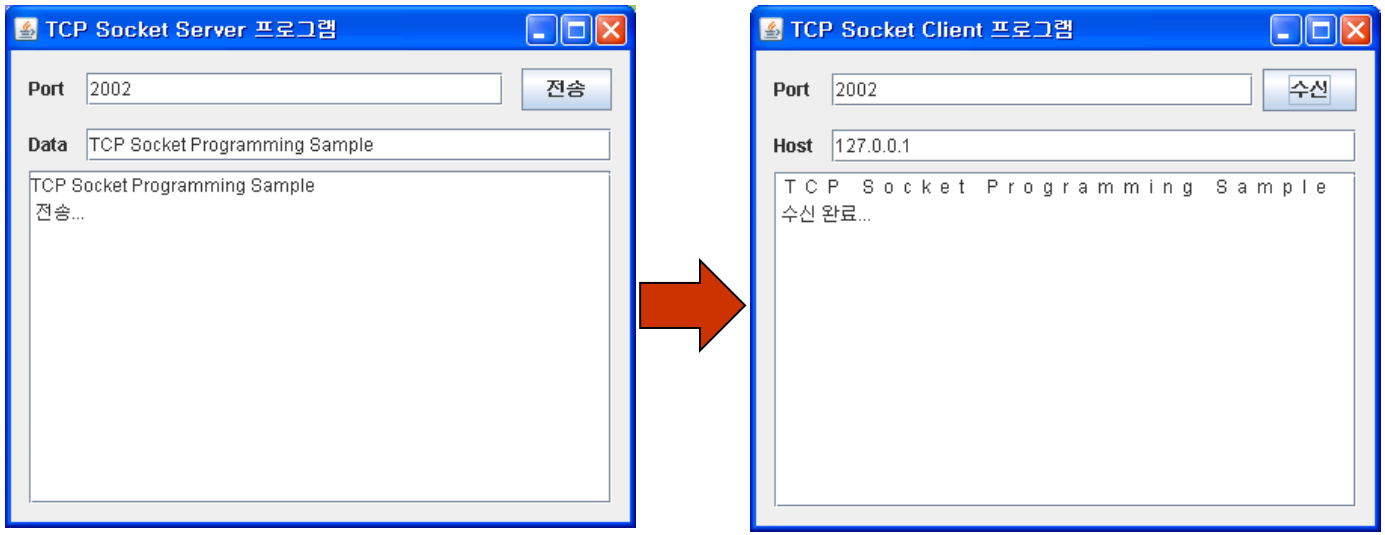
Network 프로그래밍



학습 목표

■ 이 강의를 마치면 학생들은

- ❖ Network 개요에 대하여 설명할 수 있다.
- ❖ InetAddress 클래스에 대하여 설명할 수 있다.
- ❖ URL 클래스에 대하여 설명할 수 있다.
- ❖ URLConnection 클래스에 대하여 설명할 수 있다.
- ❖ TCP Socket 프로그래밍 방법에 대하여 설명할 수 있다.
- ❖ UDP Socket 프로그래밍 방법에 대하여 설명할 수 있다.

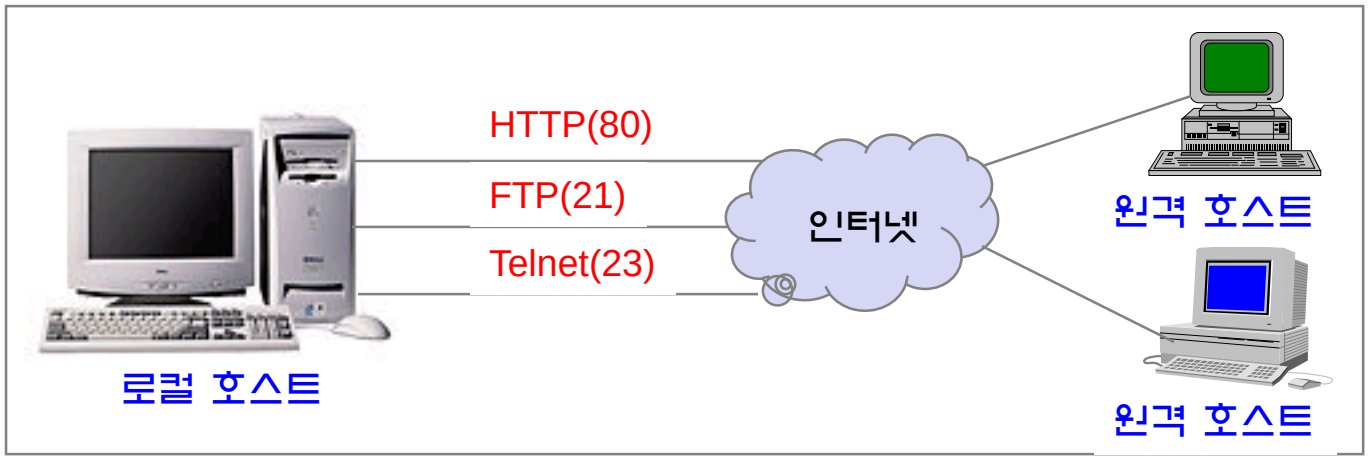
Network 개요

IP

- ❖ 기능
  - ◆ 인터넷에 접속된 컴퓨터를 구분하는 4바이트(210.119.160.100)의 숫자.

Port

- ❖ 기능
  - ◆ 외부 컴퓨터에서 전송한 데이터를 컴퓨터의 해당 프로그램으로 구분하여 배분할 때 사용하는 번호



InetAddress 클래스 (1)

InetAddress 클래스

❖ 정의

- IP 주소를 개체로 모델링한 클래스로 IP 주소와 관련된 여러 정보 제공.

❖ 특징

- InetAddress 클래스는 생성자가 없다.
- InetAddress 클래스의 메서드 InetAddress.getByName("호스트명") 메서드를 사용하여 객체를 생성한다.
- getByName() 메서드를 사용할 때 UnknowHostException에 대한 예외 처리를 해준다.

❖ 객체 생성

```
try{
    InetAddress inhatc = InetAddress.getByName("www.inhatc.ac.kr");
    InetAddress inhatcIP = InetAddress.getByName("221.154.90.151");
} catch (UnknowHostException e) {
    System.err.println(e);
}
```

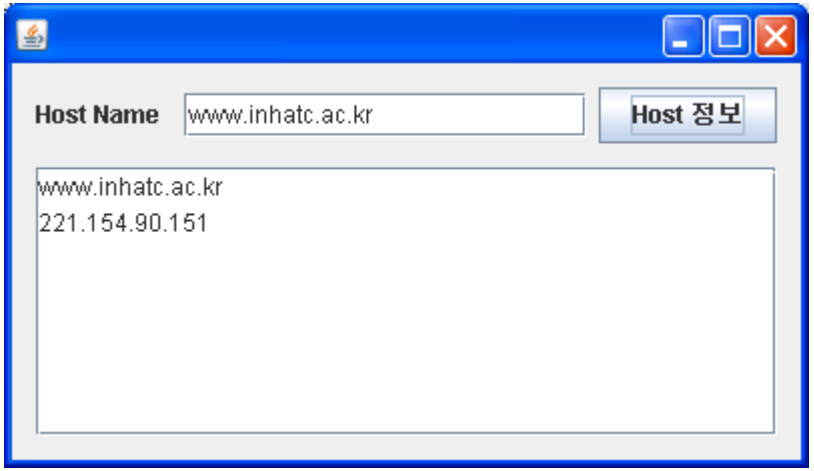
InetAddress 클래스 (2)

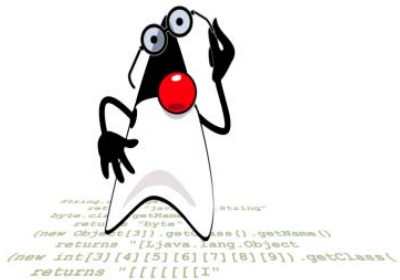
Method

Method	기 능
boolean equals(InetAddress other)	Other와 동일 객체 여부 반환
byte[] getAddress	주소를 갖는 4개 요소의 byte 배열 반환
public String getHostAddress()	컴퓨터의 주소 반환
public String getHostName()	컴퓨터 명 반환
public static InetAddress getLocalHost( ) throws UnknownHostException	현재 컴퓨터의 InetAddress 객체 반환
public static InetAddress getByName(String host) throws UnknownHostException	Host의 InetAddress 객체 반환
public static InetAddress[] getAllByName(String host) throws UnknownHostException	모든 Host의 InetAddress 객체 반환 (1개 도메인으로 다중 컴퓨터 운영)

실습 1 : InetAddress 클래스

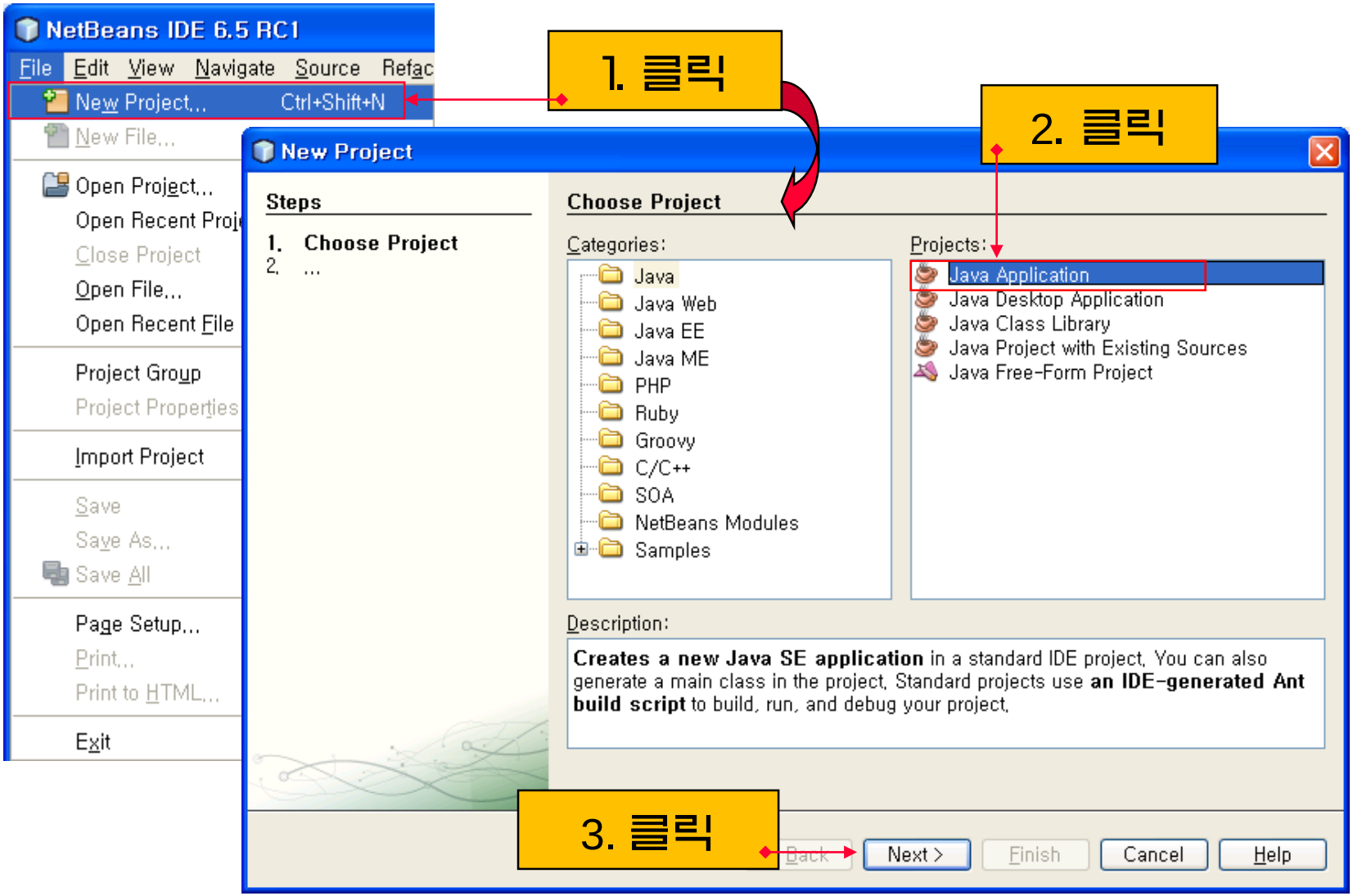
- Project Name: InetAddress_Source(실습^시간 : 30분)
  - [Host 정보] button event handler
    - 입력된 Host Name에 대한 IP 를 구하여 출력하^시오.





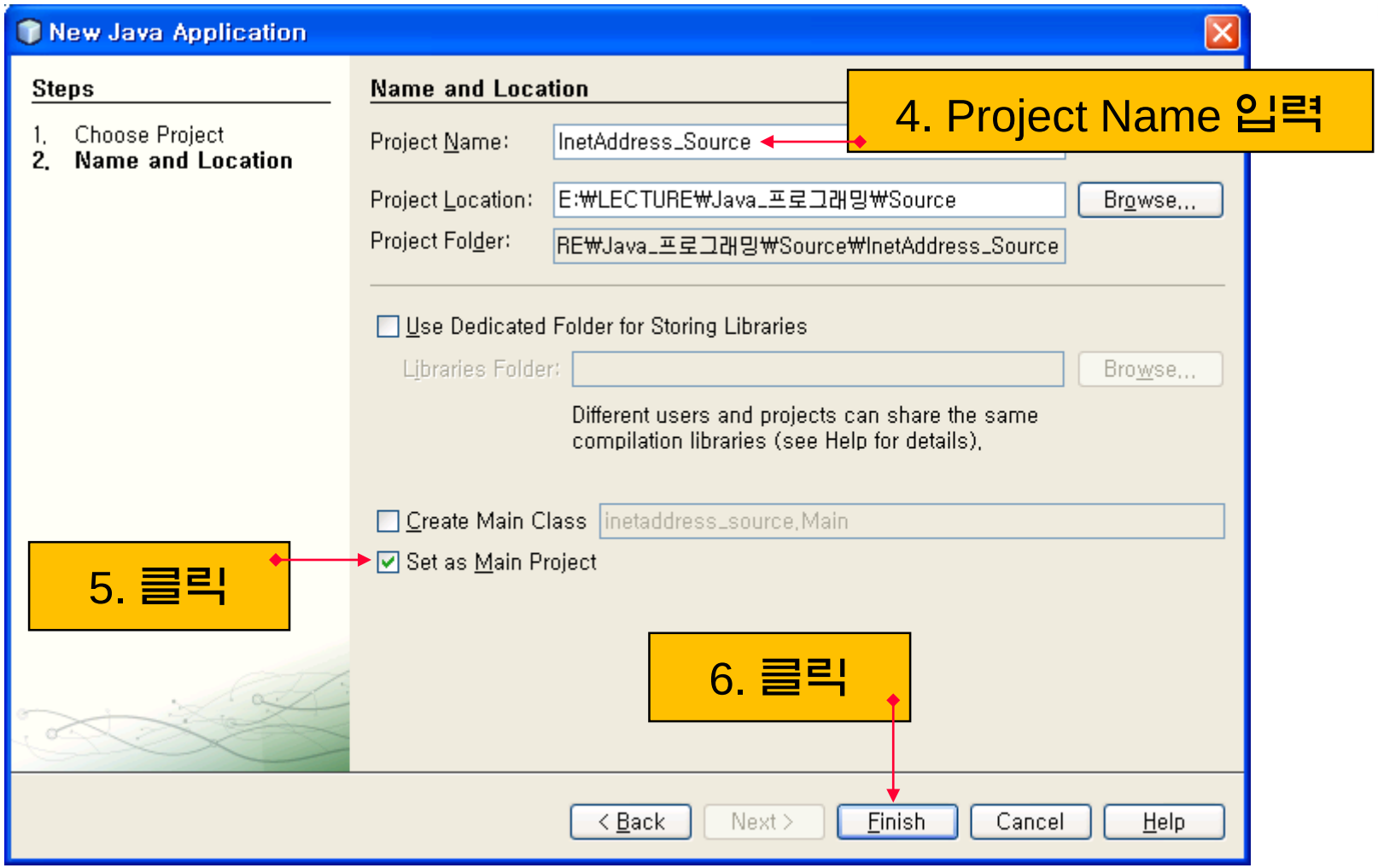
실습 1 : InetAddress 클래스 (1)

Application 생성



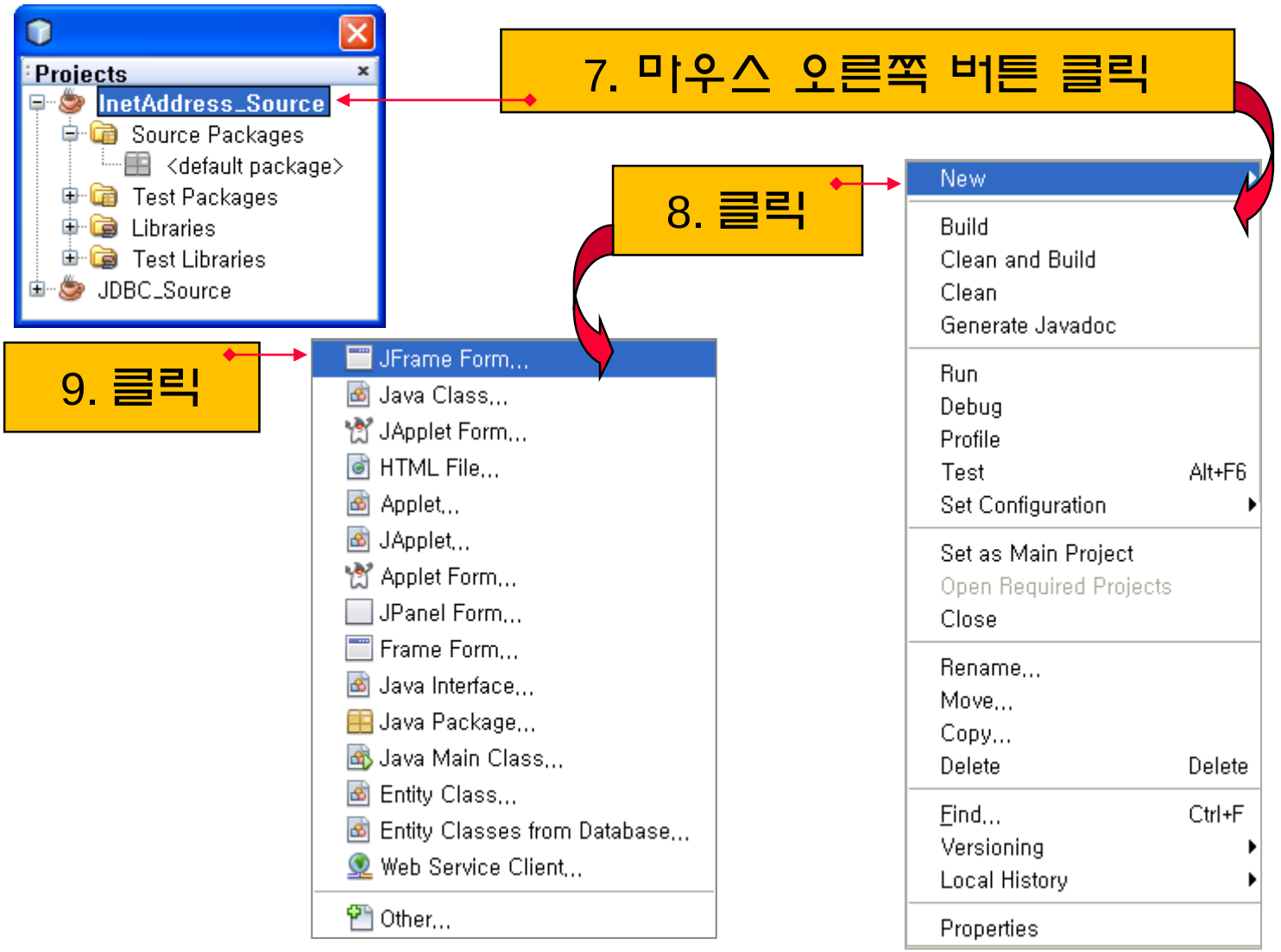
실습 1 : InetAddress 클래스 (2)

- Project Name and Location
  - Project name: InetAddress_Source



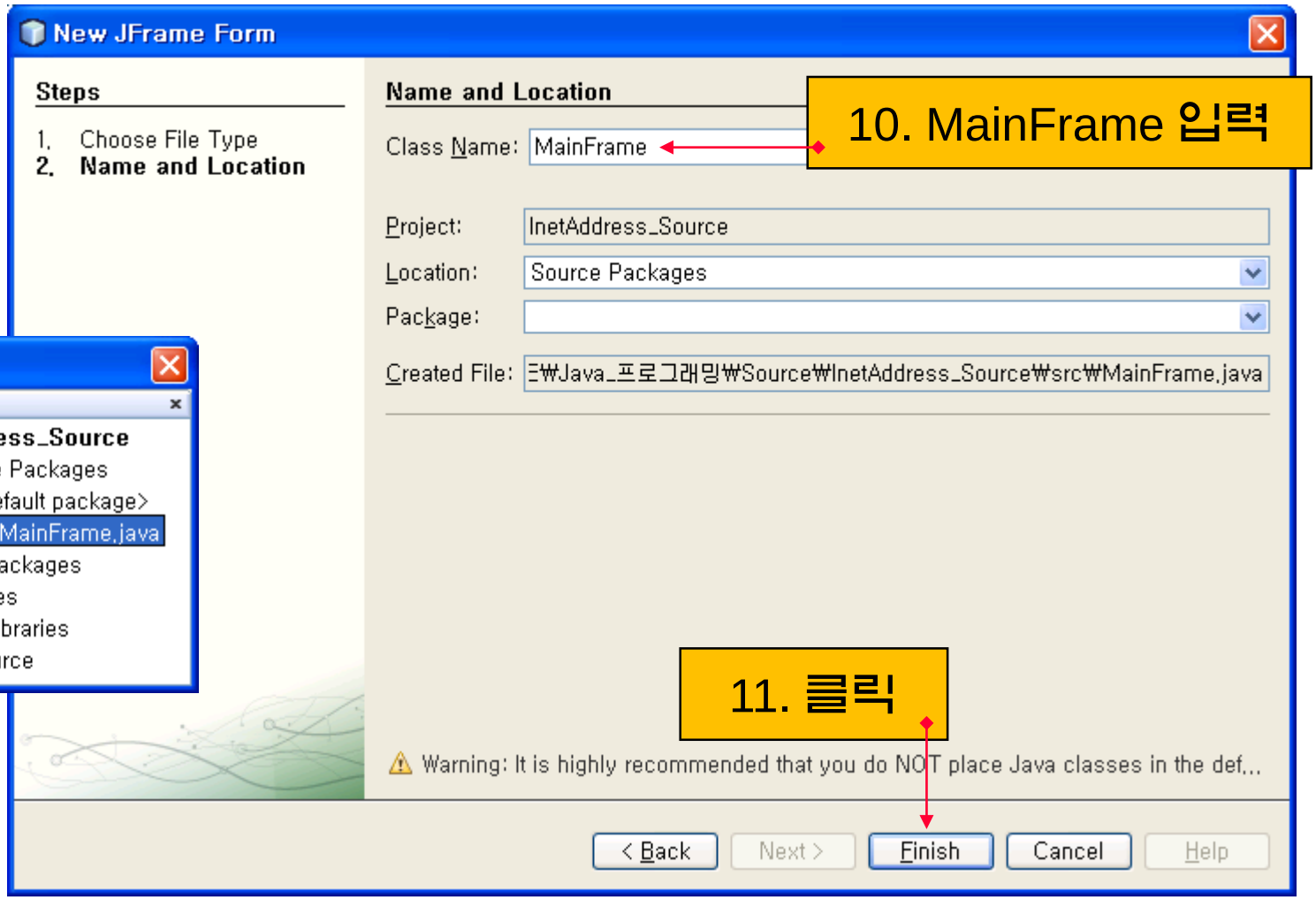
실습 1 : InetAddress 클래스 (3)

