Тема: Використання основних елементів для створення інтерфейсу  
мобільних додатків. Використання намірів (явних та неявних). Визначення  
місцеперебування за допомогою GPS датчика. Робота з Notifications.  
Використання вібрації. Робота з текстовими файлами (збереження та  
завантаження даних).  
Мета роботи: навчитися використовувати елементи інтерфейсу (Views) для  
створення інтерфейсу користувача. На практиці засвоїти роботу з намірами  
операційної системи андроїд. Навчитися використовувати датчики  
місцезнаходження та отримувати координати з них. Опанувати  
використання повідомлень (Notifications), використання апаратних клавіш  
мобільного пристрою.

Індивідуальне завдання

Розрахувати значення функції y = а ∙ exp (-b), якщо a<0, b>0; y = а ∙ tg (b),  
якщо a>0, b<0. Результат вивести за допомогою Notification.

Main.axml

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<LinearLayout

android:id="@+id/Mylay"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:text="Входные параметры:"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/textView1" />

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Введите А"

android:id="@+id/paramA" />

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Введите В"

android:id="@+id/paramB" />

<Button

android:text="Расчет значения функции"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/buttoncount" />

<Button

android:text="Сохранить в файл"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/buttonsave" />

<Button

android:text="Загрузить из файла"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/buttonload" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Выбор цвета фона" />

<RadioGroup

android:id="@+id/radioGroup"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<RadioButton

android:id="@+id/RadioButton1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Серый" />

<RadioButton

android:id="@+id/RadioButton2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Зеленый" />

</RadioGroup>

<Button

android:id="@+id/colorbtn"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Установить цвет фона" />

<Button

android:id="@+id/backgroundbtn"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Установить рисунок как фон" />

<Button

android:id="@+id/intentbtn"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Перейти к Intents" />

</LinearLayout>

</ScrollView>

Intentslayout.axml

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<LinearLayout

android:id="@+id/secondlayout"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Длительность вибрации" />

<SeekBar

android:id="@+id/mybar"

android:max="20"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/bvibro"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Вибрация" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="http (адрес)"

android:id="@+id/textView1" />

<EditText

android:id="@+id/httpaddress"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="http://www.microsoft.com" />

<Button

android:id="@+id/browserbutton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Открыть в браузере" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Телефонный номер:" />

<EditText

android:id="@+id/phonenumber"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:inputType="phone"

android:text="0955380417" />

<Button

android:id="@+id/bcallphone"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Позвонить по номеру" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Рисование маршрута (приложение GoogleMaps)"

android:id="@+id/textView2" />

<EditText

android:id="@+id/tto"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:hint="Конечная точка"

android:text="Краматорск ж/д вокзал" />

<Button

android:id="@+id/bgmaps"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Маршрут в GoogleMaps" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="GPS координаты (lat,long)"

android:id="@+id/textView2" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text=""

android:id="@+id/tlat" />

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text=""

android:id="@+id/tlong" />

</LinearLayout>

</ScrollView>

MainActivity.cs

using Android.App;

using Android.Widget;

using Android.OS;

using Java.IO;

using Android.Content;

using Android.Graphics;

using Environment = Android.OS.Environment;

using Android.Runtime;

using Android.Views;

// Разрешения для текущего приложения

[assembly: Application(Icon = "@mipmap/icon")]

[assembly: UsesPermission(Android.Manifest.Permission.WriteExternalStorage)]

[assembly: UsesPermission(Android.Manifest.Permission.ReadExternalStorage)]

[assembly: UsesPermission(Android.Manifest.Permission.Internet)]

[assembly: UsesPermission(Android.Manifest.Permission.AccessFineLocation)]

[assembly: UsesPermission(Android.Manifest.Permission.AccessCoarseLocation)]

[assembly: UsesPermission(Android.Manifest.Permission.CallPhone)]

[assembly: UsesPermission(Android.Manifest.Permission.CallPrivileged)]

[assembly: UsesPermission(Android.Manifest.Permission.Vibrate)]

namespace Lab1

{

[Activity(Label = "Lab1", MainLauncher = true)]

public class MainActivity : Activity

{

// Элементы интерфейса

EditText A, B;

Button bcounting, bsave, bload, backcolor;

LinearLayout mainLay;

RadioButton r1, r2;

Button backPicture, showIntentsActivity;

double a, b;

// Идентификатор уведомления

int NOTIFY\_ID = 101;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

// Привязываем Main Layout к классу MainActivity

SetContentView(Resource.Layout.Main);

// Находим элементы интерфейса из MainLayout.xml по Id (кнопки, поля ....)

