

##### доцент кафедры программной инженерии

##### Борисов Николай Анатольевич

##### Нижний Новгород 2022

**Содержание**

[1. Цель 3](#_Toc97973273)

[2. Постановка задачи 4](#_Toc97973274)

[3. Решение поставленной задачи 6](#_Toc97973275)

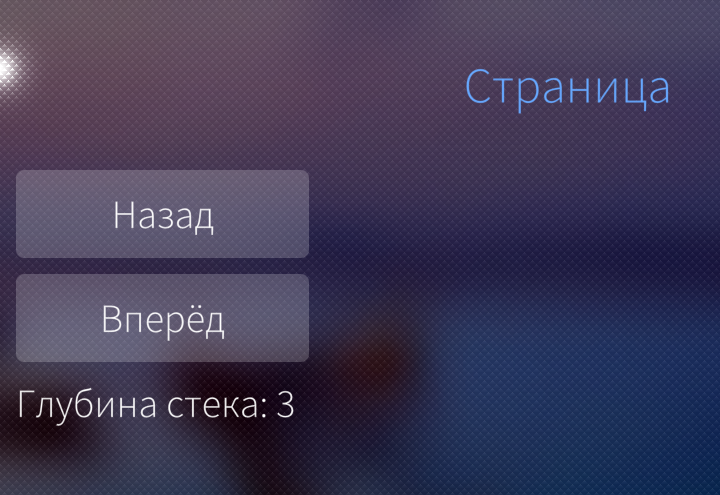
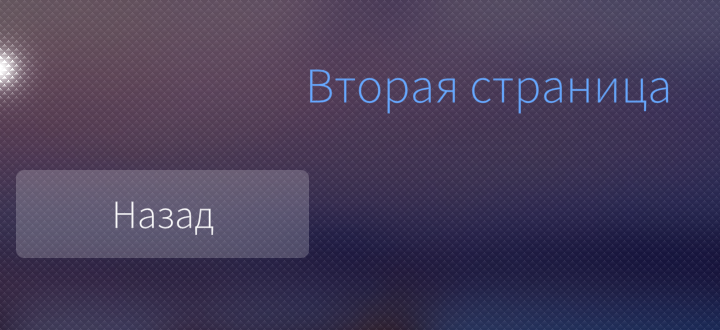
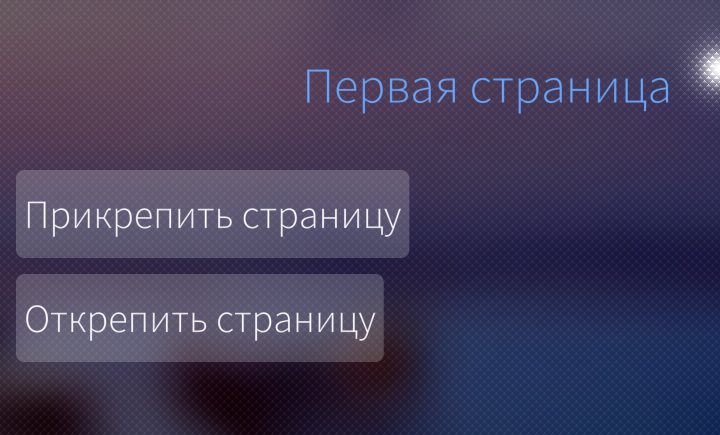
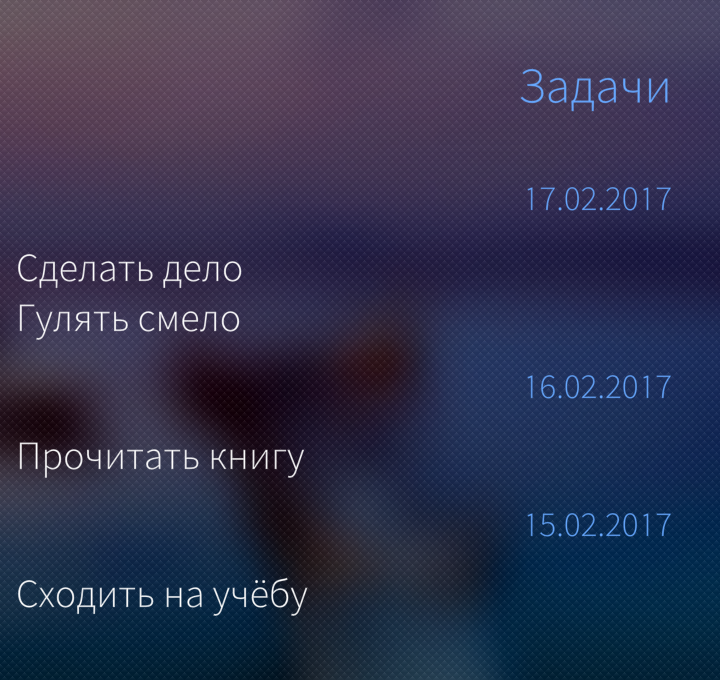
[4. Руководство пользователя 8](#_Toc97973276)

