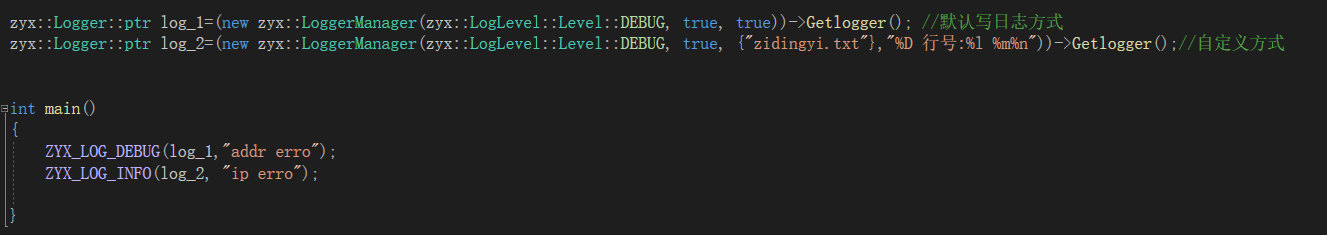
**日志器实现**

1. **总体结构**

* **class LogLevel：日志等级，只有事件的等级高于写的等级才会被记录**
* **class LogEvent：对事件的封装，如行号、线程号、错误信息等**
* **class LogFormatter：输出格式**
* **class LogAppender：日志输出(抽象类)**
* **class FileLogAppender : public LogAppender：向文件中输出**
* **class StdoutLogAppender : public LogAppender：向控制台输出**
* **class Logger : public std::enable\_shared\_from\_this<Logger>：日志器，负责将事件、输出器、格式包装在一起，实现输出**
* **class LoggerManager：进行整体的封装与调用**
* **一些宏定义，用来调用**

**调用示例：**



1. **分步介绍**