|  |
| --- |
| 概要设计说明书 |

[概要设计说明书 1](#_Toc445302602)

[1 引言 3](#_Toc445302603)

[1.1 编写目的 3](#_Toc445302604)

[1.2 范围 3](#_Toc445302605)

[1.3 定义 3](#_Toc445302606)

[2 总体设计 4](#_Toc445302607)

[2.1 需求规定 4](#_Toc445302608)

[2.2 运行环境 4](#_Toc445302609)

[2.3 基本设计概念和处理流程 4](#_Toc445302610)

[2.4 结构 5](#_Toc445302611)

[2.5 功能需求与程序的关系 6](#_Toc445302612)

[3 接口设计 6](#_Toc445302613)

[3.1 用户接口 6](#_Toc445302614)

[3.2 外部接口 6](#_Toc445302615)

[3.3 内部接口 6](#_Toc445302616)

[4 系统出错处理设计 6](#_Toc445302617)

[4.1 出错信息 6](#_Toc445302618)

[4.2 补救措施 6](#_Toc445302619)

[4.3 系统维护设计 7](#_Toc445302620)

# 1 引言

1.1 编写目的

根据简易web服务器的设计需求，构建此软件的基本结构，明确软件开发过程时所遵循的基本框架，从而更好的完成软件详细设计、编码实现以及功能测试。

1.2 范围

说明:

java web http server;

该软件是培训习题作业.

1.3 定义

http的get请求方法;

http协议中的keep-alive设置，;socket连接;

jdk1.5提供的线程池对象、NIO对象、java.util.concurrent.\*下的相关对象;

通用的工具函数apache commons相关工具包.

# 2 总体设计

2.1 需求规定

开发环境需求：

使用eclipse/egit，使用slf4j和Junit。

eclipse下载地址<ftp://ftp.succez.com/Programmer/Java/eclipse/>；

jdk下载地址<ftp://ftp.succez.com/Programmer/Java/jdk/>；

egit已经在eclipse中内置。

功能需求：

1. 只处理http的get请求方法，post或其他方法可以直接抛出异常
2. 需要处理http协议中的keep-alive设置，尽量在一个socket连接中返回多个文件的内容
3. 使用jdk1.5提供的线程池对象、NIO对象、java.util.concurrent.\*下的相关对象，通用的工具函数可以使用apache commons相关工具包
4. 通过web服务器能访问d盘下的所有文件，例如http://localhost/d/a/b.txt 就可以访问到d盘下的a目录下的b.txt文件，如果是目录，则列出目录下的文件的列表
5. 程序启动后，在控制台监控用户输入的命令，如果输入exit，等待已有任务运行完毕，再关闭需要关闭的资源，最后退出程序

性能需求：

1. 使用slf4j去输入调试日志
2. 使用junit测试相关的代码，尽量提高代码测试覆盖率

2.2 运行环境

Windows7+jre5+浏览器；

2.3 基本设计概念和处理流程

实现一个轻量级的、只处理静态文件的web服务器。

Tomcat是一个J2ee规范的参考实现，这里要做的这个服务器只需要处理静态文件即可，不支持servlet、jsp、java等动态内容，我们要做到和apache服务器的最基本的功能一致。

本系统的基本设计概念和处理流程如图：图2.1处理流程图所示。

图2.1 处理流程图

浏览器

服务器

Get请求

Socket连接

D盘数据传送

控制台

Exit命令

2.4 结构

本系统的系统元素(各层模块,子程序,公用程序等)的划分，每个系统元素的标识符和功能，各元素之间的组成关系，如图2.2程序结构图所示。

图 2.2 程序结构图

日志系统

测试系统

服务器

http协议常量映射

工具类(util)

服务器核心(server)

自定义异常(exception)

启动器(launcher)

http： http协议解析、应答、http常量定义等通信相关功能。

Util：程序工具类，读取端口、线程池配置、文件根目录等基本配置信息；封装通用逻辑方法。

Server：线程池创建和打开关闭操作，socket监听、连接建立操作，连接异常处理，get请求解析处理，应答处理等。

Exception：自定义异常信息，便于调试控制程序。

Launcher：启动服务器，初始化服务器数据。关闭服务器。

Launcher

Server

Exception

Util

http

Start/Exit命令

2.5 功能需求与程序的关系

各项功能需求的实现同各块程序的分配关系:

1，（server，exception）只处理http的get请求方法，post或其他方法可以直接抛出异常。

2，（server，http）需要处理http协议中的keep-alive设置，尽量在一个socket连接中返回多个文件的内容。

3，（server，util）使用jdk1.5提供的线程池对象、NIO对象、java.util.concurrent.\*下的相关对象，通用的工具函数可以使用apache commons相关工具包。

4，（server，util，http）通过web服务器能访问d盘下的所有文件，例如http://localhost/d/a/b.txt 就可以访问到d盘下的a目录下的b.txt文件，如果是目录，则列出目录下的文件的列表。

5，（launcher）程序启动后，在控制台监控用户输入的命令，如果输入exit，等待已有任务运行完毕，再关闭需要关闭的资源，最后退出程序。

6，（日志系统）使用slf4j去输入调试日志。

7，（测试系统）使用junit测试相关的代码，尽量提高代码测试覆盖率

# 3 接口设计

3.1 用户接口

Exit命令：控制台监控此命令用于停止服务器运行。

3.2 外部接口

见详细设计书。

3.3 内部接口

见详细设计书。

# 4 系统出错处理设计

4.1 出错信息

http除了get方法外的post等其他方法程序直接抛出异常，此异常是自定义异常类型。

4.2 补救措施

异常出现后采取的变通措施： 返回携带异常信息的静态html文件。

4.3 系统维护设计

主要是系统功能对应的测试模块。包括函数功能测试，模块功能测试等。