# 第一部分：log4j的使用

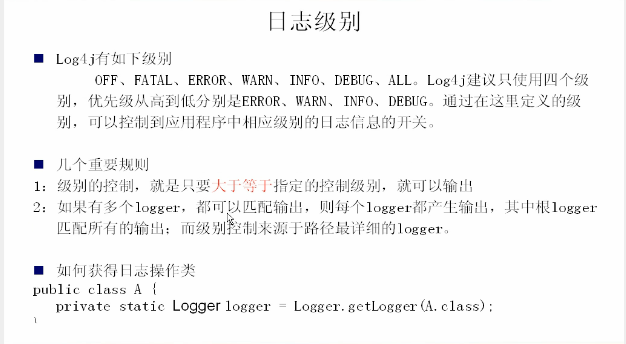
1. log4j.properties的基本用法，目的地【appender】、布局【layout】、控制单元【logger】、级别【level】  
     2、log4j.xml的用法，以及比log4j.properties的优点  
     3、log4j的eclipse插件log4e的基本使用

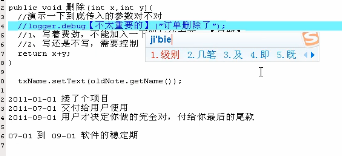
# 、日志框架概述。

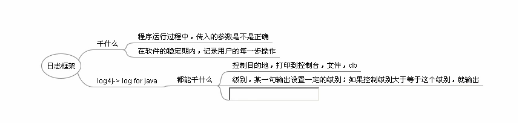
  为什么要使用日志框架？在项目未稳定期需要记录用户所有的操作，以作为验收的凭证。

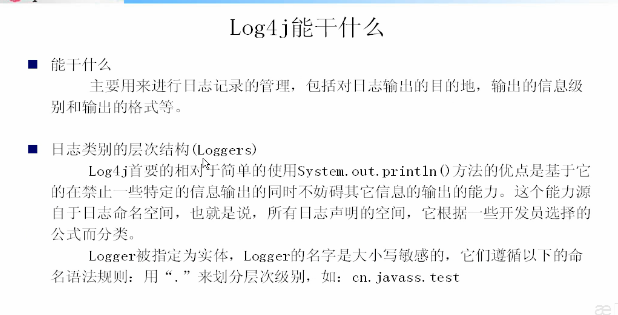
# 2、log4j.properties的基本用法。

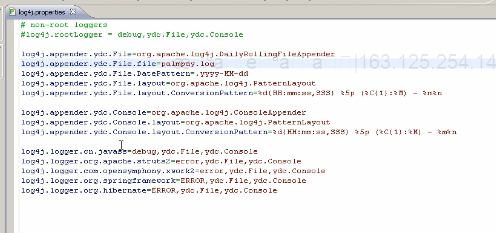
  log4j.properties中的各种概念：目的地【appender】、布局【layout】、控制单元【logger】、级别【level】。

