

Report

Assignment 2

IDV701

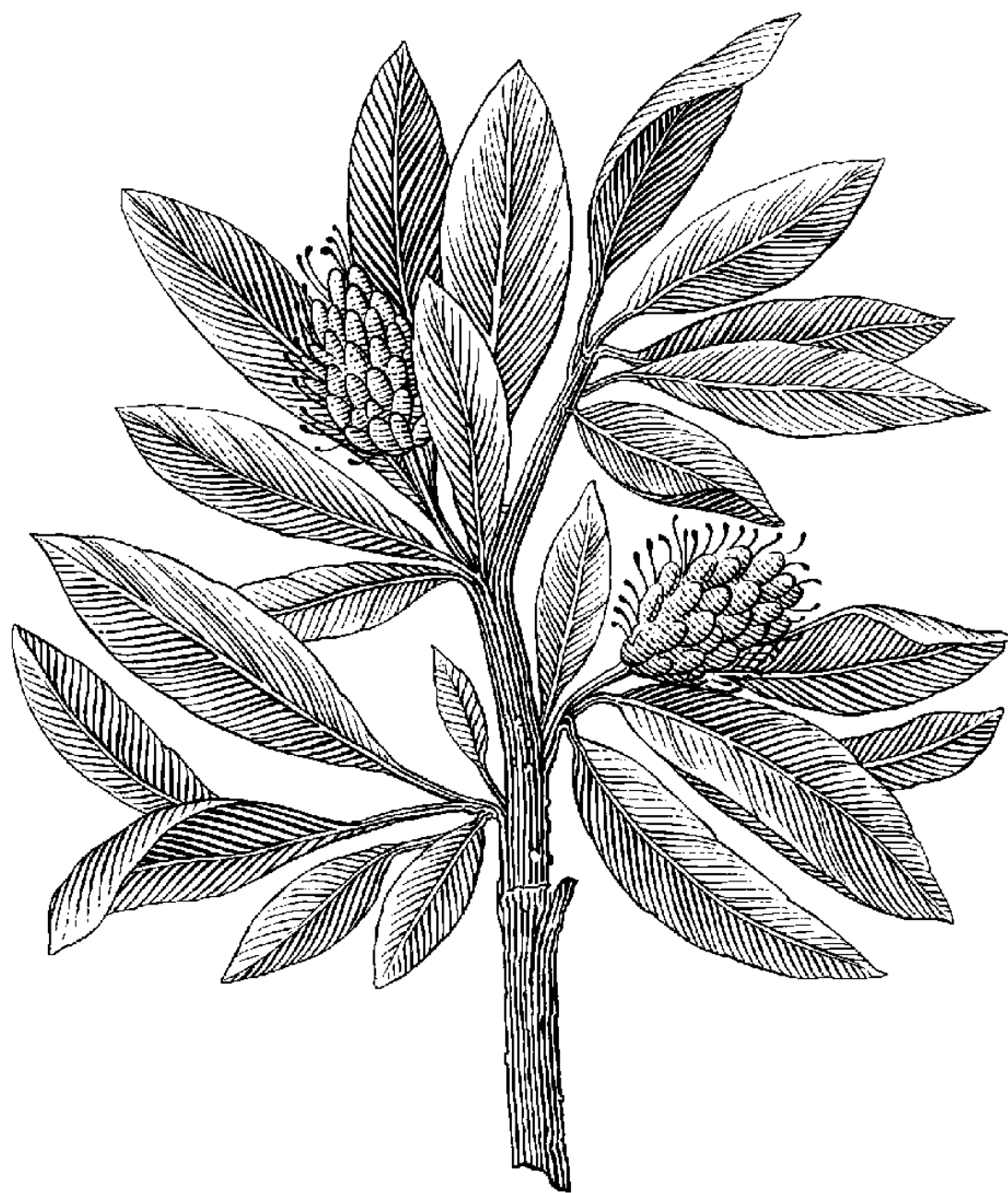
Author1: Wictor Rindebrant

Author2: Victor Lindgren

Semester: Spring 2023

Email1: wr222bp@student.lnu.se

Email2: vl222ky@student.lnu.se



- Problem 1

- Discussion

- **Working HTTP server with HTML+HTM support**

By splitting the files in the working directory, we can easily compare the last part of the splitted string, and compare it with htm or html. This way we can add support for both.

```
if (i.get(index: 1).equals(anObject: "html") || i.get(index: 1).equals(anObject: "htm")) {
```

- **Image support**

Just as we did with the HTML+HTM support, we split up the string, and compare the last part with different image format types.

```
if (i.get(index: 1).toLowerCase().equals(anObject: "png") || i.get(index: 1).toLowerCase().equals(anObject: "jpeg") || i.get(index: 1).toLowerCase().equals(anObject: "jpg")) {
```

