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**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Отчет по практическому заданию №2:

**«Инструменты разработки мобильных приложений»**

**Выполнила:**

студентка группы 381906-1

Тырина Анастасия Константиновна

Нижний Новгород

2022 г.

**Оглавление**

[Постановка задачи 3](file:///C:\Users\user\Desktop\мобилки\Тюрмина%20А.%20381806-3%20-%20отчет%20л.р.2.docx#_Toc90716626)

Руководство программиста……………………………………………………………...5

Руководство пользователя……………………………………………………………….7

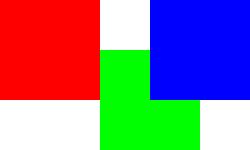
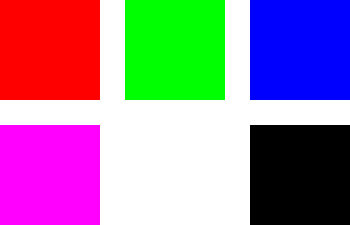
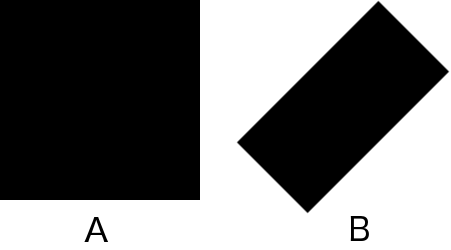
Вывод…………………………………………………………………………………....10

Приложение……………………………………………………………………………..11

# Постановка 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. 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.

# Руководство программиста

Для того, чтобы нарисовать 3 квадрата красного, зелёного и синего цветов и поместить текст “Квадрат” белого цвета по центру одного из них, использовались следующие визуальные элементы:

«**Rectangle»:**

* **id –** идентификатор объекта
* **width: – ширина объекта в пикселях**
* **height: – высота объекта в пикселях**
* **color: "green" –** цвет объекта
* **x:**  – смещение по оси х
* **y:**  – смещение по оси у

Блок «**Text»:**

* **text:** – текст надписи
* **color: "FFF"** – цвет надписи
* **font.pointSize** - размер шрифта в точках
* **anchors.centerIn:**  *–* размещение по центру

Для отрисовки пяти квадратов использовались следующие контейнеры:

* «**Column**» - компонент для размещения элементов в столбец и **«Row» -** для размещения элементов в рядсо свойством **spacing**, указывающим на размер отступа между строками и столбцами.
* «**Grid»:** для размещения элементов сеткой

Для трансформирования квадрата использовался **transform** со следующими свойствами**:**

**Translate** - перемещение элемента вдоль осей x и y

**Scale** - изменение размера элемента по x и y

**Rotation** - изменение угла наклона вдоль осей x, y или z

Для анимации квадрата использовался **ParallelAnimation,** который позволяет запускать несколько анимаций параллельнои **NumberAnimation** (анимирует изменения числовых значений)

Для ввода данных пользователем используются диалоги.

Блок **Button:**

* **text:–** название кнопки
* **width–** ширина кнопки
* **onClicked: *console*.log(parseInt(*num1*.text) + parseInt(*num2*.text)) –** обработчик события. При нажатии на кнопку в консоль выводится сумма чисел.

# Руководство пользователя

После запуска программы пользователем, открывается окно эмулятора Sailfish OS.

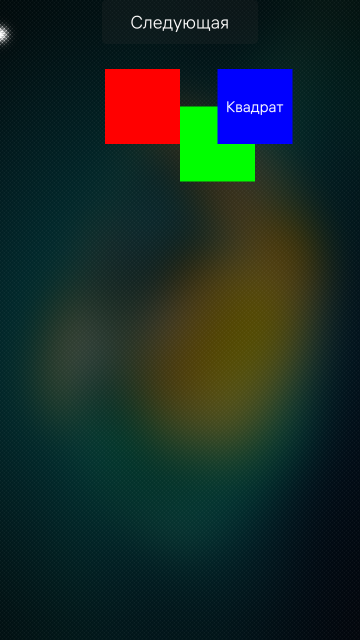


Рис 1. Интерфейс программы

По кнопке «Следующая» пользователь переходит на следующую страницу.

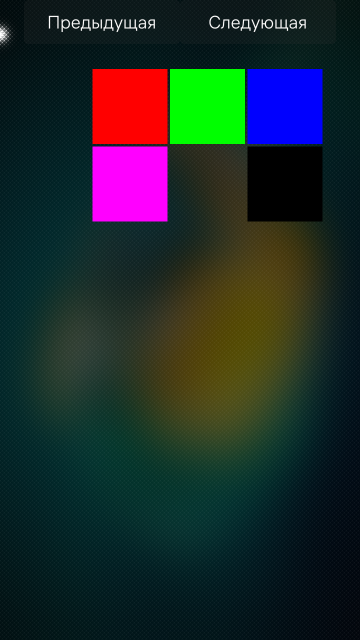


Рис 2. Интерфейс программы

При нажатии на кнопку «Открыть диалог» происходит переход в диалог.

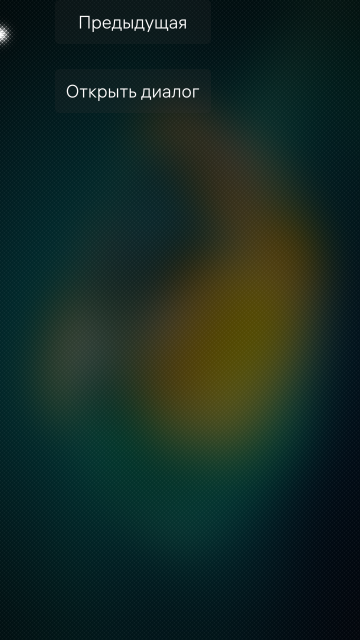


Рис 3. Интерфейс программы

После ввода цифр в текстовые поля и нажатии на кнопку «Accept» в консоль выводится их сумма.