A = (EditText)FindViewById(Resource.Id.paramA);

B = (EditText)FindViewById(Resource.Id.paramB);

bcounting = (Button)FindViewById(Resource.Id.buttoncount);

bsave = (Button)FindViewById(Resource.Id.buttonsave);

bload = (Button)FindViewById(Resource.Id.buttonload);

backcolor = (Button)FindViewById(Resource.Id.colorbtn);

mainLay = (LinearLayout)FindViewById(Resource.Id.Mylay);

r1 = (RadioButton)FindViewById(Resource.Id.RadioButton1);

r2 = (RadioButton)FindViewById(Resource.Id.RadioButton2);

backPicture = (Button)FindViewById(Resource.Id.backgroundbtn);

showIntentsActivity = (Button)FindViewById(Resource.Id.intentbtn);

// Первый переключатель нажат всегда

r1.Checked = true;

// Заполним активность цветом

mainLay.SetBackgroundColor(Color.PaleGreen);

// Привязываем обработчик OnClick к bcounting (Кнопка расчета)

bcounting.Click += delegate {

double FunResult = 0;

// Считываем параметры из полей

try

{

a = double.Parse(A.Text);

}

catch (System.FormatException)

{

// Показываем уведомление об ошибке

Toast.MakeText(this, "Значение А неверно!", ToastLength.Short).Show();

return;

}

try

{

b = double.Parse(B.Text);

}

catch (System.FormatException)

{

Toast.MakeText(this, "Значение B неверно!", ToastLength.Short).Show();

return;

}

// Рассчитываем значение функции и выводим результат в Toast

FunResult = MyFunction(a, b);

// Context текущего приложения

Context CurrentContext = ApplicationContext;

// Создаем pending Intent для вызова IntentsActivity при щелчке по уведомлению

// Pending Intent - указывает действие, которое необходимо предпринять в будущем.

// Он позволяет передать намерение другому приложению и позволить этому

// приложению выполнять этот намерение, как если бы оно имело такие же разрешения как и ваше приложение.

// Intents - это стандартный механизм обмена сообщениями в Android, который выражает намерение пользователей

// выполнять определенную работу. Они позволяют вам взаимодействовать с другими компонентами,

// определенными вами или операционной системой Android.

Intent notificationIntent = new Intent(CurrentContext, typeof(IntentsActivity));

PendingIntent contentIntent = PendingIntent.GetActivity(CurrentContext, 0, notificationIntent,

PendingIntentFlags.UpdateCurrent);

// Инициализация уведомления (notification)

Notification.Builder builder = new Notification.Builder(CurrentContext)

.SetSmallIcon(Resource.Mipmap.Icon)

.SetLargeIcon(BitmapFactory.DecodeResource(CurrentContext.Resources, Resource.Mipmap.Icon))

.SetTicker("Результат расчета!")

.SetWhen(Java.Lang.JavaSystem.CurrentTimeMillis())

.SetContentTitle("MyNotification")

.SetContentText("F(A,B)=" + FunResult.ToString())

.SetContentIntent(contentIntent)

.SetDefaults(NotificationDefaults.Sound | NotificationDefaults.Vibrate)

.SetAutoCancel(true);

// Применяем настройки уведомления

Notification notification = builder.Build();

// Отображаем уведомление с помощью NotificationManager

// Инициализация NotificationManager

NotificationManager myNotificationManager = (NotificationManager)CurrentContext.GetSystemService(Context.NotificationService);

// собственно отображение)))

myNotificationManager.Notify(NOTIFY\_ID, notification);

};

// Обработчик загрузки данных из файла

bload.Click += delegate

{

try

{

// Инициализируем путь файла на внешней карте памяти

File path = Environment.ExternalStorageDirectory;

path = new File(path.AbsolutePath + "/" + "MyData.txt");

// Если нет файла

if (!path.Exists())

{

Toast.MakeText(this, "Не могу открыть файл! " + "\n" + "Файл не существует! (;", ToastLength.Long).Show();

return;

}

BufferedReader br = new BufferedReader(new FileReader(path));