[Приложение 13](#_Toc97973277)

# Цель

: 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. 
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 поле. 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.

# Решение поставленной задачи

1. Добавление страниц и отображение глубины стека

Глубина стека отображается в label используя pageStack.depth; для получения значения.

pageStack.push(Qt.resolvedUrl("CreatePages.qml")); Создает новую страницу

pageStack.pop(); Удаляет страницу из стека

2. Прикрепленные страницы

pageStack.pushAttached(Qt.resolvedUrl("PushAttachedPage\_2.qml")); Прикрепляет страницу

pageStack.popAttached(); Открепляет страницу

3. Диалог для ввода текста

Вызов диалога и получение результата:

var dialog = pageStack.push(Qt.resolvedUrl("DialogText.qml"))

dialog.accepted.connect(function() { lb.text = "Результат: " + dialog.res

})

4. Диалог с календарем

var dialog = pageStack.push("Sailfish.Silica.DatePickerDialog",

{date: new Date()});

dialog.accepted.connect(function() {

lb.text = (dialog.date);

});

5. Диалог с часами

var dialog = pageStack.push("Sailfish.Silica.TimePickerDialog",

{hour: 10, minute: 00, hourMode:DateTime.TwentyFourHours});

dialog.accepted.connect(function() {

lb.text = (dialog.timeText);

});

6. Список задач

Используется SilicaListView с объявленными полями:

section {

property: 'data'

delegate: SectionHeader { text: section }

}

delegate: Text { text: task; color: color\_text; }

Данные для списка берутся из ListModel:

ListModel {

id: taskModel

ListElement { task: "Задание 1"; data: "20.04.2022"; color\_text: "white"; }

ListElement { task: "Задание 2"; data: "22.04.2022"; color\_text: "white"; }

ListElement { task: "Задание 3"; data: "23.04.2022"; color\_text: "white"; }

ListElement { task: "Задание 4"; data: "24.04.2022"; color\_text: "white"; }

ListElement { task: "Задание 5"; data: "24.04.2022"; color\_text: "white"; }

ListElement { task: "Задание 6"; data: "24.04.2022"; color\_text: "white"; }

}

7. Браузер

SilicaWebView {

id: webView

anchors {

top: parent.top; bottom: urlField.top;

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

}

url: "https://gbuzno51.ru/prijom-grazhdan/zapis-k-vrachu"

}

8. Просмотр задач в виде слайдов

Данные берутся из ListModel аналогичто заданию 6

SlideshowView {

id: view

anchors.centerIn: parent

height: width

itemHeight: width; itemWidth: width;

model: taskModel

delegate: Item {

width: view.itemWidth;

height: view.itemHeight;

SectionHeader { text: datas }

Text { text: task; color: color\_text ; anchors.centerIn: parent}

}

}

9. Вытягиваемые меню

PullDownMenu {

MenuLabel { text: qsTr("Верхнее меню") }

MenuItem {

text: "Пункт меню 1"

onClicked: lb.text = "Пункт верхнего меню 1"

}

MenuItem {

text: "Пункт меню 2"

onClicked: lb.text = "Пункт верхнего меню 2"

}

}

PushUpMenu {

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

MenuItem {

text: "Пункт меню 1"

onClicked: lb.text = "Пункт нижнего меню 1"

}

MenuItem {

text: "Пункт меню 2"

onClicked: lb.text = "Пункт нижнего меню 2"

