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 Математического обеспечения и суперкомпьютерных технологий**

Направление подготовки: «Фундаментальная информатика и

информационные технологии»

Профиль подготовки: «Инженерия программного обеспечения»

**ОТЧЕТ**

**По лабораторной работе №3**

по дисциплине «Технология программирования мобильных систем»

**Выполнил(а):** студент(ка) группы

381908-4

Кургузиков Кирилл

**Руководитель:**

доцент кафедры программной

инженерии

Борисов Николай Анатольевич

Нижний Новгород

2022

Содержание

[**1.Цель** 3](#_Toc97543312)

[**2.Постановка задачи** 4](#_Toc97543313)

[**3.Решение поставленной задачи** 5](#_Toc97543314)

[**4.Руководство пользователя** 6](#_Toc97543315)

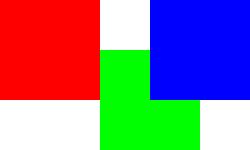
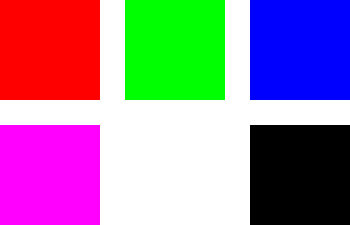
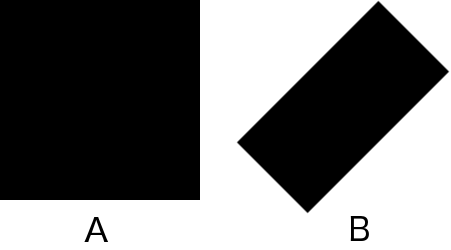
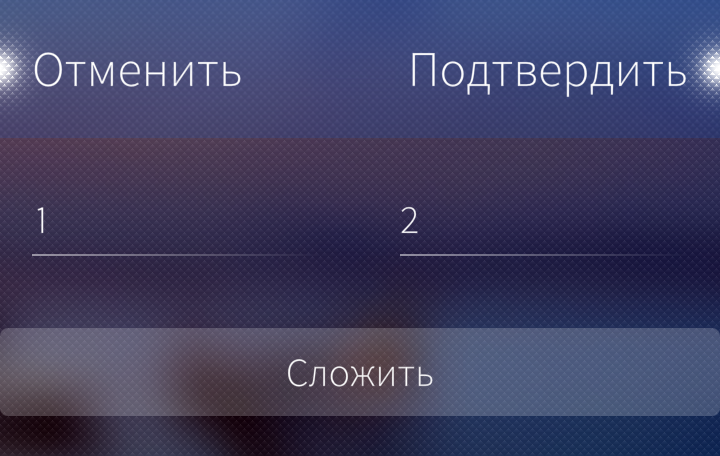
[**5.Приложение** 8](#_Toc97543316)

# **1.Цель**

Освоить базовые навыки построения пользовательских интерфейсов, позиционирования, отрисовки и перемещения элементов. Научиться анимировать элементы. Научиться создавать диалоги и взаимодействовать с ними.

# **2.Постановка задачи**

Шаги:

1. Создать новый проект со стандартной заготовкой приложения.
2. Нарисовать 3 квадрата красного, зелёного и синего цветов следующим образом:
3. Поместить текст “Квадрат” белого цвета по центру синего квадрата.
4. Нарисовать 5 квадратов с использованием Column и Row следующим образом: 
5. Нарисовать те же 5 квадратов с использованием Grid.
6. Сделать из квадрата “A” прямоугольник “B” с использованием объектов Translate, Scale и Rotation  
   
7. Нарисовать квадрат и анимировать его перемещение вниз с увеличением его размера. Документация по анимации доступна по адресу <http://doc.qt.io/qt-5/qml-qtquick-animation.html>.
8. Реализовать диалог с двумя текстовыми полями, в которые вводятся числа. После нажатия на кнопку “Подтвердить” в консоль выводится сумма чисел. Для преобразования строк к числам использовать функцию *parseInt(“42”).* Валидацией и обработкой ошибок можно пренебречь.

