日志管理:

- 1. 收集项目运行的问题
- 2. 调试代码

日志框架:

log4j(版本1x)

log4j的组成:

- 1. 日志记录器:输出日志
- 2. 输出端: 指定日志输出的位置
- 3. 日志格式化: 控制日志输出的样式

方式一(代码格式):

创建日志记录器:

Logger logger = Logger.getLogger(类名.class)

日志格式:

简单格式

SimpleLayout simpleLayout = new SimpleLayout();

Html格式

HTMLLayout htmlLayout = new HTMLLayout();

自定义格式

PatternLayout patternLayout = new PatternLayout("%r [%t] %-5p %c - %m%n")

输出端:

定期控制台

ConsoleAppender consoleAppender = new ConsoleAppender(格式名);

文件

FileAppender fileAppender = new FileAppender(格式名,文生件名);

定期生成文件

DailyRollingFileAppender dailyRollingFileAppender = new

DailyRollingFileAppender(格式名,文件名,日期)

实现文件:

logger.addAppender(输出端名)

设置日志等级

logger.setLevel(Level.DEBUG/INFO/WARN/ERROR/FATAL);

只会显示当前等级以下的日志

方式二(文档格式):

日志级别和输出端

log4j.rootLogger=日志级别,输出端名字

控制台格式的设置

log4j.appender.输出端名=org.apache.log4j.ConsoleAppender

文件格式的设置

log4j.appender.输出端名=org.apache.log4j.FileAppender 定期生成格式文件

log4j.appender.输出端名=org.apache.log4j.DailyRollingFileAppender 输出格式

log4j.appender.输出端名.layout=org,apache.log4j.格式名

当有文件生成时

log4j.appender,输出端名.File=文件名.文件格式

自定义文件中设置为

log4j.appender.文件名.layout.ConversionPattern=%r [%t] %-5p %c - %m%n