Homework1

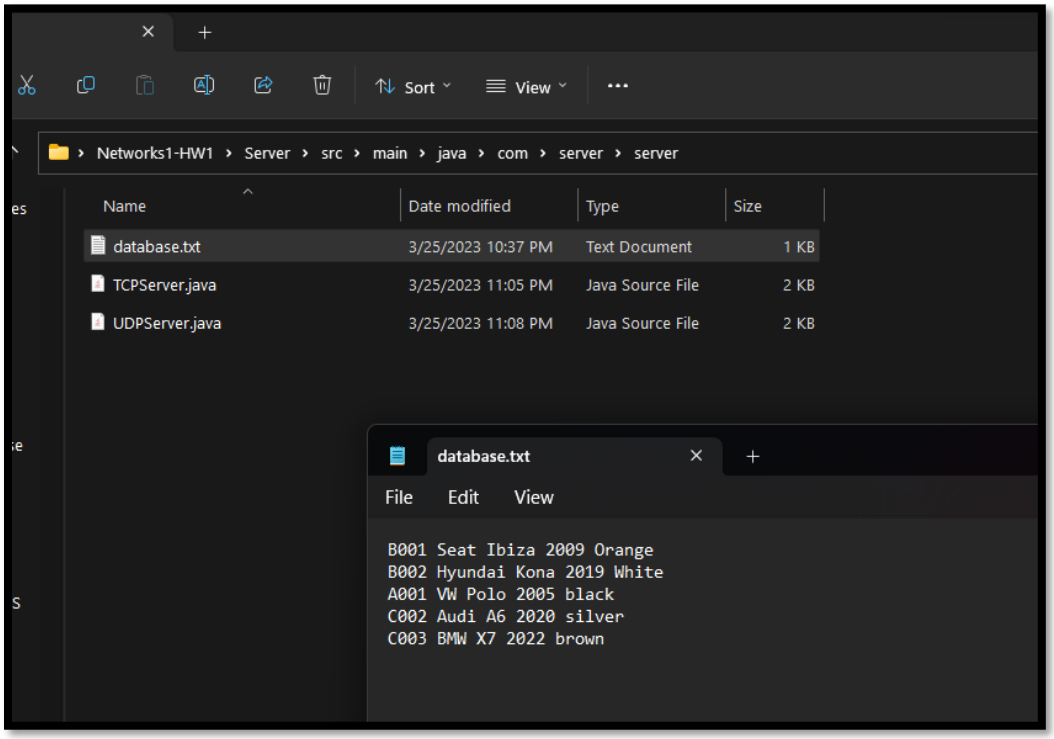
Homework1, 2nd semester 2023

25-3-2023
Networks1

Riham Muneer Katout
Dr. Saed Tarabieh

TCP part

- Stores the data in a file



- On the server side:

  - Create a welcoming socket at port 6789, and wait for contact by client

```
1   package com.server.server;
2   import java.io.*;
3   import java.net.*;
4   public class TCPServer {
5       public static void main(String argv[]) throws Exception
6       {
7           String clientSentence;
8           String answer;
9           ServerSocket welcomeSocket = new ServerSocket( port: 6789);
10          while(true) {
11              int flag=0;
12              Socket connectionSocket = welcomeSocket.accept();
```

- Create input/output streams attached to socket and read the line from the client

```
13              BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
14              DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
15              clientSentence = inFromClient.readLine();
16
```

