个 android:theme 属性，用于给当前 Activity 指定主题，Android 系统内置有很多主题可以选择，当然我们也可以定制自己的主题，而这里的@style/Theme.AppCompat.Dialog 则毫无疑问是让 DialogActivity 使用对话框式的主题。

接下来我们修改 activity_main.xml，重新定制主 Activity 的布局，将里面的代码替换成如下内容：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/startNormalActivity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start NormalActivity" />

    <Button
        android:id="@+id/startDialogActivity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start DialogActivity" />

</LinearLayout>
```

可以看到，我们在 LinearLayout 中加入了两个按钮，一个用于启动 NormalActivity，一个用于启动 DialogActivity。

最后修改 MainActivity 中的代码，如下所示：

```
class MainActivity : AppCompatActivity() {

    private val tag = "MainActivity"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        Log.d(tag, "onCreate")
        setContentView(R.layout.activity_main)
        startNormalActivity.setOnClickListener {
            val intent = Intent(this, NormalActivity::class.java)
            startActivity(intent)
        }
        startDialogActivity.setOnClickListener {
            val intent = Intent(this, DialogActivity::class.java)
            startActivity(intent)
        }
    }

    override fun onStart() {
        super.onStart()
        Log.d(tag, "onStart")
    }

    override fun onResume() {
        super.onResume()
        Log.d(tag, "onResume")
    }
```

```
    override fun onPause() {
        super.onPause()
        Log.d(tag, "onPause")
    }

    override fun onStop() {
        super.onStop()
        Log.d(tag, "onStop")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d(tag, "onDestroy")
    }

    override fun onRestart() {
        super.onRestart()
        Log.d(tag, "onRestart")
    }

}
```

在 onCreate()方法中，我们分别为两个按钮注册了点击事件，点击第一个按钮会启动 NormalActivity，点击第二个按钮会启动 DialogActivity。然后在 Activity 的 7 个回调方法中分别打印了一句话，这样就可以通过观察日志来更直观地理解 Activity 的生命周期。

现在运行程序，效果如图 3.25 所示。

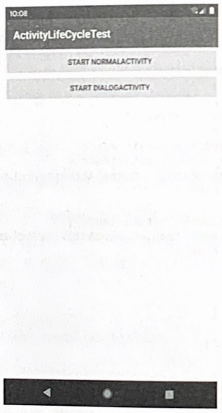

图 3.25　MainActivity 界面

这时观察 Logcat 中的打印日志，如图 3.26 所示。

图 3.26　启动程序时的打印日志

可以看到，当 MainActivity 第一次被创建时会依次执行 onCreate()、onStart()和 onResume()方法。然后点击第一个按钮，启动 NormalActivity，如图 3.27 所示。

图 3.27　NormalActivity 界面

此时的打印信息如图 3.28 所示。

图 3.28　打开 NormalActivity 时的打印日志

由于 NormalActivity 已经把 MainActivity 完全遮挡住，因此 onPause()和 onStop()方法都会得到执行。然后按下 Back 键返回 MainActivity，打印信息如图 3.29 所示。