- **Safety against escaping the root**

When we start the server, we take two arguments and one of them should contain the public directory. So in order to check the input, we add the subfolders that the public folder should always contain. By doing this we can check if the inputted folder actually contains the subfolders, and if it does not then the server will terminate.

```
file = new File(secondArg + "/a/b/c");
```

- **Exception handling**

The very first exception handling in our program happens when we check the arguments entered when we start the server. If there is not at least two arguments, then the program will terminate and print the error message.

```
public static void main(String[] args) throws IOException {
    int port = 0;
    String firstArg = "";
    String secondArg = "";
    File file = null;

    try {
        firstArg = args[0];
        secondArg = args[1];
        file = new File(secondArg + "/a/b/c");
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println(x: "Two arguments required (port, path)");
        System.exit(status: 1);
    }
}
```

Second one is when we try to take the first argument (the port one) and make it into an integer. So if the argument contains for example letters, then the program will terminate and print the error message.

```
try {
    port = Integer.parseInt(firstArg);
} catch (NumberFormatException e) {
    System.out.println(x: "Port argument has to be Integer");
    System.exit(status: 1);
}
```

Since we added the subfolders to the public directory, we now have to check if that folder actually exists. If it does not then the program will terminate and print the error message.

```
if (file.isDirectory()) {
    file = new File(secondArg);
} else {
    System.out.println("The folder '" + secondArg + "' does not exist in the current working directory.");
    System.exit(status: 1);
}
```

- **Program arguments in correct order**

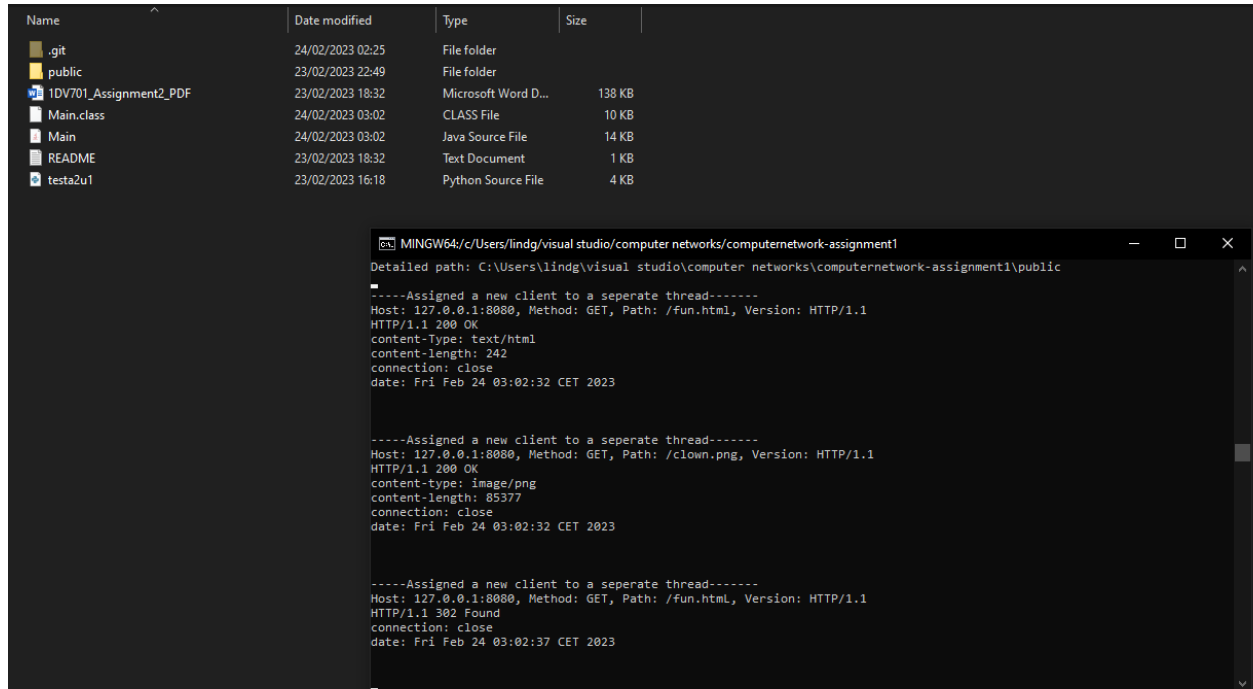
Since the argument order is (port, directory), we put the port integer as firstArg and file as secondArg.

```
try {
    firstArg = args[0];
    secondArg = args[1];
    file = new File(secondArg + "/a/b/c");
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println(x: "Two arguments required (port, path)");
    System.exit(status: 1);
}

// if first argument (port) cant be converted to an Integer, then program exits.
try {
    port = Integer.parseInt(firstArg);
} catch (NumberFormatException e) {
    System.out.println(x: "Port argument has to be Integer");
    System.exit(status: 1);
}
```

- **Console output**

We will see a new output everytime we try to load a new page. the output will contain different headers along with the path that was entered, the host ip, HTTP version.



