Practical No. 3

Title: Android program based on Intents

Aim: Create an application to demonstrate use of Intent

Introduction

Android Intent is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.

It is generally used with startActivity() method to invoke activity, broadcast receivers etc.

The **dictionary meaning** of intent is *intention or purpose*. So, it can be described as the intention to do action.

The LabeledIntent is the subclass of android.content.Intent class.

Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.

Types of Android Intents

There are two types of intents in android: implicit and explicit.

1) Implicit Intent

Implicit Intent doesn't specifiy the component. In such case, intent provides information of available components provided by the system that is to be invoked.

For example, you may write the following code to view the webpage.

```
1. Intent intent=new Intent(Intent.ACTION_VIEW);
2. intent.setData(Uri.parse("http://www.javatpoint.com"));
3. startActivity(intent);
```

2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

```
1. Intent i = new Intent(getApplicationContext(), ActivityTwo.class);
2. startActivity(i);
```

Exercise - Create android application to demonstrate Intent

Implementation:
Program:

MainActivity.java

```
package com.example.btndashboard;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button button;
    private Button button2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button=findViewById(R.id.button);
        button2=findViewById(R.id.button2);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                click_for_other();


            }
        });

        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                second();
            }
        });
    }
    public void click_for_other()
    {
        Intent intent=new Intent(MainActivity.this,Activity2.class);
        startActivity(intent);
    }
    public void second()
    {
        Intent intent2=new Intent(MainActivity.this,Activity3.class);
                startActivity(intent2);
    }

}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="123dp"
        android:layout_height="49dp"
        android:text="First"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.506"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.444" />

    <Button
        android:id="@+id/button2"
        android:layout_width="119dp"
        android:layout_height="48dp"
        android:layout_marginTop="68dp"
        android:text="second"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button"
        app:layout_constraintVertical_bias="0.011" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output: