Computer Networks: Fall 2022 Assignment #1

Computer Networks: Fall 2022 Programming Assignment 1

2021045796 김도겸

UDP의 Multicast를 이용한 P2P 방식의 오픈 채팅 프로그램을 구현한다

1. iml(Intelli J IDEA Module)

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
  <component name="NewModuleRootManager" inherit-compiler-output="true">
    <exclude-output />
    <content url="file://$MODULE_DIR$">
      <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
    </content>
    <orderEntry type="inheritedJdk" />
    <orderEntry type="sourceFolder" forTests="false" />
  </component>
</module>
```

2. 각 파일 설명

1) Peer.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.security.NoSuchAlgorithmException;


public class Peer {

    public static void main(String[] args) throws NoSuchAlgorithmException {
        InetAddress inetAddress = null;
        BufferedReader br = null;
        UDPMultiReceive udpMultiReceive = null;
        UDPMultiSend udpMultiSend = null;
        int portNumber = 0;
        String strId = "";
```

```
        try {
            // 1. IP 주소, port 번호 입력 사항 확인
            br = new BufferedReader(new InputStreamReader(System.in));
            String port = args[0];
            portNumber = Integer.parseInt(port);
            System.out.print("running - port number : " + portNumber + "\n");
            String chatName;
            String[] chatNameArr;
            while(true) {
                chatName = br.readLine();
                chatNameArr = chatName.split(" ");
                if(chatName.equals("#EXIT")) {
                    System.out.println("종료합니다.");
                    System.exit(0);
                }
                else if(chatNameArr.length == 3 && chatNameArr[0].equals("#JOIN")) {
                    break;
                }
                else {
                    continue;
                }
            }
            String address = chatNameArr[1];
            SHA256 sha256 = new SHA256();
            strId = sha256.encrypt(address);
            byte[] id = strId.getBytes();
            String finalAddress = "225." + id[61] + "." + id[62] + "." + id[63];
            inetAddress = InetAddress.getByName(finalAddress);
            // 2. 이름 입력
            br = new BufferedReader(new InputStreamReader(System.in));
            strId = chatNameArr[2];
            System.out.println("*********** " + strId + "님 접속 ***********");
            // 3. 데이터 받기
            udpMultiReceive = new UDPMultiReceive(inetAddress, portNumber);
            udpMultiReceive.start();
            // 4. 데이터 보내기
            udpMultiSend = new UDPMultiSend(inetAddress, portNumber, strId);
            udpMultiSend.start();

        } catch (UnknownHostException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Main 실행 파일, port Number을 입력받고, 채팅방 이름과 사용자의 이름을 입력받는다. data send와 data receive를 관리한다.

입력받은 채팅방 이름을 SHA256을 호출하여 채팅방 주소로 변환한다.

사용자에게 채팅방에 들어왔음을 알린다.

#EXIT를 입력받았을 때 프로그램을 종료한다.

#JOIN을 입력받았을 때 뒤에 채팅방 이름과 사용자 이름이 입력되었는지 확인된 뒤 채팅방에 연결한다.

2) UDPMultiReceived.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;


public class UDPMultiReceive extends Thread {
    private InetAddress inetAddress = null;
    private int portNumber = 0;
    private MulticastSocket multicastSocket = null;
    private DatagramPacket datagramPacket = null;

    public UDPMultiReceive(InetAddress tmpInetAddress, int tmpportNumber) {
        inetAddress = tmpInetAddress;
        portNumber = tmpportNumber;
    }

    public void run() {
        // 1. Data를 받을 버퍼 생성
        byte[] buffer = new byte[512];
        try {
            // 2. DatagramPacket을 받기 위한 Socket 생성
            multicastSocket = new MulticastSocket(portNumber);
            // 3. 그룹 등록 - 통신 가능하게 함
            multicastSocket.joinGroup(inetAddress);
            // 메시지 계속 받음
            while(true) {
                // 4. Data를 받을 Packet 생성
                datagramPacket = new DatagramPacket(buffer, buffer.length);
                // 5. 멀티캐스트에 존재하는 메시지 받음
                multicastSocket.receive(datagramPacket);
                // 6. 수신된 메시지 출력
                String msg = new String(datagramPacket.getData(), 0, datagramPacket.getLength());
                System.out.println(msg);
            }
        } catch (IOException e) {
            System.err.println(e);
            System.exit(0);
        } finally {
            multicastSocket.close();
        }
    }
}
```

수신을 담당하는 파일.

buffer와 Packet을 받기 위한 Socket을 생성한다. Group을 설정해주어 생성되는 Message를 계속 받아준다.

