Web Server 详细设计

一、引言

1. 编写目的

编写本详细设计的目的是在 Web Server 概要设计的基础上，对该程序进行详细设计，进一步明确系统结构，详细地介绍系统的各个模块，为进行后续的实现和测试作准备。

本详细设计说明书的预期读者为本项目小组成员。

2. 定义

1.类图中属性和方法之前附加的可见性修饰符：

加号（+）表示 public；减号（-）表示 private；#号表示 protected；省略这些修饰符表示具有 package 访问权限，如果属性或方法下面有下划线，则说明是静态的。

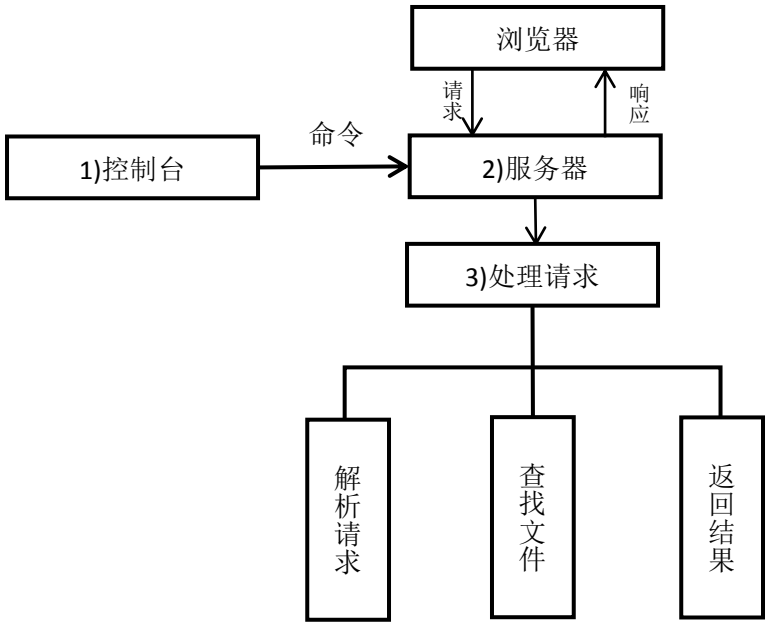
2.NIO:nio 是 New IO 的简称，在 jdk1.4 里提供的新 api 提供多路非阻塞式的高伸缩性网络 I/O 。

二、总体设计

1. 需求概述

按照需求分析文档中的规格要求，用户通过浏览器地址栏输入地址连接服务器，查看指定路径下的文件及文件夹。同时，可以在线预览文件和下载文件。

2. 程序结构



系统由 3 个大模块组成：

1.控制台：控制台输入命令，设置服务器端口号、退出程序。

2.服务器启动：服务器启动并监听指定端口，等待客户端连接。

3.请求处理：服务器在接收到客户端的请求后，对请求作出响应。

其中，请中请求处理模块分为 3 个子模块：

a.解析请求：解析客户端的请求类型，得到客户端访问的文件路径。

b.查找文件：根据文件路径，查找响应文件，存入集合中。

c.返回结果：将集合中的内容返回给客户端。

三、模块详细设计

采用基于 NIO 的 socket 服务器，避免在多用户同时访问时，多个连接需要多个线程，减少并发的线程数。

1. 控制台

1.1 模块描述

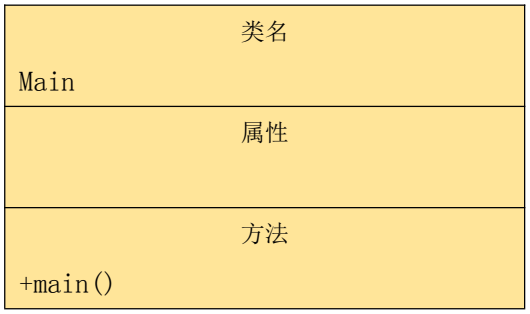
程序入口，等待用户设置服务器监听端口号。如果端口号被占用，则给出提示信息，并要求设置创建端口号；等待用户输入 exit 命令，结束程序。

