

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ИМЕНИ И.М. ГУБКИНА»

ФАКУЛЬТЕТ КОМПЛЕКСНОЙ БЕЗОПАСНОСТИ ТЭК
КАФЕДРА БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Лабораторная работа №12

по дисциплине «Специализированные языки и технологии
программирования»

на тему «Создание простого приложения для ОС Аврора»

Выполнил студент:

группы КА-22-06

Воронин Алексей Дмитриевич

Преподаватель:

Греков Владимир Сергеевич

Москва, 2025

Оглавление	
Цель работы	3
Задание 1. Статический интерфейс	4
Задание 2. Интерактивные элементы	4
Задание 3. Анимация.....	4
Часть 4 – Самостоятельная работа	4
ЗАКЛЮЧЕНИЕ	9
Контрольные вопросы	10

Цель работы

- Освоить базовые принципы работы с графическими элементами в Qt Quick.
- Научиться создавать и позиционировать элементы интерфейса в ОС Аврора.
- Познакомиться с изменением стилей и добавлением текста в элементы.

Задание 1. Статический интерфейс

Основу приложения составляют:

- Три цветных квадрата (синий, красный, зеленый)
- Горизонтальное расположение
- Текст в синем квадрате
- Кнопки управления

Задание 2. Интерактивные элементы

Необходимо добавить:

- Переключатели цвета фона
- Текстовое поле с кнопкой
- Ползунок прозрачности

Задание 3. Анимация

Реализовать

- Движение зеленого квадрата
- Изменение цветов по кнопкам
- Обновление текста

Часть 4 – Самостоятельная работа

В рамках самостоятельной работы необходимо добавить кнопки «Увеличить» и «Уменьшить». Для этого в MainPage.qml добавить следующее:

```
import QtQuick 2.6
import Sailfish.Silica 1.0
import QtMultimedia 5.6

Page {
    id: root
    allowedOrientations: Orientation.All

    property int squareSize: 100
    property real squareOpacity: 1.0
    property string blueText: "lab12"

    Rectangle {
        id: backgroundRect
        anchors.fill: parent
        color: Theme.rgb(Theme.highlightBackgroundColor, 0.0)
        z: -1
    }

    SoundEffect {
```

```

        id: soundEffect
        source: "grc:/sounds/click.wav"
    }

    SilicaFlickable {
        anchors.fill: parent
        contentHeight: column.height

        Column {
            id: column
            width: parent.width
            spacing: Theme.paddingLarge
            topPadding: Theme.paddingLarge
            bottomPadding: Theme.paddingLarge

            PageHeader {
                title: "Лабораторная работа"
            }

            Row {
                id: squaresRow
                anchors.horizontalCenter: parent.horizontalCenter
                spacing: Theme.paddingLarge
                height: squareSize

                Rectangle {
                    id: blueSquare
                    width: squareSize
                    height: squareSize
                    y: 0
                    color: "blue"
                    opacity: squareOpacity

                    Label {
                        anchors.centerIn: parent
                        text: blueText
                        color: "black"
                        font.pixelSize: Math.max(squareSize * 0.3, 12)
                    }
                }

                Rectangle {
                    id: redSquare
                    width: squareSize
                    height: squareSize
                    y: 0
                    color: "red"
                    opacity: squareOpacity
                }

                Rectangle {
                    id: greenSquare
                    width: squareSize
                    height: squareSize
                    y: 0
                    color: "green"
                    opacity: squareOpacity

                    property real startX: 0

```

```

        property real startY: 0

        Component.onCompleted: {
            startX = x
            startY = y
        }

        SequentialAnimation {
            id: moveAnimation
            running: false
            NumberAnimation { target: greenSquare; property: "y";
to: greenSquare.startY + 100; duration: 300; easing.type: Easing.InOutQuad }
            NumberAnimation { target: greenSquare; property: "x";
to: greenSquare.startX + 100; duration: 300; easing.type: Easing.InOutQuad }
            NumberAnimation { target: greenSquare; property: "y";
to: greenSquare.startY; duration: 300; easing.type: Easing.InOutQuad }
            NumberAnimation { target: greenSquare; property: "x";
to: greenSquare.startX; duration: 300; easing.type: Easing.InOutQuad }
        }
    }

    Label {
        text: "Прозрачность квадратов: " + Math.round(slider.value *
100) + "%"

        font.pixelSize: Theme.fontSizeLarge
        color: Theme.highlightColor
        anchors.horizontalCenter: parent.horizontalCenter
    }

    Slider {
        id: slider
        width: parent.width - 2 * Theme.horizontalPageMargin
        anchors.horizontalCenter: parent.horizontalCenter
        minimumValue: 0
        maximumValue: 1
        value: 1.0
        stepSize: 0.01
        onValueChanged: squareOpacity = value
    }

    Button {
        text: "Двигать"
        anchors.horizontalCenter: parent.horizontalCenter
        onClicked: {
            soundEffect.play()
            moveAnimation.start()
        }
    }

    TextField {
        id: textInput
        width: parent.width
        anchors.horizontalCenter: parent.horizontalCenter
        placeholderText: "Введите текст"
        EnterKey.enabled: text.length > 0
        EnterKey.onClicked: blueText = text
    }

    Button {
        text: "УСТАНОВИТЬ ТЕКСТ"
    }

