**Exercise 08**

**Name:** Jose Juan Sandoval

**Link to Project:** <https://github.com/Juanchiselo/CS380/tree/master/Exercises/Exercise%2008>

**Java Code**

WebServer.java

import java.io.IOException;  
import java.net.ServerSocket;  
  
public class WebServer  
{  
 public static void main(String[] args)  
 {  
 int portNumber = 8080;  
  
 try (ServerSocket serverSocket = new ServerSocket(portNumber))  
 {  
 while (true)  
 new WebServerThread(serverSocket.accept()).start();  
 }  
 catch (IOException e)  
 {  
 System.*err*.println("ERROR: Could not listen on port " + portNumber + ".");  
 System.*exit*(-1);  
 }  
 }  
}

WebServerThread.java

import java.io.\*;  
import java.net.Socket;  
import java.util.ArrayList;  
  
public class WebServerThread extends Thread  
{  
 private Socket socket = null;  
  
 */\*\*Objects for input reading\*/* private String line;  
 private ArrayList<String> request;  
 private InputStream inputStream;  
 private InputStreamReader inputStreamReader;  
 private BufferedReader bufferedReader;  
  
 */\*\*Objects for output writing\*/* private ArrayList<String> response;  
 private OutputStream outputStream;  
 private PrintWriter printWriter;  
 private String charsetName;  
  
 public WebServerThread(Socket socket)  
 {  
 super("WebServerThread");  
 this.socket = socket;  
  
 try  
 {  
 charsetName = "UTF-8";  
 inputStream = socket.getInputStream();  
 inputStreamReader = new InputStreamReader(inputStream, charsetName);  
 bufferedReader = new BufferedReader(inputStreamReader);  
  
 outputStream = socket.getOutputStream();  
 printWriter = new PrintWriter(outputStream, true);  
 }  
 catch(IOException exception)  
 {  
 System.*err*.println("ERROR: Connection lost with client "  
 + socket.getInetAddress().getHostAddress());  
 }  
 }  
  
 */\*\*  
 \* The overridden run() function belonging to the Thread class.  
 \* This is what handles the communication between the server and the client.  
 \*/* public void run()  
 {  
 try  
 {  
 request = new ArrayList<>();  
 // The main loop of execution.  
 while((line = bufferedReader.readLine()) != null)  
 {  
 if(line.contains("GET"))  
 {  
 HTTP\_RESPONSE("GET", line);  
 break;  
 }  
 }  
 printWriter.close();  
 bufferedReader.close();  
 socket.close();  
 } catch (IOException exception) {  
 System.*err*.println("ERROR: Could not read message from client.");  
 }  
 }  
  
 */\*\*  
 \* Sends a response to the client based on the request type.  
 \** ***@param*** *requestType - The type of request the client is making.  
 \** ***@param*** *line - The first line of the request.  
 \*/* public void HTTP\_RESPONSE(String requestType, String line)  
 {  
 response = new ArrayList<>();  
 String header = "HTTP/1.1 ";  
 String code;  
 String contentType = "Content-type: text/html\n";  
 String contentLength = "Content-length: ";  
  
 switch (requestType)  
 {  
 case "GET":  
 String path = "www\\" + line.split("[ ]")[1];  
 response = FileHandler.*getInstance*().readFile(path);  
  
 if(!response.isEmpty())  
 {  
 code = "200 OK\n";  
 contentLength += " 124\n\n";  
 }  
 else  
 {  
 response = FileHandler.*getInstance*().readFile("www\\404.html");  
 code = "404 Not Found\n";  
 contentLength += " 126\n\n";  
 }  
  
 header = header + code + contentType + contentLength;  
 response.add(0, header);  
 break;  
 }  
  
 // Sends the response to the client.  
 for(String responseLine : response)  
 printWriter.print(responseLine);  
 }  
}

FileHandler.java

import java.io.\*;  
import java.util.ArrayList;  
  
public class FileHandler  
{  
 private static FileHandler *instance* = null;  
 private String line;  
 private String path;  
 private FileReader fileReader;  
 private BufferedReader bufferedReader;  
  
 public static FileHandler getInstance()  
 {  
 if(*instance* == null)  
 *instance* = new FileHandler();  
 return *instance*;  
 }  
  
 private FileHandler()  
 {  
 }  
  
 private void openFile()  
 {  
 try {  
 fileReader = new FileReader(path);  
 bufferedReader = new BufferedReader(fileReader);  
 } catch(FileNotFoundException e) {  
 System.*err*.println("ERROR: Unable to open file '" + path + "'.");  
 }  
 }  
  
 private void closeFile()  
 {  
 try {  
 bufferedReader.close();  
 fileReader.close();  
 } catch(IOException e) {  
 System.*err*.println("ERROR: Unable to close file '" + path + "'.");  
 }  
 }  
  
 */\*\*  
 \* Reads the specified file.  
 \** ***@param*** *path - The path to the file to read the data from.  
 \*/* public ArrayList<String> readFile(String path)  
 {  
 this.path = path;  
 ArrayList<String> file = new ArrayList<>();  
  
 try {  
 openFile();  
 while((line = bufferedReader.readLine()) != null)  
 file.add(line);  
 } catch (IOException e) {  
 System.*err*.println("ERROR: Unable to read from file '" + this.path + "'.");  
 } finally {  
 closeFile();  
 }  
 return file;  
 }  
}