Data Communication \_ Project 2

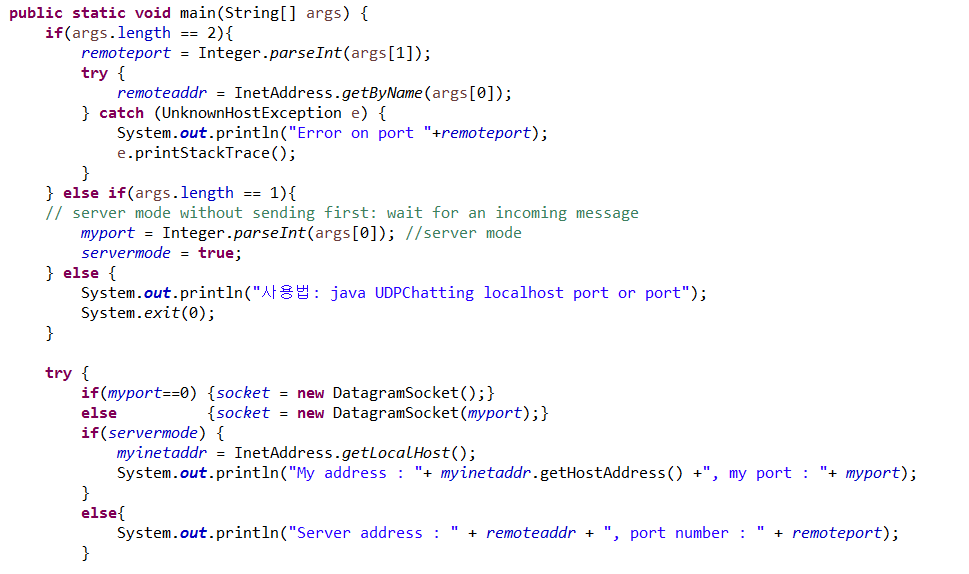
2014004693 송보석

1. 과제

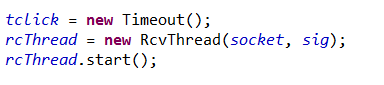
UDP Socket을 이용하여 화면에서 keyboard를 입력 받아 상대에게 UDP 패킷를 보내고 상대로부터 온 UDP 패킷을 화면에 표시하면서 UDP를 통해 메시지를 주고 받는 과제