JFrame Form 생성



실습 1 : InetAddress 클래스 (4)

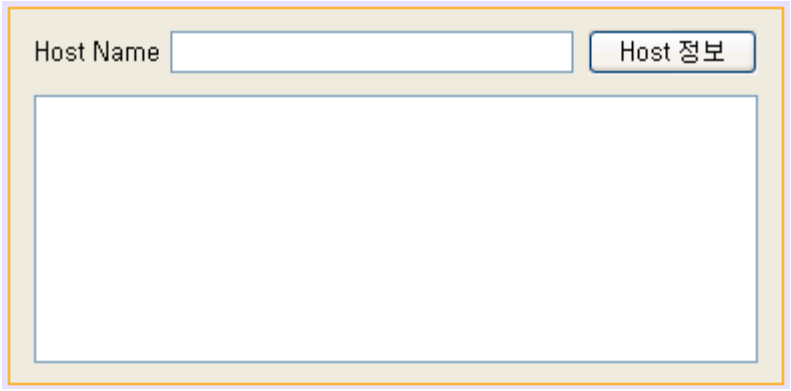
- JFrame Form Name 지정
  - MainFrame.java 생성



실습 1 : InetAddress 클래스 (5)

컨트롤 배치 및 속성 지정

- JFrame Form



12. 컨트롤 배치 및 속성 지정

컨트롤	속성 지정
jLabel1	• Variable Name : lblHostName • Text : Host Name
jTextField1	• Variable Name : txtHostName • Text :
jButton1	• Variable Name : btnHostName • Text : Host 정보
jTextArea1	• Variable Name : jTextArea1 • Text :

실습 1 : InetAddress 클래스 (6)

소스 분석

❖ MainFrame.java

MainFrame.java * – Editor

MainFrame.java *

Source | Design

```
1  /*...*/
5  import java.net.*;
6
7  /**...*/
11 public class MainFrame extends javax.swing.JFrame {
12
13     InetAddress objInetAddr;          // InetAddress 클래스 객체 선언
14
15     /** Creates new form MainFrame */
16     public MainFrame() {
17         initComponents();
18     }
19
20     /**...*/
25     @SuppressWarnings("unchecked")
26     Generated Code
```

85:13 INS

13. Coding

실습 1 : InetAddress 클래스 (7)

❖ [Host 정보] Event Handler

◆ Event

- ActionPerformed

MainFrame.java * – Editor

MainFrame.java *

Source | Design

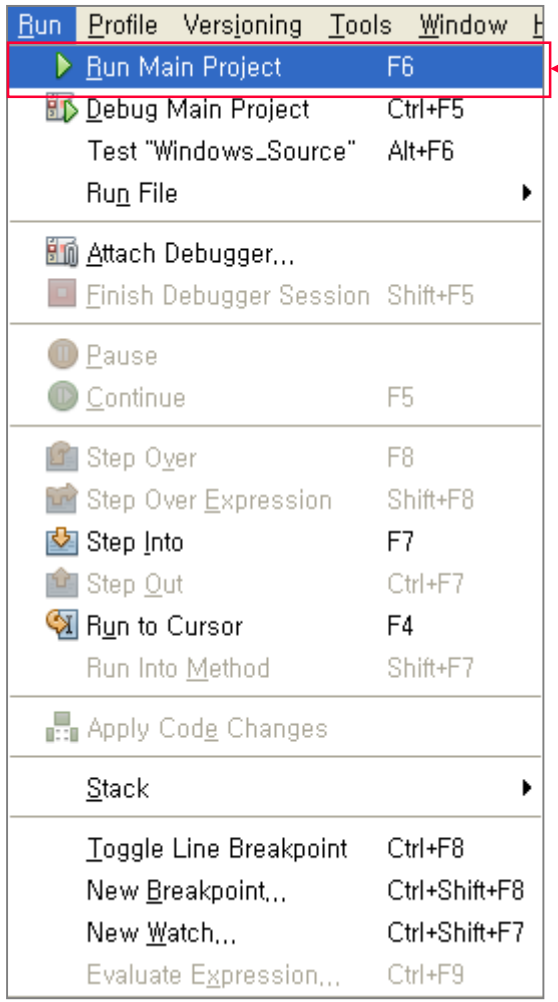
```
82  private void btnHostNameActionPerformed(java.awt.event.ActionEvent evt) {
83      String strHostName = txtHostName.getText();
84
85      try{
86
87          objInetAddr = InetAddress.getByName(strHostName);   // HostName 의 InetAddress 구함
88          String strData = objInetAddr.getHostName() + "₩n";   // Host Name 구함
89          jTextArea1.append(strData);
90          strData = objInetAddr.getHostAddress() + "₩n";       // Host Address 구함
91          jTextArea1.append(strData);
92
93      } catch (UnknownHostException ex) {
94          String  strErr = strHostName + " host가 없습니다. ₩n";
95          jTextArea1.append(strErr);
96      }
97  }
98
```

16:25 INS

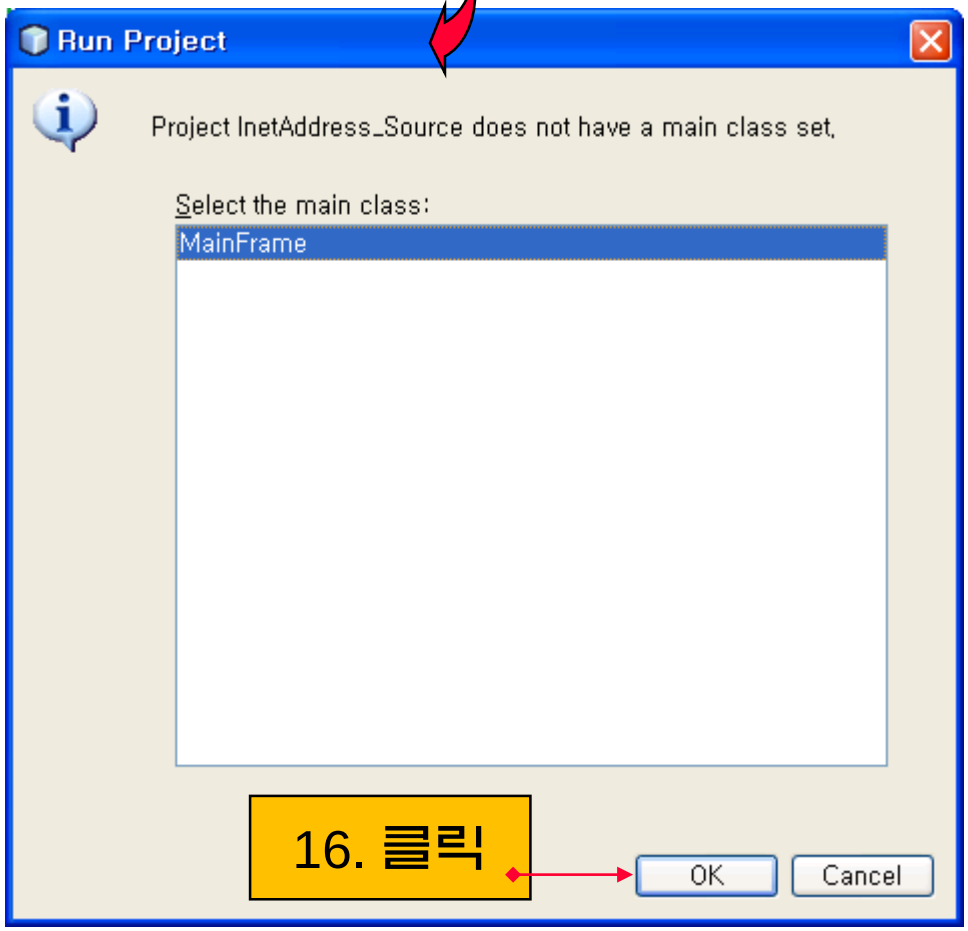
14. Coding

실습 1 : InetAddress 클래스 (8)

실행

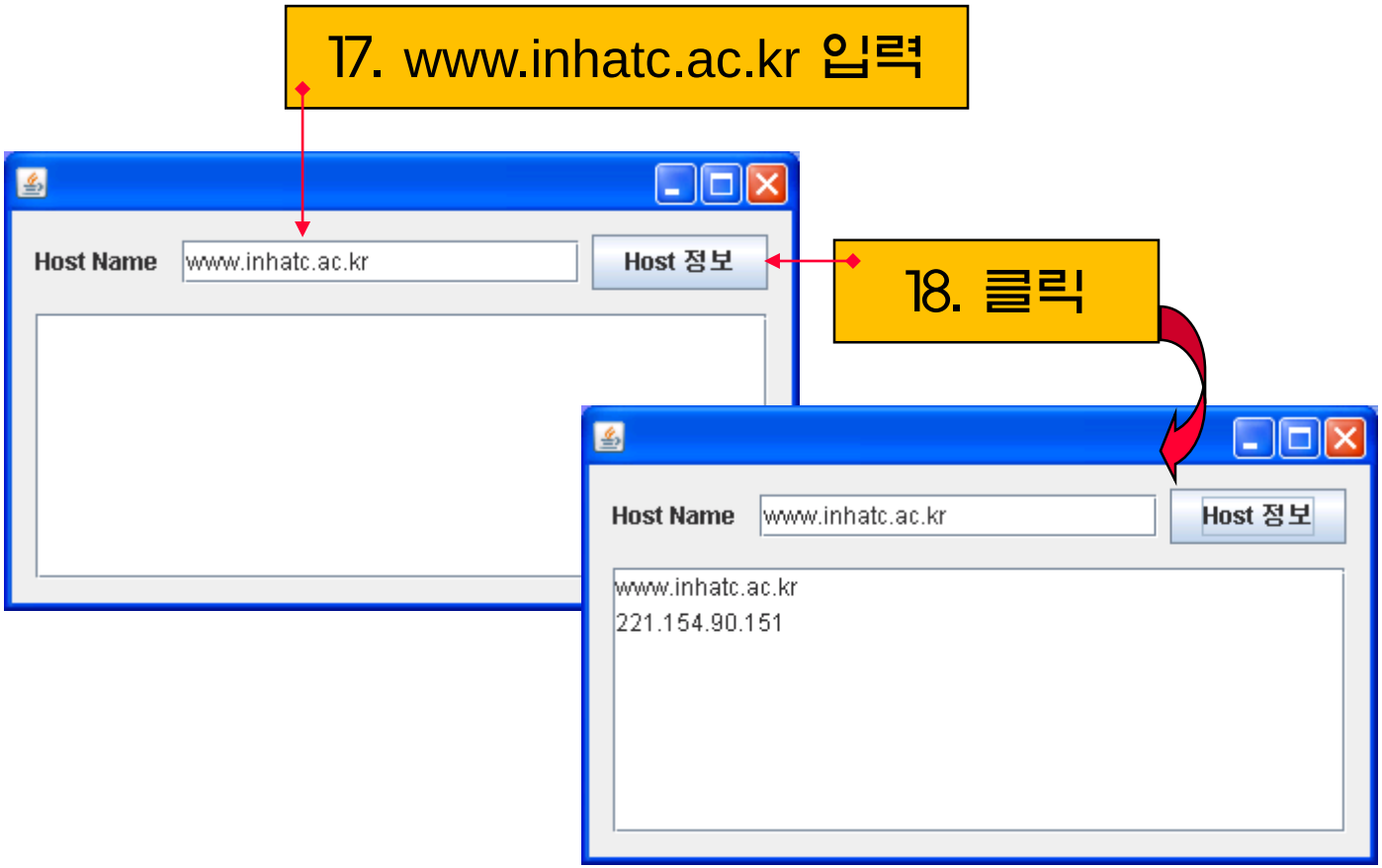


15. 클릭



실습 1 : InetAddress 클래스 (9)

실행 결과



URL 클래스 (1)

URL(Uniform Resource Locator) 클래스

❖ 기능

- ◆ 인터넷 접속을 객체로 모델링 하여 다양한 정보를 얻을 수 있는 기능 제공

❖ 구성요소

프로토콜://호스트:[포트 번호]/[파일]#[섹션]
http://www.microsoft.com file://C:\Share\test.html http://cs.inhatc.ac.kr:80/index.html ftp://ftp.kaist.ac.kt/pub/ mailto: kdhong@hotmail.com telnet://para.hana.net

URL 클래스 (2)

URL 클래스 객체 생성

public URL(String u) throws MalformedURLException
public URL(String protocol, String host, String file) throws MalformedURLException
public URL(String protocol, String host, int port, String file) throws MalformedURLException
public URL(URL context, String u) throws MalformedURLException

```
URL u = null;
try{
   u = new URL("http", "www.microsoft", 8080, "/index.html#cs");
} catch (MalformedURLException e){
   System.out.println(e);
}
```

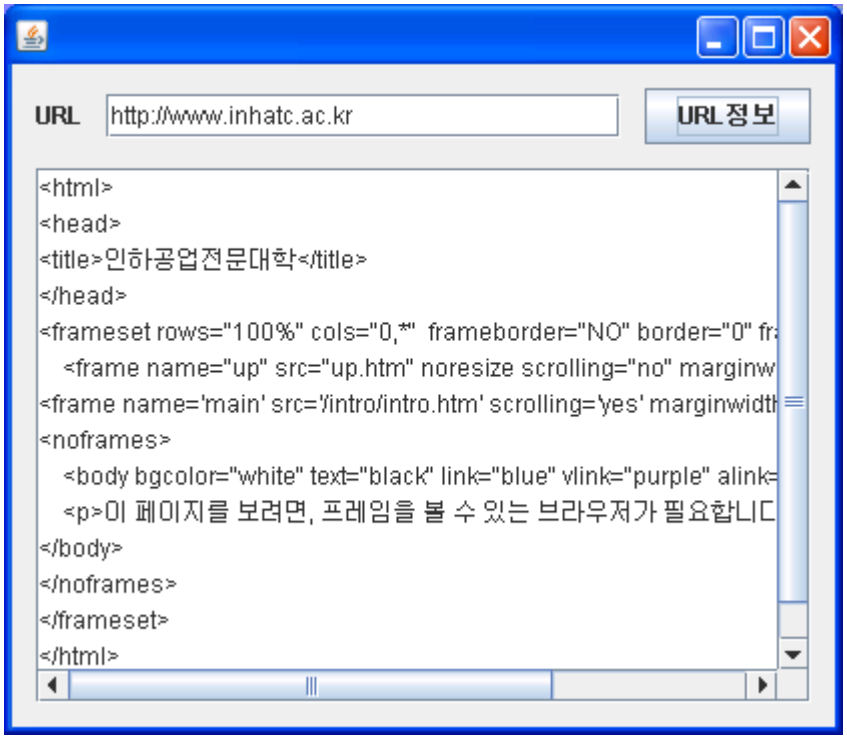
URL 클래스 (3)

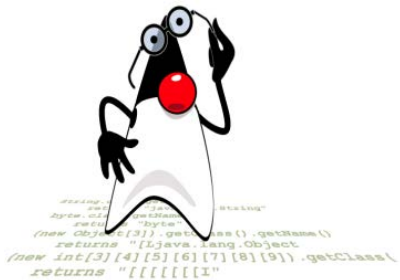
Method

Method	기 능
public String getProtocol()	Protocol 반환
public String getHost()	Host 반환
public String getPort()	Port 반환
public String getFile()	File 정보 반환
public String getRef()	Ref 정보 반환

실습 2 : URL 클래스

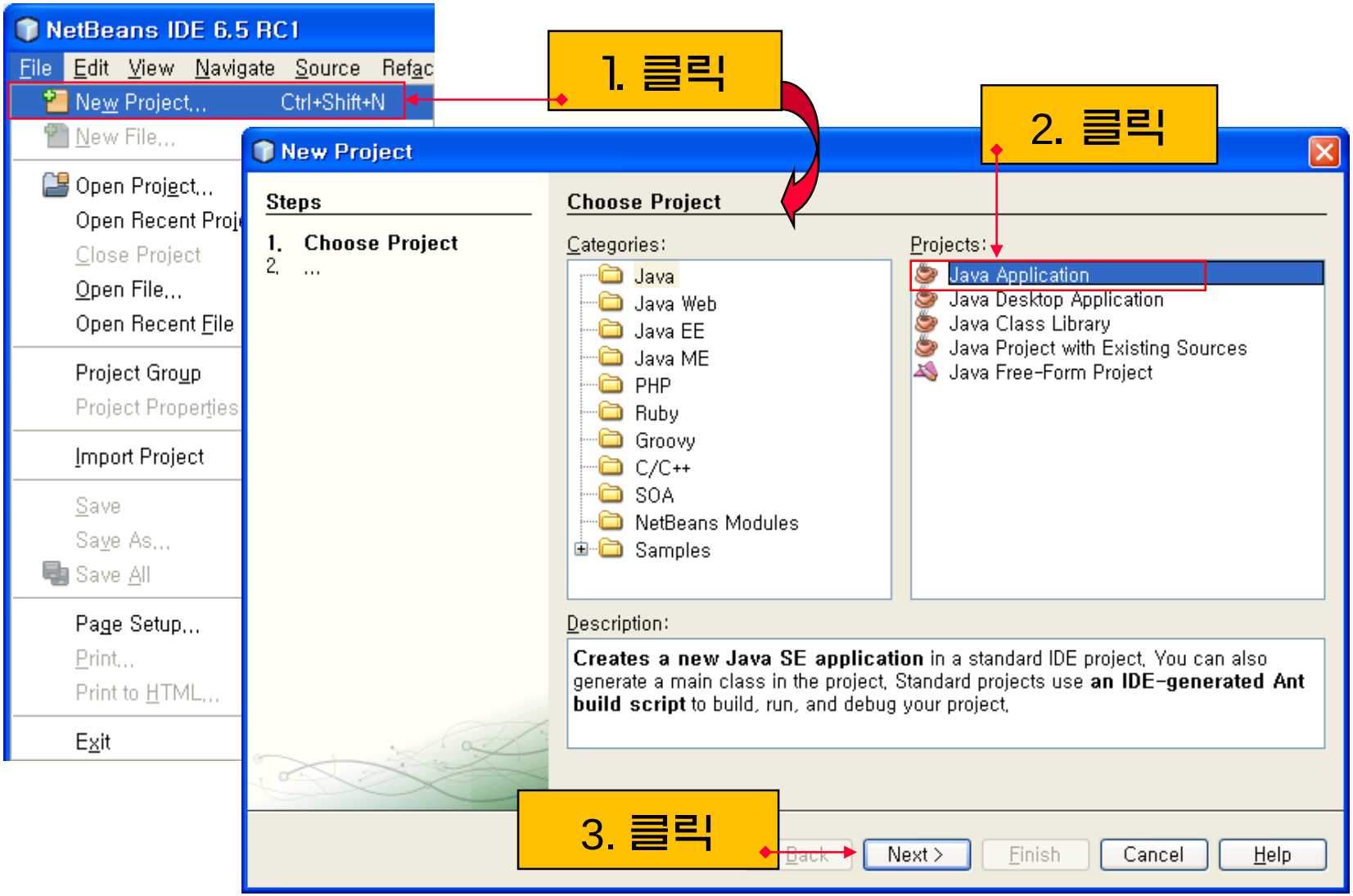
- Project Name: URL_Source(실습^시간 : 30분)
  - [URL 정보] button event handler
    - 입력된 URL에 대한 정보(html)를 jTextArea에 출력하^시오.