}

}

10. Список с контекстным меню

SilicaListView {

anchors.fill: parent

model: 15 //Количество элементов списка

delegate: ListItem {

id: delegate

menu: ContextMenu { //Контекстное меню

MenuLabel { text: "Контекстное меню" }

MenuItem {

text: "Пункт 1"

onClicked: console.log("Пункт 1. Строка списка " + index)

}

MenuItem {

text: "Пункт 2"

onClicked: console.log("Пункт 2. Строка списка " + index)

}

}

}

}

11. Обложка приложения

CoverBackground {

Label {

property int count: 0

id: label\_1

anchors.centerIn: parent

text: str

}

CoverActionList {

id: coverAction

CoverAction {

iconSource: "image://theme/icon-cover-new"

onTriggered: label\_1.count++;

}

CoverAction {

iconSource: "image://theme/icon-cover-cancel"

onTriggered: label\_1.count = 0;

}

}

}

# Руководство пользователя

# Страница с двумя кнопками “Назад” и “Вперёд”. Первая удалит текущую страницу со стека, вторая добавит новую. Также на экране нужно отображать текущую глубину стека.

# 

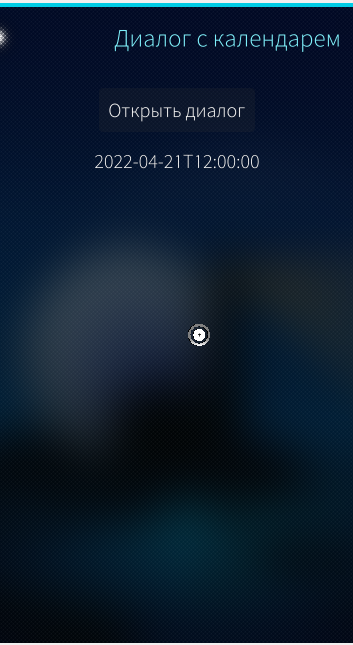
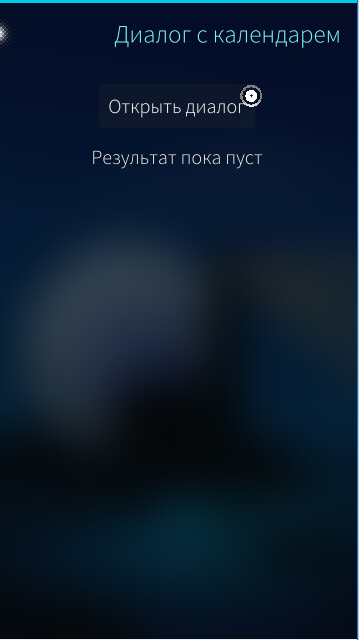
# Приложение из двух страниц. Первая страница содержит две кнопки “Добавить страницу” и “Убрать страницу”. Первая кнопка добавит вторую страницу как прикреплённую, вторая кнопка её удалит. На второй странице должна быть кнопка для возврата на первую страницу без закрытия второй.