송신자 이름과 내용이 담겨진 String msg를 출력한다.

데이터를 수신하기 위한 512byte의 버퍼를 생성한다.

3) UDPMultiSend.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;


public class UDPMultiSend extends Thread {
    private InetAddress inetAddress = null;
    private int portNumber = 0;
    private String strId = "";
    private DatagramSocket datagramSocket = null;
    private DatagramPacket datagramPacket = null;
    private BufferedReader br = null;

    public UDPMultiSend(InetAddress tmpInetAddress, int tmpportNumber, String tmpstrId) {
        inetAddress = tmpInetAddress;
        portNumber = tmpportNumber;
        strId = tmpstrId;
    }

    public void exit() {
        datagramSocket.close();
        System.exit(0);
    }

    public void run() {
        // 1. Data를 보낼 버퍼 생성
        byte[] buffer = new byte[512];
        try {
            // 2. Socket 열기
            datagramSocket = new DatagramSocket();
            // 3. Server로 메시지 전송을 위한 입력 스트림 생성
            br = new BufferedReader(new InputStreamReader(System.in));
            // 메시지 계속 전송
            while(true) {
                // 4. 메시지 입력
                String temp = br.readLine();
                String msg = strId + ": " + temp;
                byte [] b = temp.getBytes();
                buffer = msg.getBytes();
                if(temp.equals("#EXIT")) {
                    throw new CustomException("종료합니다.");
                }
                if(b[0] == '#') {
                    throw new CustomException2("명령어가 아닙니다.");
                }
                // 6. DatagramPacket에 각 정보를 담고 전송
                datagramPacket = new DatagramPacket(buffer, buffer.length, inetAddress, portNumber);
                datagramSocket.send(datagramPacket);
            }
        } catch(IOException ie) {
            System.out.println("send 오류");
        } catch(CustomException ce) {
            System.out.println(ce.getMessage());
            exit();
        } catch (CustomException2 c2) {
            System.out.println(c2.getMessage());
            run();
        }
        finally {
            datagramSocket.close();
        }
```

```
    }
}
```

송신을 담당하는 파일

buffer와 socket을 생성하여 지속적으로 입력받은 메시지를 전송한다.

String msg에는 송신자의 이름을 포함한다.

#EXIT가 입력되었을 때 예외 처리를 해주어 프로그램을 종료해준다.

메시지의 첫 글자가 #일 때, 예외 처리를 해주어 “명령어가 아닙니다.”를 출력해준다. 이 경우에는 프로그램이 종료되지 않는다.

명령어 이외에 Exception이 발생하였을 때 오류 메시지를 출력한다.

4) SHA256.java

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SHA256 {

    public String encrypt(String text) throws NoSuchAlgorithmException {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        md.update(text.getBytes());

        return bytesToHex(md.digest());
    }

    private String bytesToHex(byte[] bytes) {
        StringBuilder builder = new StringBuilder();
        for (byte b : bytes) {
            builder.append(String.format("%02x", b));
        }
        return builder.toString();
    }

}
```

입력받은 채팅방 이름을 SHA-256에 기반하여 변경한다.

String 형태로 return하며, Peer.java에서 byte로 변경하여 ip address를 지정하는데 사용한다.

64byte의 크기로 리턴하기 때문에 뒤의 배열의 뒤 3개의 byte 정보를 이용하여 주소를 만드는데 사용한다.