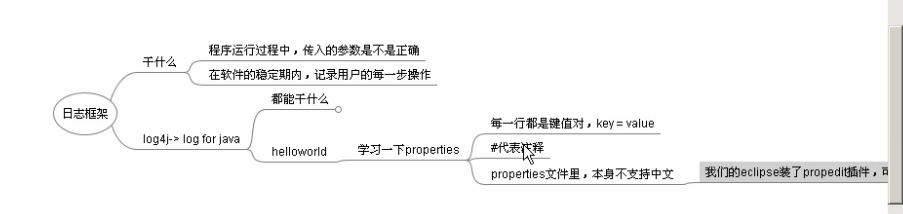


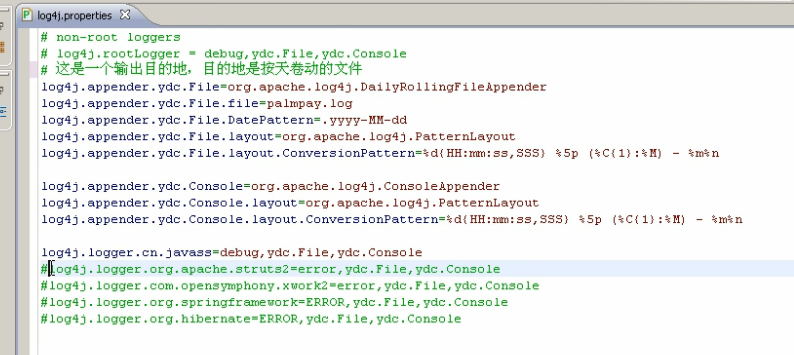


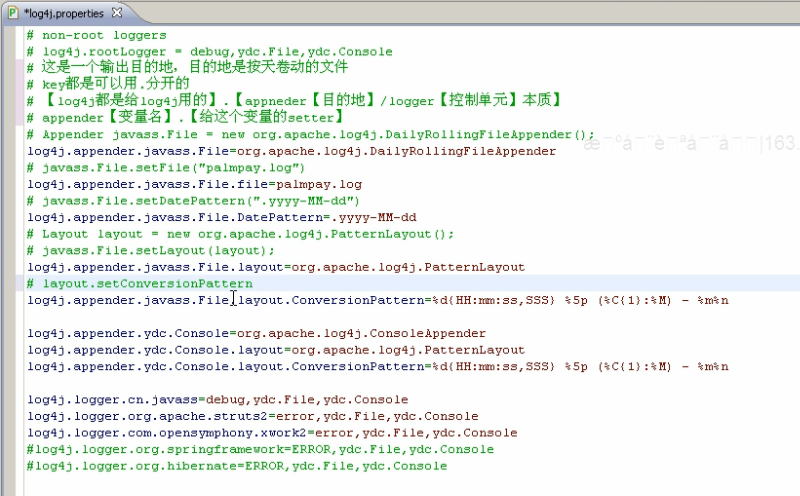


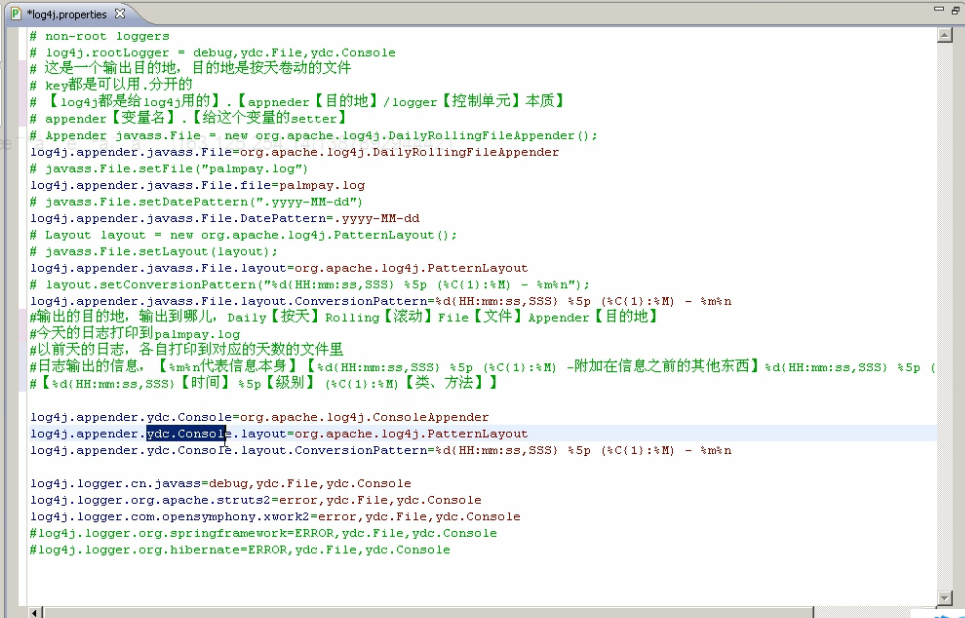


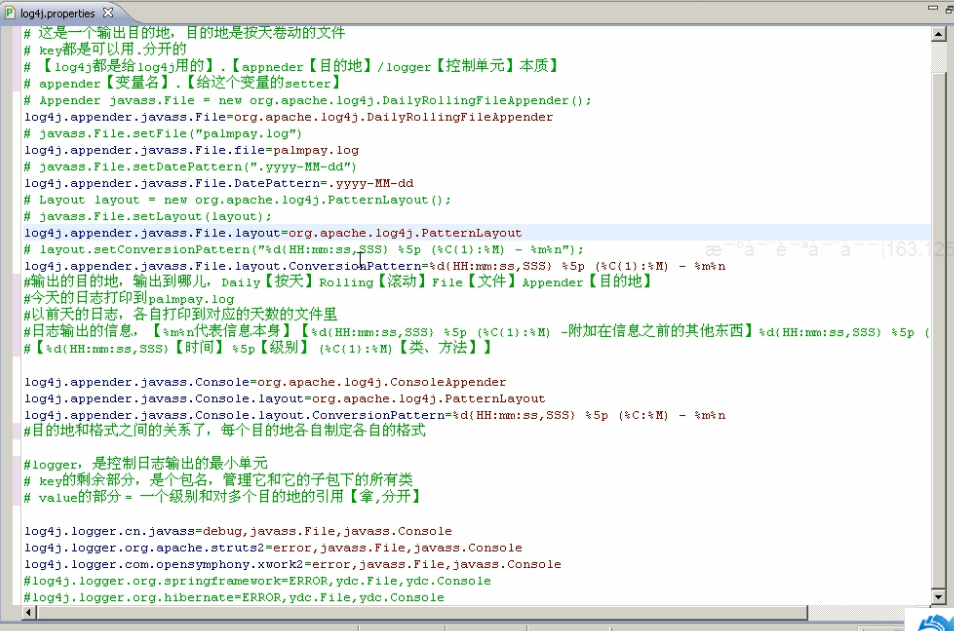


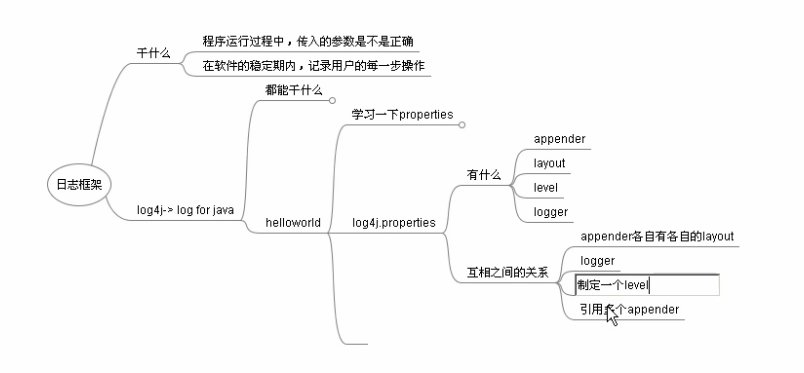


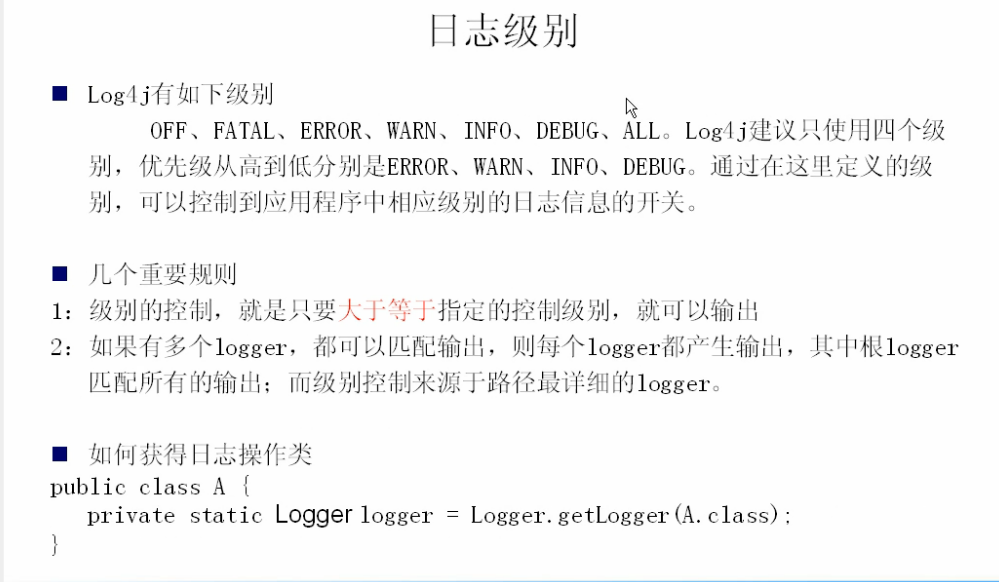


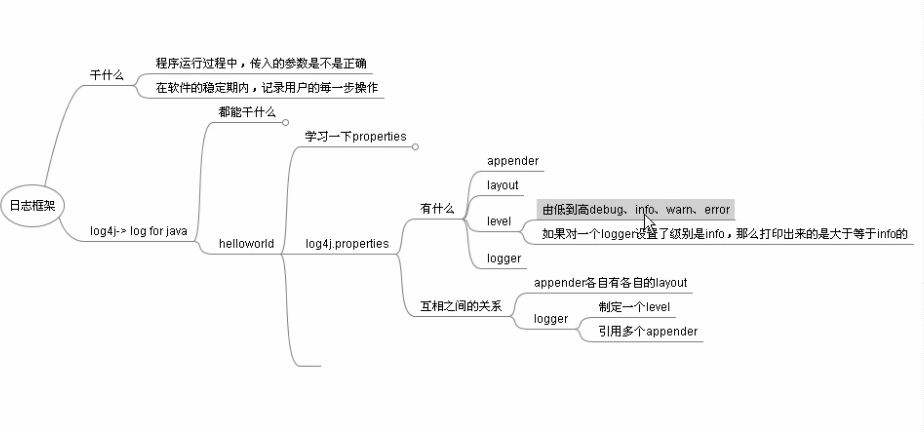


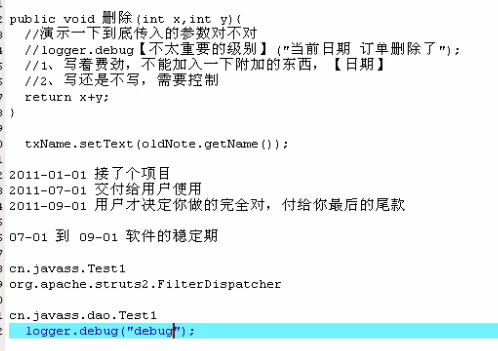


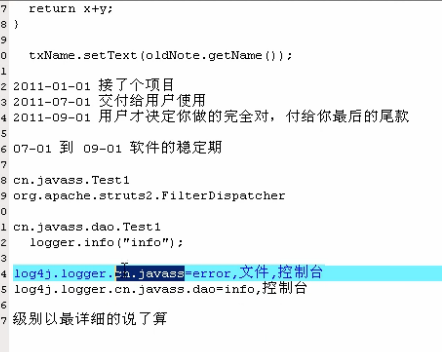






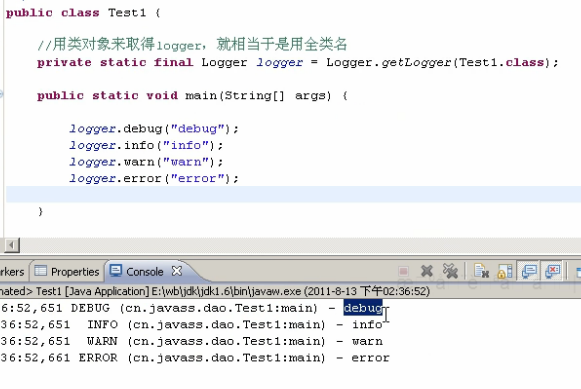
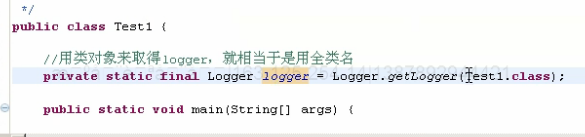






前面的info和error级别以级别说了算,为info级别;后面的各自管各自的;