1.2 模块设计

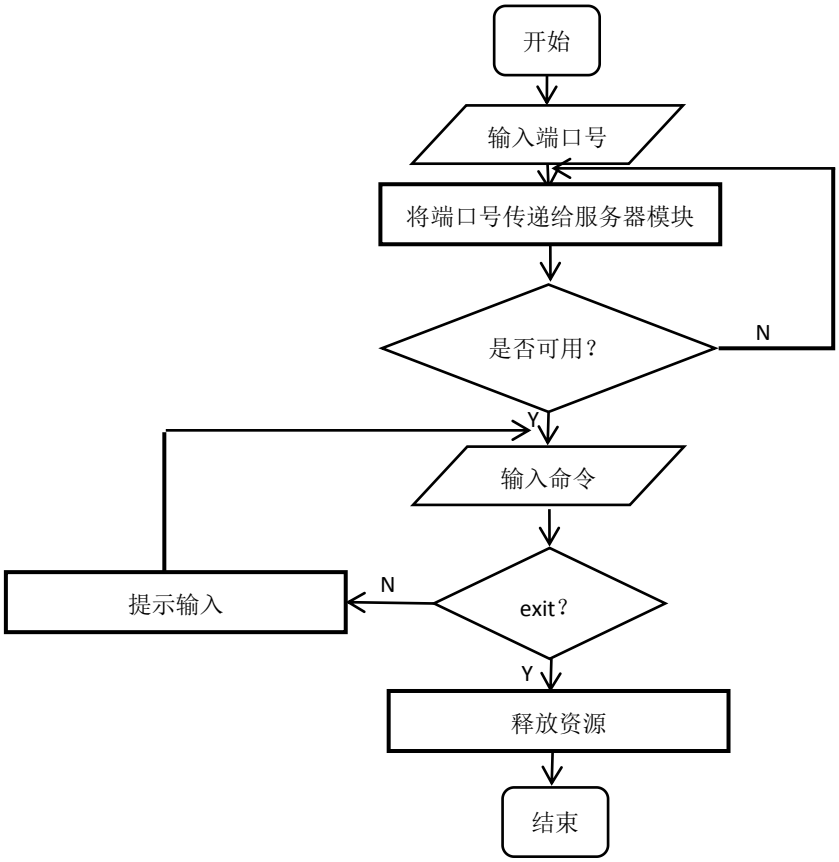
该模块中的主要文件，文件中包含的主要类和功能以及与其他模块交互的情况如下：

- Main.java：Main 是含有主函数的类，也是程序执行的入口。程序开始时，要求用户在控制台设置服务器端口号，将用户输入的端口传入到服务器模块；若服务器返回端口不可用的信息，则要求用户重新输入一个端口号。然后，等待用户控制台输入的 exit 命令，释放资源，退出程序。若用户输入的命令不可用，则提示用户输入。

1.3 类图



1.4 流程逻辑



2. 初始化服务器

2.1 模块描述

该模块负责初始化服务器，接收 Main 中传递过来的端口号，若端口号被占用，则抛出异常。若端口号可用，则监听该端口号，并等待客户端连接。

2.2 模块设计

该模块中的主要文件,文件中包含的主要类和功能以及与其他模块交互的情况如下：

- Server.java：初始化服务器，采用 NIO 的 ServerSocketChannel 监听指定的通道，并创建选择器将选择器绑定到通道,然后循环等待客户端的连接。客户端连接后，将 SelectionKey 传递到请求处理模块进行处理。

2.3 类图

类名
Server
属性
-serverSocketChannel -port -selector
方法
+server() +start() +exit()

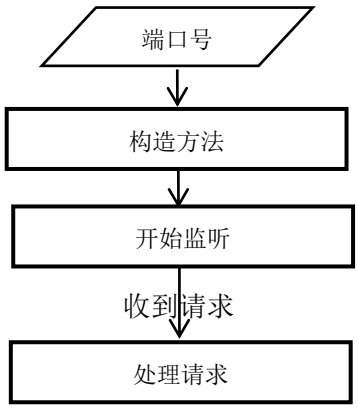
属性：

- serverSocketChannel:监听套接字的可选择通道。
- port:端口号。
- selector:选择器。

方法：

- void server(int port):构造方法，监听指定端口号。
- void start():服务开始监听指定端口。
- void exit（）:释放资源，关闭服务器。