실습 2 : URL 클래스 (1)

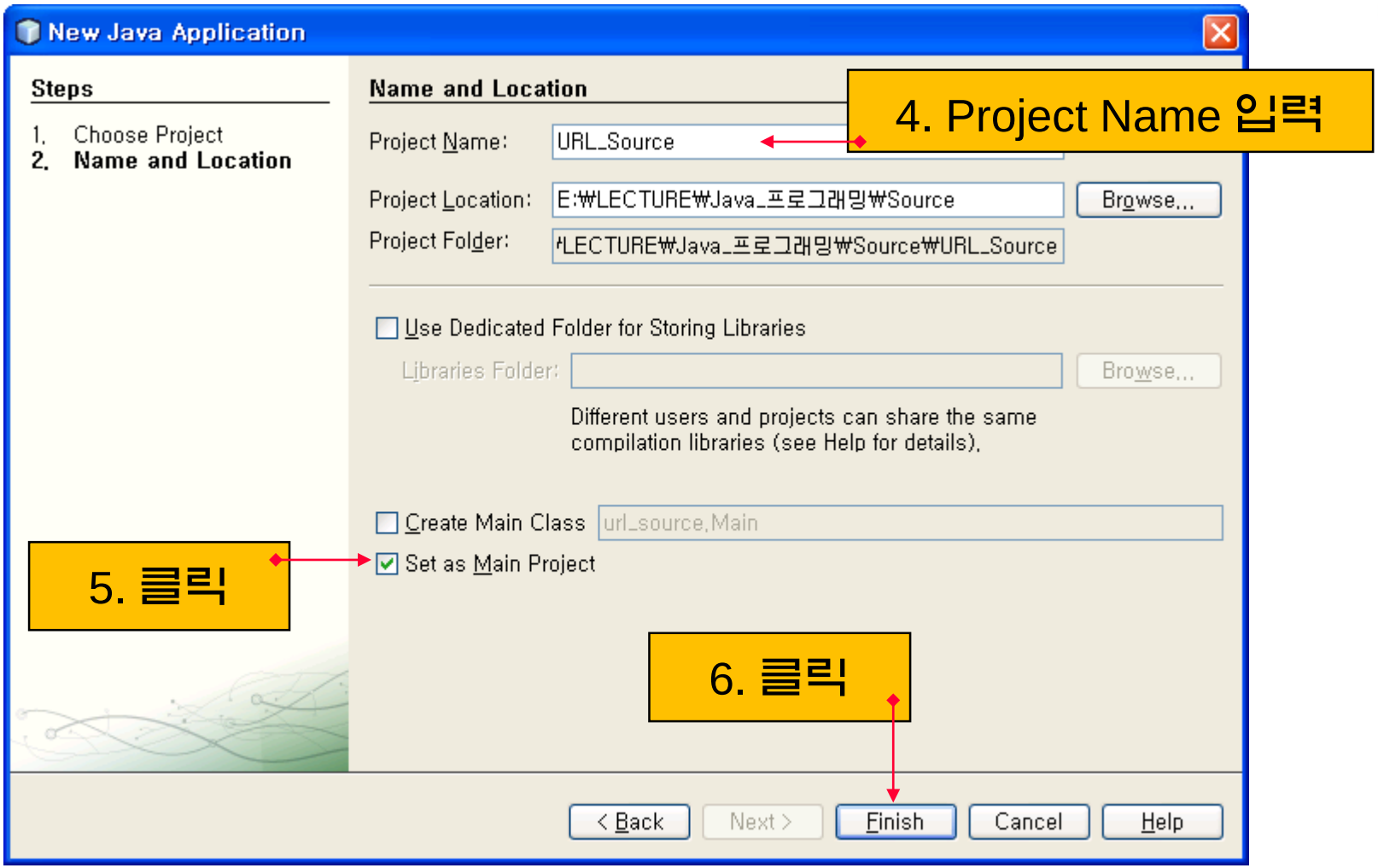
Application 생성



실습 2 : URL 클래스 (2)

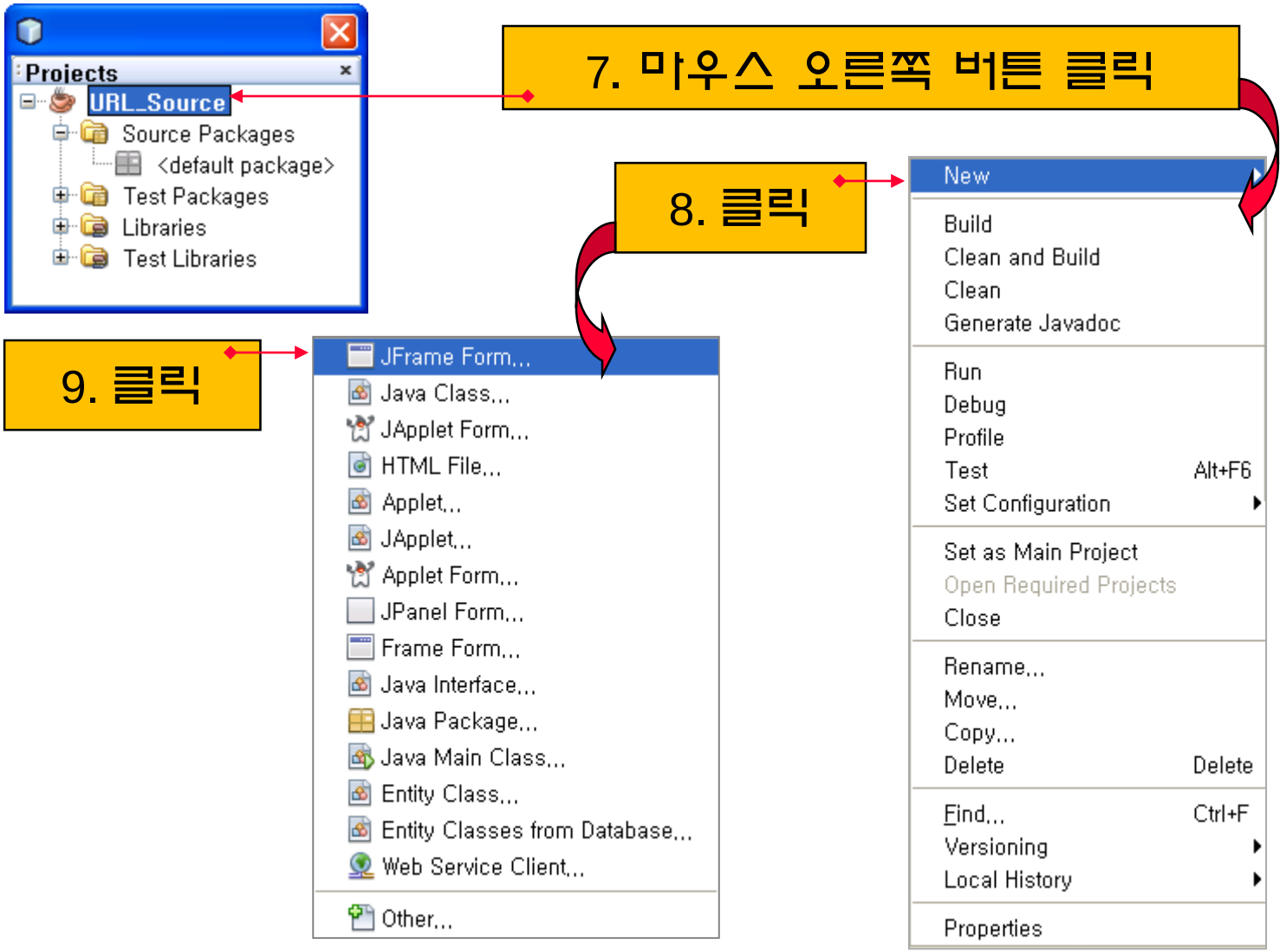
Project Name and Location

- Project name: URL_Source



실습 2 : URL 클래스 (3)

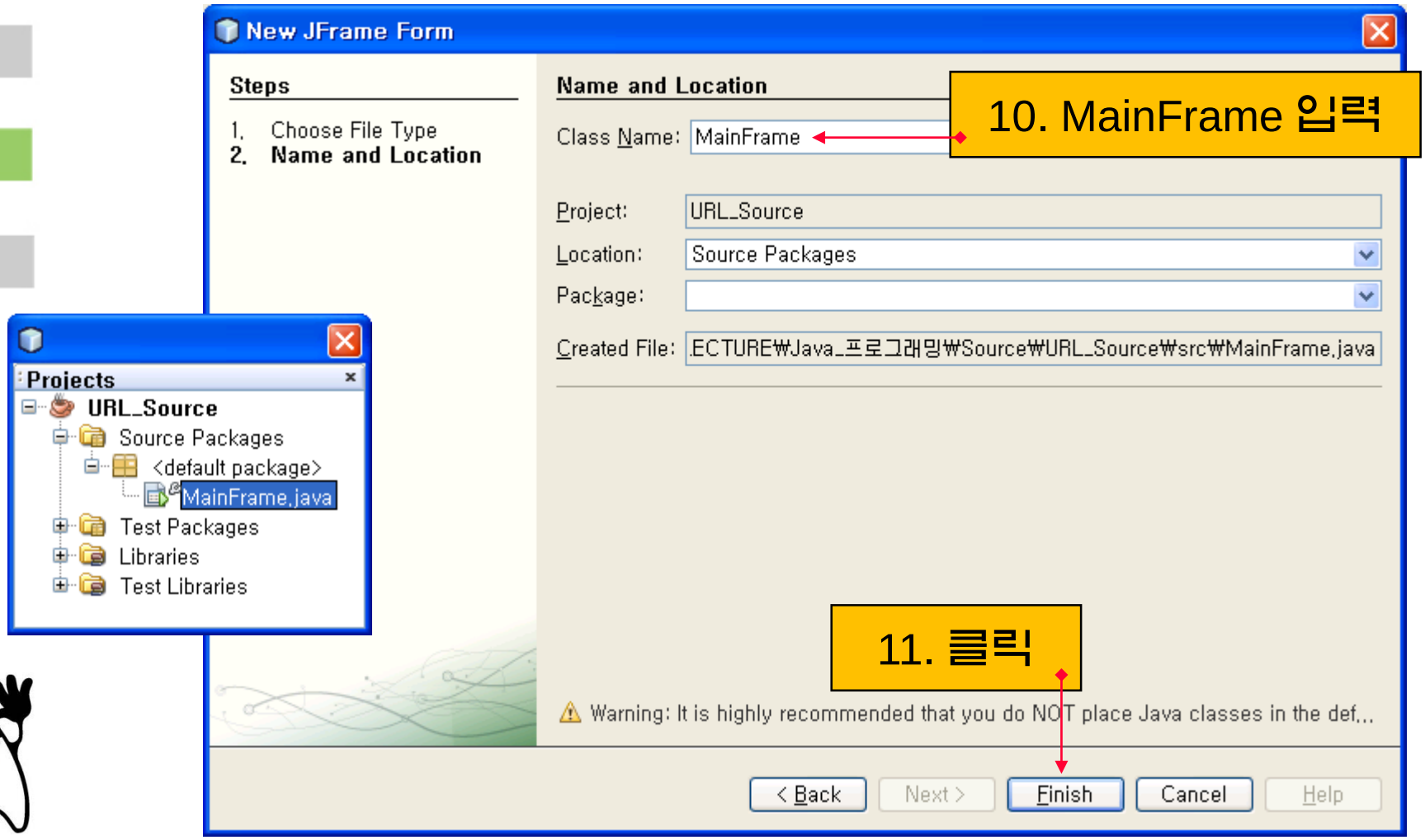
JFrame Form 생성



실습 2 : URL 클래스 (4)

JFrame Form Name 지정

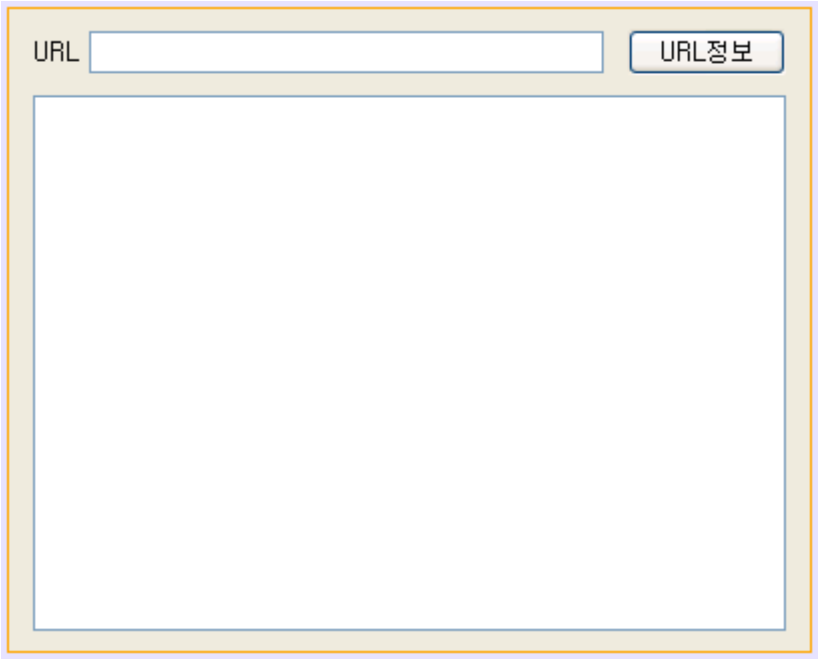
❖ MainFrame.java 생성



실습 2 : URL 클래스 (5)

컨트롤 배치 및 속성 지정

- JFrame Form



컨트롤	속성 지정
jLabel1	• Variable Name : lblURL • Text : URL
jTextField1	• Variable Name : txtURL • Text :
jButton1	• Variable Name : btnURL • Text : URL정보
jTextArea1	• Variable Name : jTextArea1 • Text :

12. 컨트롤 배치 및 속성 지정

실습 2 : URL 클래스 (6)

소스 분석

❖ MainFrame.java

MainFrame.java * – Editor

MainFrame.java *

Source | Design

```
1   /*...*/
5
6   import java.net.*;
7   import java.io.*;
8
9   /**...*/
13  public class MainFrame extends javax.swing.JFrame {
14      URL URLName;                    // URL 객체
15      InputStream  inData;            // I/O 데이터 객체
16      BufferedReader bufData;         // BufferedReader 객체
17      String  strData;
18
19      /** Creates new form MainFrame */
20      public MainFrame() {
21          initComponents();
22      }
```

91:58 INS

13. Coding

실습 2 : URL 클래스 (7)

❖ [URL정보] Event Handler

◆ Event

- ActionPerformed

MainFrame.java * - Editor

MainFrame.java *

Source | Design

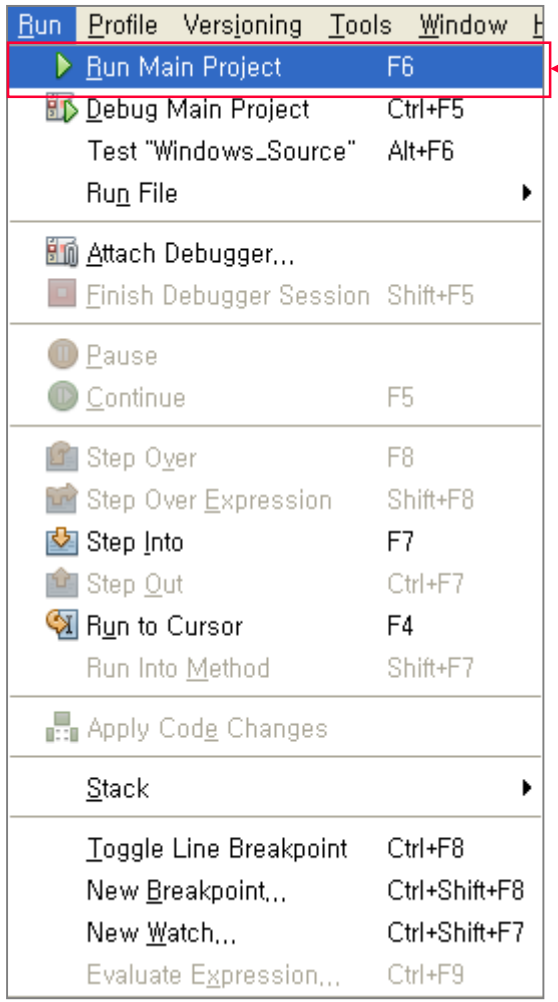
```
87  private void btnURLActionPerformed(java.awt.event.ActionEvent evt) {
88      try{
89          String strURL = txtURL.getText();    // URL 입력
90          URLName = new URL(strURL);           // URL 객체 생성
91          jTextArea1.append(strData);
92          inData = URLName.openStream(); // URL 접속 후 데이터 입력
93          bufData = new BufferedReader(new InputStreamReader(inData ));
94          while( (strData = bufData.readLine() ) != null ){
95              jTextArea1.append(strData + "₩n");
96          }
97      } catch (MalformedURLException me) {
98          System.out.println(me);
99      } catch (IOException ie) {
100         System.out.println(ie);
101     }  catch (Exception ee) {
102         System.out.println(ee);
103     }
104 }
```

109:38 INS

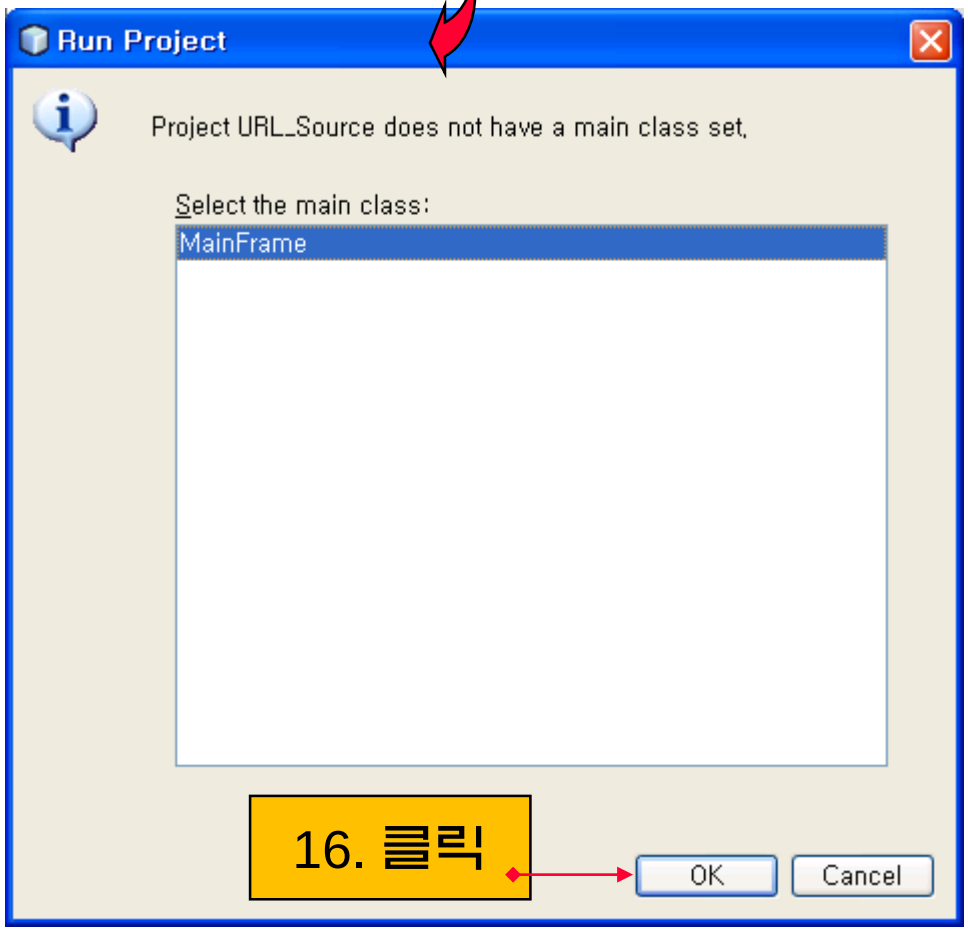
14. Coding

실습 2 : URL 클래스 (8)

실행

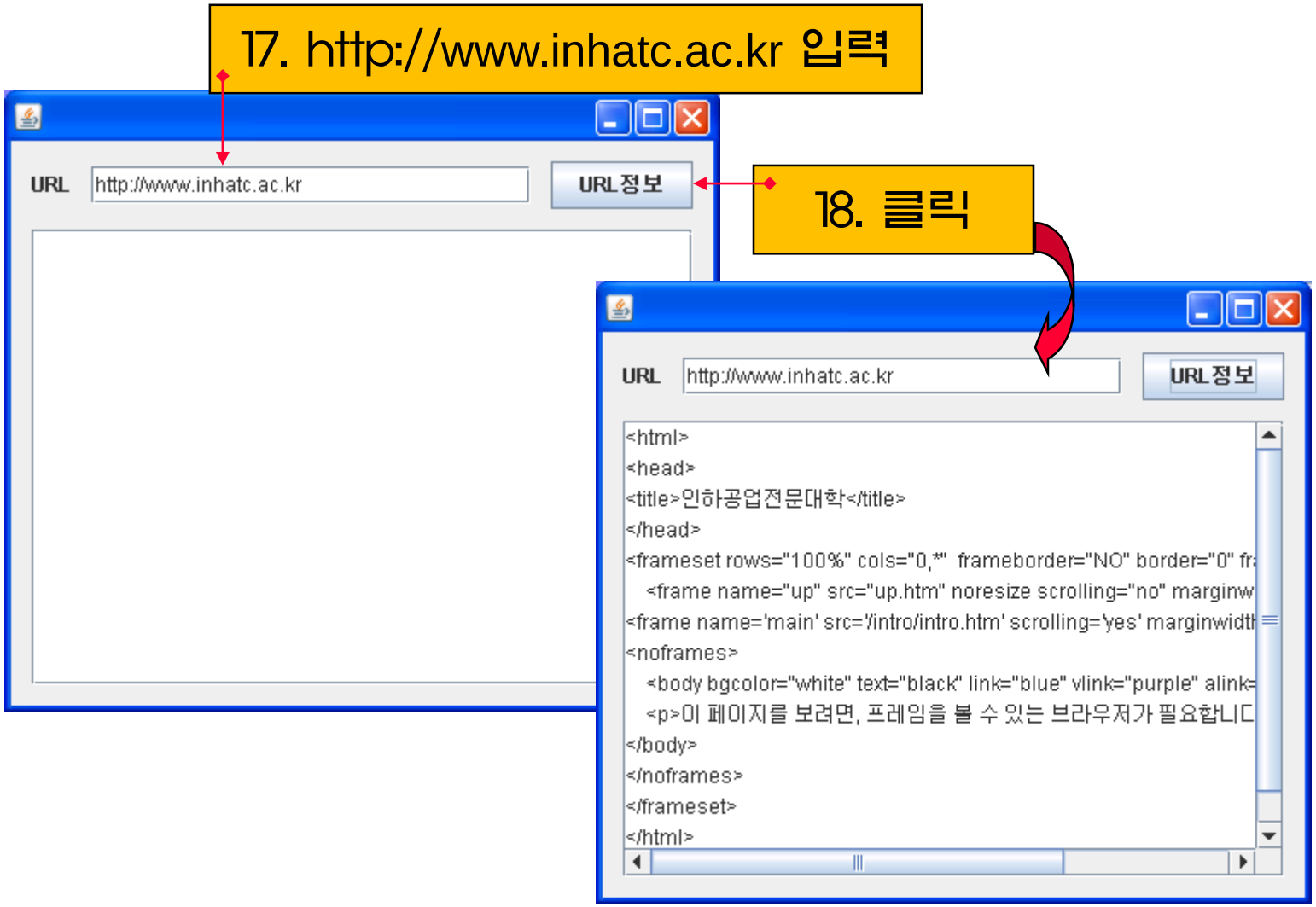


15. 클릭



실습 2 : URL 클래스 (9)

실행 결과



URLConnection 클래스 (1)

URLConnection 클래스

- 기능
  - 서버 접속을 객체로 모델링 하여 다양한 정보를 얻을 수 있는 기능 제공
- URLConnection 생성 및 접속

public URLConnection openConnection() throws IOException
URLConnection 생성

public abstract void connect() throws IOException
Network을 통한 실제 연결

URLConnection 클래스 (2)

