

# 네트워크프로그래밍-9주

## Java 소켓 프로그래밍-2

정인환교수

# 9주 : Java 소켓 프로그래밍-2

- ▶ JavaChatClient/Server 기능 추가하기
  - 이모티콘 사용하기
- ▶ ObjectOutputStream 사용하기
  - ChatMsg Object 사용하기
  - 채팅메시지 처리
  - 종료 버튼 처리
  - 이미지 첨부하기
- ▶ 과제
  - 이모티콘 사용하기 복습
  - ObjectOutputStream 사용하기 복습
  - 텀프로젝트 제안서 제출

# JavaChatClient/Server 이모티콘 사용

- ▶ JavaChatClient.java 수정
  - JTextArea 대신 JTextPane 사용

```
//private JTextArea textArea;  
private JTextPane textArea;
```

```
textArea = new JTextPane();  
textArea.setEditable(true);  
textArea.setFont(new Font("굴림체", Font.PLAIN, 14));  
scrollPane.setViewportView(textArea);
```

- AppendText(String msg) 수정
  - textArea.append(msg + "\n"); 대신

```
public void AppendText(String msg) {  
    //textArea.append(msg + "\n");  
    AppendIcon(icon1);  
    int len = textArea.getDocument().getLength();  
    // 끝으로 이동  
    textArea.setCaretPosition(len);  
    textArea.replaceSelection(msg + "\n");  
}
```

- AppendIcon(ImageIcon icon) 추가

```
ImageIcon icon1 = new ImageIcon("src/icon1.jpg");  
public void AppendIcon(ImageIcon icon) {  
    int len = textArea.getDocument().getLength();  
    // 끝으로 이동  
    textArea.setCaretPosition(len);  
    textArea.insertIcon(icon);  
}
```



# JavaChatClient.java

```
private JLabel lblUserName;  
//private JTextArea textArea;  
private JTextPane textArea;  
  
/**  
 * Create the frame.  
 */  
public JavaChatClientView(String username, String ip_addr, String port_no) {  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setBounds(100, 100, 392, 462);  
    contentPane = new JPanel();  
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
    setContentPane(contentPane);  
    contentPane.setLayout(null);  
  
    JScrollPane scrollPane = new JScrollPane();  
    scrollPane.setBounds(12, 10, 352, 340);  
    contentPane.add(scrollPane);  
  
    textArea = new JTextPane();  
    textArea.setEditable(true);  
    textArea.setFont(new Font("굴림체", Font.PLAIN, 14));  
    scrollPane.setViewportView(textArea);
```

```
    ImageIcon icon1 = new ImageIcon("src/icon1.jpg");  
    public void AppendIcon(ImageIcon icon) {  
        int len = textArea.getDocument().getLength();  
        // 끝으로 이동  
        textArea.setCaretPosition(len);  
        textArea.insertIcon(icon);  
    }  
    // 화면에 출력  
    public void AppendText(String msg) {  
        //textArea.append(msg + "\n");  
        AppendIcon(icon1);  
        int len = textArea.getDocument().getLength();  
        // 끝으로 이동  
        textArea.setCaretPosition(len);  
        textArea.replaceSelection(msg + "\n");  
    }
```

# ObjectStream 사용하기

## ▶ Client/Server/Client 사이에 byte array 대신

- Object 자체를 보내고 받을 수 있다
- String, Image, ... 등등 보낼 수 있다
- 채팅에서 Image(사진) 첨부기능 구현 가능

## ▶ ObjectStream I/O

- 변수 ois, oos 선언

```
private ObjectInputStream ois;  
private ObjectOutputStream oos;
```

- oos, ois 변환

```
oos = new ObjectOutputStream(client_socket.getOutputStream());  
oos.flush();  
ois = new ObjectInputStream(client_socket.getInputStream());
```

- Read/write

- Object obj = ois.readObject();
- ChatMsg obcm = new ChatMsg(UserName, "200", msg);
- oos.writeObject(obcm);

```
ChatMsg obcm = new ChatMsg(UserName, "200", msg);  
oos.writeObject(obcm);
```

```
Object obj = null;  
String line1 = null;  
if (this.client_socket == null)  
    return;  
// read /login user1  
if ((obj = ois.readObject()) == null)  
    return;  
if (obj instanceof ChatMsg) {  
    ChatMsg cm = (ChatMsg) obj;  
    AppendObject(cm);  
    line1 = cm.getData();  
}
```