2.4 流程逻辑



3. 请求处理

3.1 模块描述

该模块负责在服务器监听到客户端请求后，对客户端的请求作出响应。

3.2 模块设计

该模块中的主要文件，文件中包含的主要类和功能以及与其他模块交互的情况如下：

- Parser.java：获取 SelectionKey 并对 SelectionKey 进行处理。

1.测试此键的通道是否已准备好接受新的套接字连接（isAcceptable)。

若此键的通道已经准备好接受新的套接字连接，通过 ServerSocketChannel 的 accept 方法接受到此通道套接字的连接,得到一个 SocketChannel。然后，将该 SocketChannel 通过 register 方法向给定的选择器注册此通道。

2.测试此键的通道是否已准备好进行读取（isReadable)

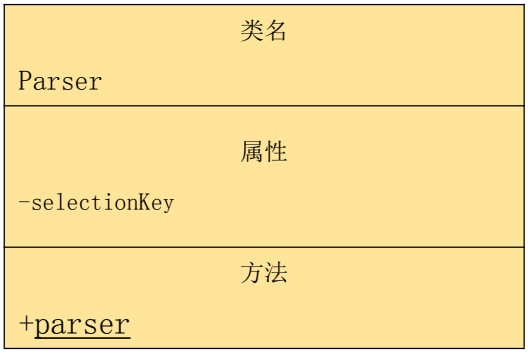
若此键的通道已经准备好进行读操作，则读取此通道中客户端传递过来的信息，将信息读取到给定的缓冲区中，得到一个 ByteBuffer，然后将 ByteBuffer 传入到 Request 中，作为一个 Http 请求。

3.测试此键的通道是否已准备好进行写入（isWritable)

若此键的通道已经准备好进行写操作，则将缓冲区中客户端请求的资源写入到通道中。

- Request.java:接受从通道写入到缓冲区中的 ByteBuffer，Resquest 解析 ByteBuffer 中的内容，得到客户端请求资源的路径，然后将路径传递给 SeekFile 类。
- SeekFile.java:接受一个文件路径，判断请求资源的类型属于文件还是文件夹。若为文件夹，则返回文件夹中的文件内容列表；若为文件，则返回一个文件的输入流；若文件不存在，则抛出异常。
- Response.java:接受从 SeekFile 中传递过来的资源,将整理后写入到缓冲区中。

3.3 类图

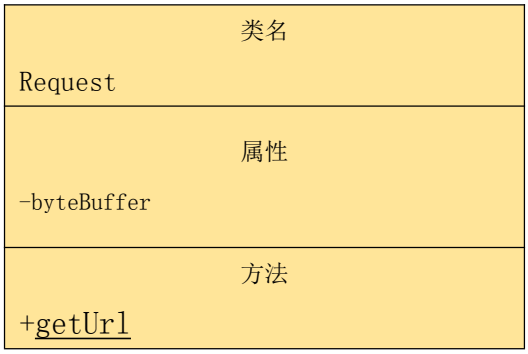


属性：

- selectionKey:SelectableChannel 在 Selector 中的注册的标记。

方法：

- void parser(SelectionKey key):获取 SelectionKey 并对 SelectionKey 进行处理。

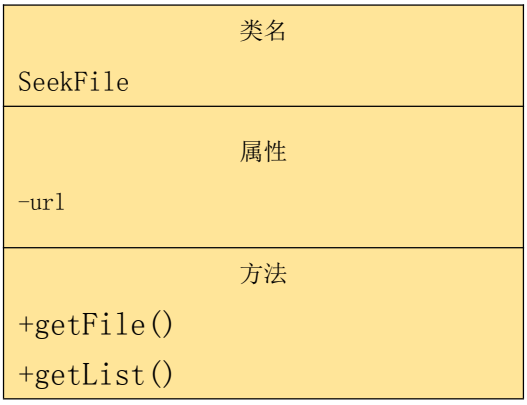


属性：

- byteBuffer:写入到缓冲区的内容。

方法：

- String getUrl(ByteBuffer buffer):解析缓冲区中浏览器的 Http 请求，获取浏览器请求访问的文件路径。



属性：

- url:文件路径

方法：

- InputStream getFile(String url):获取文件输入流。
- String[] getList(String url):获取文件夹中的内容

类名
Respon
属性
-SeekFile
方法
+writeToByteBuffer()

属性：

- SeekFile:SeeKFile 类

方法：

- void writeToByteBuffer():将浏览器请求的资源写入到缓冲区。

3.4 流程逻辑