# **3.Решение поставленной задачи**

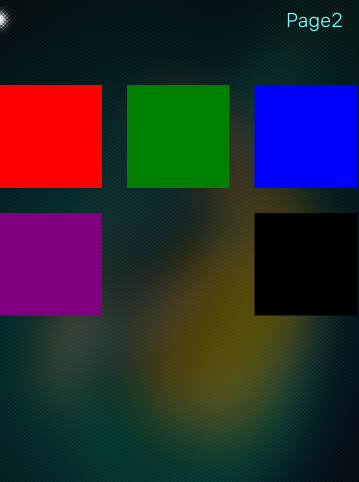
В самом начале страницы нужно добавить контейнер Column, используемый по умолчанию.

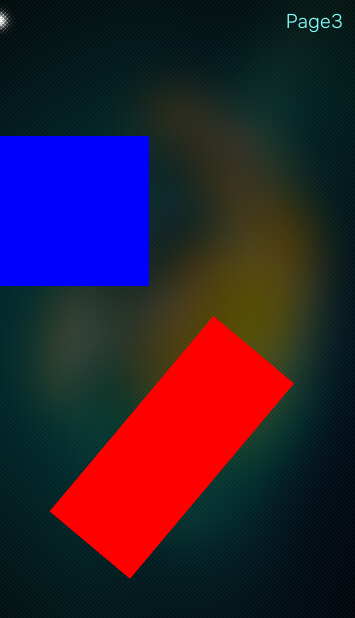
Чтобы добавить квадраты нужно добавить элемент Item в них поместить Rectangle. Чтобы появилась надпись нужно поместить Label в центр синего квадрата с помощью функции anchors.centerIn. Для отрисовки квадратов в ряд был использован контейнер Column и Row . чтобы квадраты были одинакового размера высота была задана из контейнера Column width:*column*.width/4 + 25, height:*column*.width/4 + 25. Для отрисовки квадратов с помощью Grid были добавлены rows:2; columns:3;. Для отрисовки прямоугольного квадрата как на рисунку 7, была использована функция transform. Для анимации были использованы функции ParallelAnimation PathAnimation PropertyAnimation. Для реализации диалога была добавлена кнопка, которая вызывает диалоговое окно при помощи функции OnClicked. В самом окне, для вывода суммы в консоль была добавлена функция onAccepted:.

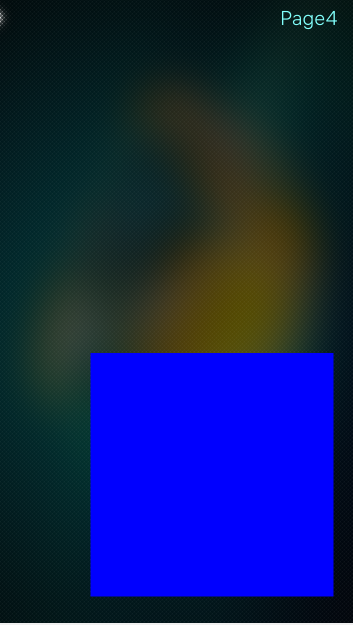
# **4.Руководство пользователя**

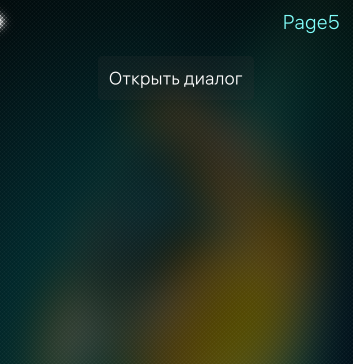
При запуске приложения пользователь увидит квадраты с 1 по 4 страницы.