# Class ChatMsg : 채팅 Object

// ChatMsg.java 채팅 메시지 ObjectOutputStream 용.

```
import java.io.Serializable;
```

```
import javax.swing.ImageIcon;
```

```
class ChatMsg implements Serializable {  
    private static final long serialVersionUID = 1L;  
    private String id;  
    private String code; // 100:로그인, 400:로그아웃, 200:채팅메시지, 300:Image  
    private String data;  
    public ImageIcon img;  
  
    public ChatMsg(String id, String code, String msg) {  
        this.id = id;  
        this.code = code;  
        this.data = msg;  
    }  
}
```

```
public String getCode() {  
    return code;  
}
```

```
public void setCode(String code) {  
    this.code = code;  
}
```

```
public String getData() {  
    return data;  
}
```

```
public String getId() {  
    return id;  
}
```

```
public void setId(String id) {  
    this.id = id;  
}
```

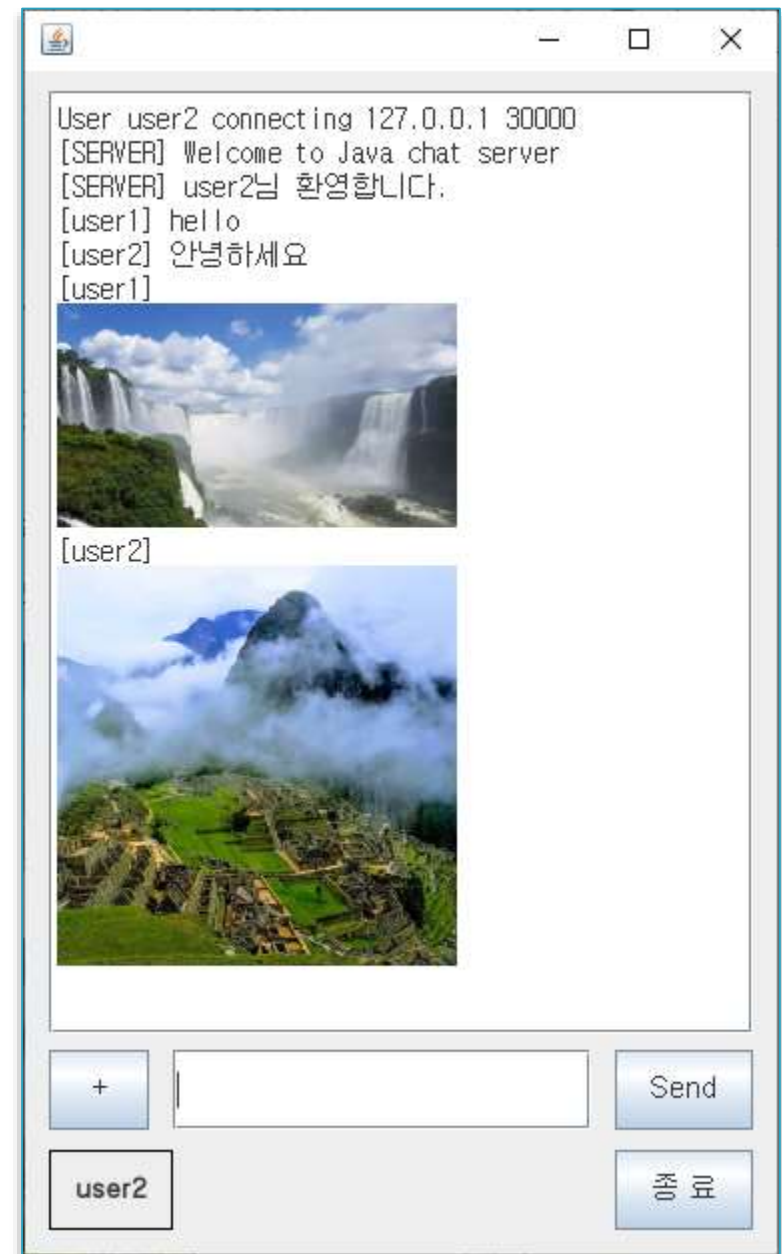
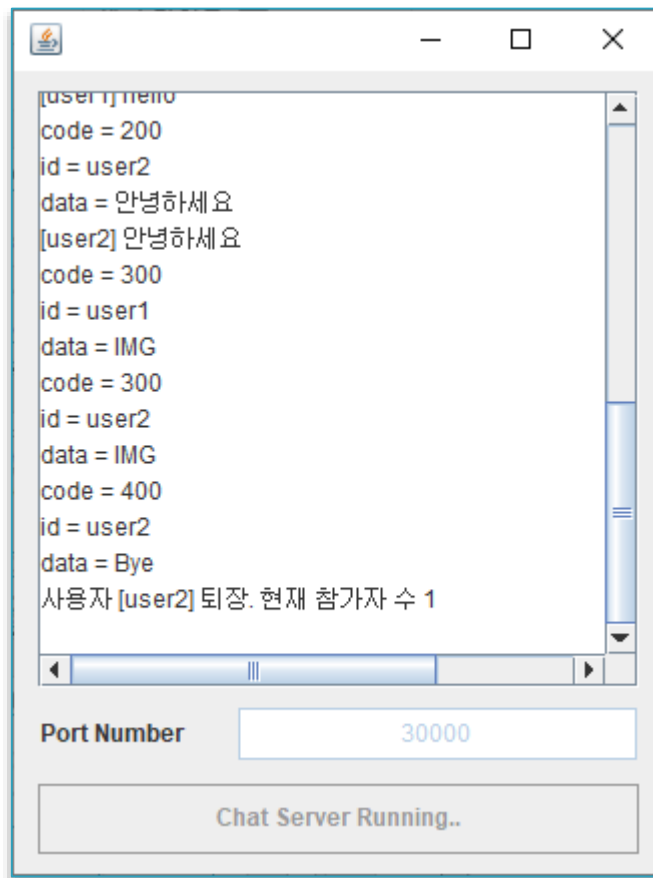
```
public void setData(String data) {  
    this.data = data;  
}
```

```
public void setImg(ImageIcon img) {  
    this.img = img;  
}
```