System.String[] pars;

pars = new System.String[2];

// Читаем строки из файла

for (int i = 0; i < 2; i++)

{

pars[i] = br.ReadLine();

}

A.Text = pars[0];

B.Text = pars[1];

Toast.MakeText(this, "Файл загружен! " + "\n" + ")))", ToastLength.Long).Show();

}

catch (IOError)

{

Toast.MakeText(this, "Ошибка ввода вывода", ToastLength.Long).Show();

}

};

// Обработчик сохранения данных в файл

bsave.Click += delegate

{

File path = Environment.ExternalStorageDirectory;

path = new File(path.AbsolutePath + "/" + "MyData.txt");

Toast.MakeText(this, "Файл сохранен!))) "+path.ToString(), ToastLength.Long).Show();

System.String[] pars;

pars = new System.String[2];

pars[0] = A.Text;

pars[1] = B.Text;

// Открываем поток для записи

BufferedWriter bw = new BufferedWriter(new FileWriter(path));

for (int i = 0; i < 2; i++)

{

// Записываем данные в файл

bw.Write(pars[i].ToString() + "\n");

}

// Закрываем поток

bw.Close();

};

// Обработчик клавиши установки фонового цвета активности

backcolor.Click += delegate

{

// Проверяем, какой из переключателей выбран

if (r1.Checked) mainLay.SetBackgroundColor(Color.DarkGray);

else mainLay.SetBackgroundColor(Color.Green);

};

// Установка рисунка как фона главной активности

backPicture.Click += BackPicture\_Click;

// Вызов второй активности IntentsActivity

showIntentsActivity.Click += delegate

{

Intent showIntentactivity = new Intent(this, typeof(IntentsActivity));

StartActivity(showIntentactivity);

};

}

private void BackPicture\_Click(object sender, System.EventArgs e)

{

// Идентификатор ресурса в Resource.Designer.cs (Back.jpg)

mainLay.SetBackgroundResource(2130837504);

}

// Наша функция)))

public double MyFunction(double a, double b)

{

double myresult=0;

if(a<0 && b>0)

{

myresult = a \* System.Math.Exp(-b);

}

else if (a > 0 && b < 0)

{

myresult = a \* System.Math.Tan(b);

}

return myresult;

}

// Обработка аппаратных клавиш (клавиша Назад)

public override bool OnKeyDown([GeneratedEnum] Keycode keyCode, KeyEvent e)

{

switch (keyCode)

{

case Keycode.Back:

ShowMessage();

return true;

break;

}

return base.OnKeyDown(keyCode, e);

}

public override bool OnKeyUp([GeneratedEnum] Keycode keyCode, KeyEvent e)

{

switch (keyCode)

{

case Keycode.Back:

return true;

break;

}

return base.OnKeyUp(keyCode, e);

}

// Вызов диалога выхода из приложения

public void ShowMessage()

{

var builder = new AlertDialog.Builder(this)

.SetPositiveButton("Да", new System.EventHandler<Android.Content.DialogClickEventArgs>((o, a) =>

{

Finish();

}))

.SetNegativeButton("Нет", new System.EventHandler<Android.Content.DialogClickEventArgs>((o, a) => { }))

.SetTitle("Подтверждение")

.SetMessage("Хотите выйти?")

.Show();

}

protected override void OnDestroy()

{

base.OnDestroy();

Toast.MakeText(this,"Пока!))) Еще увидимся)))",ToastLength.Short).Show();

}

}

}

IntentsActivity.cs

using System;

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Views;

using Android.Widget;

using Android.Locations;

namespace Lab1

{

[Activity(Label = "IntentsActivity")]

public class IntentsActivity : Activity, ILocationListener

{

Button vibrobtn, btnbrowser, btncall, btnroute;

EditText secondpoint, phonenumber, addresshttp;

SeekBar durationbar;

TextView tlatitude, tlongitude;

LinearLayout MyLayout;

// Коордианты текущего местоположения

double mylatitude, mylongitude;

LocationManager service;

bool enabled;

string provider;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

//

SetContentView(Resource.Layout.Intentslayout);

