

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: программной инженерии

Направление подготовки: «Фундаментальная информатика и
информационные технологии»

ОТЧЕТ

по лабораторной работе №1

Тема:

«Управление с итеративным обучением (УИО)»

Выполнил:

студент группы 381906-3

Михеев И. А.

Руководитель:

младший научный сотрудник

Карчков Денис Александрович

г. Нижний Новгород

2022

Содержание

Содержание	2
Цель работы	3
Постановка задачи	3
Методы решения поставленной задачи	4
Руководство пользователя	5
Приложение	6

Цель работы

Целью данной работы является: освоить процесс создания нового проекта, изучить его структуру, научиться запускать приложение. Для разработки приложения используется фреймворк для кроссплатформенной разработки мобильных приложений Xamarin. Язык программирования C#.

Постановка задачи

- 1) Создать новый проект со стандартной заготовкой приложения.
- 2) Посмотреть содержимое вкладок Qt creator (Welcome, Edit, Debug, Projects, Sailfish OS, Help). Выяснить назначение каждой из них.
- 3) Изучить структуру созданного проекта (каталоги, расположение файлов). Выяснить соглашения по размещению файлов для проектов Qt для Sailfish OS.
- 4) Изучить содержимое *.pro файла проекта. Выяснить назначение разделов файла. Документация по файлам проекта доступна по адресу <http://doc.qt.io/qt-5/qmake-project-files.html>
- 5) Изучить содержимое *.qml файлов. Выяснить назначение элементов, используемых в файле, с помощью интерактивной справки (нажать на элемент в файле, затем вызвать справку нажатием на кнопку F1 на клавиатуре).
- 6) Запустить эмулятор, освоить принципы навигации в Sailfish OS, посмотреть возможности и настройки эмулятора. Научиться осуществлять навигацию на устройстве Sailfish OS, узнать возможности настроек устройства (приложение Settings).
- 7) Собрать и запустить заготовку приложения на эмуляторе
- 8) Используя материал слайдов 35 и 39 из лекции, изменить приложение таким образом, чтобы оно содержало одно текстовое поле со счетчиком и одну кнопку, позволяющую увеличивать значение счетчика на 1. Размещению элементов на экране внимания можно не уделять.
- 9) Собрать и запустить приложение на эмуляторе. Убедиться в правильности его работы.

Методы решения поставленной задачи

activity_main.xml

Добавляем два элемента - TextView и Button на LinearLayout макет приложения.

Т.к. Размещению элементов на экране внимания можно не уделять пропускаем определение свойств элементов, только изменим свойство text у элементов на “0” у TextView и “Click Me” у Button.

MainActivity.cs

Заведем счетчик:

```
int count = 1;
```

Для того чтобы связать TextView и Button используем свойство объекта Tag:

```
button.Tag = textView;
```

Напишем функцию button_Click, которая будет увеличивать значение счетчика и обновлять значение поля TextView:

```
private void button_Click(object sender, EventArgs e)
{
    ((sender as Button).Tag as TextView).Text = string.Format("{0} clicks!", count++);
}
```

Заметим, что свойство text может принимать только строковые значения, поэтому преобразуем int count в string.

Для обработки нажатия Button используем метод EventHandler и передадим в нее нашу функцию-инкремент button_Click:

```
button.Click += new System.EventHandler(this.button_Click);
```

Таким образом, при каждом нажатии на кнопку, счетчик будет увеличиваться на единицу.

Руководство пользователя

- 1) При запуске приложения пользователь увидит кнопку и текстовое поле с начальным значением 0. (рис.1)
- 2) Каждое нажатие на кнопку увеличивает значение текстового поля на 1. (рис.2)