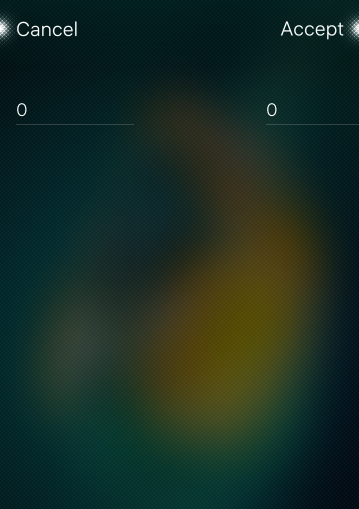




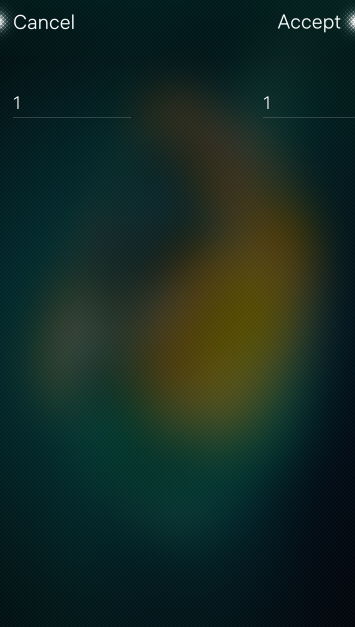




На 5 странице пользователь увидит кнопку, при нажатии кнопка вызовет диалоговое окно 



Когда пользователь введёт числа и нажмёт на “Accepted” в консоль выведется сумма чисел, иначе консоль закроется.





# **5.Приложение**

FirstPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Dialog{

property string res

DialogHeader {}

Row{

y:200;

spacing:200;

TextField{

id:*leftTF*

width:300

text:"0"

}

TextField{

id:*rightTF*

width:300

text:"0"

}

}

onRejected:{*leftTF*.text("0")

*rightTF*.text("0")}

onAccepted:{*res*=parseInt(*leftTF*.text)+parseInt(*rightTF*.text);

*console*.log("Result: " + (parseInt(*leftTF*.text )+ parseInt(*rightTF*.text)));

}

}

SecondPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page{

id:*page*

//TheeffectivevaluewillberestrictedbyApplicationWindow.allowedOrientations

allowedOrientations:Orientation.All

//ToenablePullDownMenu,placeourcontentinaSilicaFlickable

SilicaFlickable{

anchors.fill:*parent*

//PullDownMenuandPushUpMenumustbedeclaredinSilicaFlickable,SilicaListVieworSilicaGridView

PullDownMenu{

MenuItem{

text:*qsTr*("ShowPage3")

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

}

}

//PlaceourcontentinaColumn.ThePageHeaderisalwaysplacedatthetop

//ofthepage,followedbyourcontent.

PageHeader{

title:*qsTr*("Page2")

}

Grid{

y:200;

width:*parent*.width;

height:*parent*.height;

spacing:50;

rows:2;

columns:3;

Rectangle{

id:*r1*;

color:"red";

width:*parent*.width/3.5;

height:*parent*.width/3.5;

}

Rectangle{

id:*r2*;

color:"green";

width:*r1*.width;

height:*r1*.width;

}

Rectangle{

id:*r3*;

color:"blue";

width:*r1*.width;

height:*r1*.width;

}

Rectangle{

id:*r4*;

color:"purple";

width:*r1*.width;

height:*r1*.width;

}

Rectangle{

id:*r5*;

color:"transparent";

width:*r1*.width;

height:*r1*.width;

}

Rectangle{

id:*r6*;

color:"black";

width:*r1*.width;

height:*r1*.width;

}

}

}

}

ThirdPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id:*page*

//TheeffectivevaluewillberestrictedbyApplicationWindow.allowedOrientations

allowedOrientations:Orientation.All

//ToenablePullDownMenu,placeourcontentinaSilicaFlickable

SilicaFlickable{

anchors.fill:*parent*

//PullDownMenuandPushUpMenumustbedeclaredinSilicaFlickable,SilicaListVieworSilicaGridView

PullDownMenu{

MenuItem{

text:*qsTr*("ShowPage4")

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

}

}

//PlaceourcontentinaColumn.ThePageHeaderisalwaysplacedatthetop

//ofthepage,followedbyourcontent.