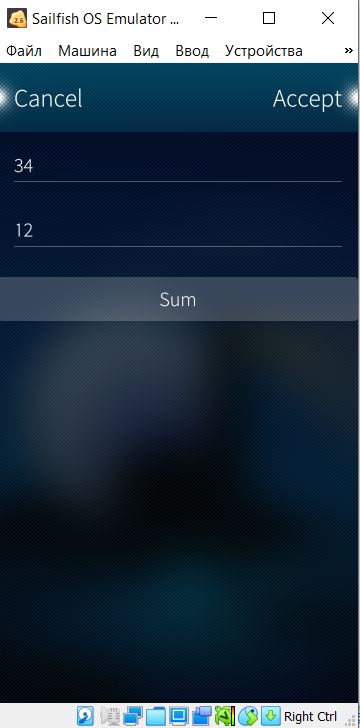


Рис 4. Интерфейс программы

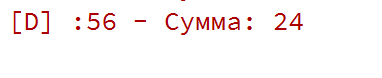


Рис 5. Вывод суммы

# Вывод

В процессе выполнения данной лабораторной работы мы освоили базовые навыки построения пользовательских интерфейсов, позиционирования, отрисовки и перемещения элементов. Научились анимировать элементы. Научились создавать диалоги и взаимодействовать с ними, а также выполнили практическое задание.

# Приложение

Page1

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page1"

property double factor: 3

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Следующая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page2.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 100 \* *factor*

height: 75 \* *factor*

Rectangle {

id: *red1*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#FF0000"

}

Rectangle {

id: *green1*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#00FF00"

anchors.top: *red1*.verticalCenter

anchors.left: *red1*.right

}

Rectangle {

id: *blue1*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#0000FF"

anchors.left: *green1*.horizontalCenter

anchors.bottom: *green1*.verticalCenter

}

Text {

text: "Квадрат"

color: "#FFF"

anchors.centerIn: *blue1*

font.pointSize: 7.5 \* *factor*

}

}

}

}

Page 2

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page2"

property double factor: 3

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Предыдущая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page1.qml"))

}

Button {

text: "Следующая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page3.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 350

height: 50 \* 2 \* *factor*

Row {

spacing: 5

Column {

spacing: 5

Rectangle {

id: *red2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#FF0000"

}

Rectangle {

id: *magenta2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#F0F"

}

}

Column {

spacing: 5

Rectangle {

id: *green2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#00FF00"

}

}

Column {

spacing: 5

Rectangle {

id: *blue2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#0000FF"

}

Rectangle {

id: *black2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#000"

}

}

}

}

}

}

Page 3 import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page3"

property double factor: 3

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Предыдущая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page2.qml"))

}

Button {

text: "Следующая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page4.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 400

height: 50 \* 2 \* *factor*

Grid {

spacing: 5

columns: 3

Rectangle {

id: *red3*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#FF0000"

}

Rectangle {

id: *green3*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#00FF00"

}

Rectangle {

id: *blue3*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#0000FF"

}

Rectangle {

id: *magenta3*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#F0F"

}

Rectangle {

id: *empty3*

width: 50 \* *factor*

height: 50 \* *factor*

opacity: 0

}

Rectangle {

id: *black3*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#000"

}

}

}

}

}

Page 4

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page4"

property double factor: 3

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Предыдущая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page3.qml"))

}

Button {

text: "Следующая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page5.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 350

height: 50 \* 2 \* *factor*

Row {

spacing: 50 \* *factor*

Rectangle {

id: *black4\_1*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#000"

}

Rectangle {

id: *black4\_2*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#000"

transform: [

Scale {

xScale: 0.5

},

Rotation {

angle: 45

}

]

}

}

}

}

}

Page 5

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page5"

property double factor: 3

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Предыдущая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page4.qml"))

}

Button {

text: "Сдедующая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page6.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 350

height: 50 \* 2 \* *factor*

Rectangle {

id: *black5*

width: 50 \* *factor*

height: 50 \* *factor*

color: "#000"

x: 50

ParallelAnimation {

running: true

NumberAnimation { target: *black5*; property: "y"; to: 100; duration: 1000; loops: Animation.Infinite }

NumberAnimation { target: *black5*; property: "scale"; to: 2; duration: 1000; loops: Animation.Infinite }

}

}

}

}

}

Page 6

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "page6"

Column {

anchors.horizontalCenter: *parent*.horizontalCenter

spacing: 50

Row {

Button {

text: "Предыдущая"

onClicked: pageStack.push(*Qt*.resolvedUrl("Page5.qml"))

}

}

Item {

anchors.horizontalCenter: *parent*.horizontalCenter

width: 500

height: 400

Button {

text: "Открыть диалог"

onClicked: *dialog*.open()

}

Dialog {

id: *dialog*

Column {

anchors.fill: *parent*

width: *parent*.width

spacing: Theme.paddingMedium

DialogHeader { }

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

width: *parent*.width / 2.2 \* 2 + 20

spacing: 20

TextField {

id: *num1*

width: *parent*.width / 2.2

}

TextField {

id: *num2*

width: *parent*.width / 2.2

}

}

}

onAccepted: function(){

var *res1* = *num1*.text - '0'

if (isNaN(*res1*)) return

var *res2* = *num2*.text - '0'

if (isNaN(*res2*)) return

*console*.log("Сумма:", *res1* + *res2*)

}

}

}

}

}