2. 코드 설명

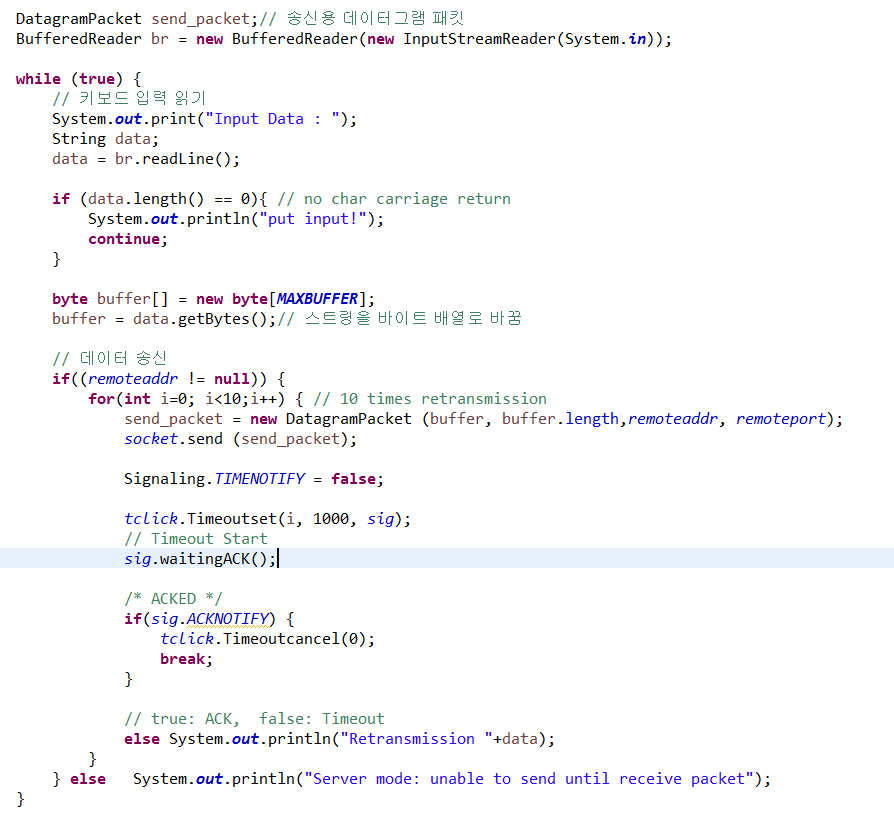
UDPChatting.java



프로그램을 실행시키면 제일 먼저 실행되는 부분이다. 인자로 받은 매개변수의 개수를 이용해 서버인지 클라이언트인지 확인한다. 인자의 길이가 2인경우(서버주소, 포트번호)는 클라이언트이므로 remoteaddr, remoteport 에 각각 그 값을 초기화 해준다. 길이가 1인 경우는 myport에 포트 번호를 초기화하고, servermode를 켜준다. 서버는 myport값으로 socket을 열고 아닌경우 임의의 포트 번호를 지정해 socket을 지정한다.

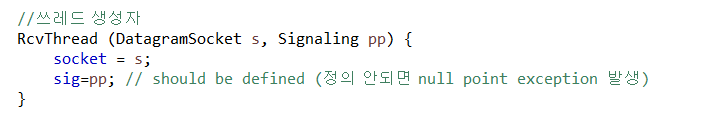


Timeout을 다루기 위한 객체를 만들고, receive를 맡을 객체를 생성해 쓰레드를 실행시킨다.

