

```

1 package Sixteenth;
2
3 import java.io.*;
4 import java.net.Socket;
5 import java.util.Date;
6 import java.util.Scanner;
7
8 public class SocketManager {           소켓서버의 필요한 송신 및 수신 과 서버에 출력될 수 있도록 하는
9     private final int ZERO = 0;       메서드 만드는 클래스.
10    private final int ONE = 1;
11    // OR 연산의 특성을 활용하기 위해 3인 경우를 4로 만들기 위함
12    private final int MAGICNUM = 3;
13
14    private OutputStream[] out;         송수신 위한 데이터
15    private InputStream[] in;          -여러개를 받기 위해서 배열을 사용한다
16
17    private PrintWriter writer;         PrintWriter = 프린트할 내용을 OutputStream으로 전달
18    private BufferedReader reader;      BufferedReader = 수신된 데이터를 저장(번호를 매기고)
19                                         ReadLine()메서드를 통해 출력이 가능
20
21    private Scanner scan;
22
23    String[] arrRockScissorPaper;
24
25    // send용 [현재 개발 관점은 클라이언트 입장]
26    public SocketManager() {
27        scan = new Scanner(System.in);  클라이언트가 서버에게송신하는 방법에 대한 클래스
28                                         -가위바위보 입력을 위한 스캐너
29                                         - 서버에게 송신을 위한 outstream
30        //in = new InputStream[ONE];
31        out = new OutputStream[ONE];
32    }

```

```

32 // recv용 [현재 개발 관점은 서버 입자]
33 public SocketManager(int num) {
34     //scan = new Scanner(System.in);
35
36     out = new OutputStream[num];
37     in = new InputStream[num];
38
39     arrRockScissorPaper = new String[num];
40 }
41
42 public void send(Socket sock) throws IOException {
43     System.out.print("숫자를 입력하세요: ");
44     String str = scan.nextLine();
45
46     out[ZERO] = sock.getOutputStream();
47     out[ZERO].write(str.getBytes());
48 }
49
50 public String convertNumber2RSP() {
51     String res = "";
52     int len = arrRockScissorPaper.length;
53
54     for(int i = ZERO; i < len; i++) {
55         res += (i + 1) + "번 사용자: ";
56
57         if (arrRockScissorPaper[i].equals("1")) {
58             res += "가위";
59         } else if (arrRockScissorPaper[i].equals("2")) {
60             res += "바위";
61         } else {
62             res += "보";
63         }
64     }
65 }

```

서버에서 클라이언트에게 보내고 클라이언트로 부터 데이터를 받기위한 (송신+수신)클래스

스캐너 통한 입력 후

클라이언트가 서버에게 보낼(송신)할 내용 1

Scanner 숫자 입력되는 int 를 String 으로 바꾸는 클래스

(초기화값)

접속자 만큼의 수를 배열안에 넣기 위해 for구문 사용

```

65         if(i == len - ONE) {
66             res += "";
67         } else {
68             res += ", ";
69         }
70     }
71
72     return res;    return String res 값
73 }               -> 다른메소드의 입력 값을 사용할 예정

```

```

74
75 public void send(Socket[] sock, int num) throws IOException {
76     for(int i = ZERO; i < num; i++) {
77         out[i] = sock[i].getOutputStream();
78
79         writer = new PrintWriter(out[i], true);
80
81         String str = convertNumber2RSP();
82
83         writer.println(str);
84     }
85 }

```

서버에서 클라이언트에게 보낼(송신) 위한 클래스  
 배열은 서버에서 여러 클라이언트들을 관리 위해 사용  
 num = server 입력값의 MAX  
 i++값들을 송신(보냄)을 저장(getOutputStream)  
 writer = 클라이언트에 출력위한 준비단계  
 Str 을 writer.print를통해 출력