# JavaObjClient/Server ObjectOutputStream이용 채팅

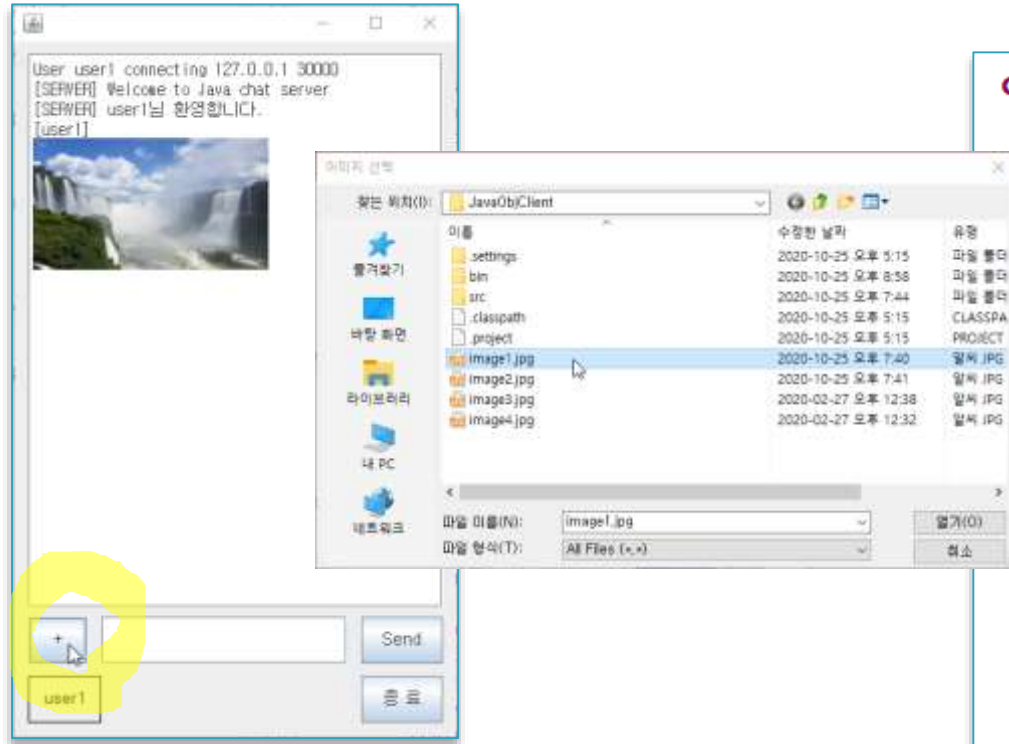
- ▶ 로그인 처리 /login 대신
  - Client
    - ChatMsg obcm = new ChatMsg(UserName, " 100 ", "Hello");
    - SendObject(obcm);
  - Server
    - obj = ois.readObject(); ChatMsg cm = (ChatMsg) obj;
    - if (cm.getCode().matches("100")) { UserName = cm.getId(); UserStatus = "0"; Login(); }
- ▶ 일반 Message 전송 (byte array 대신 Object 사용)
  - SendMessage(String msg)
    - ChatMsg obj = new ChatMsg(UserName, "200", msg);
    - oos.writeObject(obj);
- ▶ 종료 Button 처리
  - ChatMsg obj = new ChatMsg(UserName, "400", "Bye");
  - oos.writeObject(obj);
- ▶ Image 전송
  - + Button 누르면 파일 선택에서 Image 파일 선택
    - ChatMsg obcm = new ChatMsg(UserName, " 300 ", " IMG " );
    - ImageIcon img = new ImageIcon(fd.getDirectory() + fd.getFile());
    - obcm.setImg(img);
    - SendObject(obcm);
    - Oos.writeObject(ob);
- ▶ Image 화면 표시
  - obj = ois.readObject();
  - if (obj instanceof ChatMsg)
    - ChatMsg cm = (ChatMsg) obj;
  - AppendImage(cm.img);







# JavaObjClient.java – Image 전송하기



```
class ImageSendAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        // 액션 이벤트가 sendBtn일때 또는 textField 에세 Enter key 치면
        if (e.getSource() == imgBtn) {
            frame = new Frame("이미지첨부");
            fd = new FileDialog(frame, "이미지 선택", FileDialog.LOAD);
            // frame.setVisible(true);
            // fd.setDirectory(".\\");
            fd.setVisible(true);
            //System.out.println(fd.getDirectory() + fd.getFile());
            ChatMsg obcm = new ChatMsg(UserName, "300", "IMG");
            ImageIcon img = new ImageIcon(fd.getDirectory() + fd.getFile());
            obcm.setImg(img);
            SendObject(obcm);
        }
    }
}
```

# JavaObjClient.java – Image 보여주기

```
