**Part 1 Application**

You are asked to develop a client-server download centre. On your server, there are three resources that can be downloaded:

1. A computer program

2. A JPEG picture

3. An e-book

|  |  |  |  |
| --- | --- | --- | --- |
| Description for version 1 | | Resource for version 2 | |
| A computer program | The program displays a message | | myProgram.jar |
| A Jpeg picture | This picture shows a man at work | | myPicture.jpg |
| An e-book | The book is about unknowns | | myBook.txt |

**1.1 Version 1**

In this version, the server returns the brief description (not the actual resource), i.e. it returns one line of text. See Table 1 above.

**1.1.1 Behavioural specification for version 1**

The sequence of exchanges between the client and the server after establishing a connection is as follows:

1. “Server: Terms of reference. Do you accept? Y or N”

2. “Client: Y”

3. “Server: 1. computer program 2. picture 3. e-book”

4. “Client:” followed by 1 or 2 or 3

5. “Server:” followed by the brief description for the chosen resource and the following message “Another? Y or N”

6. “Client: N”

7. Connection closed

In step 6, if the client says “Y”, the scenario goes back to step 3.

In step 6, if the client enters invalid input, the server sends the client the message “Invalid input --- Another? Y or N”.

In step 4, if the client enters an invalid input, the server sends the client the message “Invalid choice --- 1. computer program 2. picture 3. e-book”.

In step 2, if the client enters “N”, the server closes the connection.

In step 2, if the client enters invalid input, the server sends client the message “Invalid input ---Terms of reference. Do you accept? Y or N”.

**1.1.2 Tasks for version 1**

Implement all the functionality specified above. The client and the server must adhere to the protocol illustrated above. The server must be a **concurrent** server.

**Please note that your program MUST follow the above exchanges exactly, i.e. with the same sequence of exchanges and the same wording. Failure to do so will lose marks.**

In addition to the above functionality, your application needs to have some monitoring functionality on the server side. These functionalities are:

 After each successful download, the server displays the client’s IP address and the client’s host name.

 The server keeps a counter for each resource downloaded. Each counter counts how many times the resource has been downloaded. The server displays the value of the counter on the server side after each successful download.

Please note that you **MUST** use the following fragment of code to increment each counter.

int tmp=yourCounter;

try {

Thread.sleep(5000);

} catch (InterruptedException ex) {

System.out.println("sleep interrupted");

}

yourCounter=tmp+1;

In the above, you can replace ‘yourCounter’ with the appropriate variable name.

Note that the reason for using this fragment of code is that it is likely to reveal potential problems associated with concurrent threads sharing the same data. It is not a piece of code that would solve the potential problems associated with concurrent threads sharing the same data. If you do not use the above fragment of code means that your version 1 may **NOT** be tested for the counter functionality.

To achieve a high mark, your program must be working correctly. It should have a cohesive structure which shows separation of concerns with simple and clear logic. The program must cope with potential problem due to multiple threads sharing/accessing data. The program must be appropriately documented.

**1.2 Version 2**

In this version, the server sends the actual resource instead of the brief description. See Table 1 at the beginning of the assignment. The three resource files are:

 myProgram.jar,

 myBook.txt and

 myPicture.jpg

You can find these files on Blackboard under the section “Assignments” and then “Resource Files”.

Note that you **MUST** put all three resource files in your NetBeans project folder. You **MUST** put the downloaded files also in the project folder and name them as

 newProgram.jar,

 newBook.txt and

 newPicture.jpg.

**The markers will assume that your program will read the input files from the project folder and will only check for the downloaded files in the project folder.**

The exchanges between the client and the server must include the client first of all accepting the terms of the reference and then selecting an option to download. The sequence of exchanges after this is up to you to define.

**Part 2 Report**

In this part, you are asked to write a report about 500 words addressing the potential problem of sharing/accessing the same data by multiple threads. Compare and contrast two valid solutions through discussion of their respective pros and cons. Explain how you have solved the problem.

**Deliverables**

 Submission is done via Blackboard. You must include

o The complete Netbeans project for version 1 and version 2.

o A report for part 2. The report must be written in Word so that the marker can insert comments electronically.