5) CustomException.java

```
public class CustomException extends RuntimeException {

    // 1. 매개 변수가 없는 기본 생성자
    CustomException() {

    }

    // 2. 예외 발생 원인(예외 메시지)을 전달하기 위해 String 타입의 매개변수를 갖는 생성자
    CustomException(String message) {
        super(message); // RuntimeException 클래스의 생성자를 호출합니다.
    }
}
```

#EXIT 예외 처리를 위한 파일

6) CustomException2.java

```
public class CustomException2 extends RuntimeException {

    // 1. 매개 변수가 없는 기본 생성자
    CustomException2() {

    }

    // 2. 예외 발생 원인(예외 메시지)을 전달하기 위해 String 타입의 매개변수를 갖는 생성자
    CustomException2(String message) {
        super(message); // RuntimeException 클래스의 생성자를 호출합니다.
    }
}
```

#EXIT를 제외한 #으로 시작하는 메시지를 예외 처리해주기 위한 파일

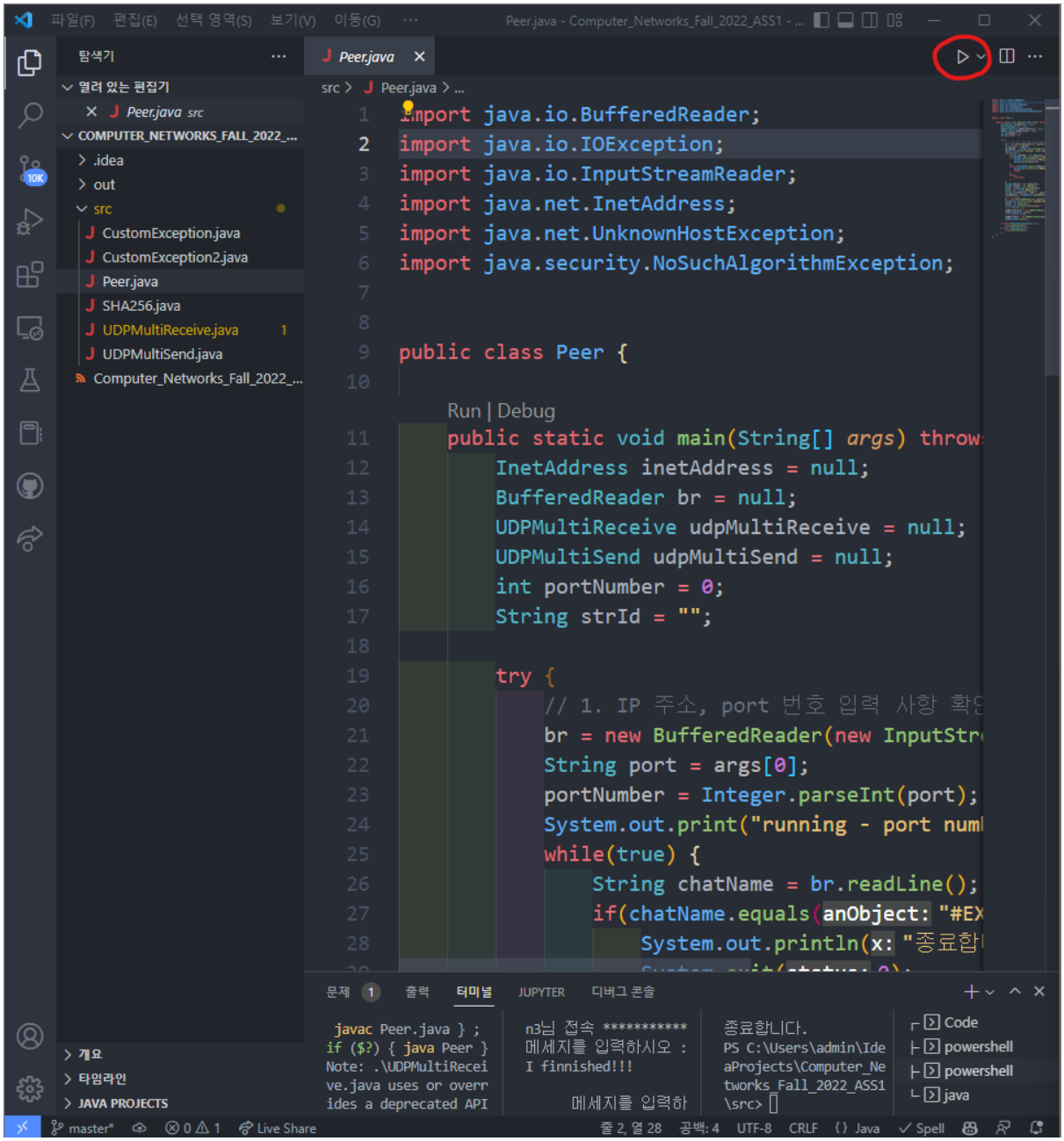
프로그램이 종료되지 않는다.