Object obcm = null;
String msg = null;
ChatMsg cm;
try {
    obcm = ois.readObject();
} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    break;
}
if (obcm == null)
    break;
if (obcm instanceof ChatMsg) {
    cm = (ChatMsg) obcm;
    msg = String.format("[%s] %s", cm.getId(), cm.getData());
} else
    continue;
switch (cm.getCode()) {
case "200": // chat message
    AppendText(msg);
    break;
case "300": // Image 첨부
    AppendText("[ " + cm.getId() + " ]");
    AppendImage(cm.img);
    break;
}
```


채팅 메시지

이미지 첨부

```
public void AppendImage(ImageIcon ori_icon) {
    int len = textArea.getDocument().getLength();
    textArea.setCaretPosition(len); // place caret at the end (with no selection)
    Image ori_img = ori_icon.getImage();
    int width, height;
    double ratio;
    width = ori_icon.getIconWidth();
    height = ori_icon.getIconHeight();
    // Image가 너무 크면 최대 가로 또는 세로 200 기준으로 축소시킨다.
    if (width > 200 || height > 200) {
        if (width > height) { // 가로 사진
            ratio = (double) height / width;
            width = 200;
            height = (int) (width * ratio);
        } else { // 세로 사진
            ratio = (double) width / height;
            height = 200;
            width = (int) (height * ratio);
        }
        Image new_img = ori_img.getScaledInstance(width, height, Image.SCALE_SMOOTH);
        ImageIcon new_icon = new ImageIcon(new_img);
        textArea.insertIcon(new_icon);
    } else
        textArea.insertIcon(ori_icon);
    len = textArea.getDocument().getLength();
    textArea.setCaretPosition(len);
    textArea.replaceSelection("\n");
    // ImageViewAction viewaction = new ImageViewAction();
    // new_icon.addActionListener(viewaction); // 내부클래스로 액션 리스너를 상속받은 클래스로
}
```