# Log 4 J = Log for Java 顾名思义这是一个java的日志工具（暂且称之为工具吧）。

　　Log4j是Apache的一个开放源代码项目，通过使用Log4j可以轻松控制日志输出到控制台或者文件。对于项目开发的阶段一般是输出到控 制台，可以随时监控程序运行情况，到了项目上线后配置监控日志文件，若程序产生异常，程序中的日志代码会输出到指定的文件中，方便项目的后期运维。因为 Log4j配置非常简单，所以上述方式还是值得参考的。

## 下面说说它的使用方法：

### 1）导入Log4j的jar包

### 2）初始化Log4j

#### application中手动初始化

#### web项目中启动时初始化

　　3）配置log4j.properties文件

　　　　指定记录器名字，设置输出方式，指定输出格式

　　4）在日志中写输出日志

注意:在配置文件路径的时候,最好用相对路径;

在对应项目在一起;方便后期使用;

4.在appender.moder1.Layout下定义的是PatternLayout才有log4j.appender.moder1.layout.ConversionPattern

   该属性自定义日志输出格式

-X号: X信息输出时左对齐；  
%p: 输出日志信息优先级，即DEBUG，INFO，WARN，ERROR，FATAL,  
%d: 输出日志时间点的日期或时间，默认格式为ISO8601，也可以在其后指定格式，比如：%d{yyy MMM dd HH:mm:ss,SSS}，输出类似：2002年10月18日 22：10：28，921  
%r: 输出自应用启动到输出该log信息耗费的毫秒数  
%c: 输出日志信息所属的类目，通常就是所在类的全名  
%t: 输出产生该日志事件的线程名  
%l: 输出日志事件的发生位置，相当于%C.%M(%F:%L)的组合,包括类目名、发生的线程，以及在代码中的行数。举例：Testlog4.main (TestLog4.java:10)  
%x: 输出和当前线程相关联的NDC(嵌套诊断环境),尤其用到像java servlets这样的多客户多线程的应用中。  
%%: 输出一个"%"字符  
%F: 输出日志消息产生时所在的文件名称  
%L: 输出代码中的行号  
%m: 输出代码中指定的消息,产生的日志具体信息  
%n: 输出一个回车换行符，Windows平台为"\r\n"，Unix平台为"\n"输出日志信息换行