6월 9일 복습

이태양


```java
import java.io.IOException;

public class ProbServer {
    public static void main(String[] args) throws IOException {
        GameStartProcess gsp = new GameStartProcess();

        gsp.createServer();
        gsp.startGame();
        gsp.startThread();
    }
}
```

```java
public class ProbClient {

    public static void main (String[]args) throws IOException {
        GameStartProcess gsp = new GameStartProcess();

        gsp.createClient();
        gsp.startThread();
    }

}
```

통신을 위해 서버 하나 클라이언트 하나 만들어주고
네트워크다이스에서 만든 함수들을 호출한다


```java
class ServerCriticalSection {
    static int currentMyDice = 0;
}

public class NetworkDiceGame {
    int myDice;
    int targetDice;

    ServerSocket servSock;
    Socket sock;

    final String SERVER_IP = "172.30.1.49";
    final int PORT = 33333;

    public NetworkDiceGame () {
        myDice = (int)(Math.random() * 6 + 1);
        targetDice = 0;
    }

    public void createServer () throws IOException {
        servSock = new ServerSocket(PORT);

        System.out.println("Server: Listening - " + PORT);
    }

    public void createClient () throws IOException {
        sock = new Socket(SERVER_IP, PORT);
    }
}
```

필요한 변수들과

아이피와 포트번호를 설정해주고

주사위값을 생성하기위해 mydice값 설정

서버소켓 servSock에 포트번호를넣고

클라이언트는 소켓에 서버아이피와 포트번호를 넣는다


```java
class GameResultThread extends NetworkDiceGame implements Runnable {
    @Override
    public void run() {
        while (true) {
            if (targetDice != 0) {
                if (myDice > targetDice) {
                    System.out.println("나의 승리");
                } else if (myDice < targetDice) {
                    System.out.println("상대편의 승리");
                } else {
                    System.out.println("무승부");
                }
                targetDice = 0;
            }
    }
}
class GameSendThread implements Runnable {

    int dice;
    Scanner scan;
    Socket sock;

    public GameSendThread (Socket sock, int dice) {
        scan = new Scanner(System.in);
        this.dice = dice;
        this.sock = sock;
        System.out.println("sock: " + sock);
    }
```

네트워크다이스게임을 상속받고
Runnable은 쓰레드가 동작할 코드를 적는곳

메소드 run()을 오버라이드해서

대소비교후 결과값 출력해주는 코드작성

게임을 보내기위해 스레드를 하나 더만들고

다이스값과 소켓값을 입력받아 전송한다


```java
@Override
public void run() {

    OutputStream out = null;
    PrintWriter writer;

    while (true) {
        System.out.print("게임을 진행하시겠습니까(y/n) ? ");
        String str = scan.nextLine();

        System.out.println("str: " + str);

        if (str.equals("y")) {
            ServerCriticalSection.currentMyDice = (int)(Math.random() * 6 + 1);

            System.out.println("내 주사위 값: " + ServerCriticalSection.currentMyDice);

            try {
                out = sock.getOutputStream();
                writer = new PrintWriter(out, autoFlush: true);
                writer.println(ServerCriticalSection.currentMyDice);
            } catch (IOException e) {
                e.printStackTrace();
            }
        } else {
            break;
        }
    }
}
```

y/n중 하나를 입력받아 게임을 진행할 지 말지 정하기 위한 코드


```
class GameStartProcess extends NetworkDiceGame {
    Thread sender;
    Thread receiver;

    public GameStartProcess () {
        super();
    }

    public void startGame () throws IOException {
        sock = servSock.accept();
        System.out.println("[" + sock.getInetAddress() + "] client connected");
    }

    // public void startServerThread
    public void startThread () {
        // new GameSendServerThread()
        sender = new Thread(new GameSendThread(sock, myDice));
        // new GameRecvServerThread()
        receiver = new Thread(new GameRecvThread(sock, myDice));

        sender.start();
        receiver.start();
    }
}
```

쓰레드 두개를 선언해주고

소켓은 서버에서 받아온 소켓이되고
그소켓에있는 주소값을 받아와 출력

스레드 시작시키는 코드