## 9주 과제1 : JavaChatClient/Server, JavaObjChatClient/Server 실습 및 추가



### ▶ JavaChatClient/Server

- 이모티콘 변경해 보기 (src/icon1.jpg 파일 변경)  (58x59 크기) 대신??

### ▶ JavaObjClient/Server

- ObjectOutputStream 사용
- 다른 이미지 첨부하기
- 종료 Button

### ▶ 추가기능 도전과제 (채팅 아이디어 참고) - 필수 아님

- 본인 Message는 우측에 나오기 (카톡 style)
- 사용자 List 화면 추가하기
  - /list 대신 GUI 버튼 사용하기 (Jbutton 사용)
  - 또는 Server가 보내주는 “[SERVER] [user2]님이 입장 하였습니다.” 를 이용 등등..
- /sleep, /wakeup 대신 GUI 버튼 사용하기 (JComboBox, JCheckBox .. 등등)
- 이모티콘 추가하기 (카톡의 이모티콘 방식 : () 안에 Text가 결정)
  - (굿)  (하하) 

# 과제 2 템프로젝트 제안서 제출

- ▶ Project 제목 (제목에 내용이 표시되어야 함)
  - (예) 배틀 테트리스 게임, 네트워크스네이크
  - (예) 한성Talk (카카오톡 모방)
- ▶ Project 팀 구성 (1인, 또는 2인)
  - 특히 2인인 경우 2명 모두 명시 (2명 모두 제출)
- ▶ 개발하려는 기능 나열
- ▶ 예상 결과물 (화면)
  - 비슷한 화면 이용 Capture 또는 직접 그리거나.. 해서 제출
- ▶ 사용언어 및 환경
  - **Java 사용 원칙, Java Low Level Socket 통신 방식이어야 함**
    - 즉, DataStream이나 ObjectOutputStream 사용해야 함
  - Android 사용 사용해도 무방함 (단, DataStream/ObjectStream 사용해야 함)

# 텀프로젝트 아이템 선정시 주의사항

- ▶ 인터넷에 Open Source 이용하는 경우
  - 예) 테트리스, 슈팅게임, 메신저 등등..
    - Copy로 확인되는 경우 F학점
  - 단, 1인용 게임 소스를 2인 네트워크용으로 확장하는 경우는 허용됨
- ▶ 텀프로젝트 유형별 필수 기능
  - 테트리스 : 2인용인 경우 좌측에 본인 화면 우측에 상대방 화면 표시
    - 3인 이상인 경우 본인 화면은 크게, 다른 화면들은 우측에 작게 표시
  - 바둑/오목
    - 반드시 무르기 기능 (무르기 요청하면 상대방이 허락하는 경우 가능) 구현
    - Mouse 사용 권장
    - 33 불가능 기능 필수
  - 채팅, 메신저, 카톡류 (필수, 추천 기능)
    - 이모티콘 필수
    - 채팅방 기능 (여러 개의 채팅방을 사용할 수 있어야 함) 필수
    - 가능한한 카톡과 유사하게 만들것
    - 메시지 저장 기능

# 텀프로젝트 제안서

- ▶ 프로젝트 제목
  - 네트워크 배틀 테트리스
- ▶ 팀원
  - 학번 이름 소속
  - 학번 이름 소속
- ▶ 기능 (예)
  - 배틀 테트리스
  - 6명 동시 Play
  - 2칸 삭제하면 공격
  - 아이템 기능
  - 점수 관리 기능
  - 채팅하기 기능



# 예상 화면