<pre> 87 public void recv(Socket[] sock, int num) throws IOException { 88     int tmp; 89 90     for(int i = ZERO; i &lt; num; i++) { 91         in[i] = sock[i].getInputStream(); 92 93         reader = new BufferedReader(new InputStreamReader(in[i])); 94 95         // 미리 변환하지 않고 문자열인 상태에서 "3"과 같은지 비교하고 96         // 같으면 바꾸고 같지 않으면 그대로 두는 형식이 더 효율적이다. 97         // 속제로 한 번 만들어보세요 ~ 98         tmp = Integer.parseInt(reader.readLine()); 99 100        if(tmp == MAGICNUM) { 101            arrRockScissorPaper[i] = Integer.toString(tmp + ONE); 102        } else { 103            arrRockScissorPaper[i] = Integer.toString(tmp); 104        } 105 106        System.out.println("msg: " + arrRockScissorPaper[i]); 107    } 108 }</pre>	<p>서버에서 클라이언트의 값들을 수신(받음)위한 클래스</p> <p>tmp는 int로 입력 받은 Scanner값들을 저장하고</p> <p>스트링으로 받은 값들을 readLine 을 통하여 인트형으로변화시켜서</p> <p>Buffer에 저장된 전송된 값을 순서에 따라 출력한다.</p> <p>가위바위보 승리 판정을 위해 보에 MAGICNUM(3)을부여</p> <p>만약 MAGICNUM이되면 i는 tmp+1이되고 아니면 tmp가됨</p> <p>그리고 println을 통해 가위바위보 번호로 서버에 출력됨</p>
--	--

```

110 public boolean canWeGetWinner(int num) {
111     // 3명 이상이 함께 가위바위보를 할 때 승패를 어떻게 판정할 것인가 ?
112     // 가위 = 1, 바위 = 2, 보 = 4라면
113     // 모든 값을 OR 했을때 7이 나올 것이다.
114     // 만약 1, 2, 3이라면 OR 결과는 3이므로
115     // 이것이 보인지 무승부인지 판정이 불가!
116     int bitOROfAllInputString = ZERO;
117
118     for(int i = ZERO; i < num; i++) {
119         bitOROfAllInputString |=
120             Integer.parseInt(arrRockScissorPaper[i]);
121     }
122
123     if(bitOROfAllInputString == 7) {
124         return false;
125     } else if(bitOROfAllInputString == 1) {
126         return false;
127     } else if(bitOROfAllInputString == 2) {
128         return false;
129     } else if(bitOROfAllInputString == 4) {
130         return false;
131     }
132
133     return true;
134 }
135 }

```

결과판정을 위한 메소드

OR연산자로 해결하기!

->그래서 '보'를 4로 만들어줌

무승부가 나오는 상황들 ->false 처리

```
1 package Sixteenth;
2
3 import java.io.IOException;
4 import java.net.Socket;
5
6 public class ClientSocketManager extends SocketManager {
7     private Socket clntSock;
8
9     public ClientSocketManager(String hostIp, int portNum)
10         throws IOException {
11
12         super();
13         clntSock = new Socket(hostIp, portNum);
14     }
15
16     public Socket getClntSock() {
17         return clntSock;
18     }
19 }
```

클라이언트 실행을 위한 클래스

- hostIp와 portNum이 입력 되면
- Socket sock이 입력되어 있는 메서드가 상속
- hostIp와 portNum이 입력된 clntSock 테이터가 생기고 이를 리턴함

29 lines (20 sloc) | 812 Bytes

```
1  package Sixteenth;
2
3  import java.io.IOException;
4
5  // 게임에 접속해서 플레이하는 고객
6  public class MainClient {
7      public static void main(String[] args) throws IOException, InterruptedException {
8          // 집에서 하는 경우 자신의 집 컴퓨터 아이피 주소                메인 클라이언트에 들어갈 정보에 대한 클래스 생성
9          // 학원인 경우는 자신의 자리 ip 주소 혹은 동일하게 해도 무방
10         String ip = "192.168.0.9";
11         int port = 7777;
12
13         ClientSocketManager csm = new ClientSocketManager(ip, port);
14
15         System.out.println("접속 요청 완료!");
16         System.out.println("가위는 1, 바위는 2, 보는 3");
17
18         csm.send(csm.getClntSock());
19
20         System.out.println("전송 완료!");
```

