|  |  |  |  |
| --- | --- | --- | --- |
| 实验项目名称 | Android创建和获取资源 | 指导教师 | 陈双平 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目编号 | 03 | 实验项目类型 | 验证型 | 实验地点 | C304 |

|  |  |
| --- | --- |
| 实验时间 | 2023年09月14日 10:00 |

|  |
| --- |
| **一、实验目的** |
| * 学会获得资源 * 学会获得view控件,并设置其属性. |

|  |
| --- |
| **二、实验环境** |
| Android Studio |
| **三、实验内容** |
| 在上次第一个APP的基础上找到相应的资源/代码文件.修改代码并回答以下问题.   1. 查看你的Android Studio设计模式，当前版本有几种布局？     有七种布局   1. PPT介绍了用代码创建Activity的布局及控件，不适用layout文件，如何用Java代码生成你的Hello world，请贴出代码     3.Android Studio中碰到编译错时，可以鼠标移动到所在行，按哪个快捷键寻求系统提示的解决方案。  Alt+Enter  4.如何通过资源名称获得资源的值，比如获得图片，字符串.  使用 getResources().getDrawable() 方法获取图片资源。  Drawable drawable = getResources().getDrawable(R.drawable.my\_image);  使用 getString() 方法获取字符串资源。  String str = getString(R.string.my\_string);  5. 字符串为什么不建议嵌入activity或者layout文件中?把Activity,布局,资源分开是一种什么样设计思想的体现?请调研Android中MVC开发，了解什么是MVC。  字符串不建议嵌入到Activity或Layout文件中的原因是要将UI和文本内容逻辑分离。  这种设计思想的体现是将不同的职责分离开来，提高代码的可维护性和可扩展性。  MVC是一种广泛应用于软件开发的设计模式，主要分为以下三个部分：  Model（模型）：负责处理数据逻辑和业务逻辑，保存数据并提供对外的接口。在Android开发中，例如使用数据库、网络请求等来处理数据操作。  View（视图）：负责展示数据和用户界面。在Android开发中，通常是通过布局文件描述UI界面。  Controller（控制器）：负责处理用户交互和调度逻辑，从View获取用户输入并将其传递给Model进行处理，然后根据处理结果更新View的显示。在Android开发中，通常是Activity或Fragment等。  采用MVC模式可以使代码更加模块化和可扩展，提高了代码的可维护性和可测试性。它能够将应用程序的不同方面解耦，使得每个组件的职责更加清晰，并且更容易进行单元测试和重构。  6.Android中要注意命名规范，设置你的TextView的id为text\_vciew\_hellow\_world.修改Android代码，通过id找到view中的TextView并获得对应的对象。设置其显示的文本内容,请编辑一个字符串资源"你好Android",并通过资源名称获得字符串,设置到textview中,    7.将你的修改commit，并push到远端库。  https://github.com/ENE574/MyFirstApplication.git |

**四、实验及分析**

在这个实验中，我们通过资源名称获取了字符串，并将其设置到TextView中显示。这样做的好处是可以将文本内容集中管理，提高代码的可维护性和可读性。

首先，在布局文件中给TextView设置了一个id，这样我们可以在Java代码中找到该TextView对应的对象。然后，在按钮的点击事件中，我们通过getResources().getString()方法获取了名为"hello\_android"的字符串资源的值，并将其设置到TextView的setText()方法中，从而实现了文本内容的更新。

**五、实验总结**

这个实验教会了我如何通过资源名称获得字符串，并使用它来设置TextView的文本内容。通过将文本放在资源文件中，我可以方便地进行多语言适配、统一管理文本内容，并且更易于维护和修改。

此外，我还学习了如何通过id找到布局文件中的控件，并操作其属性。这是Android开发中常见的操作，可以访问和修改控件的各种属性以实现界面的定制和交互。

我通过学习如何获取资源，并使其与视图控件关联起来，可以更灵活地管理文本内容，并实现更具交互性的用户界面设计。