```

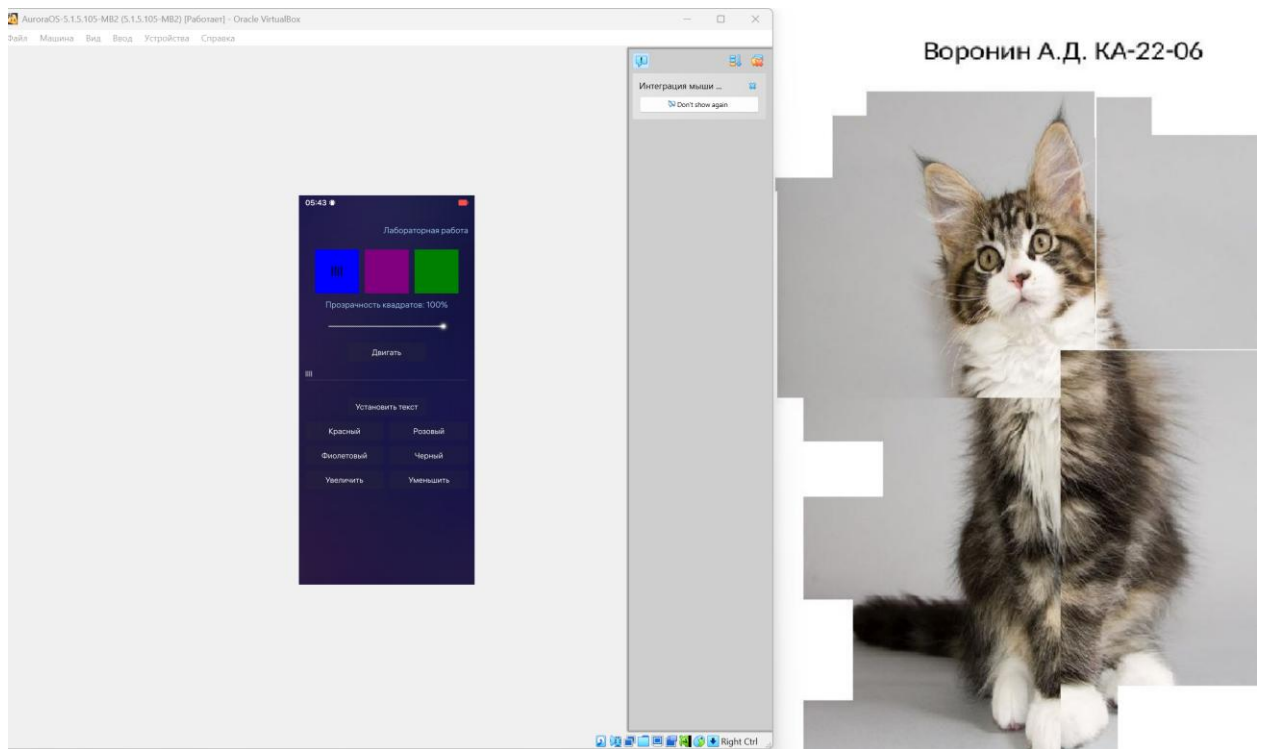



Рисунок 1 – Рабочее приложение

ЗАКЛЮЧЕНИЕ

В заключение, подводя итог проделанной работе, все поставленные в начале цели и задания данной лабораторной работы были достигнуты в полном объеме.

Контрольные вопросы

1. Способы позиционирования?

Основные способы позиционирования следующие:

Координаты (x, y) - явное указание позиции элемента относительно родителя по осям X и Y.

anchors - система привязок элемента к другим элементам или к родителю. Позволяет задавать расположение относительно сторон (left, right, top, bottom), центров (horizontalCenter, verticalCenter), а также использовать удобные свойства anchors.fill (заполнение контейнера) и anchors.centerIn (выравнивание по центру).

Layouts - специальные контейнеры для автоматического размещения элементов:

- Row - размещение в строку
- Column - размещение в колонку
- Grid - сетка
- Flow - потоковое размещение элементов с переносом

2. Перечислите основные компоненты Qt Quick и их назначение.

• Item - базовый невидимый контейнер для визуальных элементов, предоставляет общие свойства позиционирования и размеров.

• Rectangle - прямоугольник с заливкой и границей, часто используется как фон или базовый элемент интерфейса.

- Text - отображение текста с настройками шрифта и стиля.
- Image - вывод изображений.
- MouseArea - невидимая зона для обработки событий мыши и касаний.
- Button - кнопка для взаимодействия с пользователем.
- BusyIndicator - индикатор загрузки или выполнения процесса.
- ApplicationWindow - окно верхнего уровня с заголовком и панелью инструментов.

- Layouts - контейнеры для упорядоченного размещения элементов в строку, колонку или сетку.

3. Как реализована анимация движения элемента?

Анимация движения зелёного квадрата реализована через `SequentialAnimation` и несколько `NumberAnimation`, которые изменяют свойства `x` и `y` квадрата последовательно.