

Object Oriented Programming Assignment 3

1.0

Chat with QuickSearch

By Advait Raykar, Vishwa Kalyanaraman and Parv Kapoor
Roll Numbers 28, 11 and 24
CSE D
3rd Semester

Contents

| | | |
|----------|--|----------|
| 1 | Purpose of the project and concepts used | 1 |
| 2 | Hierarchical Index | 3 |
| 2.1 | Class Hierarchy | 3 |
| 3 | Class Index | 5 |
| 3.1 | Class List | 5 |
| 4 | Class Documentation | 7 |
| 4.1 | Client Class Reference | 7 |
| 4.1.1 | Detailed Description | 8 |
| 4.1.2 | Constructor & Destructor Documentation | 8 |
| 4.1.2.1 | Client(String host) | 8 |
| 4.1.3 | Member Function Documentation | 8 |
| 4.1.3.1 | allowTyping(final boolean flag) | 9 |
| 4.1.3.2 | beginProgram() | 9 |
| 4.1.3.3 | connectToServer() | 9 |
| 4.1.3.4 | endProgram() | 9 |
| 4.1.3.5 | formatForGoogle(String plain) | 9 |
| 4.1.3.6 | sendMessage(String message) | 10 |
| 4.1.3.7 | setupStreams() | 10 |
| 4.1.3.8 | showMessage(final String message) | 10 |
| 4.1.3.9 | whileChatting() | 10 |
| 4.2 | ClientDemo Class Reference | 11 |
| 4.2.1 | Detailed Description | 11 |

| | | |
|--------------|--|-----------|
| 4.3 | Server Class Reference | 12 |
| 4.3.1 | Detailed Description | 13 |
| 4.3.2 | Constructor & Destructor Documentation | 13 |
| 4.3.2.1 | Server() | 13 |
| 4.3.3 | Member Function Documentation | 13 |
| 4.3.3.1 | allowTyping(final boolean flag) | 13 |
| 4.3.3.2 | beginProgram() | 13 |
| 4.3.3.3 | endProgram() | 14 |
| 4.3.3.4 | formatForGoogle(String plain) | 14 |
| 4.3.3.5 | sendMessage(String message) | 14 |
| 4.3.3.6 | setupStreams() | 14 |
| 4.3.3.7 | showMessage(final String text) | 14 |
| 4.3.3.8 | waitForConnection() | 15 |
| 4.3.3.9 | whileChatting() | 15 |
| 4.4 | ServerStart Class Reference | 15 |
| 4.4.1 | Detailed Description | 15 |
| Index | | 17 |

Chapter 1

Purpose of the project and concepts used

This application aims to connect users over the medium of an online chat interface

There exists a Client, and a Server. The client instance connects to the server instance, thus allowing communication to take place. The communication takes place via Object Input and Output streams.

The users, can also type in search queries which will pop open an instance of a browser with the result for the query. This feature allows for users quickly look up facts, figures or to simply quell their curiosity mid conversation.

The following concepts have been utilised:

1. Inheritance
2. Error Handling
3. Swings
4. Streams
5. Abstraction

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

| | |
|-----------------------|----|
| ClientDemo | 11 |
| ServerStart | 15 |
| JFrame | |
| Client | 7 |
| Server | 12 |

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

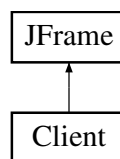
| | |
|------------------------------|----|
| Client | 7 |
| ClientDemo | 11 |
| Server | 12 |
| ServerStart | 15 |

Chapter 4

Class Documentation

4.1 Client Class Reference

Inheritance diagram for Client:



Public Member Functions

- **Client** (String host)
- void **beginProgram** ()

Private Member Functions

- void **connectToServer** () throws IOException
- void **setupStreams** () throws IOException
- void **whileChatting** () throws IOException
- void **endProgram** ()
- void **sendMessage** (String message)
- void **showMessage** (final String message)
- void **allowTyping** (final boolean flag)
- String **formatForGoogle** (String plain)

Private Attributes

- JTextField **userText**
- JTextArea **chatWindow**
- ObjectOutputStream **output**
- ObjectInputStream **input**
- String **message** = ""
- String **serverIP**
- Socket **connection**
- JTextField **search**
- Browser **br**

4.1.1 Detailed Description

The Main Client (p. 7) Class

This class is called when a **Server** (p. 12) instance is launched.