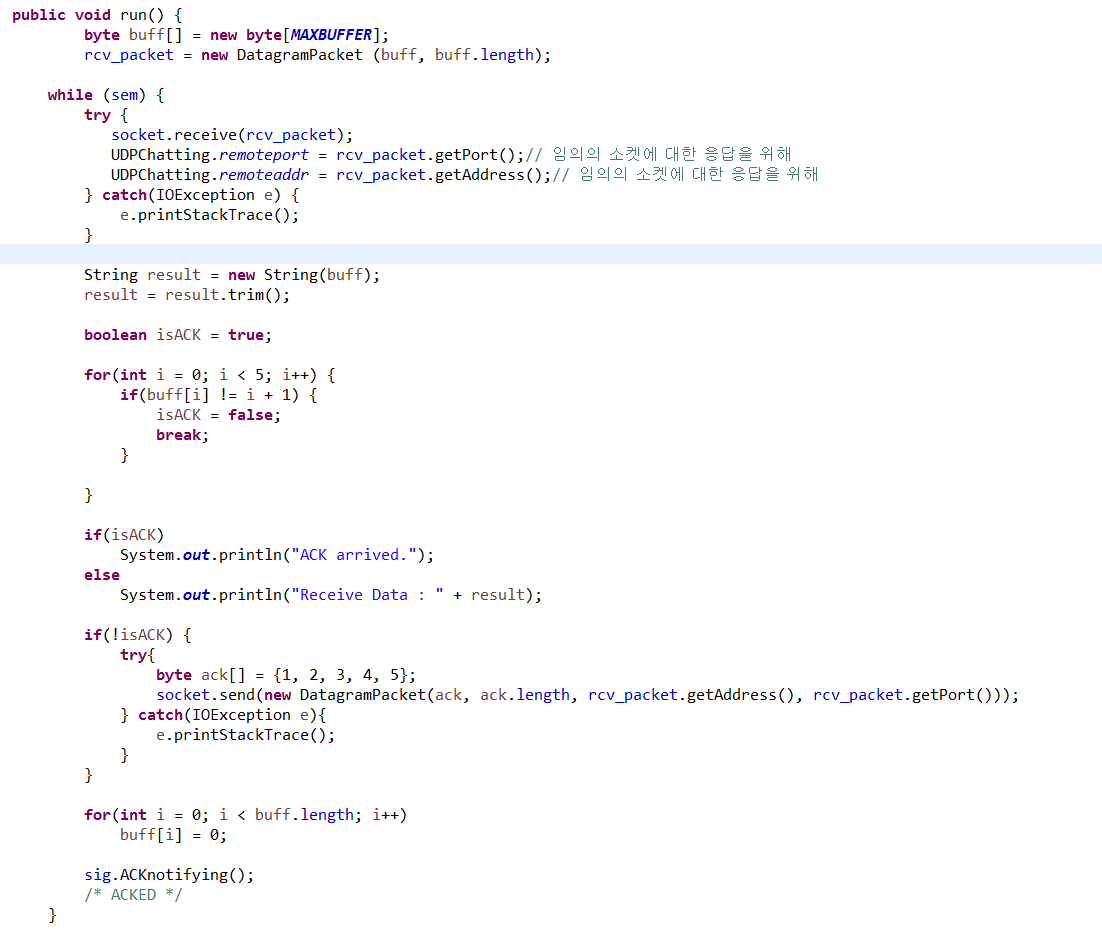


송신용 DatagramPacket send\_packet과 키보드 입력을 위한 버퍼를 선언한다. 루프 안에서 키보드 입력을 String 형 data에 넣는다. Data의 길이가 0면 다시 입력을 받고 아닐 경우 data를 byte로 변형해 바이트 배열에 넣는다. 송신은 datagrampacket 객체를 만들고, send 메소드로 보낸다. 1초를 timeout으로 설정해놓고 waiting ACK 메소드를 이용해 ack가 도달하길 기다린다. 만약 ack가 시간 안에 도착하면 ACKNOTIFY가 true로 바뀌는데 그렇다면 timeout을 초기화 시키고 빠져나간다. 그렇지 않으면 retrasmission을 하는데 10번동안 반복된다.

RcvThread.java

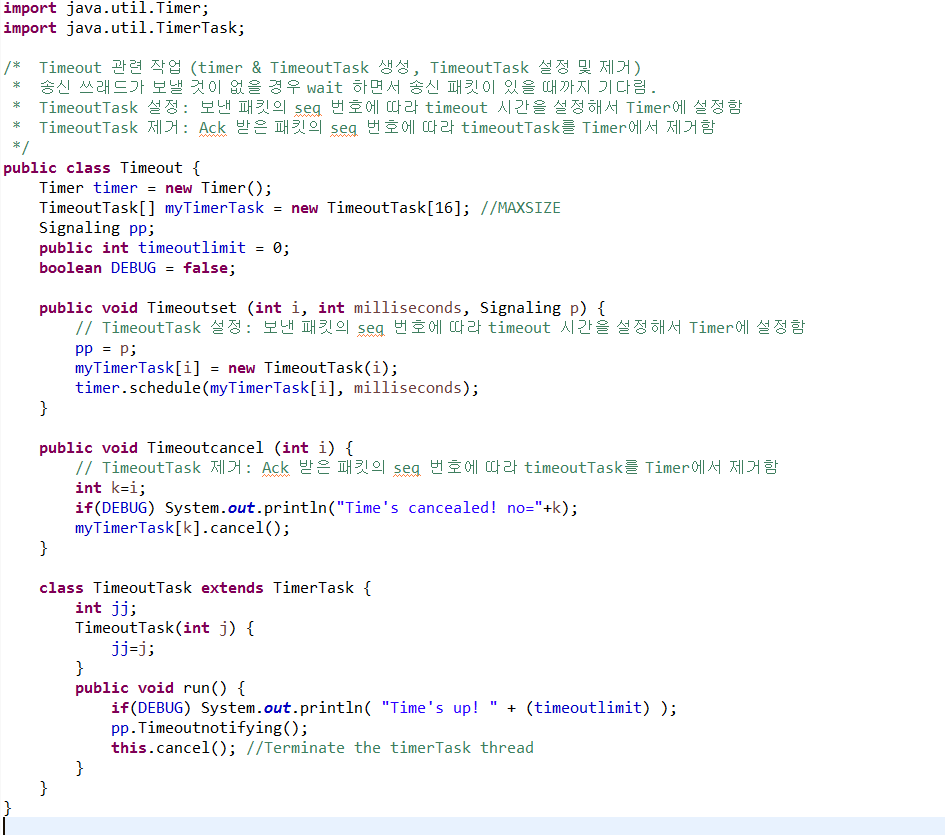


쓰레드 초기화 생성자이다. 소켓과 시그널을 받아온다.