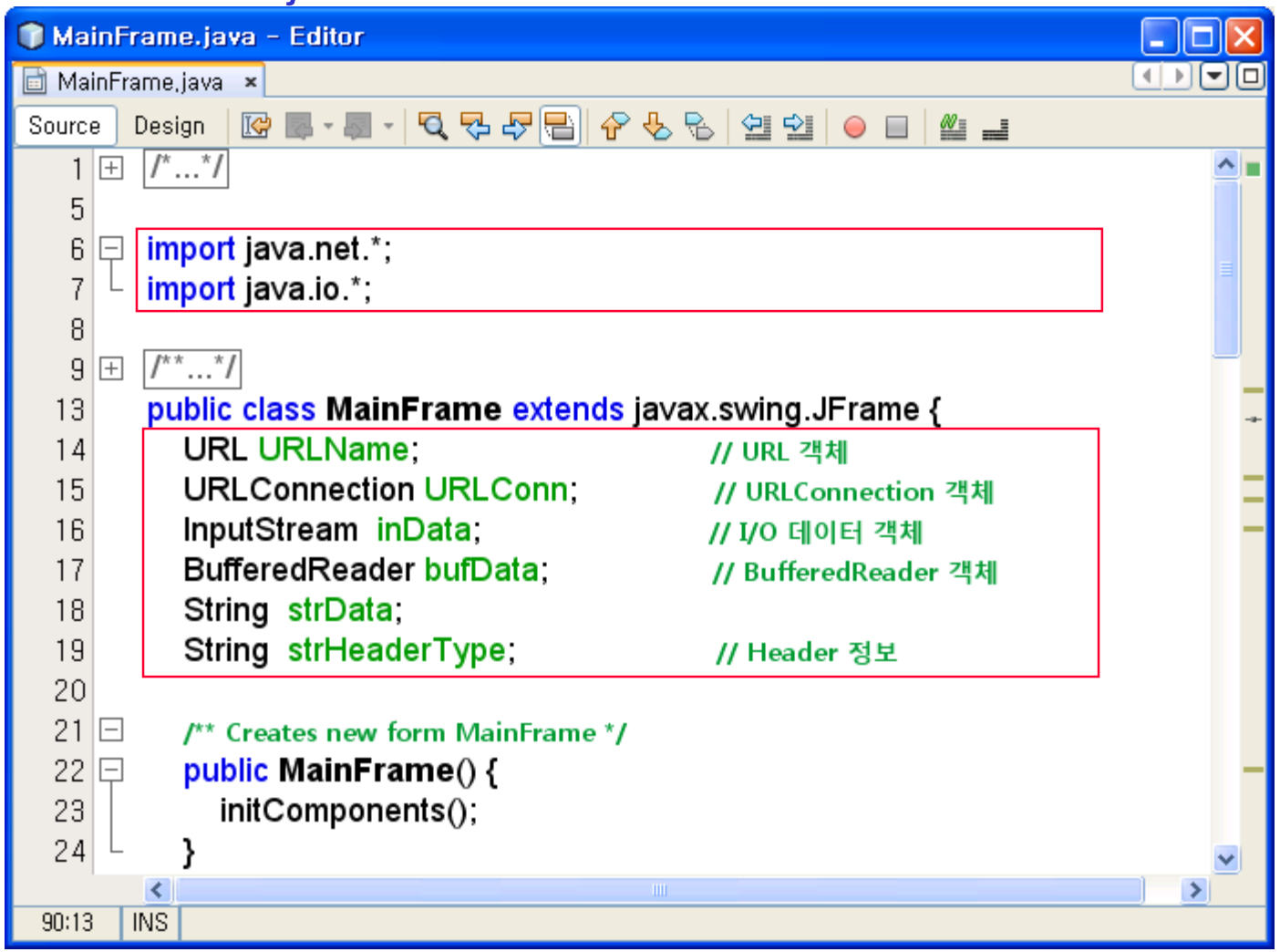
URL 클래스 객체 생성

Method	설명
InputStream getInputStream()	통신 채널의 입력 스트림 반환
OutputStream getOutputStream()	통신 채널의 출력 스트림 반환
String getHeaderField()	통신 채널의 헤더 정보 반환
URL getURL()	URL 클래스 반환

URLConnection 클래스 (3)

소스 분석

MainFrame.java



URLConnection 클래스 (4)

- [URL정보] Event Handler
  - Event
    - ActionPerformed

MainFrame.java – Editor

MainFrame.java

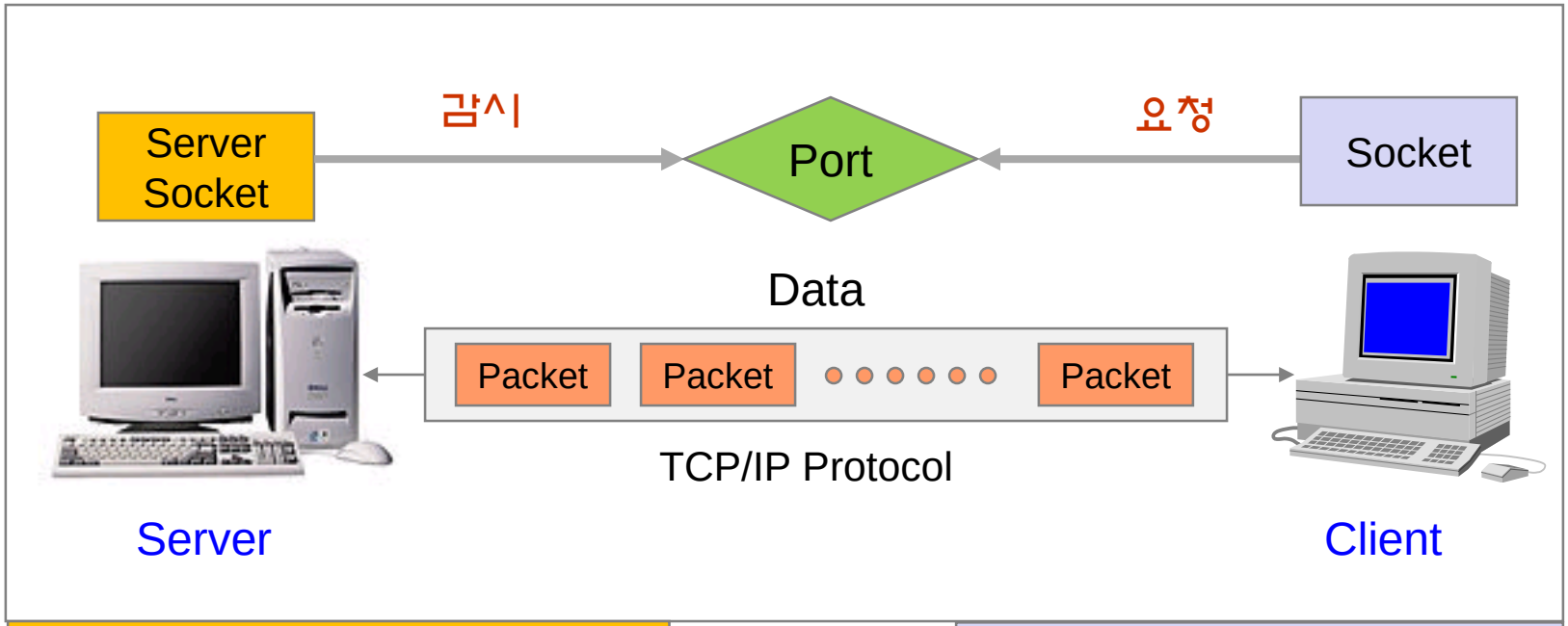
Source | Design

```
89  private void btnURLActionPerformed(java.awt.event.ActionEvent evt) {
90      try{
91          String strURL = txtURL.getText();              // URL 입력
92          URLName = new URL(strURL);                     // URL 객체 생성
93          URLConn = URLName.openConnection();            // URL Connection  객체 생성
94          URLConn.connect();                             // 실제 접속
95          strHeaderType = URLConn.getContentType();      // Header 유형 정보
96          jTextArea1.append("MIME Type : " + strHeaderType + "₩n");
97
98          inData = URLConn.getInputStream();             // 입력 stream 반환
99          bufData = new BufferedReader(new InputStreamReader(inData ));
100         while( (strData = bufData.readLine() ) != null ){
101             jTextArea1.append(strData + "₩n");
102         }
103     } catch (MalformedURLException me) {
104         System.out.println(me);
105     } catch (IOException ie) {
106         System.out.println(ie);
107     }  catch (Exception ee) {
108         System.out.println(ee);
109     }
110 }
```

12:4 INS

TCP Socket 프로그래밍 (1)

TCP Socket 프로그래밍



1. ServerSocket 객체 생성후 Client 요청 대기
2. Client 요청 발생시 Socket 객체 생성
3. Socket 객체로 I/O 스트림 생성
4. Data R/W
5. Socket 해제

1. 접속할 서버 주소와 포트번호로 Socket 객체 생성
2. Socket 객체로 I/O 스트림 생성
3. Data R/W
4. Socket 해제

TCP Socket 프로그래밍 (2)

Server 구현

❖ Server Socket 클래스 접속 처리 방법

- ◆ Server Socket 클래스 객체 생성 : 포트번호는 서버가 감시하는 포트번호
- ◆ Server Socket은 Client의 접속 요구를 감시할 뿐 data를 read/write할 수 없다.

Server Socket 생성
public ServerSocket(int port) throws IOException public Socket accept() throws IOException

[Sample Source]

```
try{
        ServerSocket  sso = new ServerSocket(2345);
        Socket so = sso.accept();
        so.close();
} catch (IOException e) {
                System.err.println(e);
}
```

TCP Socket 프로그래밍 (3)

Client 구현

❖ Socket 클래스 생성자

- ◆ Socket 클래스 객체는 한번 접속 후 접속된 원격호스트를 변경할 수 없다.
- ◆ Socket의 data를 read/write 하는 과정은 서로 독립적이다.

생성자
public Socket(String host, int port) throws UnknownHostException, IOException
public Socket(InetAddress address, int port) throws IOException
public Socket(String host, int port, InetAddress localaddress, int localport) throws IOException
public Socket(InetAddress address, int port, InetAddress localaddress, int localport) throws IOException
[Sample Source] Socket Inhaweb = new Socket("www.inhatc.ac.kr", 80);

TCP Socket 프로그래밍 (4)

Data Read

Socket에서 Data Read

```
try{
    Socket sok = new Socket("www.inhatc.ac.kr", 2002);
    InputStream inStr = sok.getInputStream();
    DataInputStream dataIS = new DataInputStream(inStr);
    String  receive_Str = dataIS.readLine();  //data 수신
    sok.close();     //Socket 접속 해제
} catch (IOException e) {
    return (new Date()).toString();
}
```

TCP Socket 프로그래밍 (5)

Data Write

Socket에서 Data Write

```
Byte[] outStr = new byte[128];
try{
        Socket  sok = new Socket("www.inhatc.ac.kr", 9);
        OutputStream outdata =  sok.getOutputStreamReader();
        while (true){
                int n = outdata.available();
                if(n> outStr.length) n = outStr.length;
                int m = outdata.read(outstr, 0, n);
                if (m == -1) break;
                outdata.write(outstr, 0, n);
        }
        sok.close();
} catch (IOException e) {

}
```

TCP Socket 프로그래밍 (6)

접속 해제

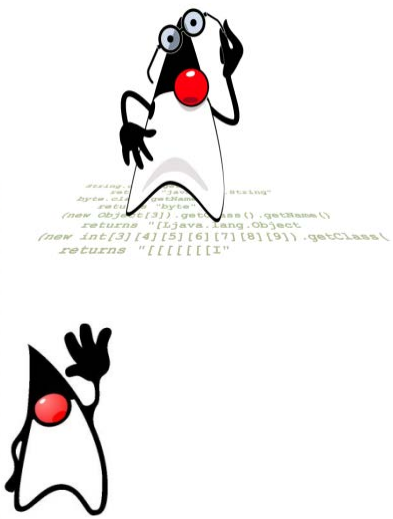
Socket 접속 해제
public synchronized void close() throws IOException

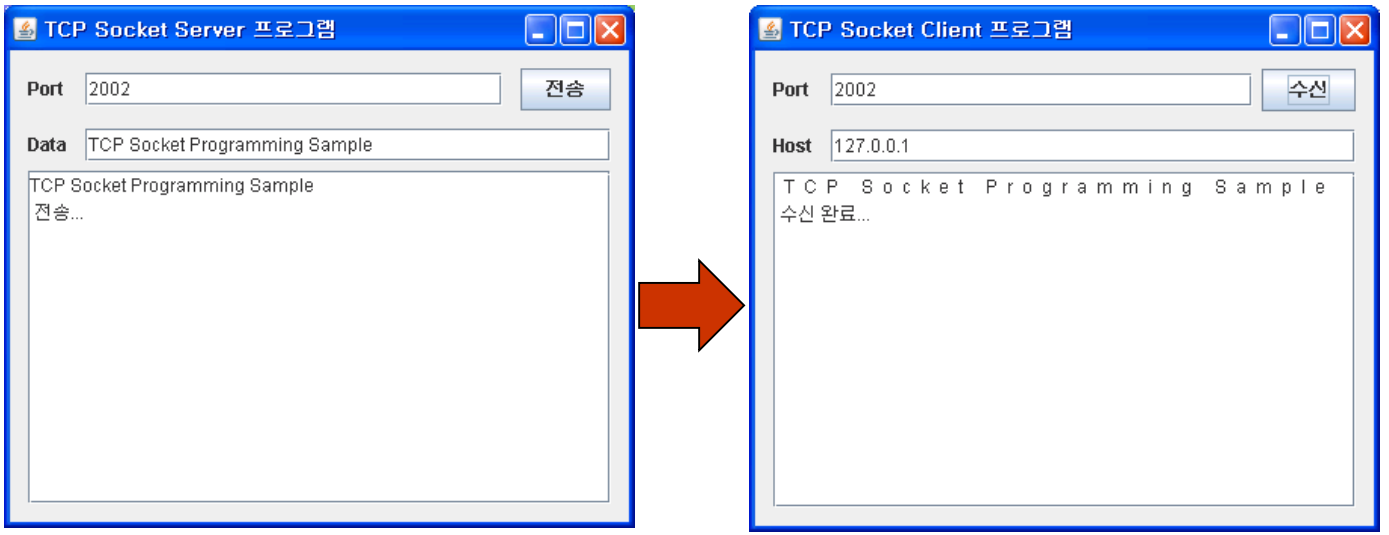
Method

Method	설명
int getPort()	현재 Socket에 접속된 컴퓨터의 포트 번호 반환
int getLocalPort()	현재 Socket을 사용하고 있는 컴퓨터의 포트 번호 반환

실습 3 : TCP Socket 프로그래밍

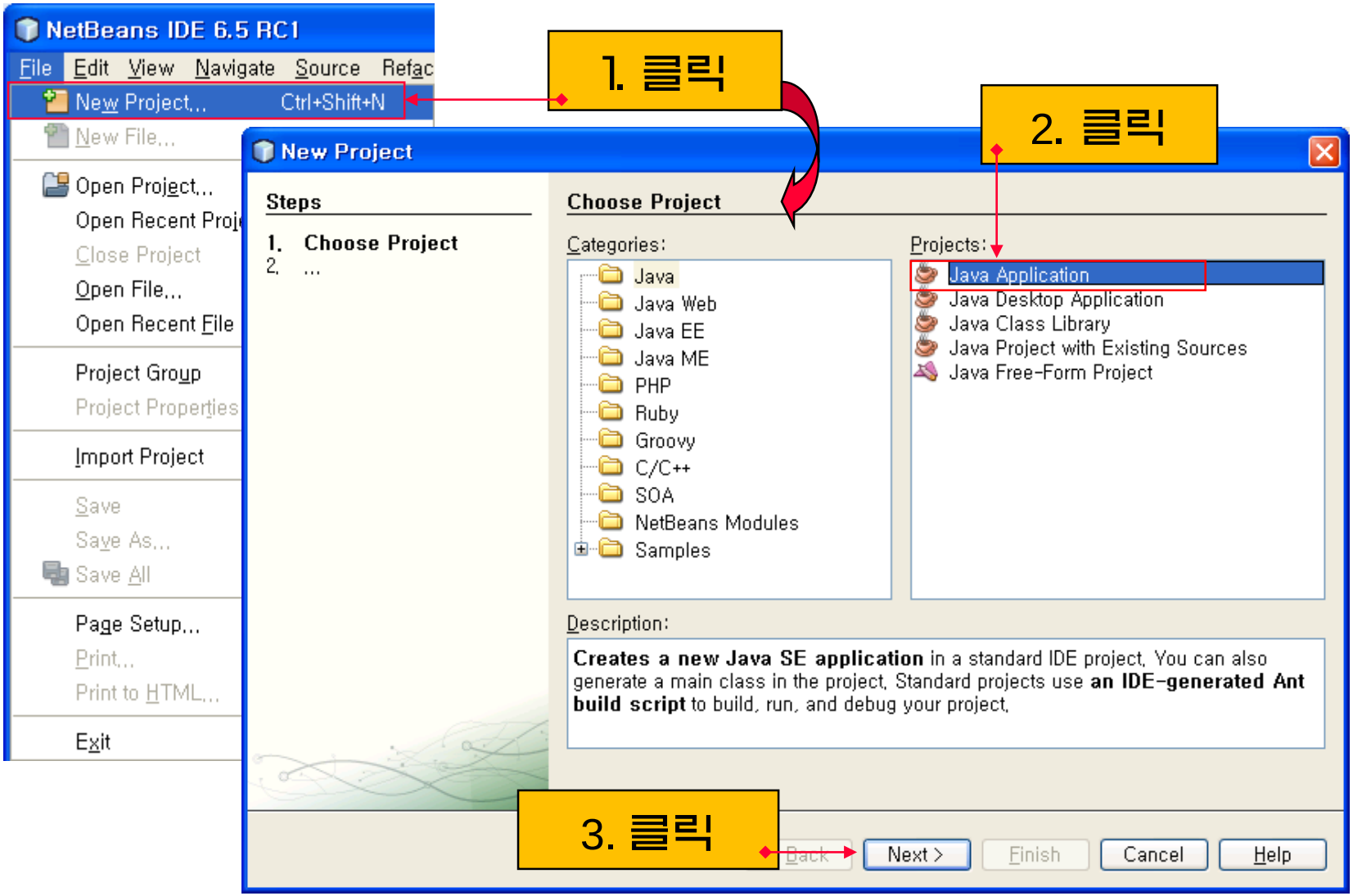
- Project Name: TCP_Source(실습^시간 : 30분)
  - Server 프로그램
    - Port, Data를 입력한 후 [전송] 버튼 클릭
  - Client 프로그램
    - Host, Port를 입력한 후 [수신] 버튼을 클릭하면 Server에서 전송한 data를 jTextArea1 컴포넌트에 출력





실습 3 : TCP Socket Server 프로그래밍 (1)

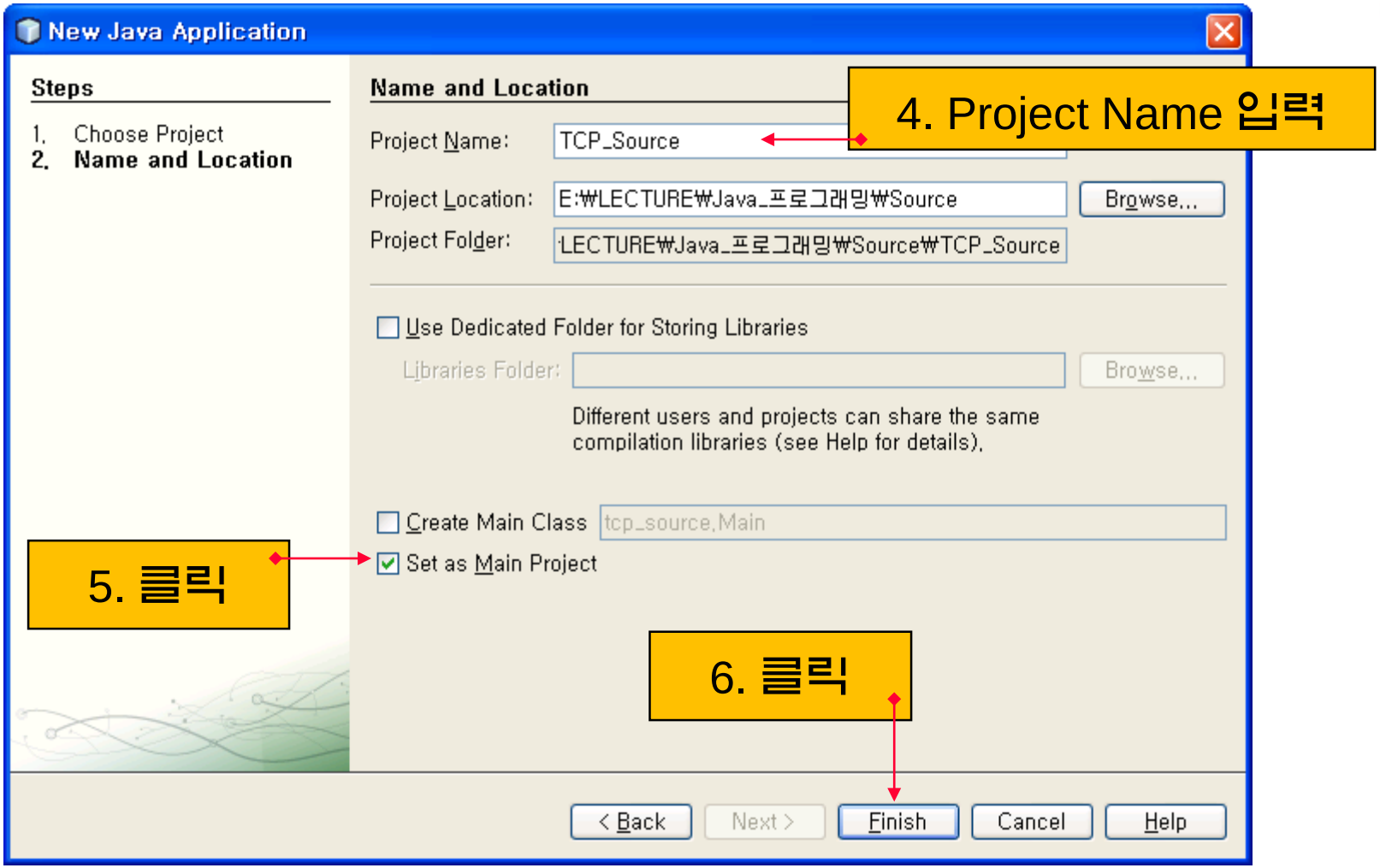
Application 생성



실습 3 : TCP Socket Server 프로그래밍 (2)

