|  |  |  |  |
| --- | --- | --- | --- |
| 实验项目名称 | Android Studio 调试APP及Demo的获取 | 指导教师 | 陈双平 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目编号 | 05 | 实验项目类型 | 验证型 | 实验地点 | C304 |

|  |  |
| --- | --- |
| 实验时间 | 2023年9月28日 10:00 |

|  |
| --- |
| **一、实验目的** |
| * 学会用stackoverflow等网站查找错误 * 学会用debug和logcat输出日志调试程序 * 学会从网上下载demo |

|  |
| --- |
| **二、实验环境** |
| Android Studio |
| **三、实验内容** |
| 调试[程序](http://www.jiaozuoye.com/joj/upload/download/94353)（上课已发，群里也有）。请描述程序中一共有几个错误，你是如何修正的。  以下问题请回答：  1. 1.Android程序Activity类的代码运行的次序和代码的先后顺序有关吗,请说明Activity类中代码的执行顺序？  在Android中，Activity类中代码的运行次序和代码的先后顺序是有关系的。Android系统会按照一定的生命周期顺序来调用Activity的生命周期方法，并在这些方法中执行相应的代码。  Activity 类中代码的执行顺序：  onCreate()：这是 Activity 的生命周期方法之一，在 Activity 创建时被调用。在该方法中，您可以进行一些初始化操作，如设置布局、绑定视图、初始化变量等。  onStart()：在 onCreate() 方法执行完毕后，紧接着调用 onStart() 方法。在 onStart() 方法中，Activity 进入可见状态，但用户仍无法与之进行交互。您可以在此方法中执行一些启动相关的操作。  onResume()：在 onStart() 方法执行完毕后，紧接着调用 onResume() 方法。在 onResume() 方法中，Activity 进入活动状态，并与用户进行交互。通常，在这里注册广播接收器、启动动画或恢复暂停的操作。  onPause()：当用户离开当前 Activity 或者有其他 Activity 起来覆盖当前 Activity 时，会调用 onPause() 方法。在 onPause() 方法中，您可以暂停正在进行的操作、保存临时数据或释放资源。需要注意的是，此方法应尽快完成，以避免阻塞其他 Activity。  onStop()：如果 Activity 不再可见，例如退到后台或者启动了别的 Activity，则调用 onStop() 方法。在 onStop() 方法中，您可以释放占用的系统资源或执行一些清理操作。  onRestart()：如果 Activity 在 onStop() 后又重新返回前台，会调用 onRestart() 方法。在该方法中，您可以执行一些与重启相关的操作。  onDestroy()：在 Activity 被销毁前，会调用 onDestroy() 方法。在 onDestroy() 方法中，您可以释放所有资源、取消注册广播接收器或其他相关操作。  2.成员变量的声明是不是一定要放在构造函数前面，其他函数放在后面？  成员变量的声明可以放在构造函数之前或之后都是合法的，这并不会影响其作用域或可见性。成员变量的声明通常放在类的顶部，无论是在构造函数前还是后，这是为了提高代码的可读性和维护性，方便开发者查看类的属性。  3.Java中比较对象是否相等要用什么运算符，比较对象的值是否相等要用什么运算符？  比较对象是否相等需要使用equals() 方法，比较对象的值是否相等需要使用相等运算符（==）。  4.Log是不是只有调试时才有输出？最终的代码包含Log代码会不会影响效率？  Log 并不仅限于调试时才有输出。Log 可以用于记录应用程序的运行状态、错误信息等，它可以在调试和生产环境中都起到重要的作用。关于 Log 代码对效率的影响，需要根据具体的实现方式和日志级别来评估。一般来说，对于生产环境，我们可以使用合适的日志级别（如警告级别或以上）来限制日志的输出，避免过多的日志信息对性能造成不必要的影响。  5.Android的调试模式和正常运行模式有什么区别？  Android 的调试模式和正常运行模式有以下区别：  应用安装方式：在调试模式下，可以通过USB连接将应用程序直接安装到设备或模拟器上进行调试。而在正常运行模式下，应用程序需要通过一般的安装方式进行安装。  日志输出：在调试模式下，可以通过Android的日志系统输出应用程序的日志信息，包括调试信息、错误信息等。而在正常运行模式下，默认情况下是无法直接查看这些日志信息的。  调试工具支持：在调试模式下，可以利用开发工具提供的调试功能，对应用程序进行调试，包括断点调试、变量监视等。而在正常运行模式下，这些调试工具的功能可能无法使用或受限。  性能和优化限制：在正常运行模式下，系统会优化应用程序的运行，以提升性能和效率。例如，可能会关闭一些调试相关的功能或限制某些操作。而在调试模式下，这些限制可能会被解除，以方便开发人员进行调试和测试。  安全限制：调试模式在一定程度上降低了设备的安全性，因为它允许开发人员更加灵活地操作设备和应用程序。因此，在正常运行模式下，为了保护用户数据和设备安全，一些操作和权限可能会受到限制。  6.程序有三种错误，分别用什么方法解决，分别对应Android Studio下方的哪一种观察子窗口  编译错误：这种错误发生在编译阶段，通常是由于语法错误、缺少引用或其他编译问题导致的。解决编译错误的方法是修复代码中的语法错误，并确保所有引用都正确导入。在 Android Studio 中，编译错误信息会显示在 "Build" 或者 "Gradle Console" 观察子窗口中。  运行时错误：这种错误发生在应用程序运行过程中，可能是由于空指针引用、数组越界、数据类型不匹配等原因导致的。解决运行时错误的方法是使用调试工具来定位并修复错误的代码，例如在 Android Studio 中使用调试器（Debugger）查看变量的值、添加断点进行逐步调试。运行时错误信息会显示在 "Logcat" 观察子窗口中。  逻辑错误：这种错误是由于代码中的逻辑错误导致的程序行为异常或与预期不符。解决逻辑错误的方法是仔细检查代码中的逻辑，包括条件判断、循环、算法等，确保代码逻辑正确。逻辑错误通常需要在代码中添加或修改条件、循环等逻辑来修复。逻辑错误没有特定的观察子窗口，开发者可以利用日志输出、断点调试等技术来帮助定位和解决问题。  请尝试编译https://github.com/THEONE10211024/ApiDemos 或者 https://github.com/lenmoyouziJiangjun/AndroidAPIDemos  描述编译过程中，你遇到的问题并如何解决。  缺少引用或依赖项错误：解决方法是打开 build.gradle 文件，检查依赖项是否正确，并更新版本号以匹配当前的环境。  语法错误或命名冲突：解决方法是检查代码中的拼写错误、语法错误、重复定义等问题，并进行修改。  Gradle 构建脚本错误：解决方法是检查构建脚本中的错误，并进行修改。  Android SDK 版本不匹配：解决方法是更新 Android SDK 版本，并重新构建项目。  文件路径或权限错误：解决方法是检查文件路径是否正确，并确保应用程序有足够的权限来访问这些文件。 |