- Read the data from “database.txt” file, at each line compare the ID with the one Given by the user, if they are matching, return the rest of the line as “make, model, Year, color”, if there is no matching ID then the flag will remain 0, so print “Vehicle is not found”, after both cases, write out line to socket.

```
try {
    FileReader fileReader = new FileReader( fileName: "C:\\Users\\Msys\\Desktop\\Networks1-HW1\\Server\\src\\main\\java\
    BufferedReader bufferedReader = new BufferedReader(fileReader);
    String line;
    while ((line = bufferedReader.readLine()) != null) {
        String []input = line.split( regex: " ");
        if(clientSentence.equals(input[0])){
            answer = input[1] + ", " + input[2] + ", " + input[3] + ", " + input[4] + '\n';
            outToClient.writeBytes(answer);
            flag=1;
            break;
        }
    }
    bufferedReader.close();
    fileReader.close();
} catch (IOException e) {
    e.printStackTrace();
}
if(flag==0) {
    answer = "Vehicle is not found" + '\n';
    outToClient.writeBytes(answer);
}
```

- On the client side
  - IP Address

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a0ed:13da:1926:981a%18
   IPv4 Address. . . . . . . . . . . : 192.168.1.104
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.254
```

  - The process: create input stream, client socket and connect to the server, and input/output stream attached to the socket read the input and send it to the server, read the response from the server and print the result. Repeat it until the user enters 0.

```java
package com.client.client;
import java.io.*;
import java.net.*;
public class TCPClient {
    public static void main(String argv[]) throws Exception
    {
        String sentence;
        String modifiedSentence;

        while(true) {
            System.out.println("Enter 0 to exit");
            Socket clientSocket = new Socket( host: "192.168.1.104",  port: 6789);
            BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
            sentence = inFromUser.readLine();
            if(sentence.equals("0")){
                clientSocket.close();
                break;
            }
            BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
            DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
            outToServer.writeBytes( s: sentence + '\n');
            modifiedSentence = inFromServer.readLine();
            System.out.println("FROM SERVER: " + modifiedSentence);
            clientSocket.close();
        }
    }
}
```

➢ Test it


"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:64635', transport: 'socket'
Enter 0 to exit
B001
FROM SERVER: Seat, Ibiza, 2009, Orange
Enter 0 to exit
A001
FROM SERVER: VW, Polo, 2005, black
Enter 0 to exit
D091
FROM SERVER: Vehicle is not found
Enter 0 to exit
0
Disconnected from the target VM, address: '127.0.0.1:64635', transport: 'socket'
Process finished with exit code 0

UDP part

- On the server side
    - Create datagram socket at port 9876, for each operation, Create space for received datagram then receive it, Get IP address port #, of sender using getAddress(), getPort() methods

```
public class UDPServer {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket serverSocket = new DatagramSocket( port: 9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while(true)
        {
            int flag=0;
            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            String sentence = new String(receivePacket.getData());
            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();
```

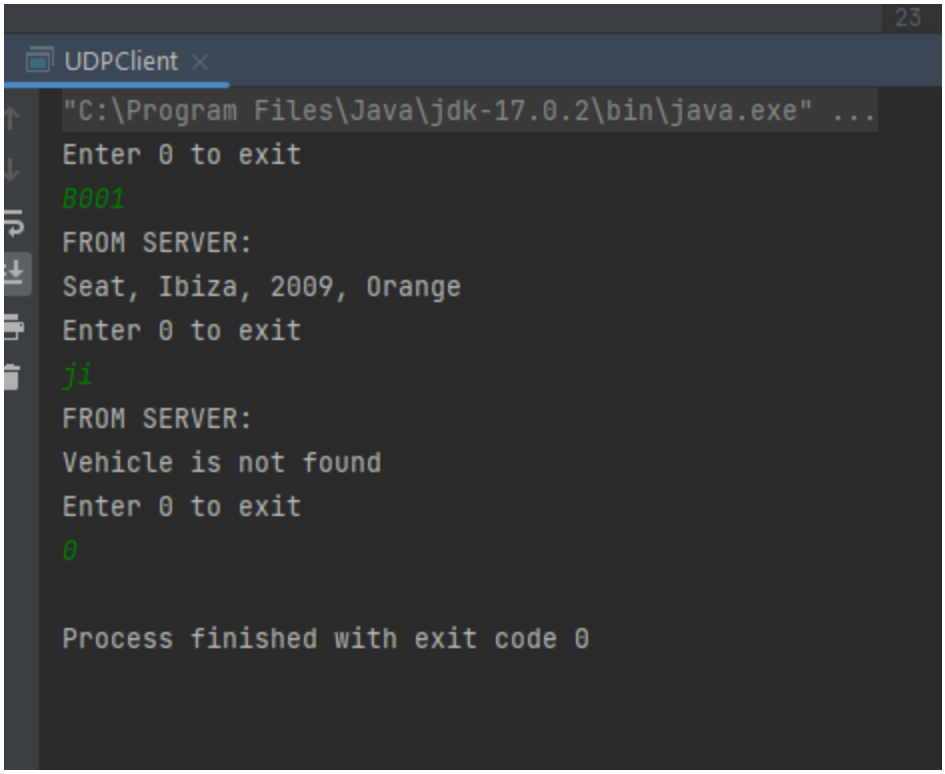
- Read the data from “database.txt” file, at each line compare the ID with the one Given by the user, if they are matching, return the rest of the line as “make, model, Year, color”, if there is no matching ID then the flag will remain 0, so print “Vehicle is not found”, after both cases, write out line to socket.