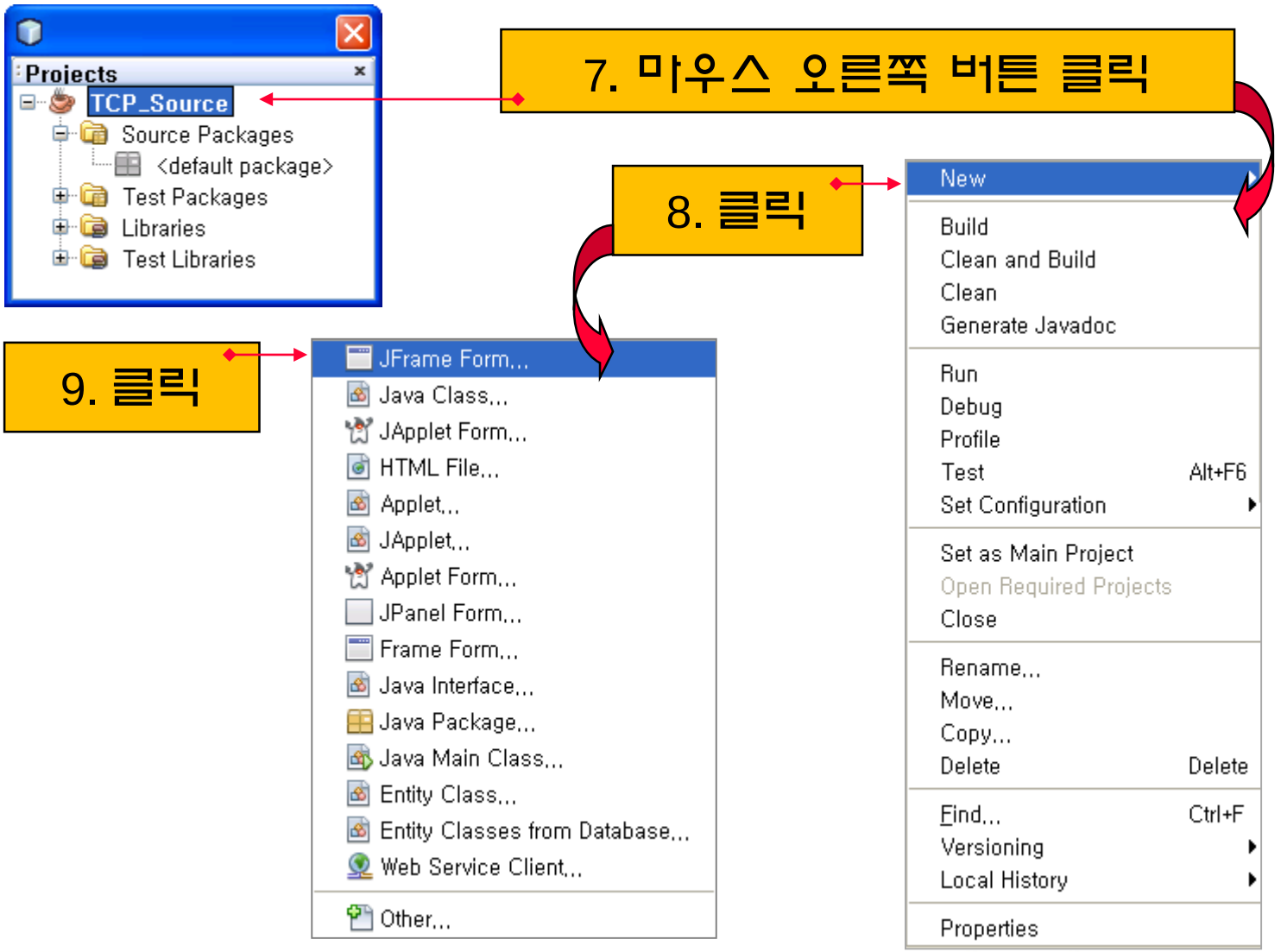
Project Name and Location

- Project name: TCP_Source



실습 3 : TCP Socket Server 프로그래밍 (3)

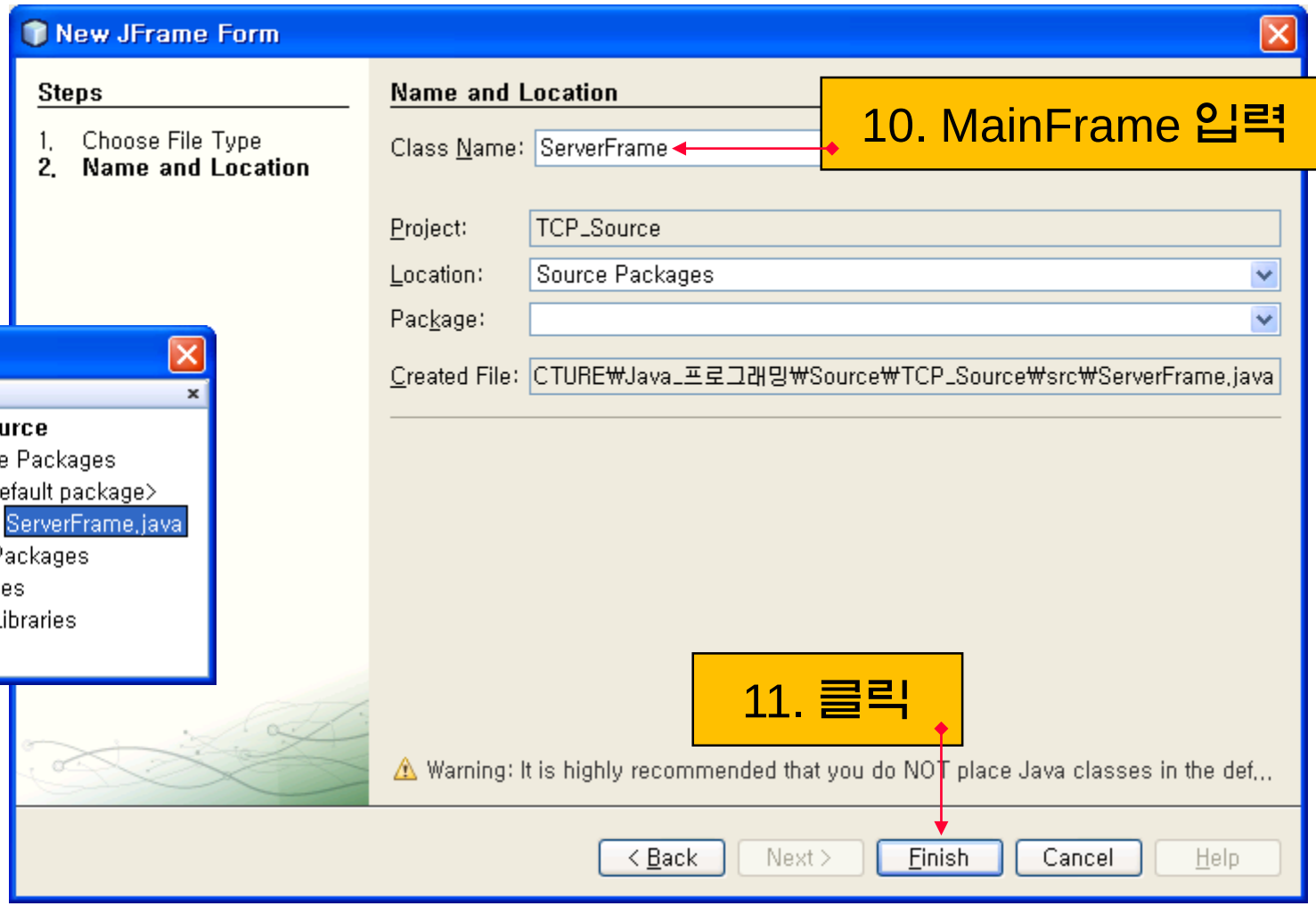
JFrame Form 생성



실습 3 : TCP Socket Server 프로그래밍 (4)

JFrame Form Name 지정

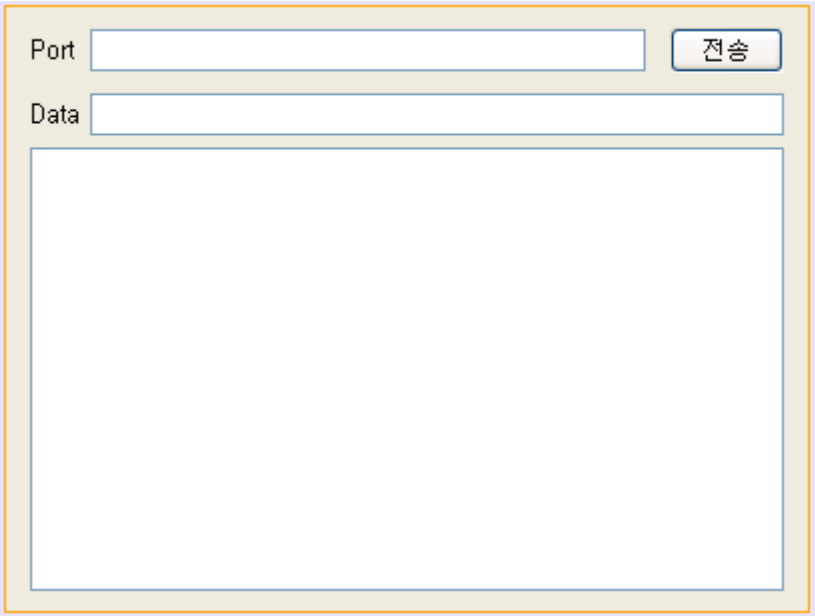
❖ ServerFrame.java 생성



실습 3 : TCP Socket Server 프로그래밍 (5)

컨트롤 배치 및 속성 지정

❖ JFrame Form



12. 컨트롤 배치 및 속성 지정

컨트롤	속성 지정
jLabel1	• Variable Name : lblPort • Text : Port
jTextField1	• Variable Name : txtPort • Text :
jLabel2	• Variable Name : lblData • Text : Data
jTextField2	• Variable Name : txtData • Text :
jButton1	• Variable Name : btnSender • Text : 전송
jTextArea1	• Variable Name : jTextArea1 • Text :

실습 3 : TCP Socket Server 프로그래밍 (6)

소스 분석

ServerFrame.java

ServerFrame.java * – Editor

ServerFrame.java *

Source | Design

13. Coding

```
1   /*...*/
5
6   import java.net.*;
7   import java.io.*;
8
9   /**...*/
13  public class ServerFrame extends javax.swing.JFrame {
14
15      /** Creates new form ServerFrame */
16      public ServerFrame() {
17          initComponents();
18          this.setTitle("TCP Socket Server 프로그램");  // Server Frame 제목 지정
19      }
20
21      /**...*/
26      @SuppressWarnings("unchecked")
27      Generated Code
```

101:1 INS

실습 3 : TCP Socket Server 프로그래밍 (7)

- [전송] Event Handler
  - Event
    - ActionPerformed

ServerFrame.java * - Editor

ServerFrame.java *

Source | Design

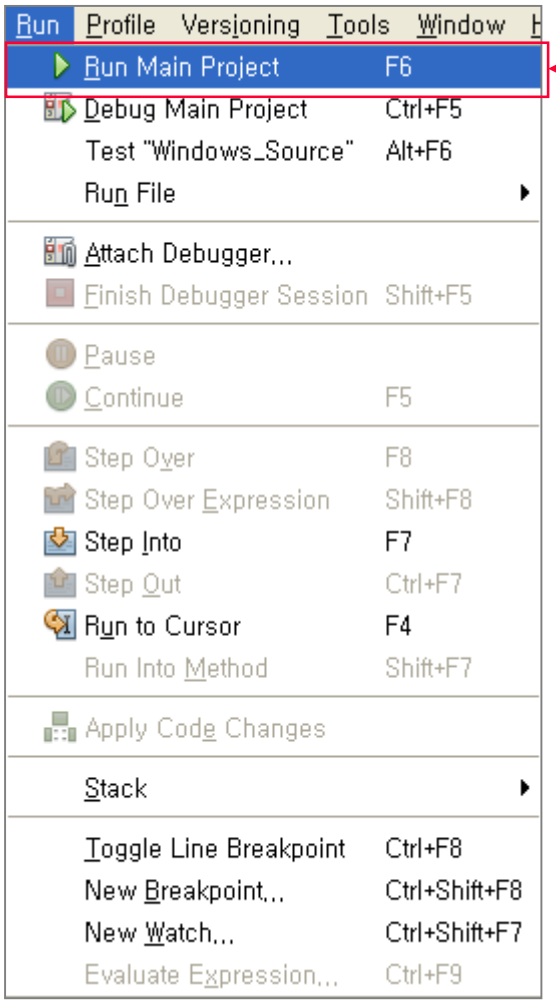
```
96  private void btnSenderActionPerformed(java.awt.event.ActionEvent evt) {
97      try{
98          String strPort = txtPort.getText();            // 입력 Port 번호 반환
99          int iPortNum = Integer.parseInt(strPort);      // Port 번호 생성
100         String strData = txtData.getText();            // 입력 Data 반환
101
102         ServerSocket Server_SKT = new ServerSocket(iPortNum); // Server Socket 생성
103         Socket Client_SKT = Server_SKT.accept();       // Client 요청으로 Socket 객체 생성
104         // 데이터 출력을 위한 Stream  객체 생성
105         OutputStream OutStream = Client_SKT.getOutputStream();
106         DataOutputStream DOutStream = new DataOutputStream(OutStream);
107         DOutStream.writeChars(strData);
108         jTextArea1.append(strData + "₩n 전송... ₩n");
109         Client_SKT.close();
110     } catch (Exception ee) {
111         jTextArea1.setText("");
112         jTextArea1.append(" Host 접속 불가... ");
113         System.err.println(" Exception : " + ee);
114     }
115 }
```

20:1 INS

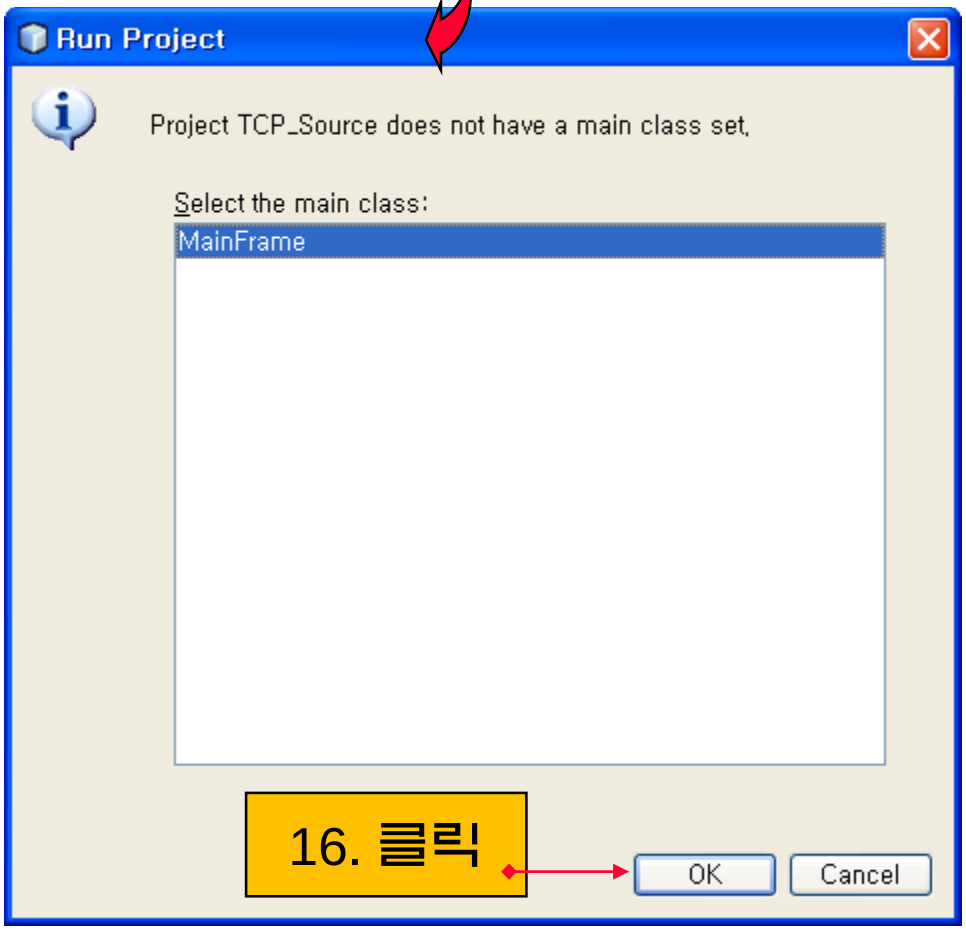
14. Coding

실습 3 : TCP Socket Server 프로그래밍 (8)

실행

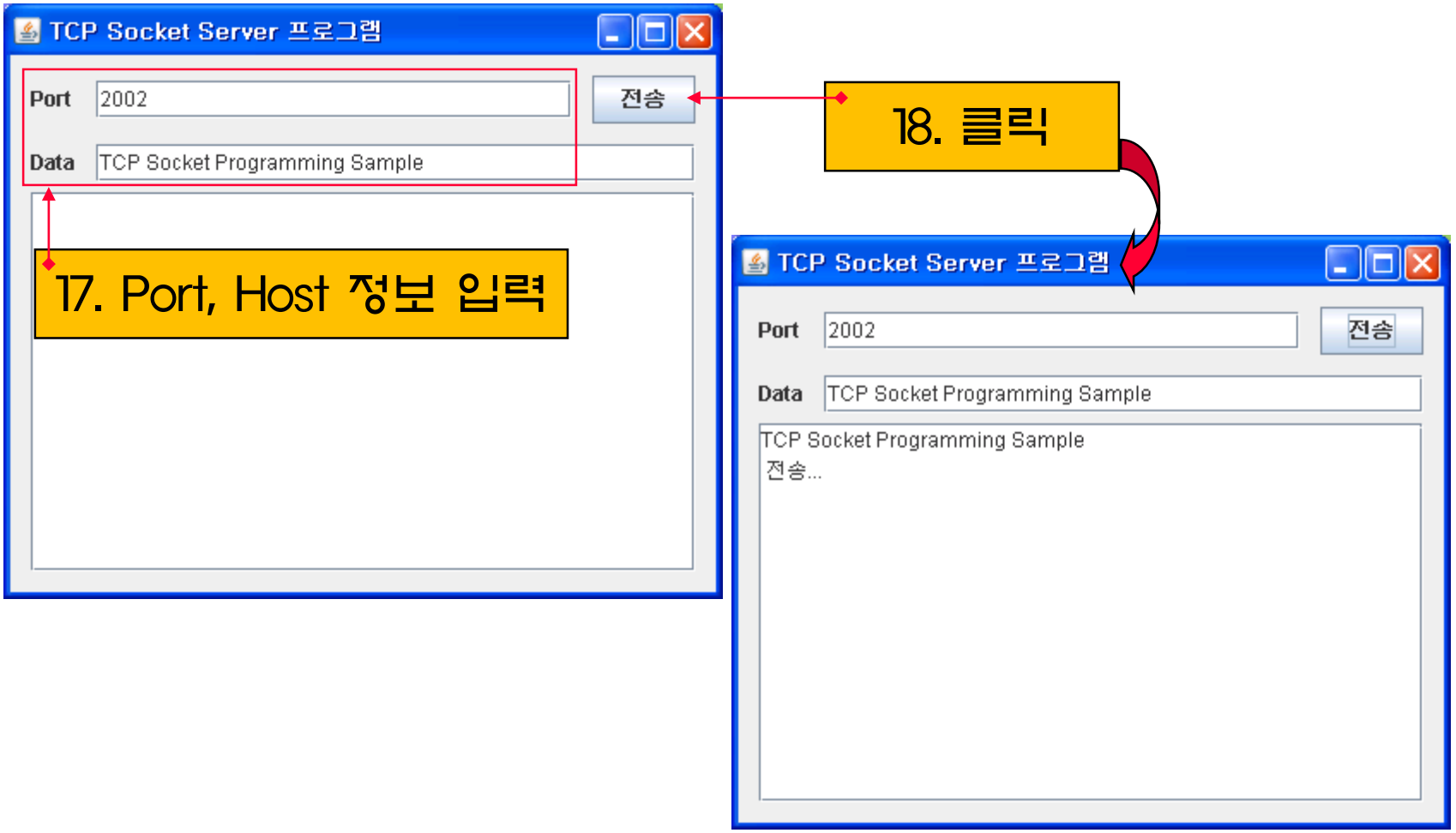


15. 클릭



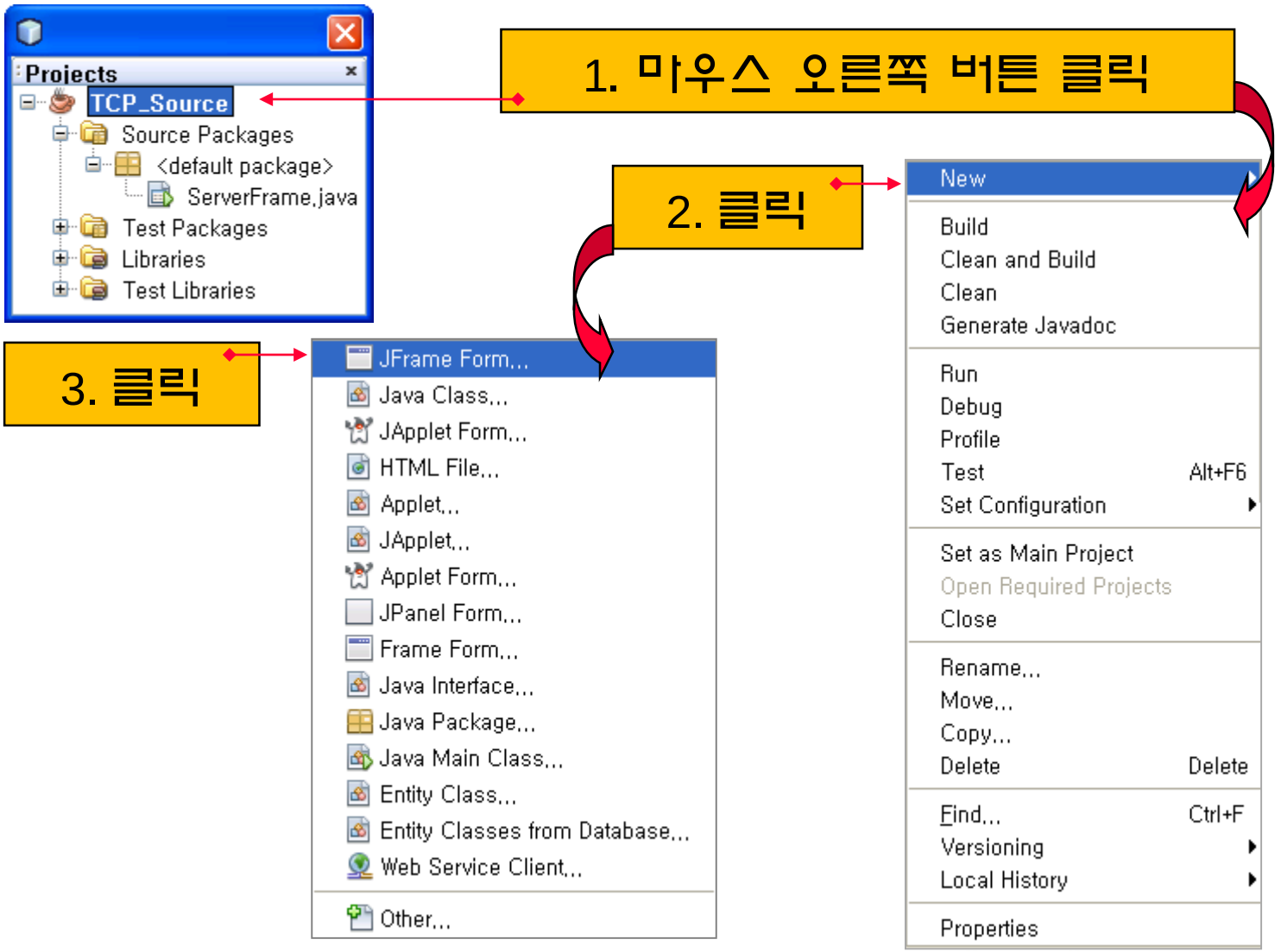
실습 3 : TCP Socket Server 프로그래밍 (9)