**四、实验及分析**

通过使用 Stack Overflow 等网站查找错误和解决方案，可以快速获得其他开发者的经验和指导，有助于解决编程中遇到的问题。这些在线社区提供了广泛的知识和经验，能够帮助开发者更好地理解和解决各种技术难题。

利用调试工具和日志输出进行程序调试，是一种非常有效的方法。调试工具可以让开发者逐步跟踪程序的执行过程，观察变量的值，定位错误的代码行，从而更准确地找到和修复问题。日志输出作为一种调试手段，可以输出关键数据和信息，帮助开发者理解程序执行路径和状态，从而快速排查问题所在。

通过下载示例代码并学习其中的实现方法，可以拓宽开发者的视野，学习到新的技术和最佳实践。通过观察和模仿示例代码，开发者可以更好地掌握各种开发技术，并将其应用于自己的项目中。

**五、实验总结**

本实验通过使用 Stack Overflow 等网站查找错误、使用调试工具和日志输出进行程序调试，以及从网上下载示例代码，让开发者能够更好地解决编程中遇到的问题，提高开发效率和技术水平。

Stack Overflow 等在线社区是宝贵的资源，可以让开发者与全球范围内的开发者互动和分享，快速获取解决方案。然而，在使用这些资源时，需要注意筛选和理解答案的质量，选择适合自己问题的最佳解决方案。

调试工具和日志输出是开发过程中必不可少的工具，能够帮助定位和修复错误。通过深入学习和熟练运用调试工具和日志输出，可以提高排错的效率和准确性。

从网上下载示例代码是学习和掌握新技术的一种重要途径。通过仔细观察示例代码的实现方式和设计思路，开发者可以积累经验和灵感，并将其应用到自己的项目中，提高代码质量和开发效率。