PageHeader{

title:*qsTr*("Page3")

}

Item{

y:300;

Rectangle{

id:*r1*;

color:"blue"

width:300;

height:300;

}

Rectangle{

id:*r2*;

color:"red"

width:300;

height:300;

transform:[Translate{x:800},Scale{xScale:0.7;yScale:1.7},Rotation{angle:40}]

}

}

}

}

FourthPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page{

id:*page*

//TheeffectivevaluewillberestrictedbyApplicationWindow.allowedOrientations

allowedOrientations:Orientation.All

//ToenablePullDownMenu,placeourcontentinaSilicaFlickable

SilicaFlickable{

anchors.fill:*parent*

//PullDownMenuandPushUpMenumustbedeclaredinSilicaFlickable,SilicaListVieworSilicaGridView

PullDownMenu{

MenuItem{

text:*qsTr*("ShowPage5")

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

}

}

PageHeader{

title:*qsTr*("Page4")

}

Rectangle{

id:*blueRect*;

color:"blue";

x:50;

y:100;

width:300;

height:300;

ParallelAnimation{

id:*animation*

loops: Animation.Infinite;

PathAnimation{

path:Path{

PathCurve {x:200; y:750}

PathCurve {x:200; y:100}

}

duration:4000;

target:*blueRect*;

}

PropertyAnimation{

target:*blueRect*

property:"width"

from:300;

to:500;

duration:2000;

}

PropertyAnimation{

target:*blueRect*

property:"height"

from:300;

to:500;

duration:2000

}

running:*animation*;

}

ParallelAnimation{

id:*animation1*

loops: Animation.Infinite;

PropertyAnimation{

target:*blueRect*

property:"width"

from:500;

to:300;

duration:2000;

}

PropertyAnimation{

target:*blueRect*

property:"height"

from:500;

to:300;

duration:2000

}

running:*animation*;

}

}

}

}

FifthPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page{

id:*page*

//TheeffectivevaluewillberestrictedbyApplicationWindow.allowedOrientations

allowedOrientations:Orientation.All

//ToenablePullDownMenu,placeourcontentinaSilicaFlickable

SilicaFlickable{

anchors.fill:*parent*

//PullDownMenuandPushUpMenumustbedeclaredinSilicaFlickable,SilicaListVieworSilicaGridView

PullDownMenu{

MenuItem{

text:*qsTr*("ShowPage1")

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

}

}

//PlaceourcontentinaColumn.ThePageHeaderisalwaysplacedatthetop

//ofthepage,followedbyourcontent.

Column{

id:*column*

width:*page*.width

spacing:Theme.paddingLarge

PageHeader{

title:*qsTr*("Page5")

id:*header*

}

Button{

text:"Открыть диалог"

x:250;

y:300;

anchors{

top:*header*.bottom

horizontalCenter:*parent*.horizontalCenter

}

onClicked:{

vardialog=pageStack.push(*Qt*.resolvedUrl("DialogPage.qml"),{"res":*header*.title})

}

}

}

}

}

DialogPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Dialog{

property string res

DialogHeader {}

Row{

y:200;

spacing:200;

TextField{

id:*leftTF*

width:300

text:"0"

}

TextField{

id:*rightTF*

width:300

text:"0"

}

}

onRejected:{*leftTF*.text("0")

*rightTF*.text("0")}

onAccepted:{*res*=parseInt(*leftTF*.text)+parseInt(*rightTF*.text);

*console*.log("Result: " + (parseInt(*leftTF*.text )+ parseInt(*rightTF*.text)));

}

}