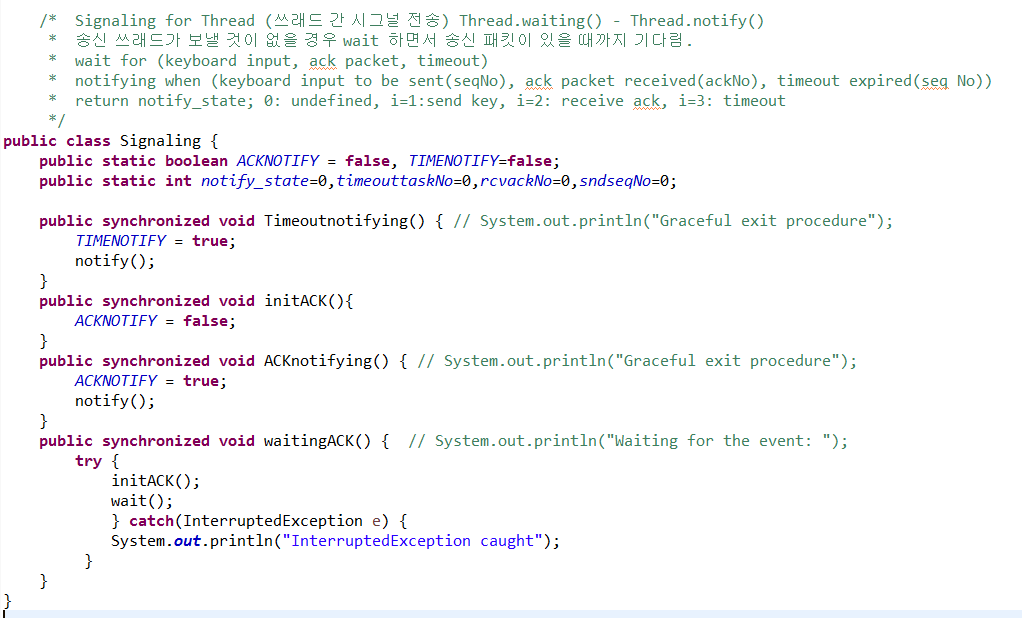
쓰레드의 실행 메소드이다. Receive 메소드로 패킷을 받아오면 소켓의 주소와 포트를 UDPChatting의 변수에 넘겨준다. Buffer를 스트링으로 변환하고, 받아온 값이 ack인지 확인한다. Ack는 1,2,3,4,5로 이루어져있는 변수이기 때문에 루프를 이용해 확인해 ack가 맞다면 isACK를 true로 아니면 false로 초기화한다. 그리고는 ACK라면 ACK가 도착했다고 출력하고 아니라면 받아온 메시지를 출력한다. ack배열을 1,2,3,4,5로 초기화하고 ack를 클라이언트에게 보내준다. 그리고 버퍼를 초기화해주고 ACKnotifying 메소드를 이용해 wait 하고 있는 서버를 꺠워주고 ACKNOTIFY를 true로 바꿔준다.

Timeout.java



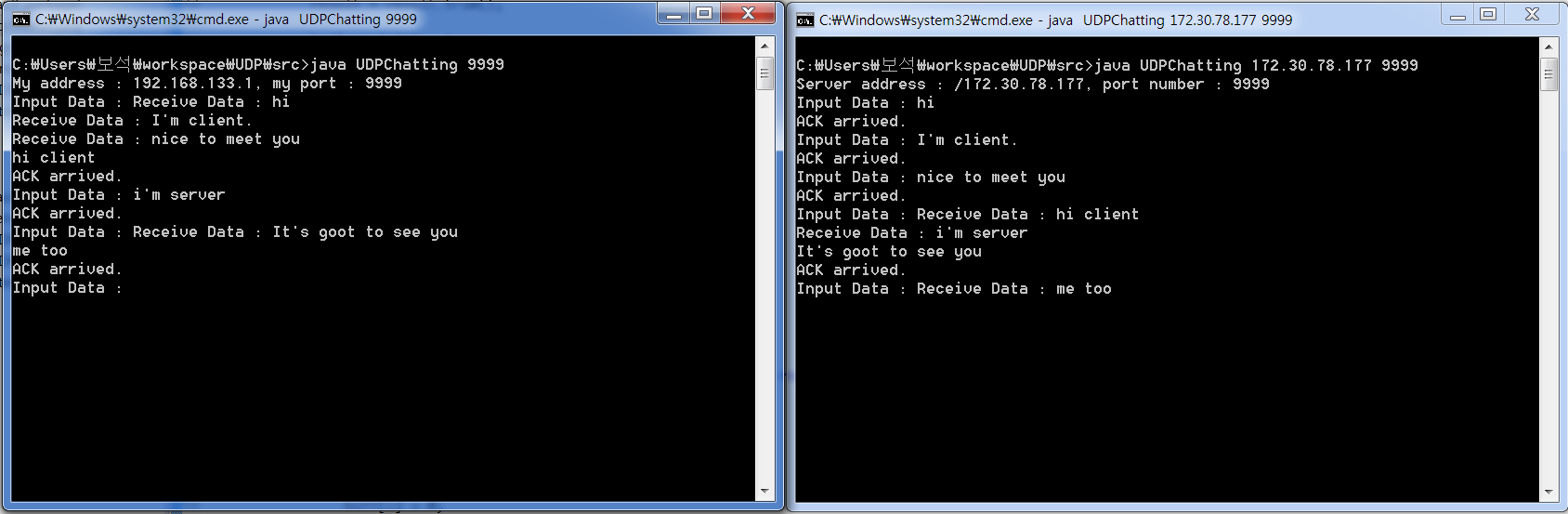
타임아웃 설정과 해제를 위한 클래스이다. Timeoutset 메소드는 패킷의 seq 번호와 timeout 시간 그리고 signaling 을 받아와 타임아웃을 설정한다. Timecancel은 설정해놨던 timeout을 해지하는 메소드이다.