3. 프로그램 실행

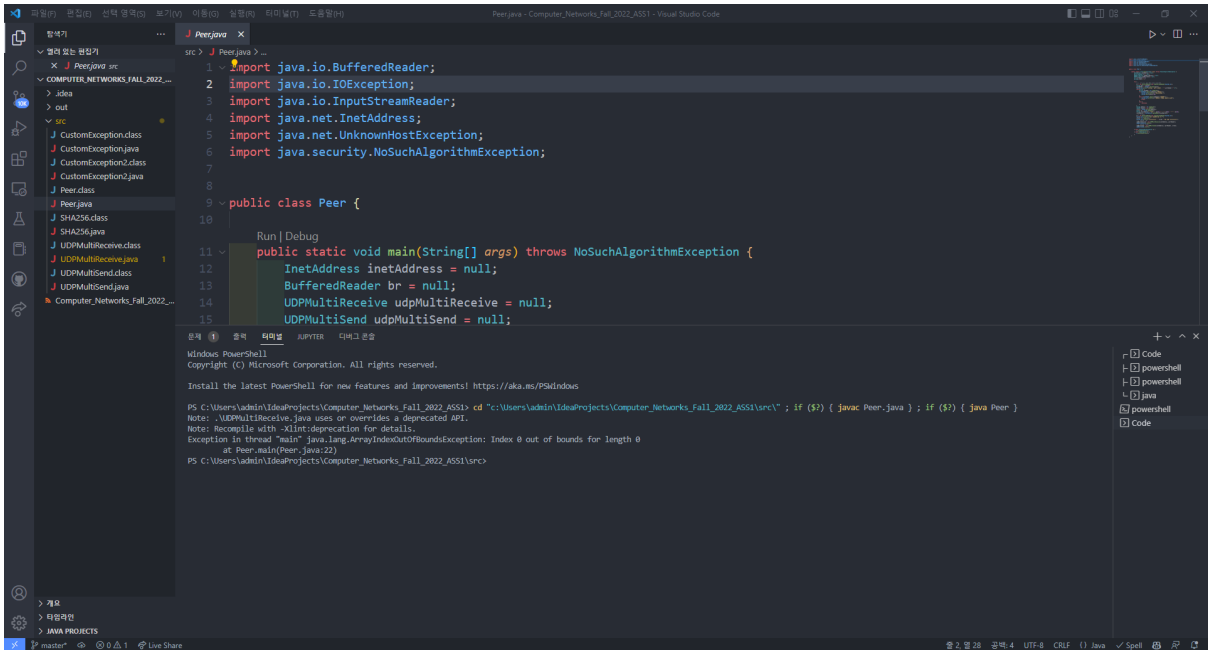
파일 압축 해제

\Assignment1_2021045796_김도겸\Computer_Networks_Fall_2022_ASS1\src

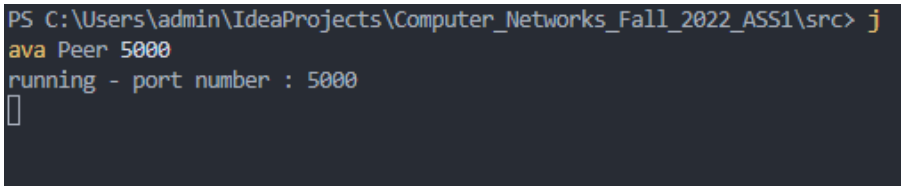
로 들어간다.



Peer.java에 들어가서 run code를 한다.(빨간색 동그라미로 표시된 부분)



Peer.class를 비롯한 6개의 class가 생성된다.



터미널에 들어가서

> 💡 java Peer PortNo

을 입력하면 프로그램이 시작된다.

PortNo는 Port Number로 int 형태의 Port Number를 입력해주어야한다.

ex) java Peer 5000

ex) java Peer 4000

프로그램이 제대로 실행되었다면 “running - port number : “와 함께 Port Number을 출력한다.

4. 구현된 기능

1) #JOIN

#JOIN (채팅방 이름) (사용자 이름)

을 입력받을 때 프로그램이 시작한다.