실행 결과



실습 3 : TCP Socket Client 프로그래밍 (1)

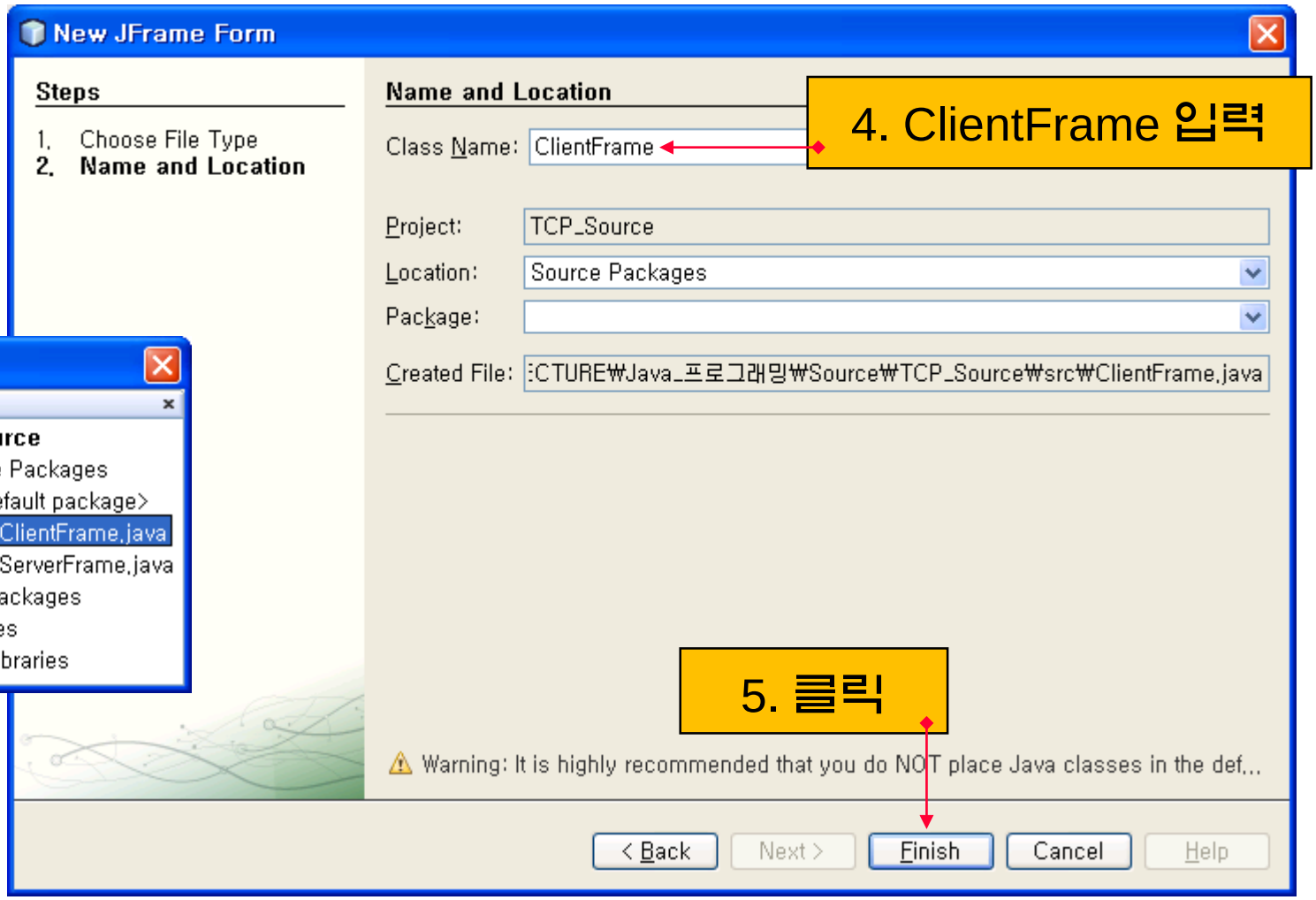
JFrame Form 생성



실습 3 : TCP Socket Client 프로그래밍 (2)

JFrame Form Name 지정

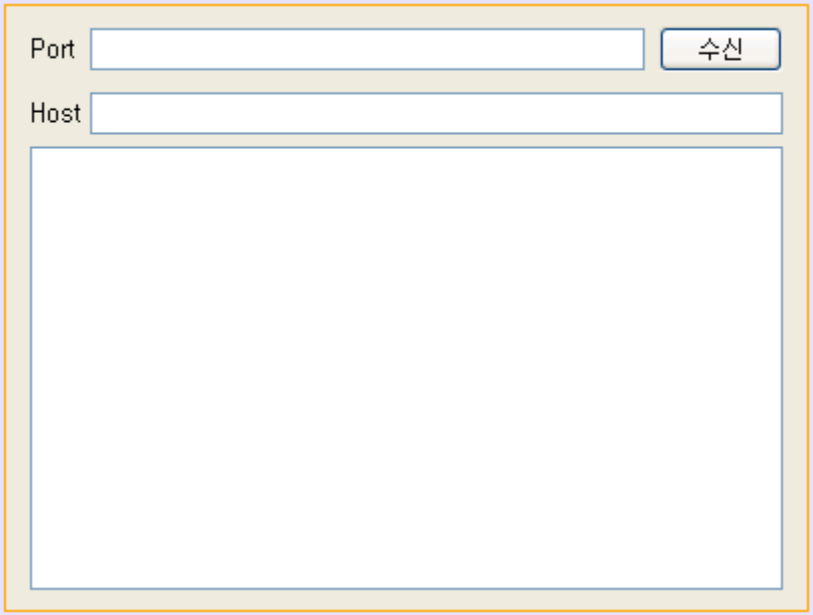
ClientFrame.java 생성



실습 3 : TCP Socket Client 프로그래밍 (3)

컨트롤 배치 및 속성 지정

❖ JFrame Form



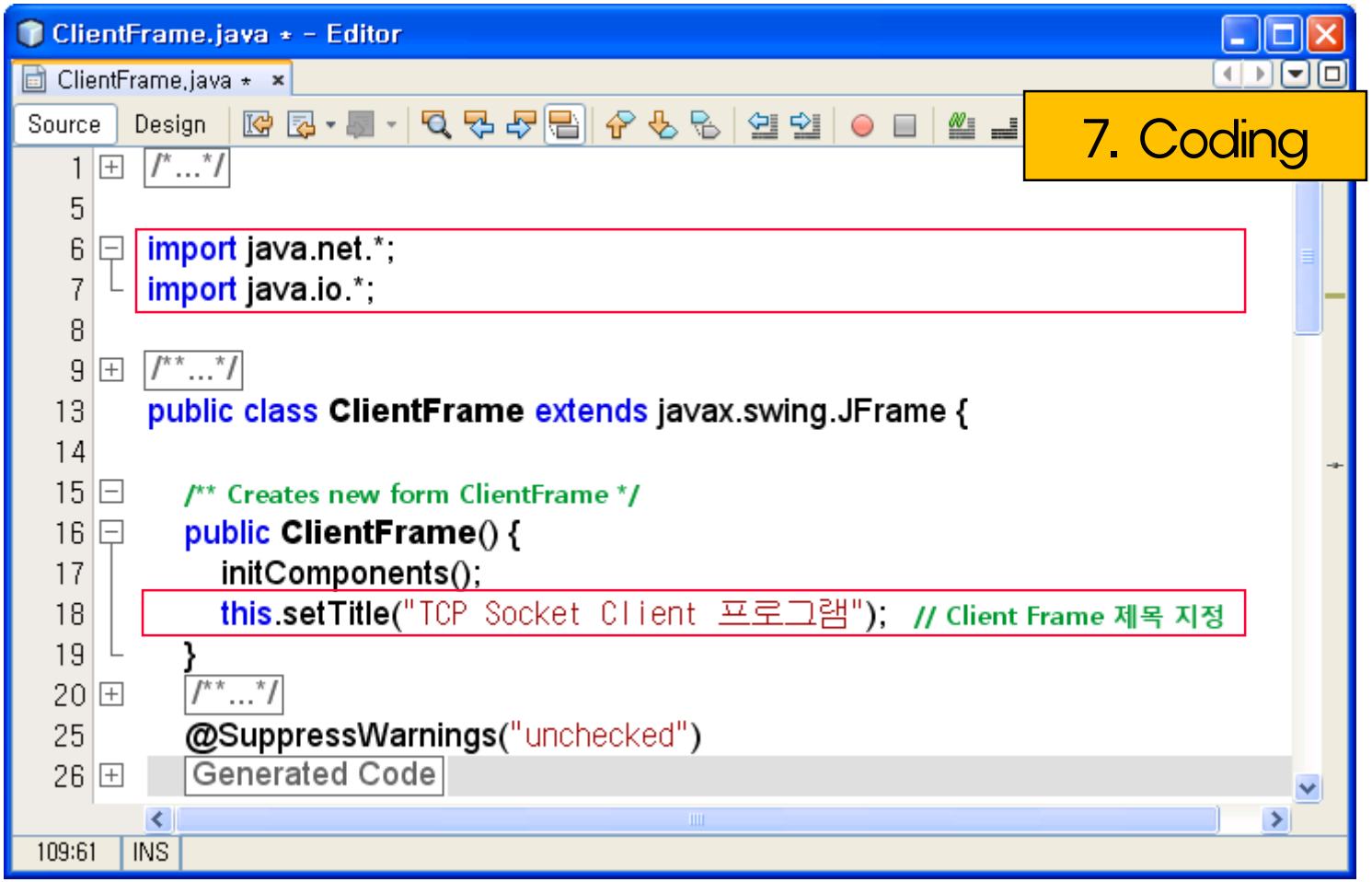
6. 컨트롤 배치 및 속성 지정

컨트롤	속성 지정
jLabel1	• Variable Name : lblClientPort • Text : Port
jTextField1	• Variable Name : txtClientPort • Text :
jLabel2	• Variable Name : lblHost • Text : Host
jTextField2	• Variable Name : txtHost • Text :
jButton1	• Variable Name : btnReceiver • Text : 수신
jTextArea1	• Variable Name : jTextArea1 • Text :

실습 3 : TCP Socket Client 프로그래밍 (4)

소스 분석

ClientFrame.java



실습 3 : TCP Socket Client 프로그래밍 (5)

- [수신] Event Handler
  - Event
    - ActionPerformed

ClientFrame.java * - Editor

ClientFrame.java *

Source | Design

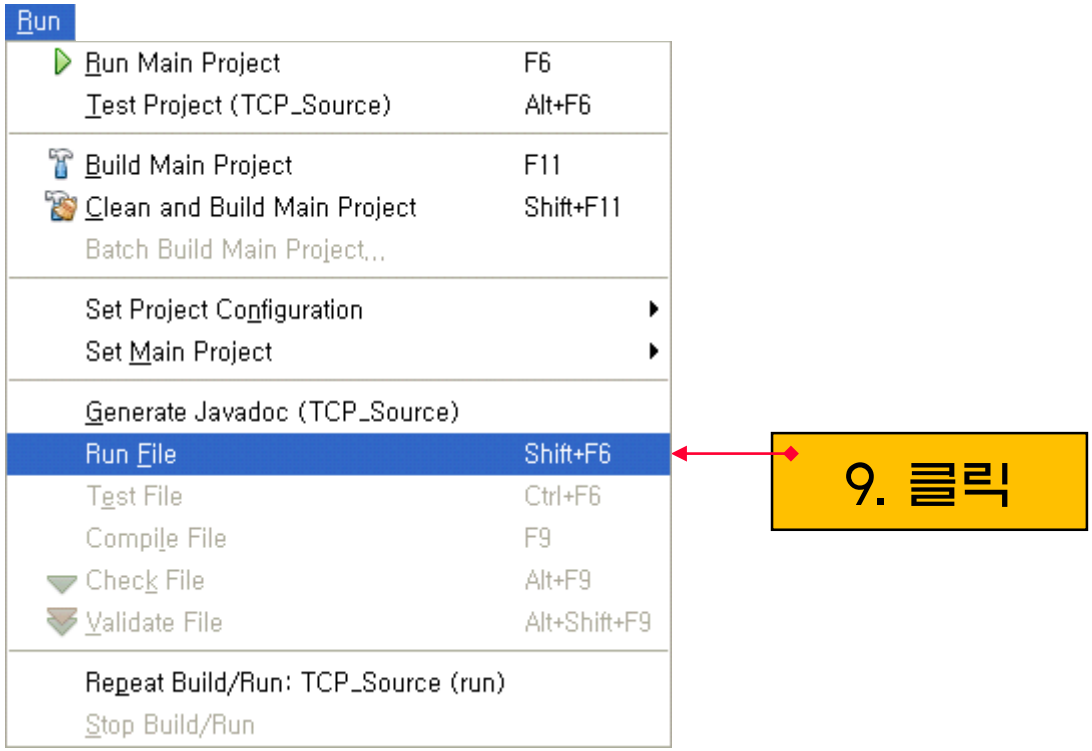
```
97  private void btnReceiverActionPerformed(java.awt.event.ActionEvent evt) {
98      try{
99          String strClientPort = txtClientPort.getText();          // Port 번호 반환
100         int iPortNum = Integer.parseInt(strClientPort);          // Port 번호 생성
101         String strHost = txtHost.getText();                      // Host IP 반환
102
103         // 입력 port, Host로 Server Socket 생성
104         Socket Client_SKT = new Socket(strHost, iPortNum);
105         // 데이터 입력을 위한 Stream  객체 생성
106         InputStream InStream = Client_SKT.getInputStream();
107         DataInputStream DInStream = new DataInputStream(InStream);
108         String strReData = DInStream.readLine();          //Data 수신
109         jTextArea1.append(strReData + "₩n 수신 완료... ₩n");
110         Client_SKT.close();
111     } catch (Exception ee) {
112         jTextArea1.setText("");
113         jTextArea1.append(" Data 수신 불가... ");
114         System.err.println(" Exception : " + ee);
115     }
116 }
```

17:26 | INS

8. Coding

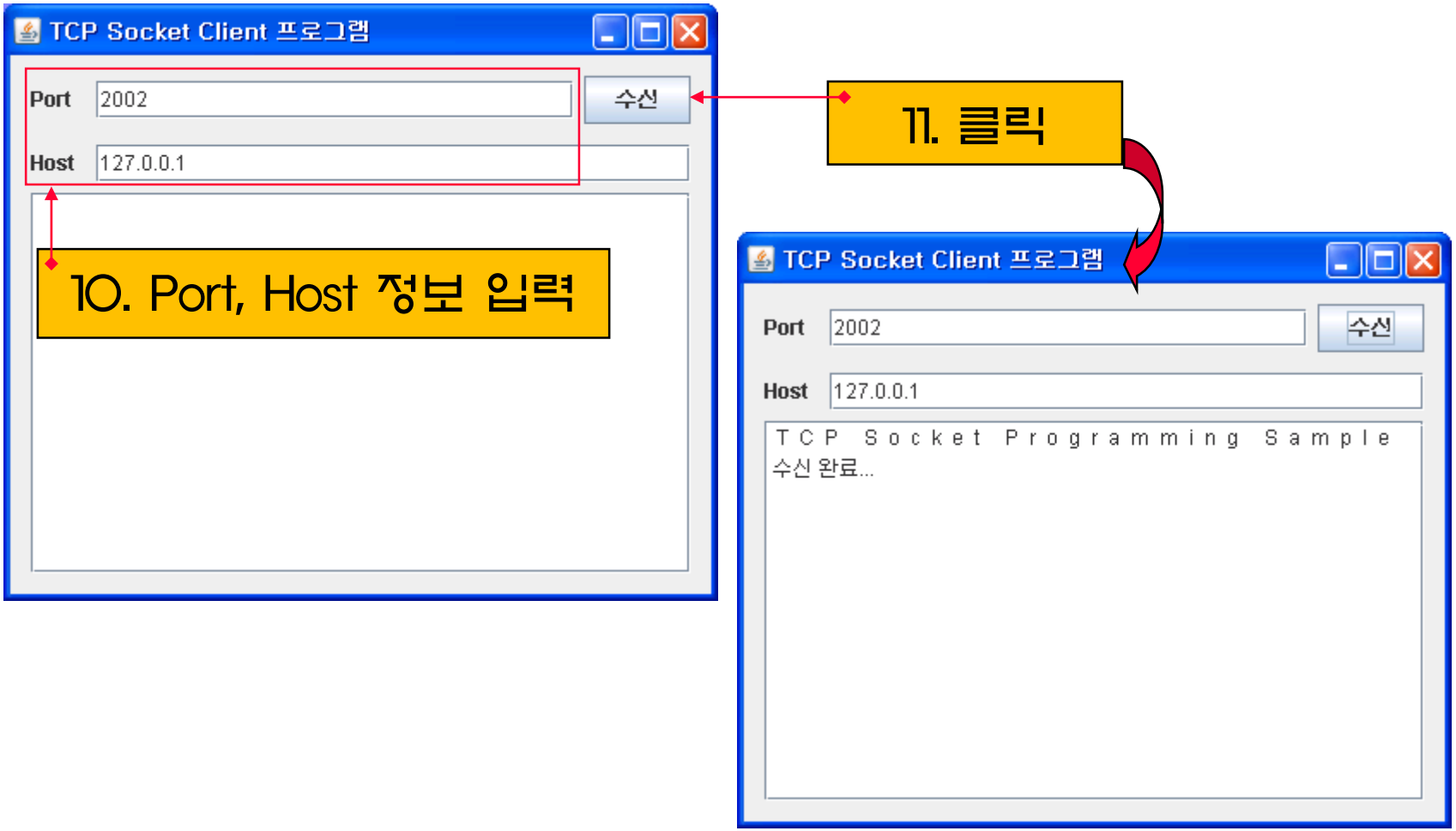
실습 3 : TCP Socket Client 프로그래밍 (6)

실행



실습 3 : TCP Socket Client 프로그래밍 (7)

실행 결과



UDP Socket 프로그래밍 (1)

UDP(User Datagram Protocol)

❖ 특징

- ◆ TCP Socket에 비해 안정성과 신뢰성이 낮지만, 접속 설정을 하지 않아도 되므로 네트워크에 부담을 주지 않는다.

❖ Datagram Socket 클래스 특징

- ◆ Datagram Packet 송신 및 수신
- ◆ 동시에 다른 주소와 Port로 데이터 전송 가능
  - 주소 및 port에 대한 정보가 socket에 포함
- ◆ UDP Port 번호는 TCP Port 번호와 독립적으로 사용된다.

UDP Socket 프로그래밍 (2)

DatagramSocket 클래스

❖ 생성자

생성자
DatagramSocket() throws SocketException
DatagramSocket(int port) throws SocketException
DatagramSocket(int port, InetAddress laddr) throws SocketException

❖ DatagramSocket 클래스의 Method

Method	설 명
void receive(DatagramPacket dgram) throws IOException	현재의 Socket에서 data를 읽어 dgram packet에 저장
void send(DatagramPacket dgram) throws IOException	현재의 Socket을 통해 dgram packet 전송
void close() throws IOException	Socket 접속 해제

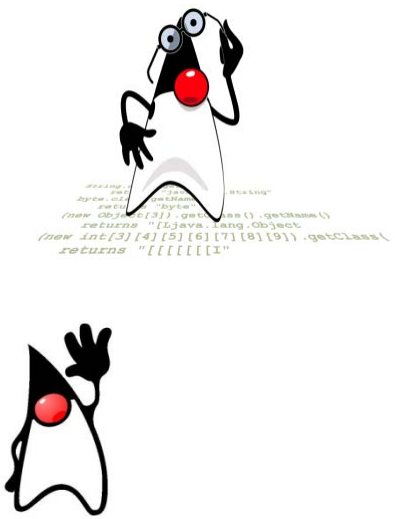
UDP Socket 프로그래밍 (3)

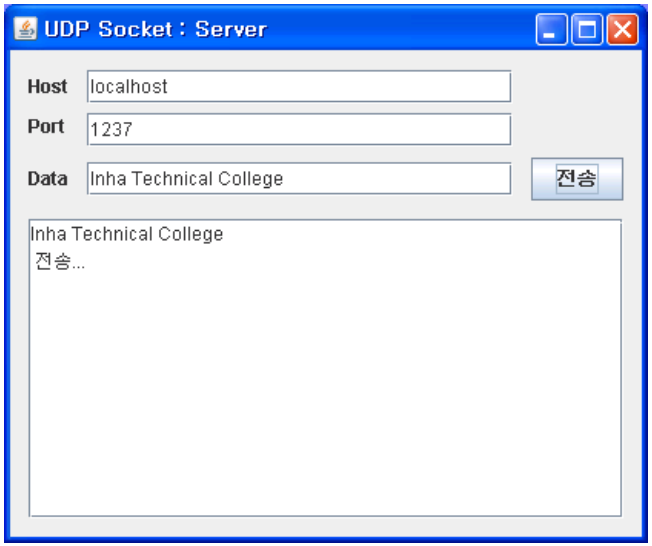
DatagramPacket 클래스

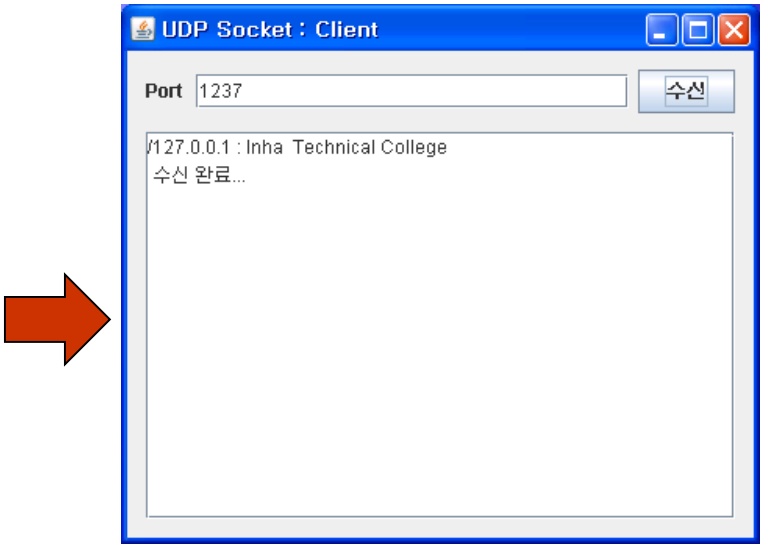
생성자	
Receiver	DatagramPacket(byte[ ] buffer, int size) • buffer: 수신된 데이터 저장 • size: buffer 크기
Sender	DatagramPacket(byte[ ] buffer, int size, InetAddress addr, int port) • buffer: 송신된 데이터 저장 • size: buffer 크기 • addr: data를 송신할 컴퓨터 주소 • port: 포트 번호

실습 4 : UDP Socket 프로그래밍

- Project Name: UDP_Source(실습^시간 : 30분)
  - Server 프로그램
    - Port, Data를 입력한 후 [전송] 버튼 클릭
  - Client 프로그램
    - Host, Port를 입력한 후 [수신] 버튼을 클릭하면 Server에서 전송한 data를 jTextArea1 컴포넌트에 출력