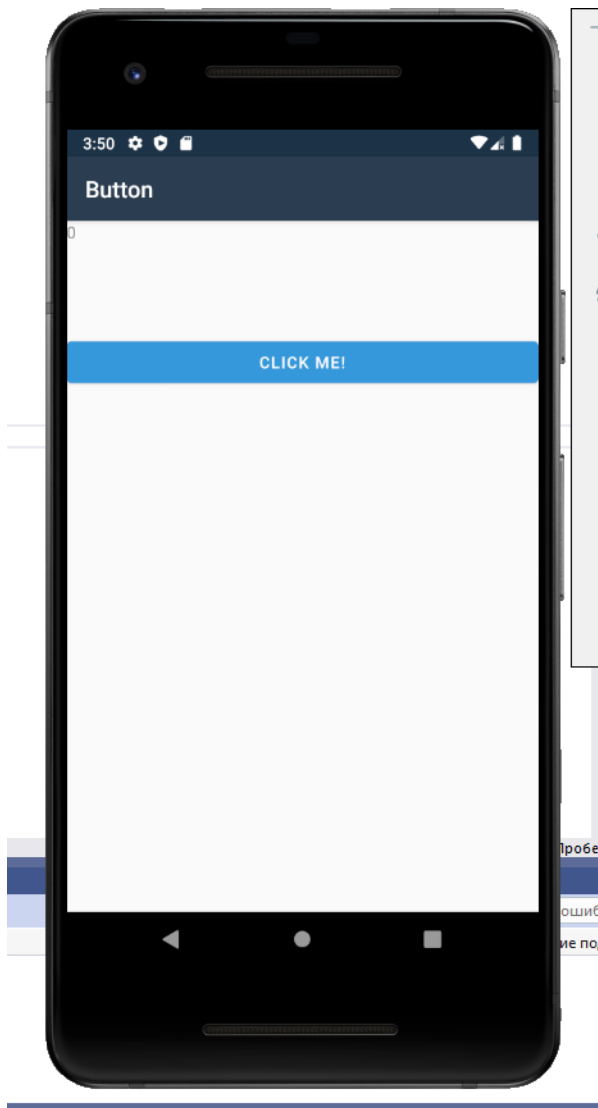


рис.1

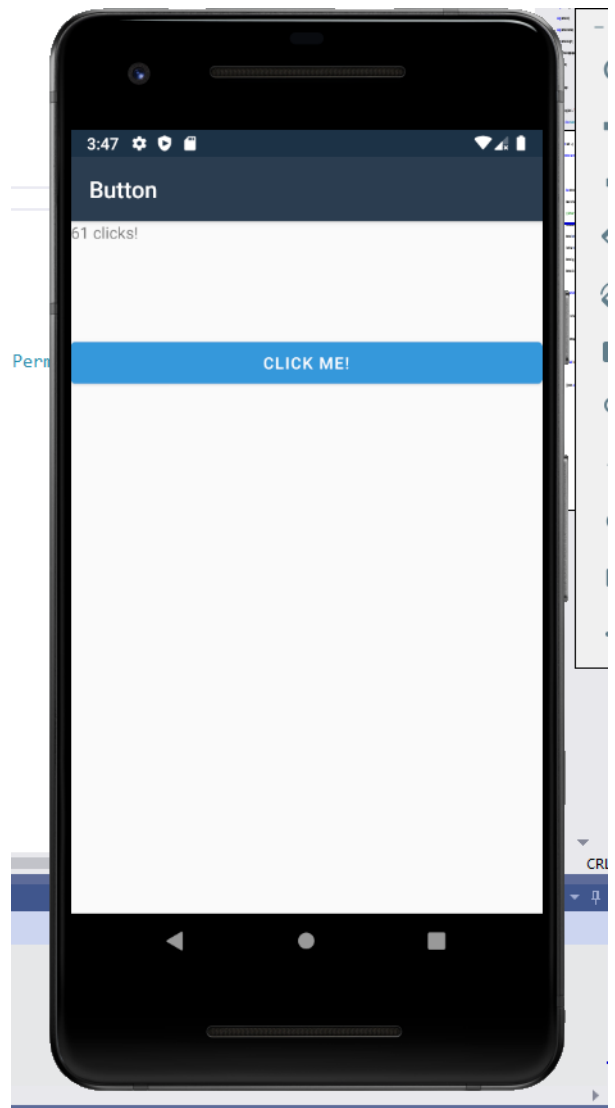


рис.2

Приложение

activity_main.xml - разметка

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:minWidth="25px"
    android:minHeight="25px"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/linearLayout1">
    <TextView
        android:text="0"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView1" />
    <Button
        android:text="Click Me!"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:layout_marginTop="80.0dp" />
</LinearLayout>
```

MainActivity.cs - код

```
using Android.App;
using Android.OS;
using Android.Runtime;
using Android.Widget;
using AndroidX.AppCompat.App;
using System;
```

```

namespace App1
{
    [Activity(Label = "@string/app_name", Theme = "@style/AppTheme", MainLauncher = true)]
    public class MainActivity : AppCompatActivity
    {
        int count = 1;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            Xamarin.Essentials.Platform.Init(this, savedInstanceState);
            SetContentView(Resource.Layout.activity_main);
            Button button = FindViewById<Button>(Resource.Id.button1);
            TextView textview = FindViewById<TextView>(Resource.Id.textView1);
            button.Tag = textview;
            button.Click += new System.EventHandler(this.button_Click);
        }

        public override void OnRequestPermissionsResult(int requestCode, string[] permissions,
        [GeneratedEnum] Android.Content.PM.Permission[] grantResults)
        {
            Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions,
            grantResults);

            base.OnRequestPermissionsResult(requestCode, permissions, grantResults);
        }

        private void button_Click(object sender, EventArgs e)
        {
            ((sender as Button).Tag as TextView).Text = string.Format("{0} clicks!", count++);
        }
    }
}

```