Android 日志工具 Log

1. 实训目的

（1）掌握 Android 日志工具 Log 的用法；

（2）理解 Log 与 System.out 的区别。

2. 实训要求

（1）创建项目 HelloWorld；

（2）在 HelloWorld 项目中使用 Log 类的相关方法输出信息；

（3）完成实验报告。

3. 实训指导

（一）使用 Android 的日志工具 Log

Android 中的日志工具类是 Log（android.util.Log），这个类中提供了如下几个方法来供我们打印日志。

（1） Log.v()

这个方法用于打印那些最为琐碎的，意义最小的日志信息。对应级别 verbose，是 Android 日志里面级别最低的一种。

（2） Log.d()

这个方法用于打印一些调试信息，这些信息对你调试程序和分析问题应该是有帮助的。对应级别 debug，比 verbose 高一级。

（3） Log.i()

这个方法用于打印一些比较重要的数据，这些数据应该是你非常想看到的，可以帮你分析用户行为的那种。对应级别 info，比 debug 高一级。

（4） Log.w()

这个方法用于打印一些警告信息，提示程序在这个地方可能会有潜在的风险，最好去修复一下这些出现警告的地方。对应级别 warn，比 info 高一级。

（5） Log.e()

这个方法用于打印程序中的错误信息，比如程序进入到了 catch 语句当中。当有错误信息打印出来的时候，一般都代表你的程序出现严重问题了，必须尽快修复。对应级别 error，比 warn 高一级。

（二）Log 与 System.out 的区别

为什么使用 Log 而不使用 System.out？

为什么 System.out.println()方法会这么遭大家唾弃呢，首先日志打印不可控制、打印时间无法确定、不能添加过滤器、日志没有级别区分。System.out 就是一个懒女孩，它只给你返回就是了，管你干嘛用的，什么时候用，你的要求级别是什么，如此不负责任，所以建议大家少用。

(1) System.out.println()方法日志打印不可控制、打印时间无法确定、不能添加过滤器、日志没有级别区分。

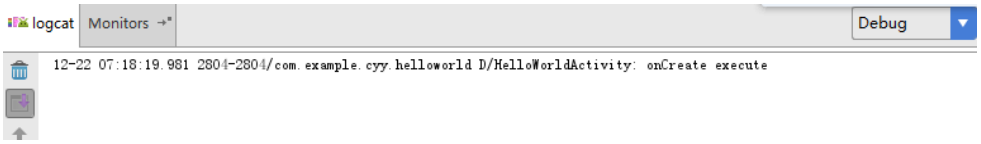
(2) Log 工具可以根据日志的重要性使用不同的方法打印。

(3) logcat 可以添加过滤器、进行级别控制。

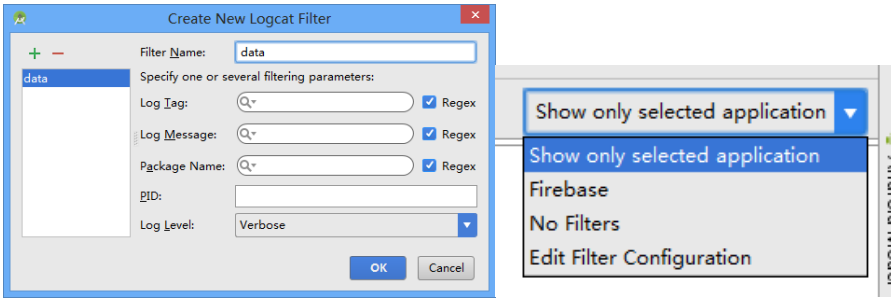
案例：

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.hello_word_layout);
        Log.d("HelloWorldActivity","onCreate execute");
}
```

运行结果：



使用 logcat：



运行后观察 Android Monitor：