|  |  |  |  |
| --- | --- | --- | --- |
| 实验项目名称 | 事件响应 | 指导教师 | 陈双平 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目编号 | 04 | 实验项目类型 | 验证型 | 实验地点 | C304 |

|  |  |
| --- | --- |
| 实验时间 | 2021年09月21日 10:00 |

|  |
| --- |
| **一、实验目的** |
| * 理解Android中事件的响应 * 理解Android中三种设置事件的方式 |

|  |
| --- |
| **二、实验环境** |
| Android Studio |
| **三、实验内容** |
| 在你上次实验的基础上，修改布局文件。在ConstraintLayout中添加2个TextView,这两个TextView 在一排，分别居中。其中一个的Text为“Hello”，另一个的Text为"JNU",这两个TextView下方放一个Button,按钮文本为"change text"，每次点击这个按钮，则交换两个TextView中的文本,同时分别用Toast和AlertDialog显示“交换成功”。请注意Android编程的命名规范，修改代码后commit，并push你的代码。完成后回答以下问题：  https://github.com/ENE574/MyFirstApplication.git  1.本实验中的布局，可不可以用LinearLayout实现类似效果。  可以，设置android:orientation="horizontal"属性使得两个TextView在一行中排列，使用android:gravity="center"属性使得两个TextView居中显示。  2.ConstraintLayout和RelativeLayout有什么区别  ConstraintLayout是一种更强大和灵活的布局容器。它利用约束来定义视图之间的相对位置关系，可以实现复杂的布局需求。而RelativeLayout则相对简单，通过控制视图相对于父容器或其他视图的相对位置来布局。  ConstraintLayout具有更好的性能优化能力。相比于RelativeLayout，使用ConstraintLayout可以减少层级嵌套，提高布局性能。当布局复杂、层级嵌套较多时，使用ConstraintLayout会更高效。  ConstraintLayout提供了可视化的编辑器，使得布局编辑更加直观和方便。您可以在可视化编辑器中拖动和调整视图的位置和约束关系。而RelativeLayout则需要手动设置控件的相对位置属性，相对繁琐一些。  ConstraintLayout的更新支持更好。在Android Studio的布局编辑器中，当您修改布局时，ConstraintLayout会智能地自动计算并调整视图之间的约束关系，以适应变化。而RelativeLayout可能需要手动调整视图的位置属性。  3.至少用两种方式设置Button的事件，并阐述下四种设置事件的方式各有什么优缺点，会不会相互覆盖，比如设置了第一种再设置第四种，会不会两个函数都执行？  使用匿名内部类：  优点：简单直接，适用于处理简单的按钮点击事件。  缺点：每次都需要创建一个新的匿名内部类对象，可能导致代码冗长。  使用Lambda表达式：  优点：简洁，可以减少代码量。  缺点：只有在支持Java 8及更高版本的设备上才能使用。  在Activity中实现View.OnClickListener接口：  优点：可以集中处理多个按钮的点击事件，逻辑清晰。  缺点：在同一个Activity中，如果一个按钮设置了OnClickListener，另一个按钮又通过android:onClick属性指定了点击事件，会导致这两种方式都执行。这可能会导致逻辑混乱和重复执行。  在布局文件中使用android:onClick属性：  优点：直接在布局文件中指定点击事件处理方法，方便快捷。  缺点：只能通过方法名来指定点击事件处理方法，如果涉及到复杂逻辑或需要多个按钮共享相同的点击事件处理逻辑时，不太方便。  相互覆盖的情况：  当你为一个Button设置多个点击事件处理方法时，只有最后一个设置的方法会被执行，之前设置的方法都会被覆盖。 |

**四、实验及分析**

1.在ConstraintLayout中添加了两个TextView和一个Button，实现了“Hello”和“JNU”两个TextView在一排并居中显示的效果。

2.每次点击按钮时，通过交换两个TextView的文本，实现了文本内容的互换。

3.使用Toast可以在屏幕底部以简洁的方式显示交换成功的提示信息。

4.使用AlertDialog可以以弹窗的形式显示交换成功的提示信息，给用户更直观的反馈。

5.在AlertDialog中设置了一个“确定”按钮，供用户关闭弹窗。

**五、实验总结**

1.实验结果表明，在ConstraintLayout中可以方便地进行布局设计，并使用TextView和Button等控件实现相应功能。

2.通过设置点击事件处理方法，可以实现对按钮点击事件的响应。

3.可以使用Toast来在屏幕底部显示简洁的提示信息，也可以使用AlertDialog来展示更直观的弹窗式的提示信息。

4.这个实验可以验证界面上控件之间的互动效果，并展示了一种简单的用户反馈机制。