# 

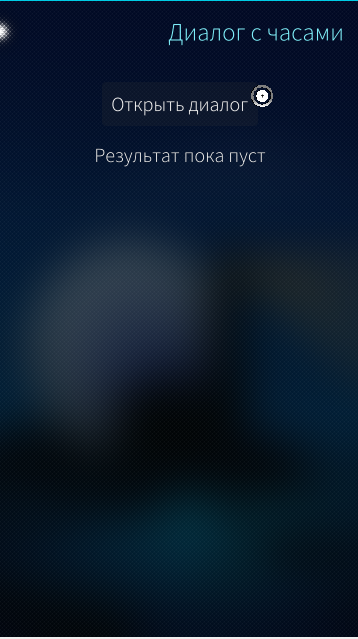
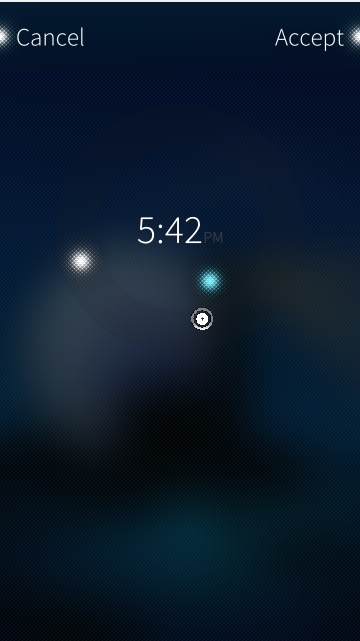
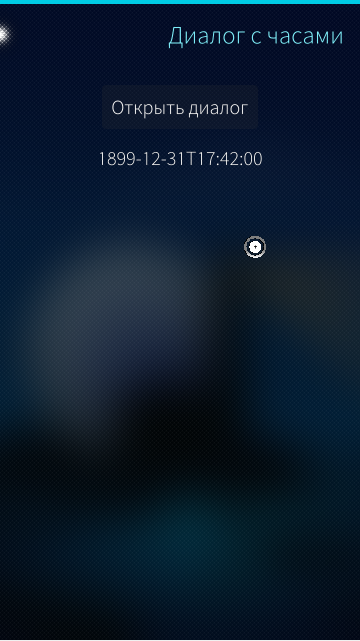
# Приложение с одной кнопкой и текстовом поле. После нажатия на кнопку отображается диалог для ввода текста. После согласия с результатом введённый текст отображается в текстовое поле.

# 

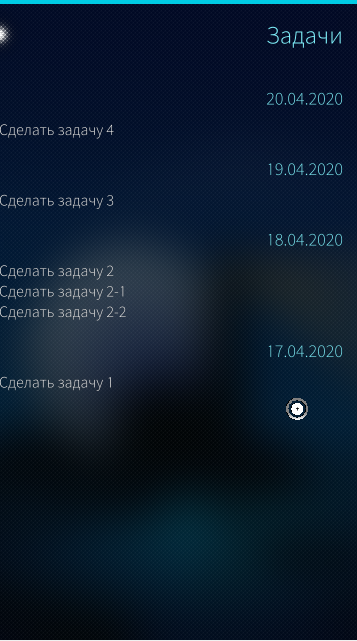
1. Приложение с одной кнопкой и текстовым полем. После нажатия на кнопку отображается диалог для выбора даты. После согласия с результатом ввода выбранная дата отображается в текстовое поле.



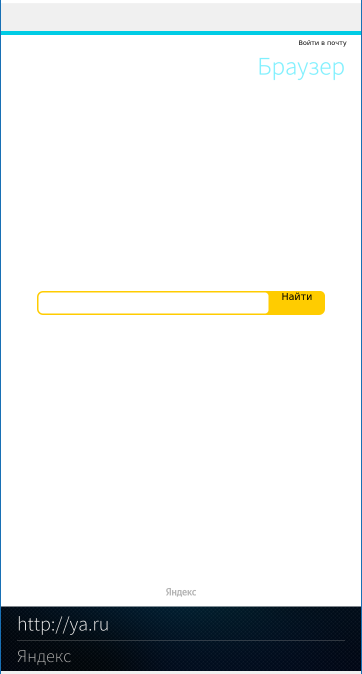
1. Приложение с одной кнопкой и текстовым полем. После нажатия на кнопку отображается диалог для выбора времени. После согласия с результатом ввода выбранное время отображается в текстовом поле.

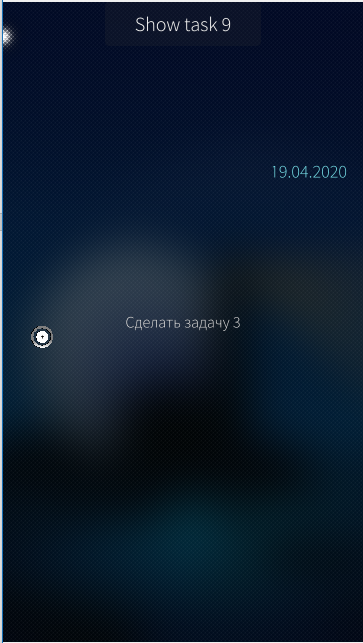
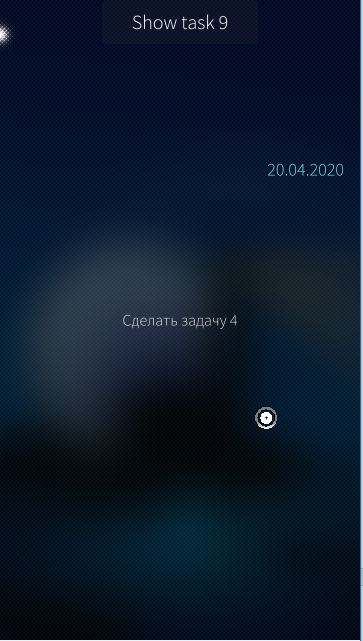
1. Приложение со списком SilicaListView, из задач на неделю. Задачи должны содержать дату и описание. В списке задачи группировать по датам.