Author

Advait Raykar, Vishwa Kalyanaraman, Parv Kapoor

Version

1.0

Since

5th November 2017

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Client.Client (String *host*)

The constructor initialises window elements, and adds the appropriate event handlers.

Parameters

| | |
|-------------|-------------------------------------|
| <i>host</i> | The host id for finding the server. |
|-------------|-------------------------------------|

4.1.3 Member Function Documentation

4.1.3.1 void Client.allowTyping (final boolean *flag*) [private]

Should typing be allowed?

Parameters

| | |
|-------------|------------------|
| <i>flag</i> | True/False value |
|-------------|------------------|

4.1.3.2 void Client.beginProgram ()

This method when called does the following three things:

- Tries to connect to the server
- Sets up all the required streams
- Makes sure messages are sent and recieved properly

4.1.3.3 void Client.connectToServer () throws IOException [private]

Attempts to connect to the server at PORT 5050 which is what is used in the **Server** (p. 12) source.

Exceptions

| | |
|--------------------|--|
| <i>IOException</i> | |
|--------------------|--|

4.1.3.4 void Client.endProgram () [private]

Closes all the streams after severing connections with the client.

4.1.3.5 String Client.formatForGoogle (String *plain*) [private]

Formats the user query into a form that yeilds search results in the browser

Parameters

| | |
|--------------|-------------------------------|
| <i>plain</i> | The plaintext to be searched. |
|--------------|-------------------------------|

4.1.3.6 void Client.sendMessage (String *message*) [private]

Sends the message to the output stream object

Parameters

| | |
|----------------|--|
| <i>message</i> | This is the message that is to be sent downstream. |
|----------------|--|

4.1.3.7 void Client.setupStreams () throws IOException [private]

Creates an

`ObjectOutputStream`

object and an

`ObjectInputStream`

object for sending and receiving messages

Exceptions

| | |
|--------------------|--|
| <i>IOException</i> | |
|--------------------|--|

4.1.3.8 void Client.showMessage (final String *message*) [private]

This prints out the message on the chat window so it's visible to the user.

Parameters

| | |
|-------------|--------------------------|
| <i>text</i> | The text to be displayed |
|-------------|--------------------------|

4.1.3.9 void Client.whileChatting () throws IOException [private]

Manages calls to

`showMessage()`

Note, the program quits connection when user inputs *xoxo* which is slang for '*hearts and kisses*'

Exceptions

| | |
|--------------------|--|
| <i>IOException</i> | |
|--------------------|--|

The documentation for this class was generated from the following file:

- Client.java

4.2 ClientDemo Class Reference

Static Public Member Functions

- static void **main** (String[] args)

4.2.1 Detailed Description

The class that calls the **Client** (p. 7) class.

Instantiates the **Client** (p. 7) class object and starts the **Client** (p. 7). The host passed in the constructor is ip of the localhost, so that the program can run locally. This can be changed per convenience.

Author

Advait Raykar, Vishwa Kalyanaraman, Parv Kapoor

Version

1.0

Since

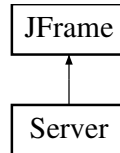
5th November 2017

The documentation for this class was generated from the following file:

- ClientDemo.java

4.3 Server Class Reference

Inheritance diagram for Server:



Public Member Functions

- **Server** ()
- void **beginProgram** ()

Private Member Functions

- void **waitForConnection** () throws IOException
- void **setupStreams** () throws IOException
- void **whileChatting** () throws IOException
- void **endProgram** ()
- void **sendMessage** (String message)
- void **showMessage** (final String text)
- void **allowTyping** (final boolean flag)
- String **formatForGoogle** (String plain)

Private Attributes

- JTextField **userText**
- JTextArea **chatWindow**
- ObjectOutputStream **output**
- ObjectInputStream **input**
- JTextField **search**
- Browser **br**
- ServerSocket **server**
- Socket **connection**

4.3.1 Detailed Description

The Main **Server** (p. 12) Class

This class is called when a **Server** (p. 12) instance is launched.