// Связываем Intentslayout с бизнес-логикой приложения

vibrobtn = (Button)FindViewById(Resource.Id.bvibro);

btnbrowser = (Button)FindViewById(Resource.Id.browserbutton);

btncall = (Button)FindViewById(Resource.Id.bcallphone);

btnroute = (Button)FindViewById(Resource.Id.bgmaps);

secondpoint = (EditText)FindViewById(Resource.Id.tto);

phonenumber = (EditText)FindViewById(Resource.Id.phonenumber);

addresshttp = (EditText)FindViewById(Resource.Id.httpaddress);

durationbar = (SeekBar)FindViewById(Resource.Id.mybar);

tlatitude = (TextView)FindViewById(Resource.Id.tlat);

tlongitude = (TextView)FindViewById(Resource.Id.tlong);

MyLayout = (LinearLayout)FindViewById(Resource.Id.secondlayout);

// Максимальное значение SeekBar

durationbar.Max = 1000;

durationbar.Progress = 500;

// Присваиваем слою цвет

// Идентификатор ресурса в Resource.Designer.cs (Back.jpg)

MyLayout.SetBackgroundResource(2130837504);

btnroute.Click += delegate

{

String geoUriString = "geo:" + mylatitude.ToString() + "," + mylongitude.ToString() + "?q="+secondpoint.Text+"&z=4";

Android.Net.Uri geoUri = Android.Net.Uri.Parse(geoUriString);

var mapIntent = new Intent(Intent.ActionView);

mapIntent.SetData(geoUri);

StartActivity(mapIntent);

};

vibrobtn.Click += delegate

{

// Инициализация вибровызова

Vibrator mvibrator;

mvibrator = (Vibrator)GetSystemService(VibratorService);

// Вибрация вибромотора определенный интервал времени

mvibrator.Vibrate(durationbar.Progress);

};

// Звоним по номеру

btncall.Click += delegate

{

var intent = new Intent(Intent.ActionCall);

intent.SetData(Android.Net.Uri.Parse("tel:" + phonenumber.Text.ToString()));

StartActivity(intent);

};

// Открываем веб-страницу в браузере

btnbrowser.Click += delegate

{

var openlink = new Intent(Intent.ActionView);

openlink.SetData(Android.Net.Uri.Parse(addresshttp.Text.ToString()));

StartActivity(openlink);

};

// Настройка и включение GPS

service = (LocationManager)GetSystemService(Context.LocationService);

enabled = service.IsProviderEnabled(LocationManager.NetworkProvider);

if (enabled)

{

Toast.MakeText(this, "GPS доступен", ToastLength.Long).Show();

}

else

{

Toast.MakeText(this, "GPS не доступен", ToastLength.Long).Show();

return;

}

var criteria = new Criteria { Accuracy = Accuracy.Fine };

provider = service.GetBestProvider(criteria, false);

var location = service.GetLastKnownLocation(provider);

if (null != location)

{

Toast.MakeText(this, "Используемый провайдер '" + provider + "'", ToastLength.Long).Show();

OnLocationChanged(location);

}

else

{

Toast.MakeText(this, "Местоположение не определено", ToastLength.Long).Show();

}

}

protected override void OnResume()

{

base.OnResume();

if (enabled) service.RequestLocationUpdates(

provider,

1000, // Минимальное время между обновлениями

1, // Минимальное расстяние в м

this);

}

protected override void OnPause()

{

base.OnPause();

if (enabled) service.RemoveUpdates(this);

}

public void OnProviderEnabled(string provider)

{

}

public void OnProviderDisabled(string provider)

{

}

public void OnStatusChanged(string provider, Availability status, Bundle extras)

{

}

// Определяем текущие координаты

public void OnLocationChanged(Android.Locations.Location location)

{

var latitude = location.Latitude;

var longtitude = location.Longitude;

mylatitude = latitude;

mylongitude = longtitude;

tlatitude.Text = mylatitude.ToString();

tlongitude.Text = mylongitude.ToString();

}

// Обработка аппаратных клавиш (клавиша Назад)

public override bool OnKeyDown([GeneratedEnum] Keycode keyCode, KeyEvent e)

{

switch (keyCode)

{

case Keycode.Back:

Finish();

return true;

break;

}

return base.OnKeyDown(keyCode, e);

}

public override bool OnKeyUp([GeneratedEnum] Keycode keyCode, KeyEvent e)

{

switch (keyCode)

{

case Keycode.Back:

return true;

break;

}

return base.OnKeyUp(keyCode, e);

}

}

}