#JOIN 만 입력받았을 때는 작동하지 않는다.

소문자 #join에는 작동하지 않는다.

#JOIN과 #EXIT를 제외한 나머지 명령어에는 반응하지 않는다.

채팅방과 사용자 이름에 띄어쓰기가 포함되어서는 안된다.

```
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
123123
kakao
#join kakao kimdogyeom
#JOIN
#JOIN kakao kimdogyeom
********** kimdogyeom님 접속 **********
```

2) #EXIT

채팅방에 들어가기 전에 #EXIT를 입력하여 종료할 수 있다.

소문자 #exit에는 작동하지 않는다.

```
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
#exit
#EXIT
종료합니다.
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src>
```

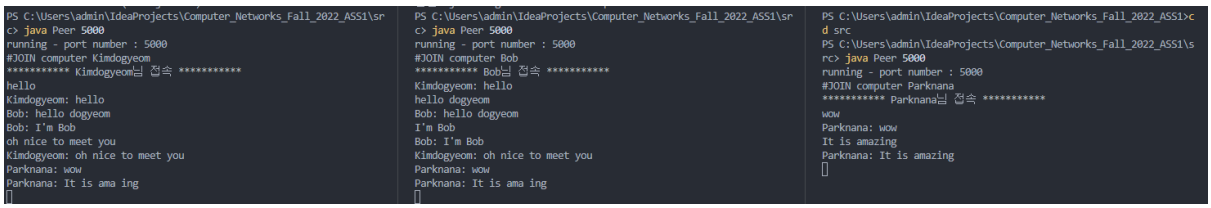
채팅방에 들어가서도 #EXIT를 입력하여 종료할 수 있다.

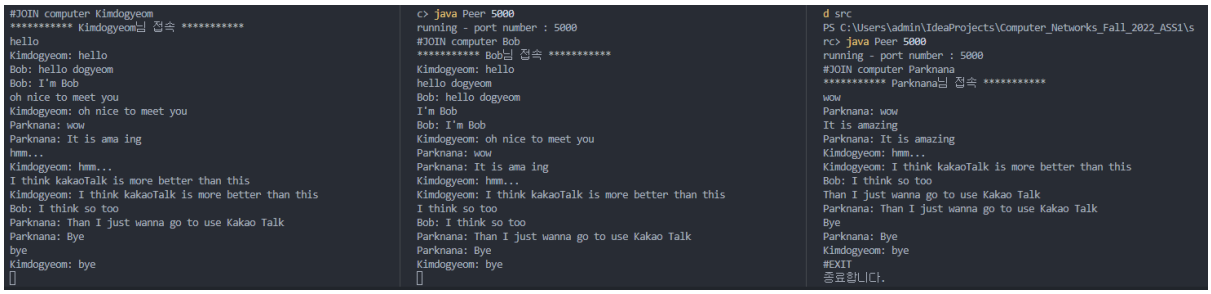
소문자 #exit에는 작동하지 않는다.

```
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
123123
kakao
#join kakao kimdogyeom
#JOIN
#JOIN kakao kimdogyeom
********** kimdogyeom님 접속 **********
hello
kimdogyeom: hello
#exit
명령어가 아닙니다.
#EXIT
종료합니다.
```

3) 채팅


********** Kimdogyeom님 접속 **********
hello
Kimdogyeom: hello
Bob: hello dogyeom
Bob: I m Bob
oh nice to meet you
Kimdogyeom: oh nice to meet you
********** Bob님 접속 **********
Kimdogyeom: hello
hello dogyeom
Bob: hello dogyeom
I'm Bob
Bob: I'm Bob
Kimdogyeom: oh nice to meet you


PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
#JOIN computer Kimdogyeom
********** Kimdogyeom님 접속 **********
hello
Kimdogyeom: hello
Bob: hello dogyeom
Bob: I'm Bob
oh nice to meet you
Kimdogyeom: oh nice to meet you
Parknana: wow
Parknana: It is ama ing
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
#JOIN computer Bob
********** Bob님 접속 **********
Kimdogyeom: hello
hello dogyeom
Bob: hello dogyeom
I'm Bob
Bob: I'm Bob
Kimdogyeom: oh nice to meet you
Parknana: wow
Parknana: It is ama ing
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1>cd src
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
#JOIN computer Parknana
********** Parknana님 접속 **********
wow
Parknana: wow
It is amazing
Parknana: It is amazing