Author

Advait Raykar, Vishwa Kalyanaraman, Parv Kapoor

Version

1.0

Since

5th November 2017

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `Server.Server ()`

This is the constructor, which initialises the actual window, adds the necessary action listeners, and adds the swing elements to the window.

4.3.3 Member Function Documentation

4.3.3.1 `void Server.allowTyping (final boolean flag)` `[private]`

Should typing be allowed?

Parameters

| | |
|-------------|------------------|
| <i>flag</i> | True/False value |
|-------------|------------------|

4.3.3.2 `void Server.beginProgram ()`

This method when called does the following three things:

- Waits for a client to connect
- Sets up all the required streams
- Makes sure messages are sent and recieved properly

4.3.3.3 void Server.endProgram () [private]

Closes all the streams after severing connections with the client.

4.3.3.4 String Server.formatForGoogle (String *plain*) [private]

Formats the user query into a form that yeilds search results in the browser

Parameters

| | |
|--------------|-------------------------------|
| <i>plain</i> | The plaintext to be searched. |
|--------------|-------------------------------|

4.3.3.5 void Server.sendMessage (String *message*) [private]

Sends the message to the output stream object

Parameters

| | |
|----------------|--|
| <i>message</i> | This is the message that is to be sent downstream. |
|----------------|--|

4.3.3.6 void Server.setupStreams () throws IOException [private]

Creates an

ObjectOutputStream

object and an

ObjectInputStream

object for sending and recieving messages

Exceptions

| | |
|-------------|--|
| IOException | |
|-------------|--|

4.3.3.7 void Server.showMessage (final String *text*) [private]

This prints out the message on the chat window so it's visible to the user.

Parameters

| | |
|-------------|--------------------------|
| <i>text</i> | The text to be displayed |
|-------------|--------------------------|

4.3.3.8 void `Server.waitForConnection ()` throws `IOException` [private]

Waits for connection on starting the server.

Exceptions

| | |
|--------------------|--|
| <i>IOException</i> | |
|--------------------|--|

4.3.3.9 void `Server.whileChatting ()` throws `IOException` [private]

Manages calls to

`showMessage ()`

Note, the program quits connection when user inputs *xoxo* which is slang for '*hearts and kisses*'

Exceptions

| | |
|--------------------|--|
| <i>IOException</i> | |
|--------------------|--|

The documentation for this class was generated from the following file:

- `Server.java`

4.4 ServerStart Class Reference

Static Public Member Functions

- static void **main** (`String[]` args)

4.4.1 Detailed Description

The class that calls the **Server** (p. 12) class.

Instantiates the **Server** (p. 12) class object and starts the server.

Author

Advait Raykar, Vishwa Kalyanaraman, Parv Kapoor

Version

1.0

Since

5th November 2017

The documentation for this class was generated from the following file:

- ServerStart.java

Index

- allowTyping
 - Client, 8
 - Server, 13
- beginProgram
 - Client, 9
 - Server, 13
- Client, 7
 - allowTyping, 8
 - beginProgram, 9
 - Client, 8
 - connectToServer, 9
 - endProgram, 9
 - formatForGoogle, 9
 - sendMessage, 9
 - setupStreams, 10
 - showMessage, 10
 - whileChatting, 10
- ClientDemo, 11
- connectToServer
 - Client, 9
- endProgram
 - Client, 9
 - Server, 13
- formatForGoogle
 - Client, 9
 - Server, 14
- sendMessage
 - Client, 9
 - Server, 14
- Server, 12
 - allowTyping, 13
 - beginProgram, 13
 - endProgram, 13
 - formatForGoogle, 14
 - sendMessage, 14
 - Server, 13
 - setupStreams, 14
 - showMessage, 14
 - waitForConnection, 15
 - whileChatting, 15
- ServerStart, 15
- setupStreams
 - Client, 10
 - Server, 14
- showMessage
 - Client, 10
 - Server, 14
- waitForConnection
 - Server, 15
- whileChatting
 - Client, 10
 - Server, 15