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**

**Кафедра программной инженерии**

**Практическое работа №2**

**Тема:**

**«Основы конструирования интерфейсов пользователя»**

**Выполнил**:  
студент группы 381808—2  
   
Жафяров Олег Ильязович

**Проверил:**

Карчков Денис Александрович

Нижний Новгород

2021

**СОДЕРЖАНИЕ**

1. [Описание задачи 3](#_Toc59472872)
2. [Демонстрация работы 5](#_Toc59472873)
3. [Заключение 7](#_Toc59472881)
4. [Приложение](#_Toc59472887) 8

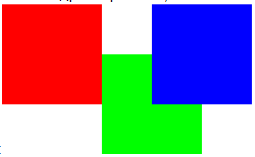
**ОПИСАНИЕ ЗАДАЧИ**

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

Шаги:

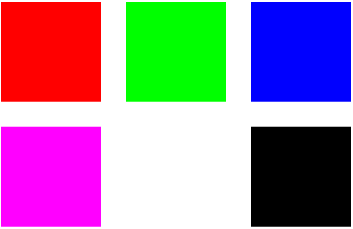
1) Создать новый проект со стандартной заготовкой приложения.

2) Нарисовать 3 квадрата красного, зелёного и синего цветов следующим образом:



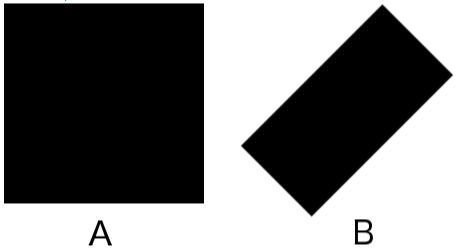
3) Поместить текст “Квадрат” белого цвета по центру синего квадрата.

4) Нарисовать 5 квадратов с использованием Column и Row следующим образом:



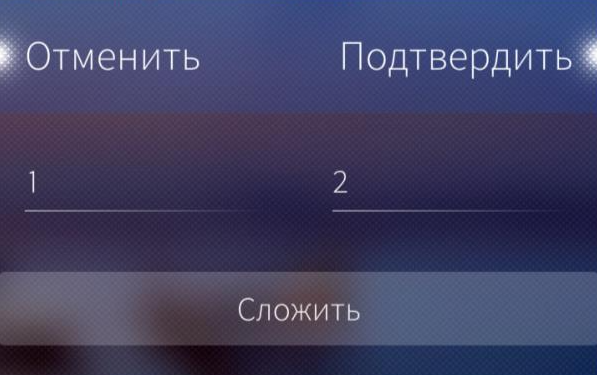
5) Нарисовать те же 5 квадратов с использованием Grid.

6) Сделать из квадрата “A” прямоугольник “B” с использованием объектов Translate, Scale и Rotation

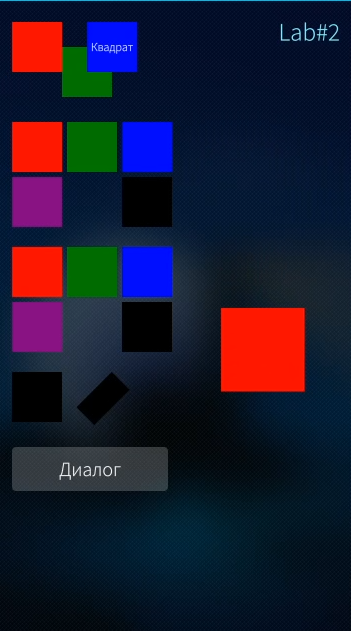


7) Нарисовать квадрат и анимировать его перемещение вниз с увеличением его размера. Документация по анимации доступна по адресу http://doc.qt.io/qt-5/qml-qtquick-animation.html.

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.



**ДЕМОНСТРАЦИЯ РАБОТЫ**







# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы был создан проект со стандартной заготовкой приложения, нарисованы 3 квадрата разного цвета, нарисованы 5 квадратов с использованием Column, Row и Grid. Из квадрата создан прямоугольник с использованием Translate, Scale и Rotation. Нарисован и анимирован квадрат. Реализован диалог, который выводит сумму чисел.

# Приложение

**1. Ссылка на проект:** https://github.com/Oleg-ui/Qt/tree/main/Lab2

**2. Код проекта:**

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 2")

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

}

}

contentHeight: *column*.height

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Lab#2")

}

}

Item{

x: Theme.horizontalPageMargin

y: 50

Rectangle{

width: 100

height: 100

color: "red"

}

Rectangle{

transform: Translate{y: 50; x:100}

width: 100

height: 100

color: "green"

}

Rectangle{

transform: Translate{x: 150}

width: 100

height: 100

color: "blue"

Text{

color:"white"

font.pointSize: 18

anchors.centerIn: *parent*

text: *qsTr*("Квадрат")

}

}

}

Row{

x: Theme.horizontalPageMargin

spacing:10

y: 250

Column{

spacing:10

Rectangle{

width: 100

height: 100

color: "red"

}

Rectangle{

width: 100

height: 100

color: "purple"

}

}

Column{

spacing:10

Rectangle{

width: 100

height: 100

color: "green"

}

}

Column{

spacing:10

Rectangle{

width: 100

height: 100

color: "blue"

}

Rectangle{

width: 100

height: 100

color: "black"

}

}

}

Grid{

x: Theme.horizontalPageMargin

y: 500

columns: 3

rows: 2

spacing: 10

Rectangle{width: 100; height: 100; color: "red"}

Rectangle{width: 100; height: 100; color: "green"}

Rectangle{width: 100; height: 100; color: "blue"}

Rectangle{width: 100; height: 100; color: "purple"}

Rectangle{width: 100; height: 100; color: "transparent"}

Rectangle{width: 100; height: 100; color: "black"}

}

Item{

y: 750

x: Theme.horizontalPageMargin

Rectangle{

width: 100

height: 100

color: "black"

}

Rectangle{

transform: [Scale{xScale: 0.5; yScale: 1}, Rotation{angle:45}, Translate{x: +200}]

width: 100

height: 100

color: "black"

}

}

Rectangle{

id: *redRect*

y: 250

x: 450

width: 10

height: 10

color: "red"

ParallelAnimation{

running: true

loops: Animation.Infinite

NumberAnimation{

target: *redRect*

property: "width"

from: 10

to: 200

duration: 2000

}

NumberAnimation{

target: *redRect*

property: "height"

from: 10

to: 200

duration: 2000

}

NumberAnimation{

target: *redRect*

property: "y"

from: 250

to: 700

duration: 2000

}

}

}

Component{

id: *dialog*

Dialog{

DialogHeader{}

Column{

anchors.centerIn: *parent*

spacing: 10

TextField{

id: *field1*

width: *parent*.width

inputMethodHints: *Qt*.ImhFormattedNumbersOnly

placeholderText: *label*

color: Theme.highlightColor

}

TextField{

id: *field2*

width: *parent*.width

inputMethodHints: *Qt*.ImhFormattedNumbersOnly

placeholderText: *label*

color: Theme.highlightColor

}

Button{

text: "Сложить"

onClicked: *console*.log(parseInt(*field1*.text) + parseInt(*field2*.text));

}

}

onAccepted: *console*.log(parseInt(*field1*.text) + parseInt(*field2*.text));

onRejected: helloLabel.text = "Отменил действие"

}

}

Item{

x: Theme.horizontalPageMargin

y: 900

Button{

text: "Диалог"

onClicked: {

*pageStack*.push(*dialog*)

}

}

}

}

}