

## 实验 4：活动的生命周期

### 1. 实验名称：活动的生命周期

### 2. 实验目的

了解并掌握活动的生命周期方法，通过记录日志的方式理解活动的生命周期，同时掌握日志工具 LogCat 的使用方法。

### 3. 实验内容：

- (1) 创建 android 应用程序，并重新活动的生命周期方法；
- (2) 使用 LogCat 查看日志，通过日志信息了解活动的生命周期；

### 4. 实验步骤：

#### 任务 1：完成一个 android 应用程序的开发与运行

- (1) 新建一个项目 LabActivityLifeCycle，允许 Android Studio 帮我们自动创建活动，创建的活动名和布局名都使用默认值：MainActivity 和 activity\_main。
- (2) 主活动创建完成，还需要分别再创建两个子活动，NormalActivity 和 DialogActivity，对应的布局文件为 normal\_layout 和 dialog\_layout。
- (3) 新建 normal\_layout.xml 文件，代码如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="This is a normal activity"
        android:textSize="26dp"
    />
</LinearLayout>
```

- (4) 再新建一个 dialog\_layout.xml 文件，代码如下所示：

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="This is a dialog activity"
        android:textSize="30dp"
    />
</LinearLayout>

```

- (5) NormalActivity 继承自 AppCompatActivity, 加载 normal\_layout 布局:

```

public class NormalActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.normal_layout);
    }
}

```

- (6) DialogActivity 继承自 AppCompatActivity, 在 DialogActivity 中加载 dialog\_layout 布局:

```

public class DialogActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dialog_layout);
    }
}

```

- (7) 在 AndroidManifest.xml 的标签中注册两个活动:

```

<activity android:name=".NormalActivity" />

```

```

    <activity android:name=".DialogActivity"
    android:theme="@style/Theme.AppCompat.Dialog"></activity>

```

- (8) 修改 activity\_main.xml, 加入两个按钮, 一个用于启动 NormalActivity, 一个用于启动 DialogActivity:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <Button
        android:id="@+id/start_normal_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start NormalActivity" />
    <Button
        android:id="@+id/start_dialog_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start DialogActivity" />
</LinearLayout>

```

- (9) 修改 MainActivity 中的代码, 如下所示:

```

public class MainActivity extends AppCompatActivity {
    public static final String TAG = "MainActivity";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, "onCreate");
        setContentView(R.layout.activity_main);
        Button startNormalActivity = (Button)
        findViewById(R.id.start_normal_activity);
        Button startDialogActivity = (Button)
        findViewById(R.id.start_dialog_activity);
    }
}

```

```

        startNormalActivity.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this,
NormalActivity.class);
        startActivity(intent);
    }
});
        startDialogActivity.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this,
DialogActivity.class);
        startActivity(intent);
    }
});
    }

@Override
protected void onStart() {
    super.onStart();
    Log.d(TAG, "onStart");
}

@Override
protected void onResume() {
    super.onResume();
    Log.d(TAG, "onResume");
}

@Override
protected void onPause() {
    super.onPause();
    Log.d(TAG, "onPause");
}

```

```

@Override
protected void onStop() {
    super.onStop();
    Log.d(TAG, "onStop");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "onDestroy");
}

@Override
protected void onRestart() {
    super.onRestart();
    Log.d(TAG, "onRestart");
}
}

```

(10) 运行程序，观察 LogCat 中的日志打印信息。

- a) 点击第一个按钮，启动 NormalActivity，观察 LogCat 中的日志打印信息。
- b) 按下 Back 键返回 MainActivity，观察 LogCat 中的日志打印信息。
- c) 再点击第二个按钮，启动 DialogActivity，观察 LogCat 中的日志打印信息。
- d) 按下 Back 键返回 MainActivity，观察 LogCat 中的日志打印信息。
- e) 在 MainActivity 按下 Back 键退出程序，观察 LogCat 中的日志打印信息。

## 5. 提交要求:

- (1) 独立完成实验并撰写实验报告;
- (2) 将工程文件夹打包为压缩文件;
- (3) 将实验报告和压缩后的文件打包，文件命名方式：学号\_姓名，请使用 winrar 或 winzip 压缩，不要用快压;
- (4) 按要求及时提交到 MOOC 系统中。