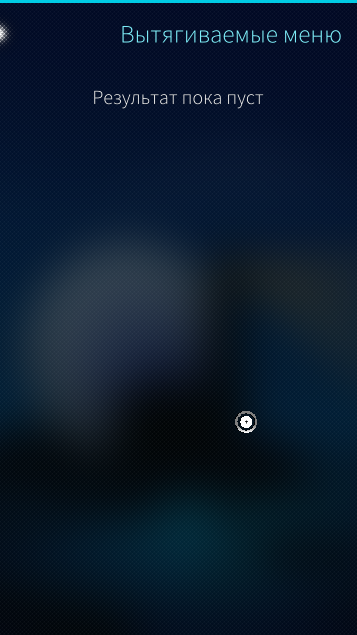
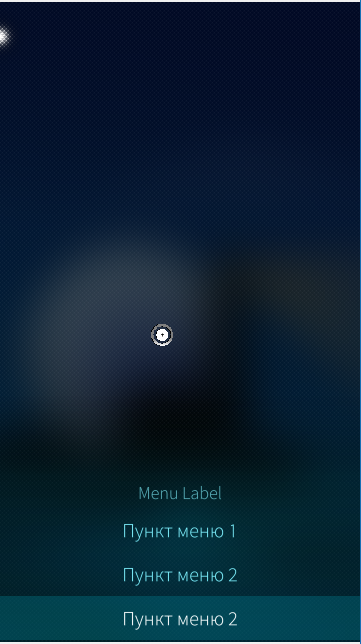
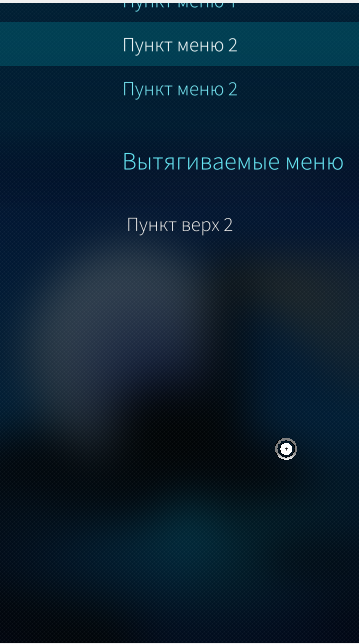
1. Приложение с SilicaWebView для доступа к вашему любимому сайту.



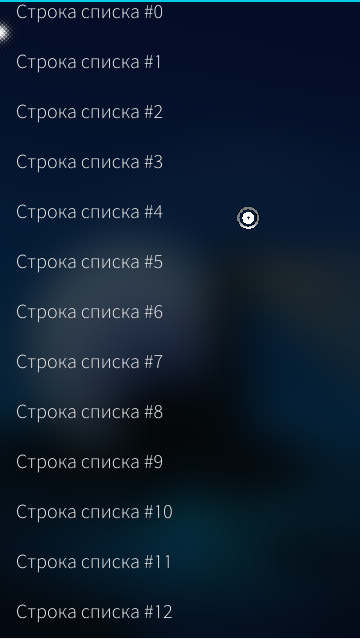
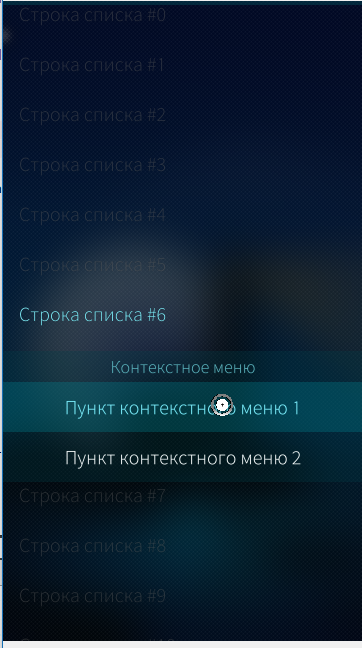
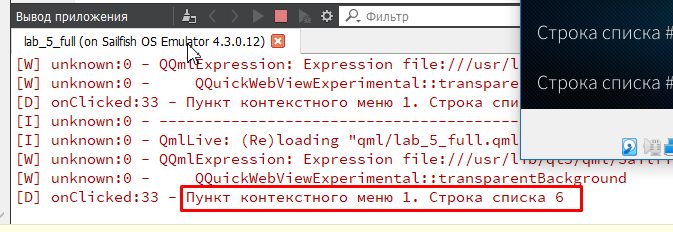
1. Использовать SlideshowView для просмотра и перелистывания задач на неделю. На одном слайде – одна задача.