The image shows a file explorer window on the left and a terminal window on the right. The file explorer displays a directory with files like .git, public, 1DV701_Assignment2_PDF, Main.class, Main, README, and testa2u1. The terminal window, titled 'MINGW64/c/Users/lindg/visual studio/computer networks/computernetwork-assignment1', shows detailed HTTP logs for three requests. Each log entry starts with '-----Assigned a new client to a seperate thread-----', followed by the host, method, path, and version. The first request is a GET for /fun.html, returning 200 OK with content-type text/html and content-length 242. The second request is a GET for /clown.png, returning 200 OK with content-type image/png and content-length 85377. The third request is a GET for /fun.html, returning 302 Found. All connections are closed, and the date is Fri Feb 24 03:02:32 CET 2023.

Name	Date modified	Type	Size
.git	24/02/2023 02:25	File folder	
public	23/02/2023 22:49	File folder	
1DV701_Assignment2_PDF	23/02/2023 18:32	Microsoft Word D...	138 KB
Main.class	24/02/2023 03:02	CLASS File	10 KB
Main	24/02/2023 03:02	Java Source File	14 KB
README	23/02/2023 18:32	Text Document	1 KB
testa2u1	23/02/2023 16:18	Python Source File	4 KB

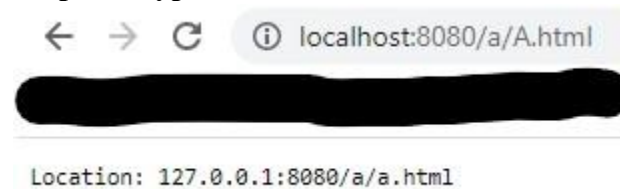
```
MINGW64/c/Users/lindg/visual studio/computer networks/computernetwork-assignment1
Detailed path: C:\Users\lindg\visual studio\computer networks\computernetwork-assignment1\public
-----Assigned a new client to a seperate thread-----
Host: 127.0.0.1:8080, Method: GET, Path: /fun.html, Version: HTTP/1.1
HTTP/1.1 200 OK
content-Type: text/html
content-length: 242
connection: close
date: Fri Feb 24 03:02:32 CET 2023

-----Assigned a new client to a seperate thread-----
Host: 127.0.0.1:8080, Method: GET, Path: /clown.png, Version: HTTP/1.1
HTTP/1.1 200 OK
content-type: image/png
content-length: 85377
connection: close
date: Fri Feb 24 03:02:32 CET 2023

-----Assigned a new client to a seperate thread-----
Host: 127.0.0.1:8080, Method: GET, Path: /fun.html, Version: HTTP/1.1
HTTP/1.1 302 Found
connection: close
date: Fri Feb 24 03:02:37 CET 2023
```

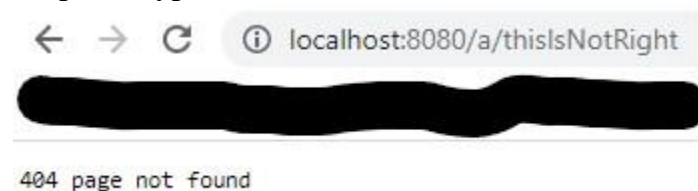
- Problem 2

Response type 302:



```
Server source file exists!  
Detailed path: C:\Users\wicea\OneDrive\Dokument\ComputerNetwork\Assignment1\computernetwork-assignment1\public  
  
-----Assigned a new client to a separate thread-----  
Host: localhost:8080, Method: GET, Path: /a/A.html, Version: HTTP/1.1  
HTTP/1.1 302 Found  
connection: close  
date: Fri Feb 24 03:37:29 CET 2023
```

Response type 404:



```
Server source file exists!  
Detailed path: C:\Users\wicea\OneDrive\Dokument\ComputerNetwork\Assignment1\computernetwork-assignment1\public  
  
-----Assigned a new client to a separate thread-----  
Host: localhost:8080, Method: GET, Path: /a/thisWillNotWork, Version: HTTP/1.1  
HTTP/1.1 404 Not Found  
connection: close  
date: Fri Feb 24 03:36:19 CET 2023  
  
Server request file does not exists!
```

- Discussion

Response type 302:

Type in any of the file names inside the public folder with an uppercase letter and it will give an response message of the type 302, that will also tell the user the correct url with lowercase letters.

Response type 404:

Type in anything in the url after “localhost:8080/” and if it’s not anything that exists in the public folder, then it should give an Error 404.