```
String answer = sentence.toUpperCase();
try {
    FileReader fileReader = new FileReader( fileName: "C:\\Users\\Msys\\Desktop\\Networks1-HW1\\Server\\src\\n
    BufferedReader bufferedReader = new BufferedReader(fileReader);
    String line;
    String sen="";
    for(int i=0;i<sentence.length();i++){
        char in=sentence.charAt(i);
        if(in>='a'&&in<='z' || in>='A'&&in<='Z' || in>='0'&&in<='9' || in==','||in==' ')
            sen+=in;
    }
    while ((line = bufferedReader.readLine()) != null) {
        String []input = line.split( regex: " ");
        if(sen.equals(input[0])){
            answer = input[1] + ", " + input[2] + ", " + input[3] + ", " + input[4] + '\n';
            sendData = answer.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);
            serverSocket.send(sendPacket);
            flag=1;
            break;
        }
    }
    bufferedReader.close();
    fileReader.close();
} catch (IOException e) {
    e.printStackTrace();
}
if(flag==0) {
    answer = "Vehicle is not found" + '\n';
    sendData = answer.getBytes();
    DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);
    serverSocket.send(sendPacket);
}
```

- On the client side
  - The process: create input stream, client socket, then Translate hostname to IP address using DNS. Create datagram with data-to-send, length, IP address, port Send datagram to server, Read datagram from server
    Repeat it until the user enters 0.

```
while(true) {
    System.out.println("Enter 0 to exit");
    BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
    DatagramSocket clientSocket = new DatagramSocket();
    InetAddress IPAddress = InetAddress.getByName( host: "192.168.1.104");
    byte[] sendData = new byte[1024];
    byte[] receiveData = new byte[1024];
    String sentence = inFromUser.readLine();
    if(sentence.equals("0")){
        clientSocket.close();
        break;
    }
    sendData = sentence.getBytes();
    DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress,  port: 9876);
    clientSocket.send(sendPacket);
    DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
    clientSocket.receive(receivePacket);
    String result = new String(receivePacket.getData());
    System.out.println("FROM SERVER:" );
    for(int i=0;i<result.length();i++){
        char in=result.charAt(i);
        if(in>='a'&&in<='z' || in>='A'&&in<='Z' || in>='0'&&in<='9' || in==','||in==' ')
            System.out.print(in);
    }
    System.out.println();
    clientSocket.close();
}
```

- Test it


UDPClient
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" ...
Enter 0 to exit
B001
FROM SERVER:
Seat, Ibiza, 2009, Orange
Enter 0 to exit
ji
FROM SERVER:
Vehicle is not found
Enter 0 to exit
0
Process finished with exit code 0