1. Приложение с вытягиваемыми меню сверху и снизу и текстовым полем. После выбора какого-либо элемента меню, его название отобразить в текстовом поле.

1. Приложение со списком и контекстным меню. После выбора элемента контекстного меню отобразить в консоли название выбранного элемента меню и индекс элемента списка.

# Приложение

Страница 1

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuItem {

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

onClicked: *pageStack*.animatorPush(*Qt*.resolvedUrl("PushAttachedPage.qml"))

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Стек страниц. Задание 1")

}

Button {

text: "Добавить";

anchors.horizontalCenter: *parent*.horizontalCenter;

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

}

Button {

text: "Удалить";

anchors.horizontalCenter: *parent*.horizontalCenter;

onClicked: *pageStack*.pop();

}

Label {

anchors.horizontalCenter: *parent*.horizontalCenter;

text: "Глубина стека: " + *pageStack*.depth;

}

}

}

}

Страница 2

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 3")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("PageDialogText.qml"))

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Прикрепленные страницы")

}

Label {

anchors.right: *parent*.right;

text: "Первая страница"

}

Button {

text: "Добавить страницу";

anchors.horizontalCenter: *parent*.horizontalCenter;

onClicked: pageStack.pushAttached(*Qt*.resolvedUrl("PushAttachedPage\_2.qml"));

}

Button {

text: "Удалить странциу";

anchors.horizontalCenter: *parent*.horizontalCenter;

onClicked: pageStack.popAttached();

}

}

}

}

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

Label {

anchors.right: *parent*.right;

text: "Вторая страница"

}

Button {

text: "Назад";

anchors.horizontalCenter: *parent*.horizontalCenter;

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

}

}

}

Страница 3

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 4")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("PageDialogCalendar.qml"))

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Диалог с текстом")

}

Button {

text: "Открыть диалог"

anchors.horizontalCenter: *parent*.horizontalCenter

onClicked: {

var *dialog* = pageStack.push(*Qt*.resolvedUrl("DialogText.qml"))

*dialog*.accepted.connect(function() { *lb*.text = "Результат: " + *dialog*.res

})

}

}

Label {

id: *lb*;

anchors.horizontalCenter: *parent*.horizontalCenter;

text: "Результат пока пуст";

}

}

}

}

Страница 4

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 5")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("PageDialogWatch.qml"))

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Диалог с календарем")

}

Button {

text: "Открыть диалог"

anchors.horizontalCenter: *parent*.horizontalCenter

onClicked: {

var *dialog* = pageStack.push("Sailfish.Silica.DatePickerDialog",

{date: new *Date*()});

*dialog*.accepted.connect(function() {

*lb*.text = (*dialog*.date);

});

}

}

Label {

id: *lb*;

anchors.horizontalCenter: *parent*.horizontalCenter;

text: "Результат пока пуст";

}

}

}

}

Страница 5

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 6")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("TaskList.qml"))

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Диалог с часами")

}

Button {

text: "Открыть диалог"

anchors.horizontalCenter: *parent*.horizontalCenter

onClicked: {

var *dialog* = pageStack.push("Sailfish.Silica.TimePickerDialog",

{hour: 13, minute: 42});

