日志构件选择文档（上机实践 10）

本次上机实践要求统一日志功能，分级别，同时日志可以动态配置。为了达到这个目的，我们小组两位成员分别研究了 log4j 和 logback 构件，并分别独立完成课程实践要求功能，最后选定的 logback 作为日志构件，两种日志构件的具体实现细节和选择理由分述如下：

log4j：

综述和选择理由：log4j 是使用 JAVA 编写的流行的日志框架，主要由三部分组成：通过 logger 采集日志信息；通过 appender 将日志发布到不同地方；通过 layout 以不同的风格格式化日志信息。其线程安全，配置简单和日志分层的特点是此次作业研究的理由。

使用过程：

1. 分别在 client 端和 server 端添加 log4j 依赖（使用 Maven）；
2. 在两处分别配置 log4j.properties 文件，文实现日志分级的效果，我们将日志分为 info 信息和 error 信息，每次使用完成后，client 端和 server 端都会生成对应的使用日志，同时规定日志格式，日志等级等信息；
3. 在具体的代码实现中，在所有必要输出日志信息的地方（如连接 server 是否成功，账号是否被注册，数据库是否连接成功等）添加必要的 log 信息，同时根据这些信息的不同对输出的 log 分等级，程序运行时，console 端会及时输出日志信息，运行结束后，系统会生成 log 文件夹，记录该次运行的所有必要的日志信息；
4. 对于动态可配置的要求，添加了 setLevel 函数，每次运行时可以动态设定日志输出等级；

logback：

综述和选择理由：logback 可以认为是 log4j 的一个改良版本，分成三个模块：logback-core，logback-classic 和 logback-access。其中 logback-core 是其他两个模块的基础模块。

使用过程：

1. 在 Client 与 Server 端添加 logback 依赖（用到了 logback-core 和 logback-classic）；
2. 添加配置文件 logback.xml；
3. 在项目中需要记录日志的地方使用不同方法来记录不同级别的日志，包括 trace，debug，info，warn，error 五个级别；
4. 系统会根据配置好的时间间隔对配置文件进行扫描，当配置文件更改后，logback 会自动重新加载配置。

总结：在 log4j 的使用过程中，由于日志等级是最初写好在 properties 文件里面的，虽然可以通过 setLevel 方法来实现调用，但其对于动态可配置的实现并不是非常强大，最后我们选用 logback 作为此次实践的日志构件。