

# 네트워크프로그래밍-11주

## Java UDP 소켓, Android 소켓 통신

정인환교수

# 11주 Java Upd 소켓, Android 소켓

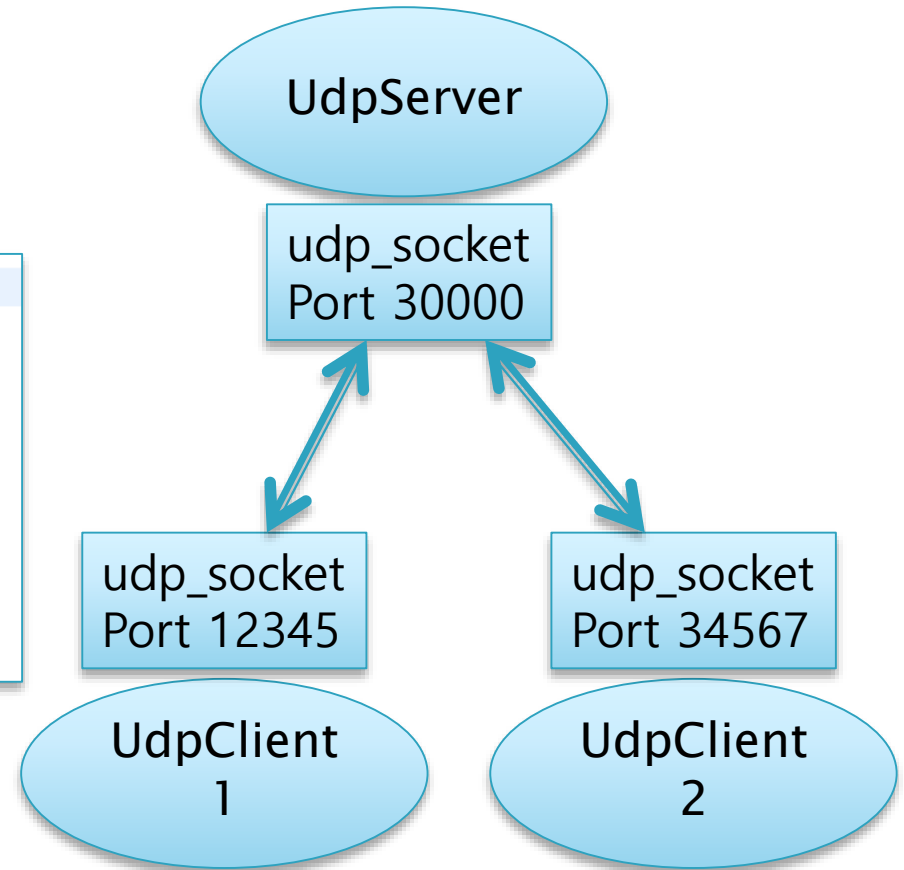
- ▶ JavaUdpEchoClient/Server
- ▶ AndroidChat Client
  - Android Socket 환경 설정
  - Java/Android ObjectOutputStream 사용
- ▶ Android 화면 처리
  - Keyboard 내리기
  - TextView 사용 채팅 메시지 자동 스크롤
- ▶ Image 처리
  - Java/Android 호환 Bitmap, byte array
- ▶ ObjectOutputStream 사용

# UDP 송신 (JavaUdpEchoClient/Server)

```
private DatagramSocket udp_socket; // 연결소켓
```

```
udp_socket = new DatagramSocket();
```

```
public void send_Message(String str) { // 서버로 메시지를 보내는 메소드
    byte[] bb = new byte[128];
    bb = str.getBytes();
    DatagramPacket udp_packet = new DatagramPacket(bb, bb.length, ip_addr, port);
    try {
        udp_socket.send(udp_packet);
    } catch (IOException e) {
        textArea.append("메세지 송신 에러!!\n");
    }
}
```