*dialog*.accepted.connect(function() {

*lb*.text = (*dialog*.time);

});

}

}

Label {

id: *lb*;

anchors.horizontalCenter: *parent*.horizontalCenter;

text: "Результат пока пуст";

}

}

}

}

Страница 6

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

ListModel {

id: *taskModel*

ListElement { task: "Сделать задачу 4"; data: "20.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 3"; data: "19.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 2"; data: "18.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 2-1"; data: "18.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 2-2"; data: "18.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 1"; data: "17.04.2020"; color\_text: "white"; }

}

SilicaListView {

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 7")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("Browser.qml"))

}

}

anchors.fill: *parent*

model: *taskModel*

header: PageHeader { title: "Задачи" }

section {

property: 'data'

delegate: SectionHeader { text: section }

}

delegate: Text { text: task; color: color\_text; }

}

}

Страница 7

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

SilicaWebView {

PullDownMenu {

MenuItem {

text: *qsTr*("Show task 8")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("TaskListShowView.qml"))

}

}

PageHeader {

title: *qsTr*("Браузер")

}

id: *webView*

anchors {

top: parent.top; bottom: *urlField*.top;

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

}

url: "http://ya.ru"

}

TextField {

id: *urlField*

anchors {

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

bottom: *parent*.bottom

}

text: "http://ya.ru"

label: *webView*.title

EnterKey.onClicked: *webView*.url = *text*

}

}

Страница 8

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

ListModel {

id: *taskModel*

ListElement { task: "Сделать задачу 4"; datas: "20.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 3"; datas: "19.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 2"; datas: "18.04.2020"; color\_text: "white"; }

ListElement { task: "Сделать задачу 1"; datas: "17.04.2020"; color\_text: "white"; }

}

Button {

text: "Show task 9";

anchors.horizontalCenter: *parent*.horizontalCenter;

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

}

SlideshowView {

id: *view*

anchors.centerIn: *parent*

height: *width*

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

model: *taskModel*

delegate: Item {

width: *view*.itemWidth;

height: *view*.itemHeight;

SectionHeader { text: datas }

Text { text: task; color: color\_text ; anchors.centerIn: *parent*}

}

}

}

Страница 9

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

contentHeight: *column*.height

PullDownMenu {

MenuLabel { text: *qsTr*("Верхнее меню") }

MenuItem {

text: *qsTr*("Show task 10")

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("ListAndContextMenu.qml"))

}

MenuItem {

text: "Пункт меню 1"

onClicked: *lb*.text = "Пункт верх 1"

}

MenuItem {

text: "Пункт меню 2"

onClicked: *lb*.text = "Пункт верх 2"

}

MenuItem {

text: "Пункт меню 2"

onClicked: *lb*.text = "Пункт верх 3"

}

}

PushUpMenu {

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

MenuItem {

text: "Пункт меню 1"

onClicked: *lb*.text = "Пункт низ 1"

}

MenuItem {

text: "Пункт меню 2"

onClicked: *lb*.text = "Пункт низ 2"

}

MenuItem {

text: "Пункт меню 2"

onClicked: *lb*.text = "Пункт низ 2"

}

}

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Вытягиваемые меню")

}

Label {

id: *lb*;

anchors.horizontalCenter: *parent*.horizontalCenter;

text: "Результат пока пуст";

}

}

}

}

Страница 10

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

PullDownMenu {

MenuItem {

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

onClicked: pageStack.animatorPush(*Qt*.resolvedUrl("CreatePages.qml"))

}

}

SilicaListView {

anchors.fill: *parent*

model: 20

delegate: ListItem {

id: *delegate*

Label {

x: Theme.paddingLarge

text: "Строка списка #" + index

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

}

menu: ContextMenu {

MenuLabel { text: "Контекстное меню" }

MenuItem {

text: "Пункт контекстного меню 1"

onClicked: *console*.log("Пункт контекстного меню 1. Строка списка " + index)

}

MenuItem {

text: "Пункт контекстного меню 2"

onClicked: *console*.log("Пункт контекстного меню 2. Строка списка " + index)

}

}

}

}

}

}