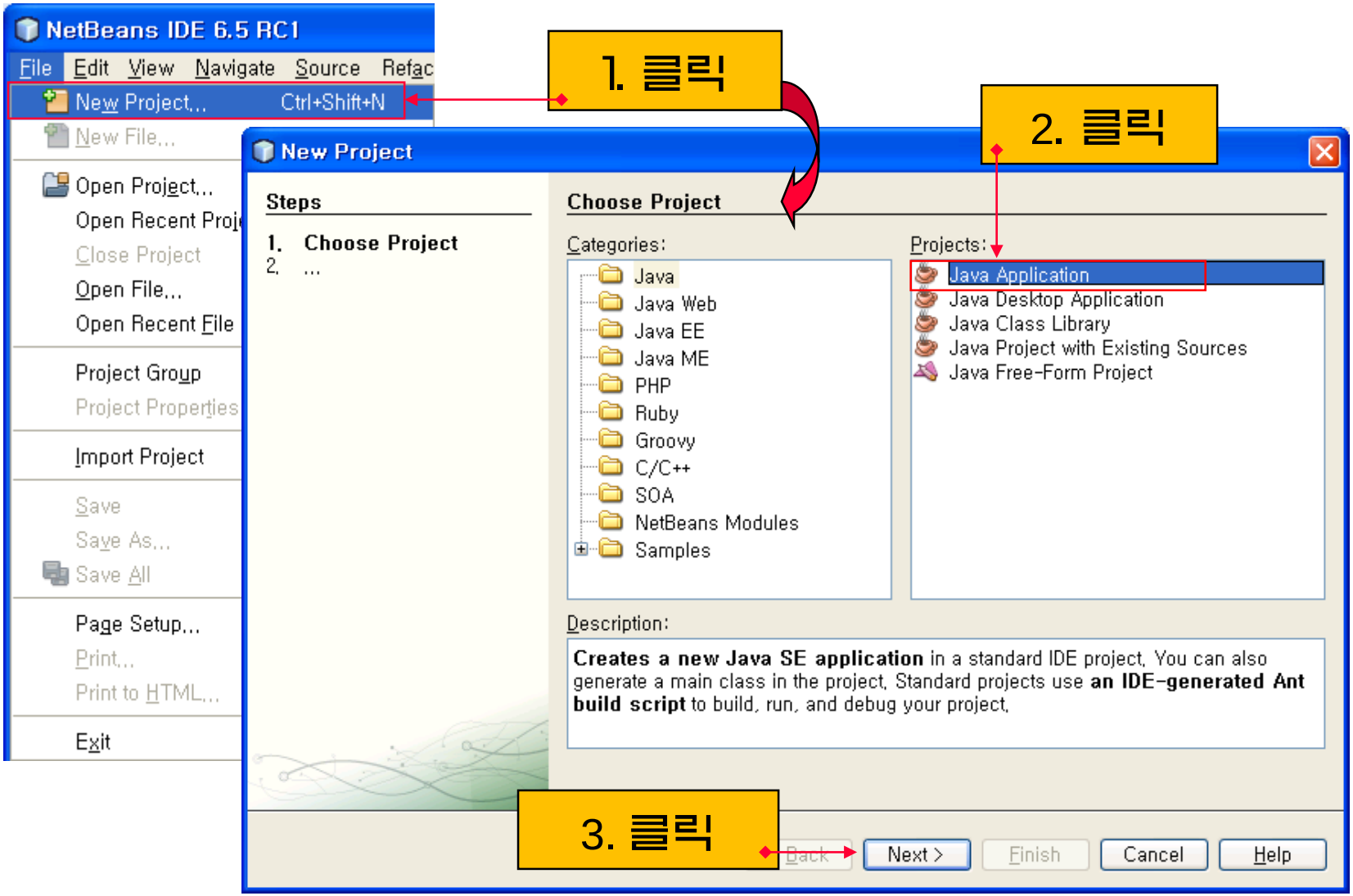






실습 4 : UDP Socket Server 프로그래밍 (1)

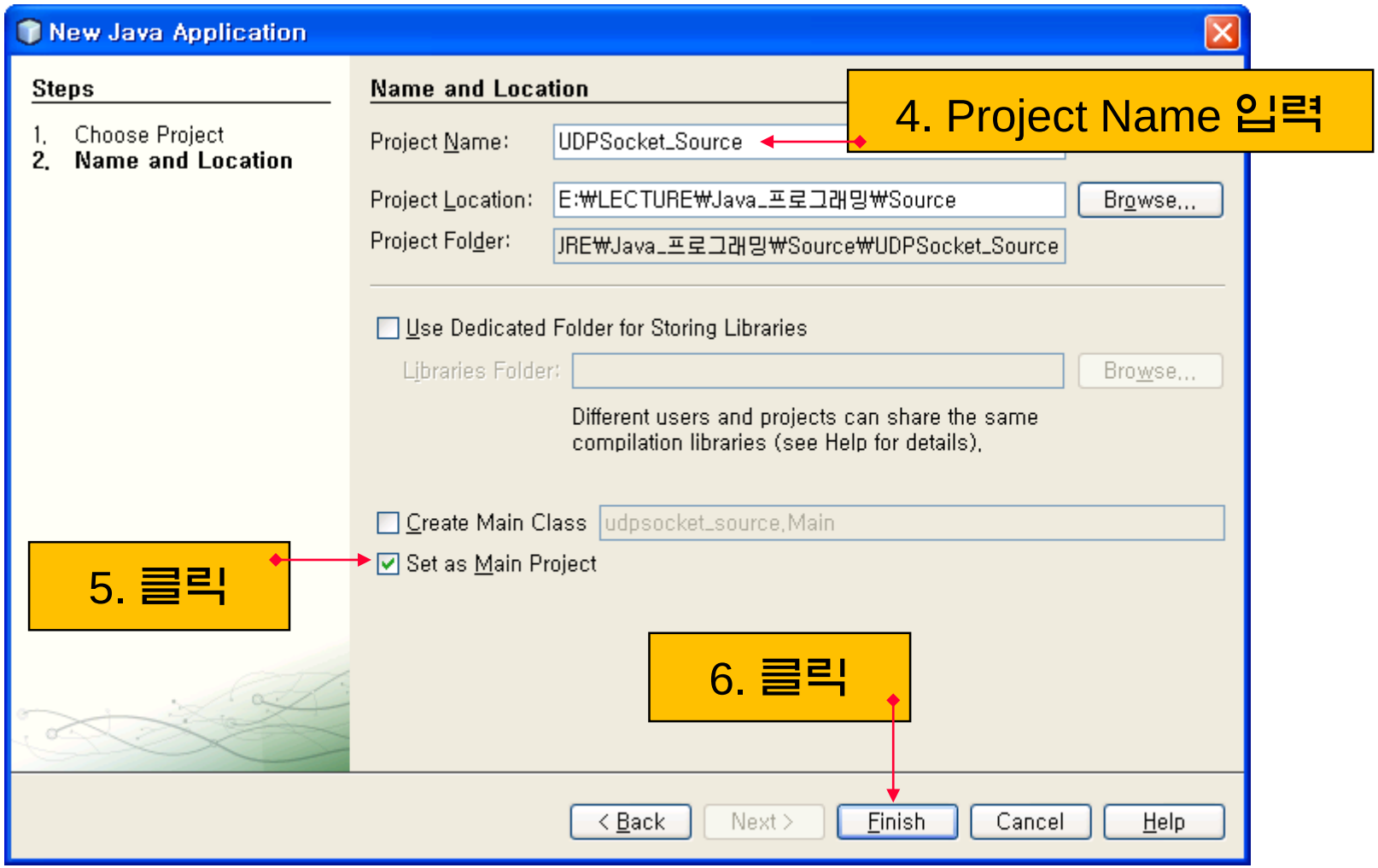
Application 생성



실습 4 : UDP Socket Server 프로그래밍 (2)

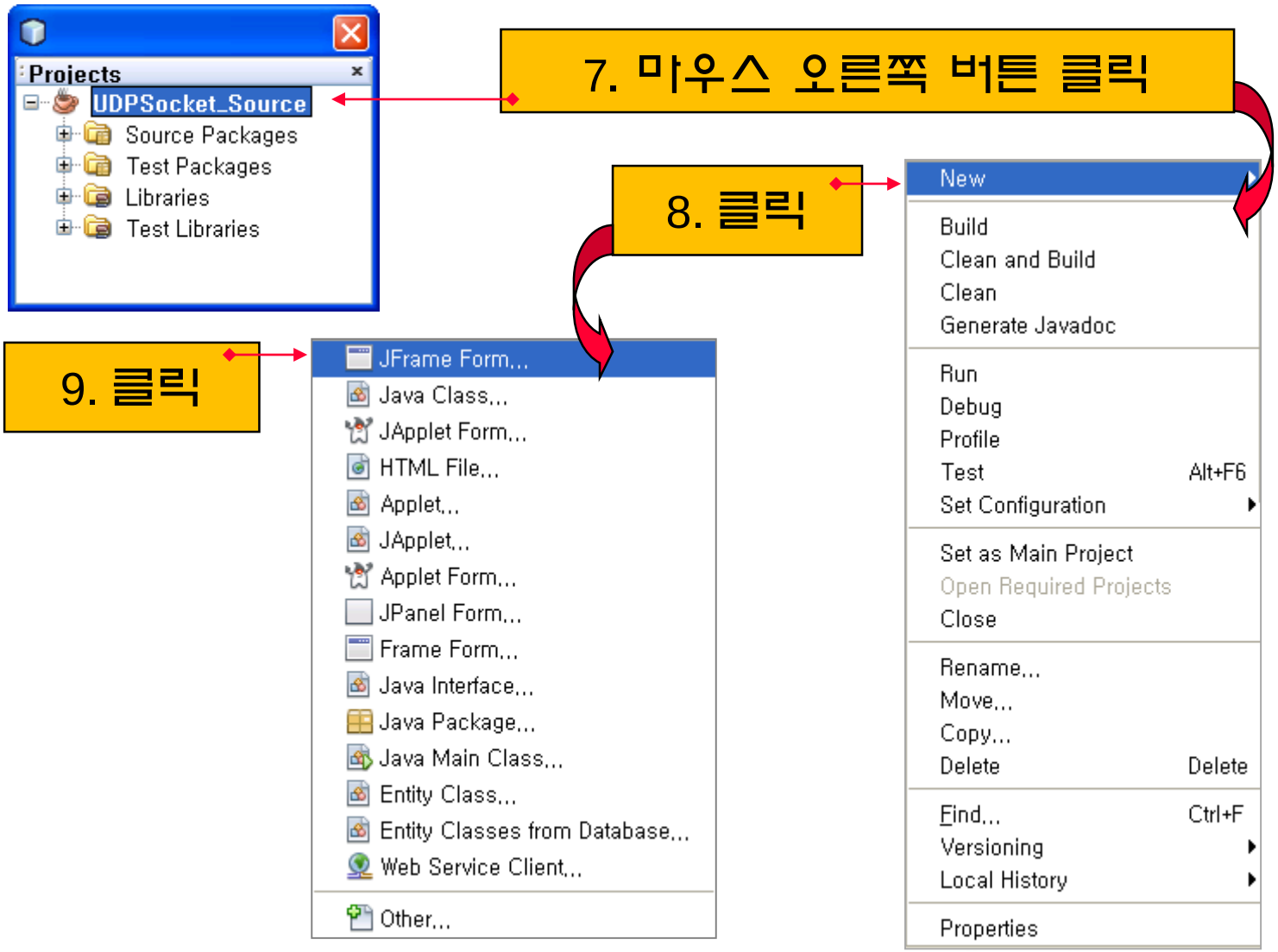
Project Name and Location

- Project name: UDPSocket_Source



실습 4 : UDP Socket Server 프로그래밍 (3)

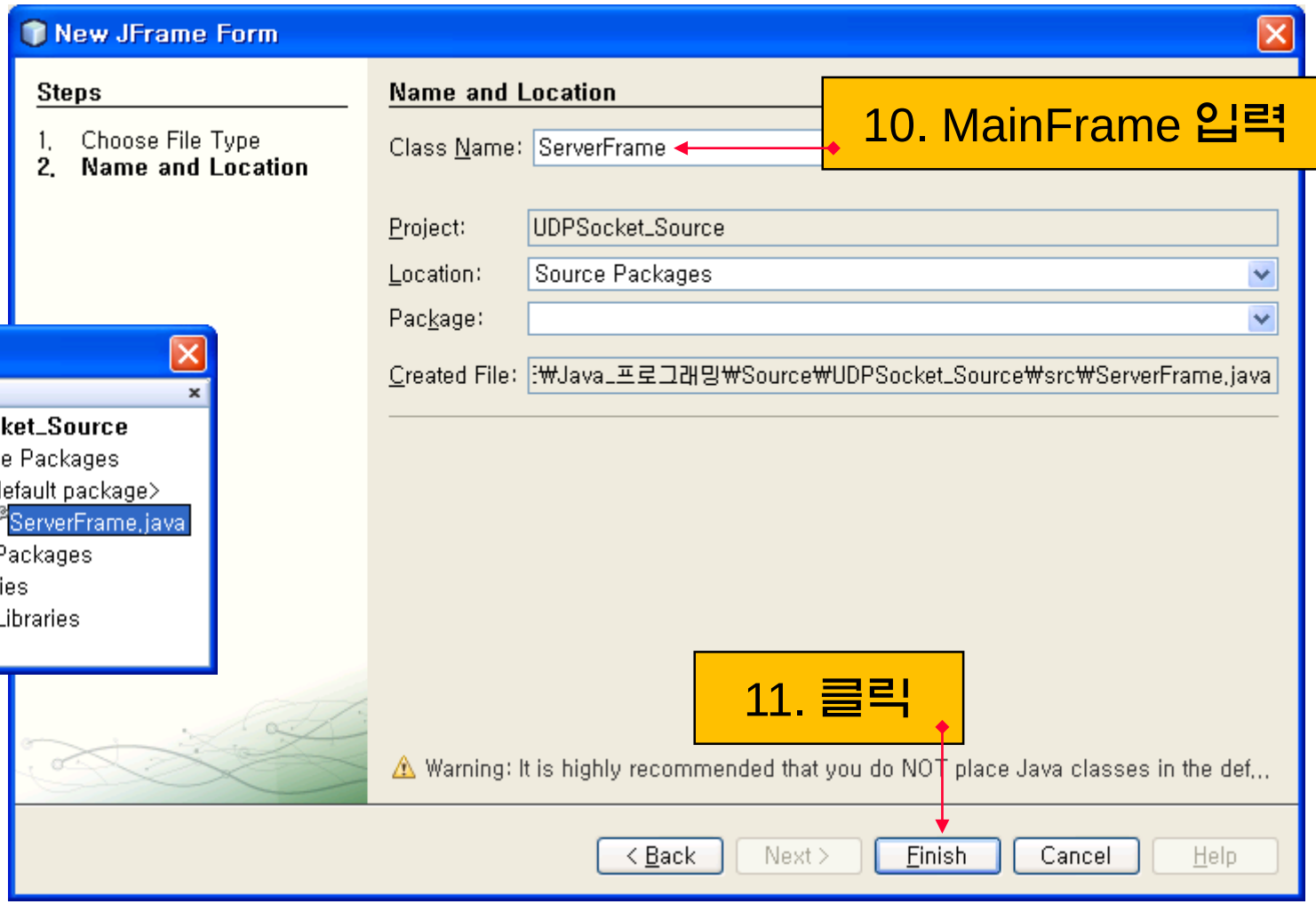
JFrame Form 생성



실습 4 : UDP Socket Server 프로그래밍 (4)

JFrame Form Name 지정

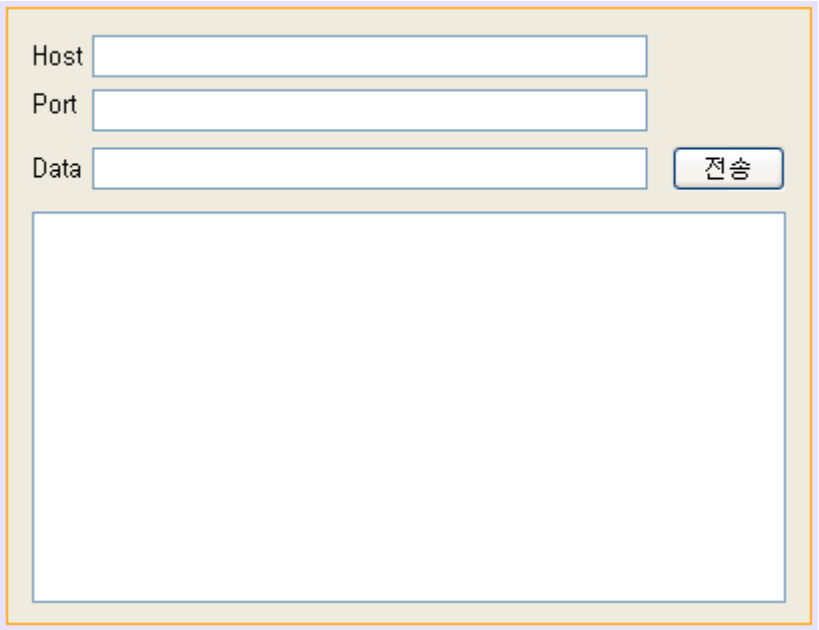
❖ ServerFrame.java 생성



실습 4 : UDP Socket Server 프로그래밍 (5)

컨트롤 배치 및 속성 지정

❖ JFrame Form



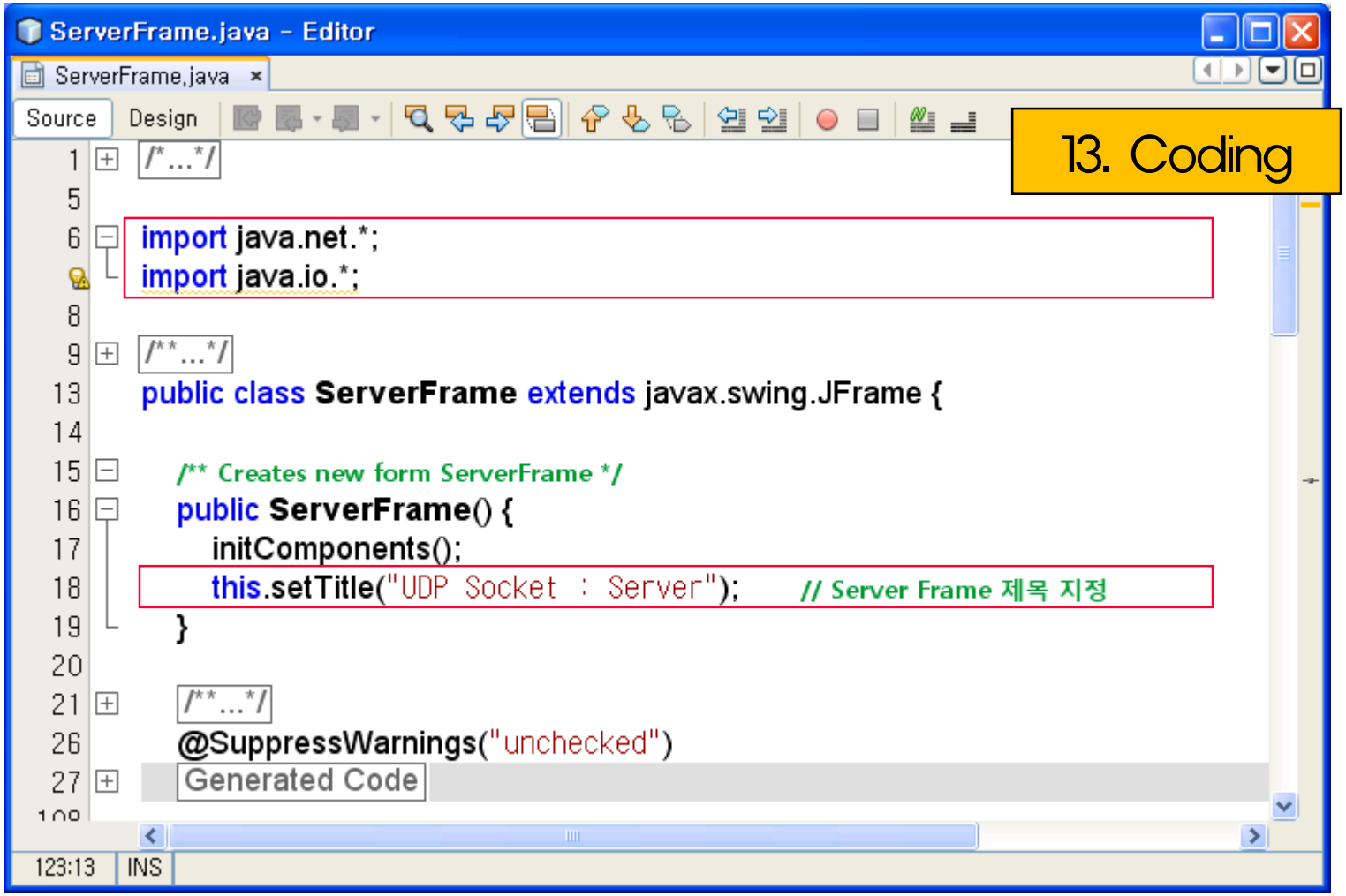
12. 컨트롤 배치 및 속성 지정

컨트롤	속성 지정
jLabel1	• Variable Name : lblHost • Text : Host
jTextField1	• Variable Name : txtHost • Text :
jLabel2	• Variable Name : lblPort • Text : Port
jTextField2	• Variable Name : txtPort • Text :
jLabel3	• Variable Name : lblData • Text : Data
jTextField3	• Variable Name : txtData • Text :
jButton1	• Variable Name : btnSender • Text : 전송
jTextArea1	• Variable Name : jTextArea1 • Text :

실습 4 : UDP Socket Server 프로그래밍 (6)

소스 분석

ServerFrame.java



실습 4 : UDP Socket Server 프로그래밍 (7)