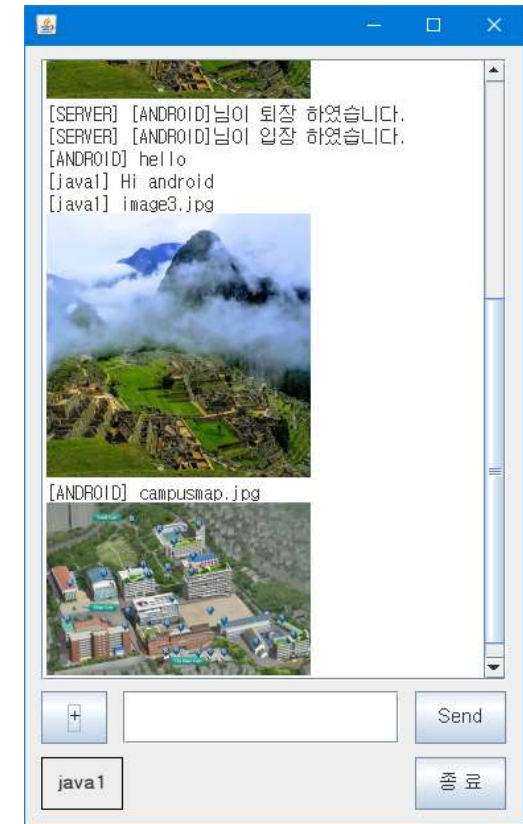
# UDP 수신

```
send_Message(id); // 정상적으로 연결되면 나의 id를 전송
Thread th = new Thread(new Runnable() { // 스레드를 돌려서 서버로부터 메시지를 수신
    @SuppressWarnings("null")
    @Override
    public void run() {
        byte[] bb = new byte[128];
        DatagramPacket udp_packet = new DatagramPacket(bb, bb.length);
        while (true) {
            for(int i=0; i<bb.length; i++)
            {
                bb[i] = 0;
            }
            try {
                udp_socket.receive(udp_packet);
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            String msg = new String(bb);
            msg = msg.trim();
            textArea.append(msg + "\n");
            textArea.setCaretPosition(textArea.getText().length());
        } // while문 끝
    } // run메소드 끝
});
```

# Android 채팅 Client

JavaObjectServer

JavaObjectClient



# Android Socket Programming

- ▶ Internet 사용을 위해 Android Manifest 추가
  - `<uses-permission android:name="android.permission.INTERNET"></uses-permission>`
- ▶ IP Address
  - Emulator (10.0.2.2) → PC (127.0.0.1)
  - `final String ip_addr = "10.0.2.2"; // Emulator PC의 127.0.0.1`
  - `final String ip_addr = "192.168.0.xxx"; // 실제 Phone으로 테스트 하는 경우`
- ▶ Android ObjectOutputStream Java 호환성 문제
  - class ChatMsg 를 보내고 받을 수 없음
  - String, byte[] 등 기본 자료형(Serializable Object)은 ObjectOutputStream이 가능
  - SendChatMsg(), ReadChatMsg() 안에서 writeObject, readObject를 여러 번 호출
- ▶ Keyboard 내리기
- ▶ TextView 사용, 자동 scroll
- ▶ Image 처리 : Image를 모두 byte array 로 처리
  - ObjectOutputStream을 이용 송/수신
  - Java : File → BufferedImage → byte[] 전송
  - Android : byte[] 수신 → Bitmap → setImageBitmap