#JOIN computer Kimdogyeom
********** Kimdogyeom님 접속 **********
hello
Kimdogyeom: hello
Bob: hello dogyeom
Bob: I'm Bob
oh nice to meet you
Kimdogyeom: oh nice to meet you
Parknana: wow
Parknana: It is ama ing
hmm...
Kimdogyeom: hmm...
I think kakaoTalk is more better than this
Kimdogyeom: I think kakaoTalk is more better than this
Bob: I think so too
Parknana: Than I just wanna go to use Kakao Talk
Parknana: Bye
bye
Kimdogyeom: bye
c> java Peer 5000
running - port number : 5000
#JOIN computer Bob
********** Bob님 접속 **********
Kimdogyeom: hello
hello dogyeom
Bob: hello dogyeom
I'm Bob
Bob: I'm Bob
Kimdogyeom: oh nice to meet you
Parknana: wow
Parknana: It is ama ing
Kimdogyeom: hmm...
Kimdogyeom: I think kakaoTalk is more better than this
I think so too
Bob: I think so too
Parknana: Than I just wanna go to use Kakao Talk
Parknana: Bye
Kimdogyeom: bye
d src
PS C:\Users\admin\IdeaProjects\Computer_Networks_Fall_2022_ASS1\src> java Peer 5000
running - port number : 5000
#JOIN computer Parknana
********** Parknana님 접속 **********
wow
Parknana: wow
It is amazing
Parknana: It is amazing
Kimdogyeom: hmm...
Kimdogyeom: I think kakaoTalk is more better than this
Bob: I think so too
Than I just wanna go to use Kakao Talk
Parknana: Than I just wanna go to use Kakao Talk
Bye
Parknana: Bye
Kimdogyeom: bye
#EXIT
종료합니다.


********** Kimdogyeom님 접속 **********
hello
Kimdogyeom: hello
Bob: hello dogyeom
Bob: I'm Bob
oh nice to meet you
Kimdogyeom: oh nice to meet you
Parknana: wow
Parknana: It is amazing
hmm...
Kimdogyeom: hmm...
I think kakaoTalk is more better than this
Kimdogyeom: I think kakaoTalk is more better than this
Bob: I think so too
Parknana: Than I just wanna go to use Kakao Talk
Parknana: Bye
bye
Kimdogyeom: bye
Bob: bye

```
c> java Peer 5000
running - port number : 5000
#JOIN computer Bob
********** Bob님 접속 **********
Kimdogyeom: hello
Parknana: It is amazing
Kimdogyeom: hmm...
Kimdogyeom: I think kakaoTalk is more better than this
I think so too
Bob: I think so too
Parknana: Than I just wanna go to use Kakao Talk
Parknana: Bye
Kimdogyeom: bye
bye
Bob: bye
#exit
명령어가 아닙니다.
#e
명령어가 아닙니다.
#
명령어가 아닙니다.
#EXIT
종료합니다.
```

```
rc> java Peer 5000
running - port number : 5000
#JOIN computer Parknana
********** Parknana님 접속 **********
wow
Parknana: wow
It is amazing
Parknana: It is amazing
Kimdogyeom: hmm...
Kimdogyeom: I think kakaoTalk is more better than this
Bob: I think so too
Than I just wanna go to use Kakao Talk
Parknana: Than I just wanna go to use Kakao Talk
Bye
Parknana: Bye
Kimdogyeom: bye
#EXIT
종료합니다.
```

4) #명령어

#EXIT를 제외하고 #으로 시작하는 메시지의 경우에는 “명령어가 아닙니다.”를 출력합니다.

#EXIT를 실행하면 “종료합니다.”를 출력하고 프로그램을 종료합니다.

```
Kimdogyeom: I think kakaoTalk is more better than this
I think so too
Bob: I think so too
Parknana: Than I just wanna go to use Kakao Talk
Parknana: Bye
Kimdogyeom: bye
bye
Bob: bye
#exit
명령어가 아닙니다.
#e
명령어가 아닙니다.
#
명령어가 아닙니다.
#EXIT
종료합니다.
```