Signaling.java

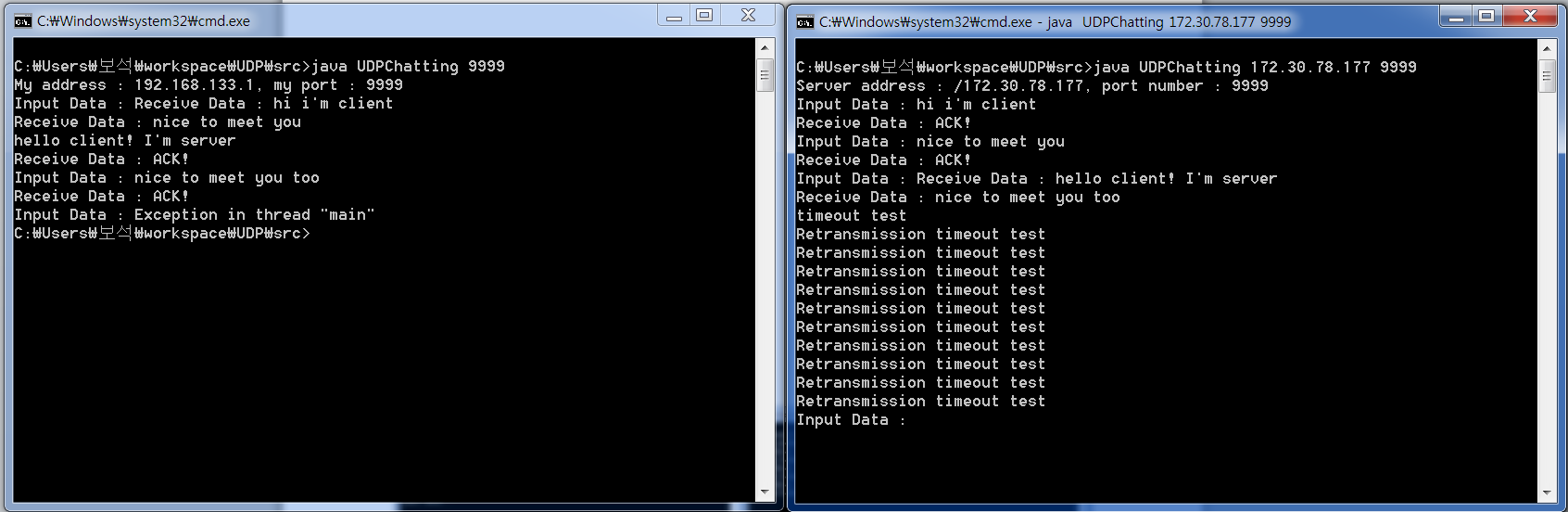


Timeout을 다루기 위한 클래스이다. Timeoutnotifying은 타임아웃이 설정되어 있다면 TIMENOTIFY를 true로 초기화한다. ACKnotifying은 ACK가 전송되면 ACKNOTIFY를 true로 초기화해 ACK가 전송됬음을 알려준다. watingACK는 ACKNOTIFY를 false로 하고 notify가 올 때까지 wait 한다.

3. 실행화면



정상 구동



Timeout