# Java/Android ObjectOutputStream 통신

## ▶ Java/Java 기존 Code

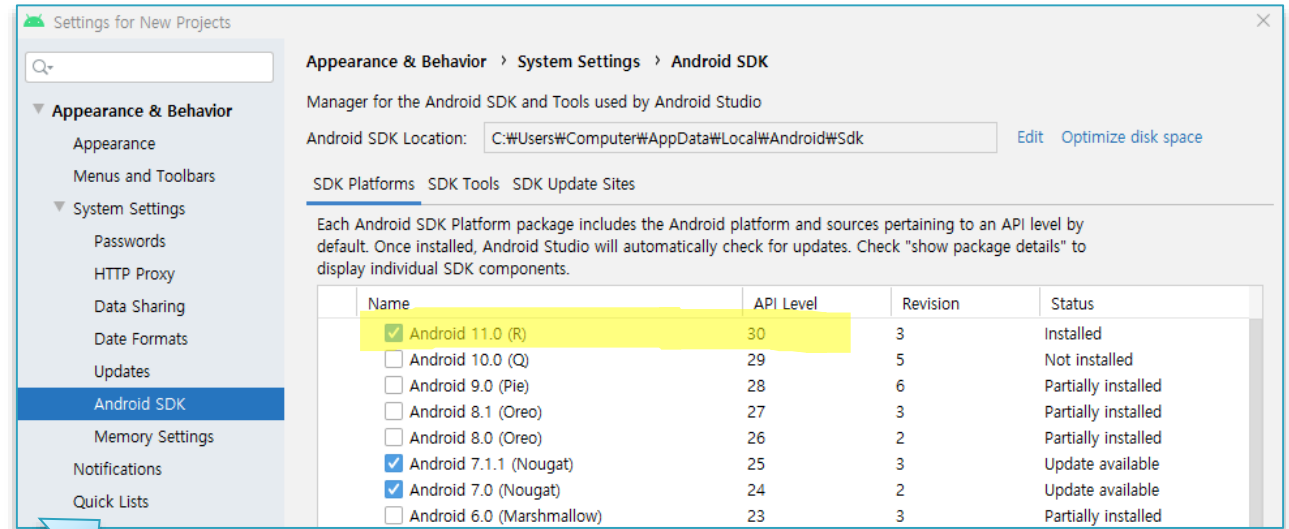
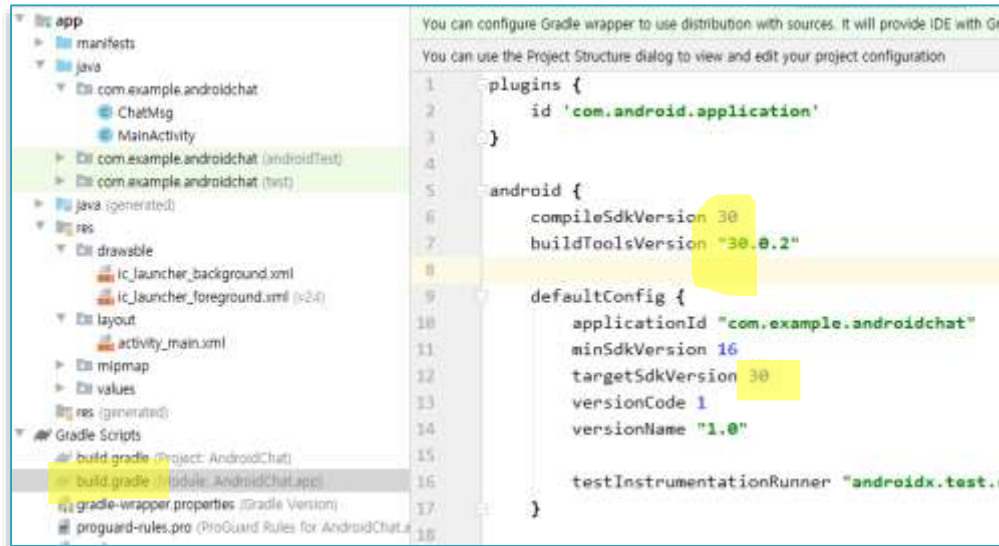
- `ChatMsg cm; oos.writeObject(cm);`
- `ChatMsg cm = (ChatMsg) ois.readObject();`

## ▶ Java/Android 용으로 수정

- `class ChatMsg {`
  - `String code, String UserName, String data, byte[] imgbytes;`
- `}`
- `SendChatMsg(ChatMsg cm)`
  - `oos.writeObject(cm.code); oos.writeObject(cm.UserName);`  
`oos.writeObject(cm.data)`
  - `if (cm.code.equals("300")) oos.writeObject(cm.imgbytes);`
- `public ChatMsg ReadChatMsg()`
  - `cm.code = ois.readObject(); cm.UserName = ois.readObject();`
  - `cm.data = ois.readObject();`
  - `if (cm.code.equals("300")) cm.imgbytes = ois.readObject();`

# Android Studio SDK Version 확인 및 설정

- ▶ NetP10-Android.zip 압축 풀고 Android Studio 로 Open
- ▶ AndroidChat SDK Version 확인
  - Android Studio Project List에서 build.gradle(Module AndroidChat.app)



- ▶ Android SdkVersion 30 설치
  - Android Studio > Tools > SDK Manager 실행 > System Settings > Android SDK
  - Anroid 11.0(R) 선택, 확인, 설치
- ▶ File > Sync Project with Gradle Files 로 Version 동기화