|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| 1. 理解Android中事件的响应 2. 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：  1.本实验中的布局，可不可以用LinearLayout实现类似效果。  2.ConstraintLayout和RelativeLayout有什么区别  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？ |

# 四、实验及分析

已修改代码并push到远程仓库中：<https://github.com/qxf-72/MyFirstApplication>

## 问题一：

本实验中的布局，可以用LinearLayout实现类似效果。具体方法如下：

1. 首先加入一个垂直的LinearLayout。
2. 然后再垂直的LinearLayout中加入两个水平的LinearLayout。
3. 其中一个水平的LinearLayout加入两个textview，另一个加入button。

这样就实现了类似的效果。

## 问题二：

两种布局的主要区别如下：

1. 依赖关系管理：
2. ConstraintLayout通过约束关系（constraints）来定义子视图之间的相对位置。这种方式更加灵活，可以创建复杂的布局。
3. RelativeLayout则依赖于子视图之间的相对位置，通常是相对于父视图或其他子视图的ID。在处理复杂的布局时，ConstraintLayout通常更容易使用。
4. 性能：
5. ConstraintLayout可以通过优化约束关系来减少视图层次的深度，从而提高渲染性能。这对于复杂的布局特别有用。
6. RelativeLayout通常需要更多的视图层次，因为它是基于视图之间的相对位置，可能导致性能较差，特别是在包含大量子视图的情况下。
7. 兼容性：
8. ConstraintLayout从Android Studio 2.3及更高版本开始引入，应用需要兼容较旧的Android版本时，可能会出现不兼容的现象。
9. RelativeLayout在较旧的Android版本中也可用，因此在这些版本上更容易实现兼容性。

## 问题三：

设置Button的事件的方式主要有：

1. **匿名内部类**：简单，适用于处理较小的事件逻辑。可以在一个按钮上设置多个不同的点击事件处理。但是匿名内部类可能会导致代码变得冗长，特别是当有多个按钮需要处理时。如果需要在多个地方重复使用相同的事件处理逻辑，会导致代码重复。
2. **OnClickListener类**：可以将点击事件处理逻辑封装到单独的类中，提高代码的可维护性和重用性。可以在多个按钮上使用相同的事件处理类。但是需要创建新的类，可能增加项目中的类的数量。
3. **Lambda表达式**：简洁，减少了冗余代码。适用于处理简单的点击事件逻辑。但是对于较复杂的事件处理，Lambda表达式可能不够灵活。
4. **XML布局中设置onClick属性**：可以在XML布局中直接指定按钮的点击事件处理方法，减少了Java代码的复杂性。适用于简单的点击事件。但是无法在代码中动态更改点击事件处理，因为事件处理方法是在XML中静态设置的。

# 五、实验总结

本次实验中，通过编写简单的交换两个textview中string的demo，并回答相关拓展问题，本人对于Android Studio不同布局的特性有了进一步了解，同时还初步掌握了让button产生事件响应的多种方法，并对这几种方法的优缺点有了一定的了解。