**Server**

User instructions:

1. The server must be turned on at first.

2.Getting these flags must be correspondent actions.

|  |  |
| --- | --- |
| Flag | Function |
| --Encode | Change the function to encode |
| -- Decode | Change the function to decode |
| -- Shutdown | Shutdown the server |

**Client**

User instructions:

1. Make sure the server has already turned on.(If there is no any server, show error message.)



2. Type the sentence and click enter. Will get the sentence after de/encoding.

3.Sending the specific flags can control the server.

|  |  |
| --- | --- |
| Flag | Function |
| --Encode | Change the function to encode |
| -- Decode | Change the function to decode |
| -- Shutdown | Shutdown the server |

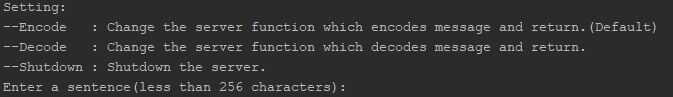
4. The maximum length of sentence is 256. (If it’s more than 256 characters, it will a show warning message.



**Screenshot**

Client:

At beginning.



Enter “Hello World”, and then encode to “Ifmmp!Xpsme”



Enter “Test ”, and then encode to “Uftu!!”



Change to decode mode.



Enter “Uftu!!”, and then decode to “Test ”.



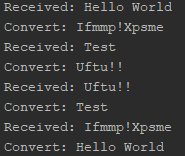
Enter “Ifmmp!Xpsme”, and then decode to “Hello World”.



Change to encode mode.



Server:



**Source code:**

Server source code

class ServerClass {  
  
 //region Indication  
 String EncodeMessage = "--Encode";  
 String DecodeMessage = "--Decode";  
 String ShutdownMessage = "--Shutdown";  
 //endregion  
  
  
 void ServerStart() throws Exception {  
 String clientSentence =null;  
 String SendToClientSentence;  
 ServerSocket welcomeSocket = new ServerSocket(9576);  
 String newStr;  
 Boolean EncodeFlag = true;  
  
 Socket connectionSocket = welcomeSocket.accept();  
 BufferedReader inFromClient = new BufferedReader(new InputStreamReader( connectionSocket.getInputStream() ));  
 DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

while (true) {  
 clientSentence = inFromClient.readLine();  
 if (clientSentence.equals(ShutdownMessage))  
 break;  
 else if(clientSentence.equals(EncodeMessage)){  
 EncodeFlag = true;  
 newStr="Now, it's available to encode sentences.";  
 }  
 else if(clientSentence.equals(DecodeMessage)){  
 EncodeFlag = false;  
 newStr="Now, it's available to decode sentences.";  
 }  
 else {  
 System.*out*.println("Received: " + clientSentence);  
 if (EncodeFlag)  
 newStr = StringEncoder(clientSentence);  
 else  
 newStr = StringDecoder(clientSentence);  
 System.*out*.println("Convert: " + newStr);  
 }  
  
 SendToClientSentence = newStr + '\n';  
 outToClient.writeBytes(SendToClientSentence);  
 }  
 inFromClient.close();  
 outToClient.close();  
 connectionSocket.close();  
  
 }  
 String StringEncoder(String input){  
 String NewStr = "";  
 char NewChar ;  
 char achar ;  
  
 for (int index =0; index<input.length();index++){  
 achar = input.charAt(index);  
 if ((int)achar == 255)  
 NewChar = (char)0;  
 else NewChar = (char)((achar)+1);  
 NewStr += NewChar;  
 }  
 return NewStr;  
 }  
 String StringDecoder(String input){  
 String NewStr = "";  
 char NewChar ;  
 char achar ;  
  
 for (int index =0; index<input.length();index++){  
 achar = input.charAt(index);  
 if ((int)achar == 0)  
 NewChar = (char)255;  
 else NewChar = (char)((achar)-1);  
 NewStr += NewChar;  
 }  
 return NewStr;  
 }  
  
}

**Client source code:**

public class ClientClass {  
  
 //region Indication  
 String EncodeMessage = "--Encode";  
 String DecodeMessage = "--Decode";  
 String ShutdownMessage = "--Shutdown";  
 //endregion  
  
 public void ClientStart() throws Exception {  
 String sentence;  
 String ReceivedSentence;  
  
 Socket clientSocket = null;  
 try {  
 clientSocket = new Socket("localhost", 9576);  
 } catch (ConnectException e) {  
 System.out.println("Fail to connect to the server.");  
 return;  
 }  
  
 BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));  
 DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());  
 BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  
  
 System.out.println("Setting:");  
 System.out.println(EncodeMessage+" : Change the server function which encodes message and return.(Default)");  
 System.out.println(DecodeMessage+" : Change the server function which decodes message and return.");  
 System.out.println(ShutdownMessage+" : Shutdown the server.");  
  
 while(true){  
 System.out.print("Enter a sentence(less than 256 characters): ");  
 sentence = inFromUser.readLine();  
  
 if (sentence.length()> 256){  
 System.out.println("The sentence is longer than 256. Slice the sentence.");  
 continue;  
 }  
 if ((SentenceParser(sentence) ==1))  
 break;  
 outToServer.writeBytes(sentence + '\n');  
 ReceivedSentence = inFromServer.readLine();  
 System.out.println("From Server: " + ReceivedSentence);  
 }  
  
 outToServer.writeBytes(ShutdownMessage + '\n');  
 inFromUser.close();  
 inFromServer.close();  
 outToServer.close();  
 clientSocket.close();  
 }  
  
 int SentenceParser(String sentence){  
 if (sentence.equals(ShutdownMessage))  
 return 1;  
 else return 0;  
 }  
}