IP와port를 입력하고 ClientSocketManager 클래스  
데이터 상자 만들어줌

클라이언트에 출력될 것들 print해주기

getClntSock = In / OutputStream  
(현재는 out만존재 / in은 주석처리???)

```

1 package Sixteenth;
2
3 import java.io.IOException;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6
7 public class ServerSocketManager extends SocketManager {
8     private ServerSocket servSock;
9     private Socket[] cIntSockArr;
10
11     private int cIntCnt;
12     private int maxCInt;
13
14     public ServerSocketManager(int portNum, int max)
15         throws IOException {
16         super(max);
17         System.out.printf("%d 명이 접속해야 게임을 시작할 수 있습니다.\n", max);
18
19         servSock = new ServerSocket(portNum);
20
21         cIntCnt = 0;
22         maxCInt = max;
23
24         cIntSockArr = new Socket[max];
25     }
26
27     public void waitForClientRequest() throws IOException {
28         System.out.println("사용자 접속을 대기합니다.");
29
30         for(int i = 0; i < maxCInt; i++) {
31             cIntSockArr[cIntCnt++] = servSock.accept();
32         }
33     }
34 }

```

17 줄의 `super(max);`에 대해:   
 "부모의 생성자는 호출"   
 "부모의 생성자에서 무조건 나와야 함!"   
 "Int 생성자가 없는 곳은 max로 치대?"

서버 실행을 위한 클래스  
 ServerSocket - 서버 생성을 위한 클래스??

서버에 들어올 두명 / 개 클라이언트들을 관리 하기 위해 배열을 사용

사용자가 여러명이므로 accept부분을 반복문 사용



```
37     public void checkEachIpAddressInfo() {
38         for(int i = 0; i < maxClnt; i++) {
39             System.out.println(
40                 "[" + clntSockArr[i].getInetAddress() +
41                 "] client connected"
42             );
43         }
44     }
45
46     public Socket[] getClntSockArr() {
47         return clntSockArr;
48     }
49
50     public int getMaxClnt() {
51         return maxClnt;
52     }
53 }
```

클라이언트 macClnt개가모두 연결 되면 연결 됐다고 프린트하는 객체

```

1 package Sixteenth;
2
3 import java.io.IOException;
4
5 // 혼자서 서버 클라이언트를 테스트하는 방법
6 // 프로젝트 최상위를 보면 전체 경로가 보이는데
7 // 터미널을 열어서 이 경로로 이동한다.
8 // 이동후에 cd out/production/GitAssociated 를 입력한다.
9 // java Sixteenth.MainClient 를 입력하면
10 // 우리가 IntelliJ에서 실행하듯이 명령어로 Client를 동작시킬 수 있다.
11 // 터미널을 3개 띄우면 클라이언트를 3개 실행할 수 있으므로
12 // New Tab을 하던 New Window를 하던해서 3개 우고 작업하면 된다.
13
14 // 서버의 역할: 클라이언트들에게 데이터를 중개
15 public class MainServer {
16     private static final int MAX = 3;
17
18     public static void main(String[] args)
19         throws IOException, InterruptedException {
20         ServerSocketManager ssm =
21             new ServerSocketManager(7777, MAX);
22
23         ssm.waitForClientRequest();
24
25         System.out.println("모두 접속 완료!");
26
27         ssm.checkEachIpAddressInfo();
28
29         ssm.recv(ssm.getCIntSockArr(), ssm.getMaxCInt());

```

ssm은 SocketManager를 상속받음

클라이언트 <sup>송신</sup> → 서버.

getCIntSockArr(max), getMaxCInt(max)

```
31 // 이제 서버가 할 일은
32 // 클라이언트들에게 승패 결과와
33 // 각자가 무엇을 제출했는지 표시해주는 것
34 if(ssm.canWeGetWinner(ssm.getMaxCnt())) {
35     System.out.println("승패가 결정되었습니다.");
36 } else {
37     System.out.println("무승부: 게임을 다시 시작합니다.");
38 }
39
40 ssm.send(ssm.getCntSockArr(), ssm.getMaxCnt());
41
42 System.out.println("모든 사용자에게 입력 결과 전달 완료!");
43 }
44 }
```

서버 → 클라이언트  
통신