- [전송] Event Handler
  - Event
    - ActionPerformed

ServerFrame.java - Editor

ServerFrame.java

Source | Design

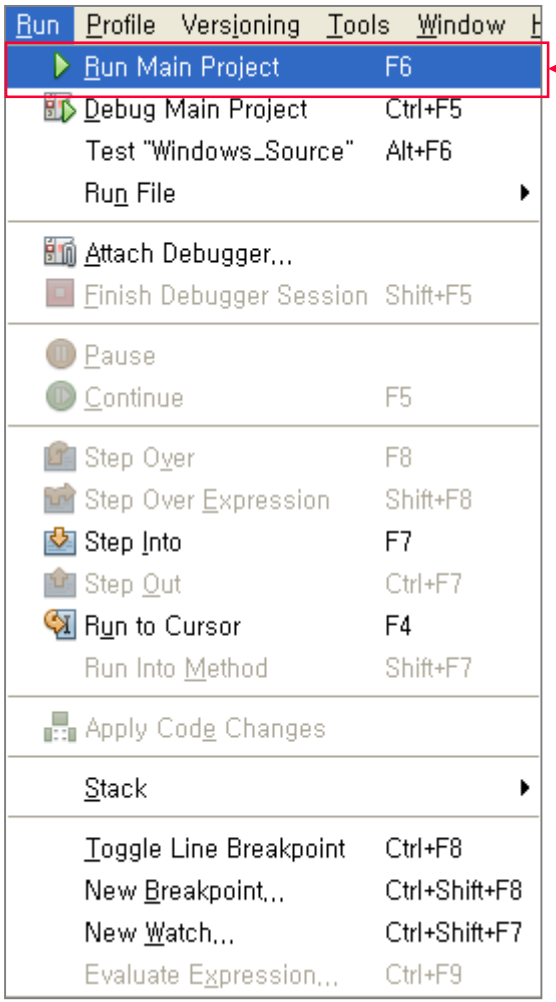
```
109  private void btnSenderActionPerformed(java.awt.event.ActionEvent evt) {
110      try{
111          DatagramSocket DGSocket = new DatagramSocket();
112          InetAddress inetAddr = InetAddress.getByName(txtHost.getText()); // Host 주소 지정
113
114          String strPort = txtPort.getText();          // 입력 Port 번호 반환
115          int iPortNum = Integer.parseInt(strPort);    // Port 번호 생성
116          String strData = txtData.getText();          // 입력 Data 반환
117          byte blnData[ ] = strData.getBytes();        // 입력 Data 변환
118
119          //Datagram Packet 생성
120          DatagramPacket DGPacket = new DatagramPacket(blnData, blnData.length, inetAddr, iPortNum);
121          DGSocket.send(DGPacket);                     // Packet 전송
122          jTextArea1.append(strData + "₩n 전송... ₩n");
123
124      } catch (Exception ee) {
125          jTextArea1.setText("");
126          jTextArea1.append(" Data 전송 실패... ");
127          System.err.println(" Exception : " + ee);
128      }
129  }
```

145:46 | INS

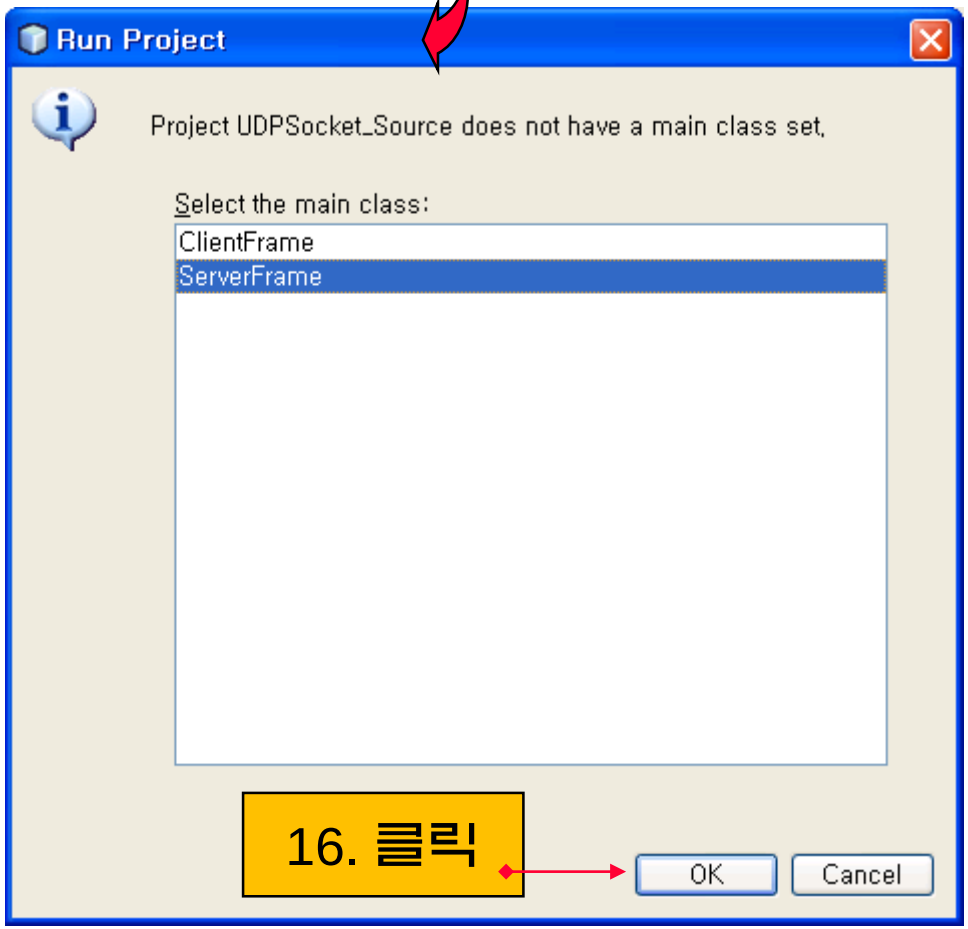
14. Coding

실습 4 : UDP Socket Server 프로그래밍 (8)

실행

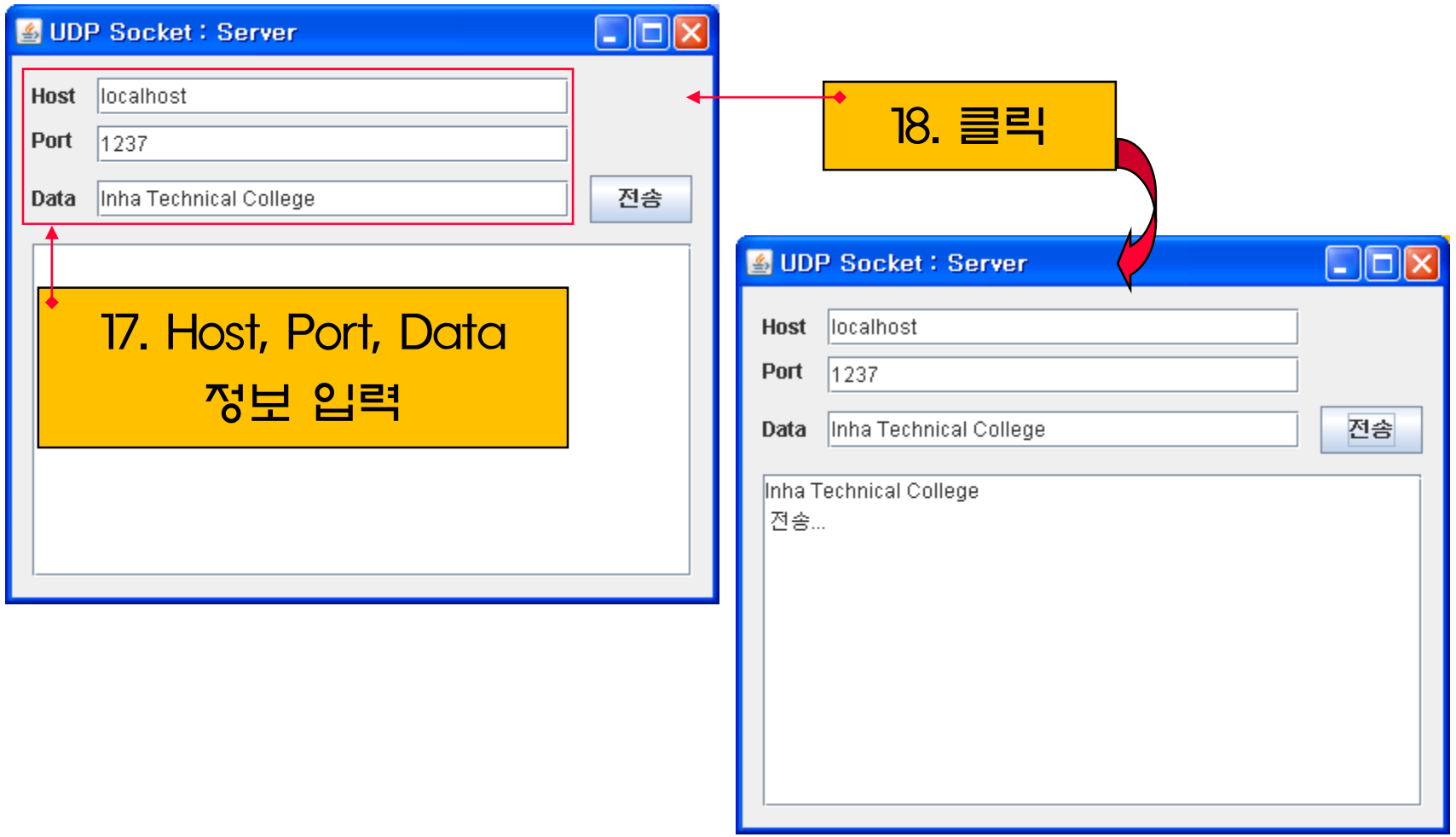


15. 클릭



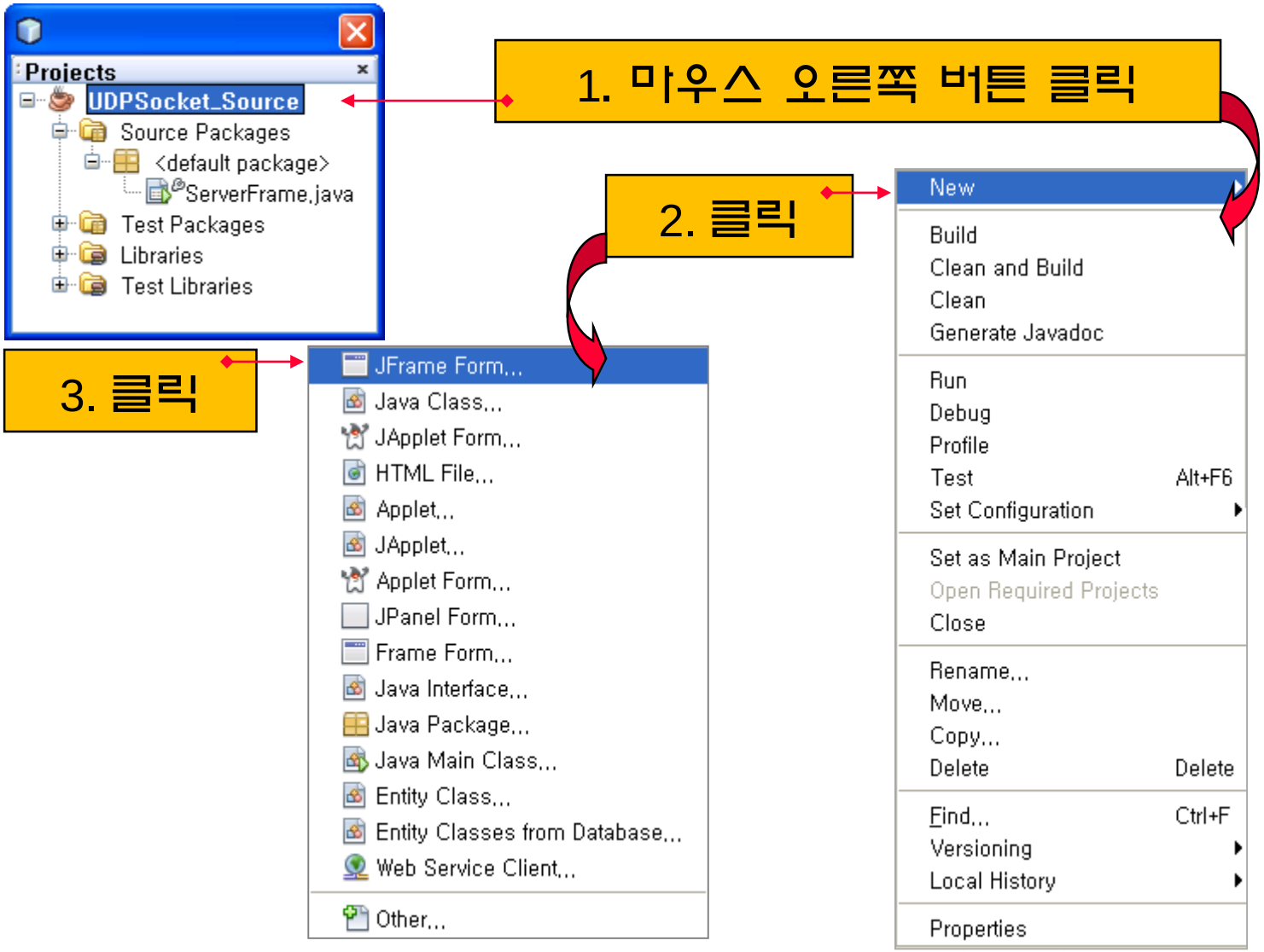
실습 4 : UDP Socket Server 프로그래밍 (9)

실행 결과



실습 4 : UDP Socket Client 프로그래밍 (1)

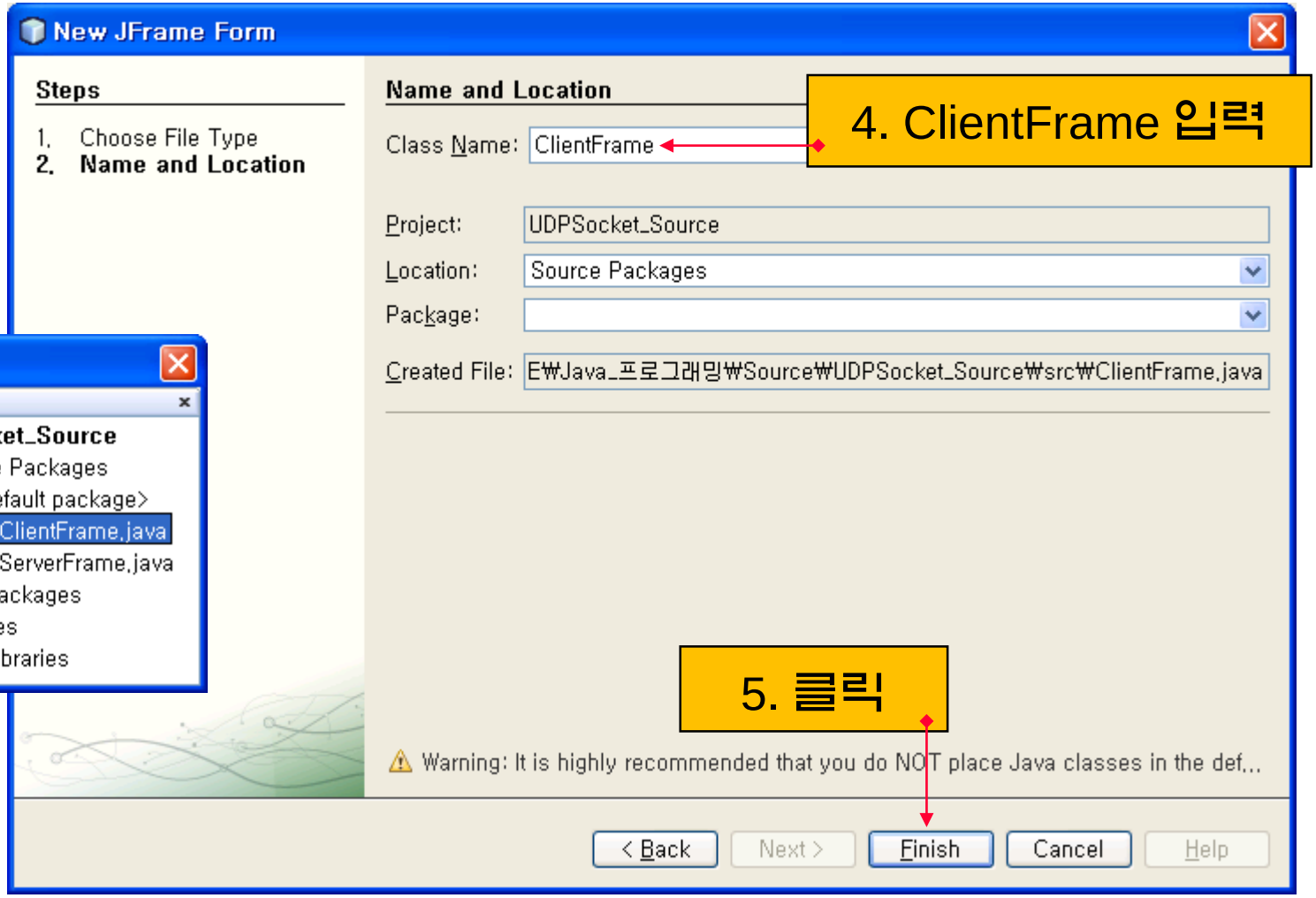
JFrame Form 생성



실습 4 : UDP Socket Client 프로그래밍 (2)

JFrame Form Name 지정

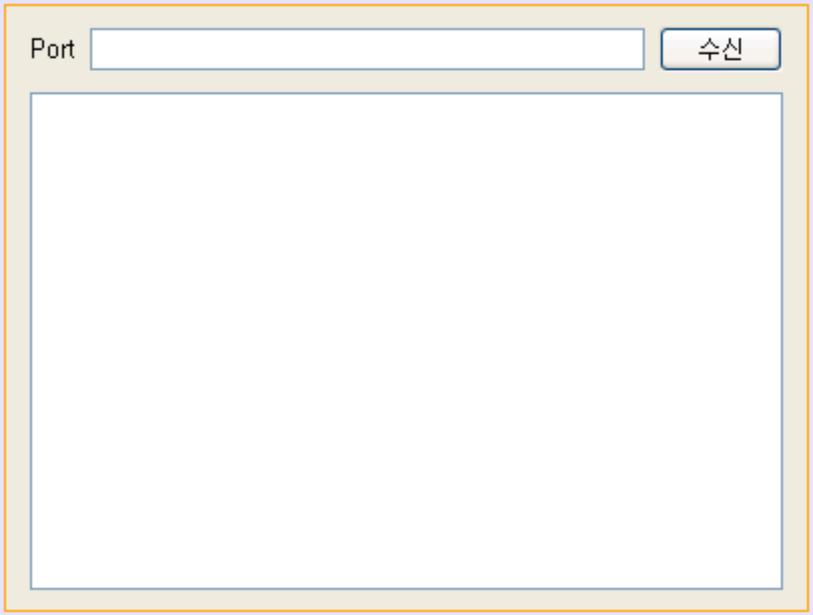
❖ ClientFrame.java 생성



실습 4 : UDP Socket Client 프로그래밍 (3)

컨트롤 배치 및 속성 지정

❖ JFrame Form



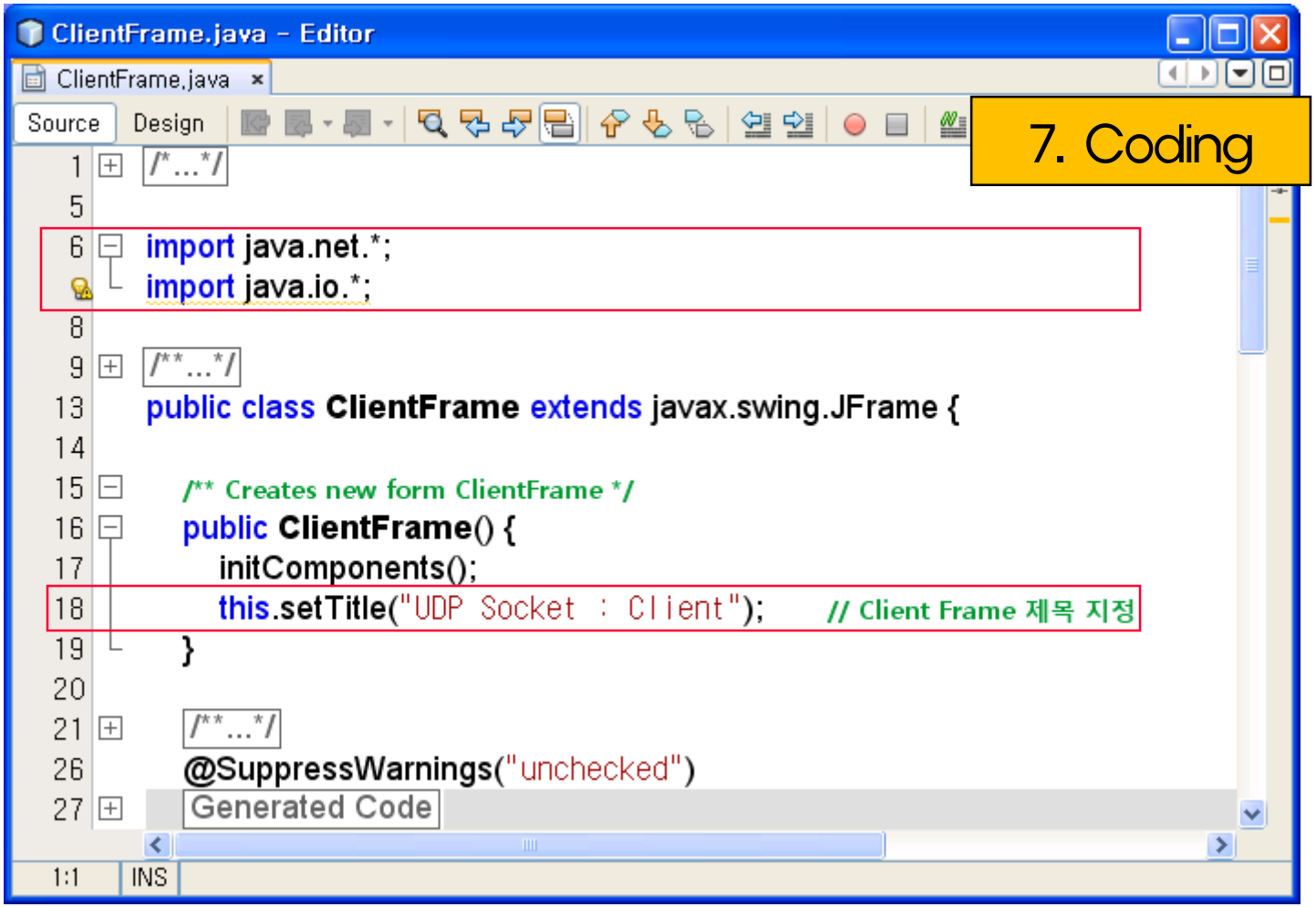
컨트롤	속성 지정
jLabel1	• Variable Name : lblClientPort • Text : Port
jTextField1	• Variable Name : txtClientPort • Text :
jButton1	• Variable Name : btnReceiver • Text : 수신
jTextArea1	• Variable Name : jTextArea1 • Text :

6. 컨트롤 배치 및 속성 지정

실습 4 : UDP Socket Client 프로그래밍 (4)

소스 분석

ClientFrame.java



실습 4 : UDP Socket Client 프로그래밍 (5)

- [수신] Event Handler
  - Event
    - ActionPerformed

ClientFrame.java - Editor

ClientFrame.java

Source | Design

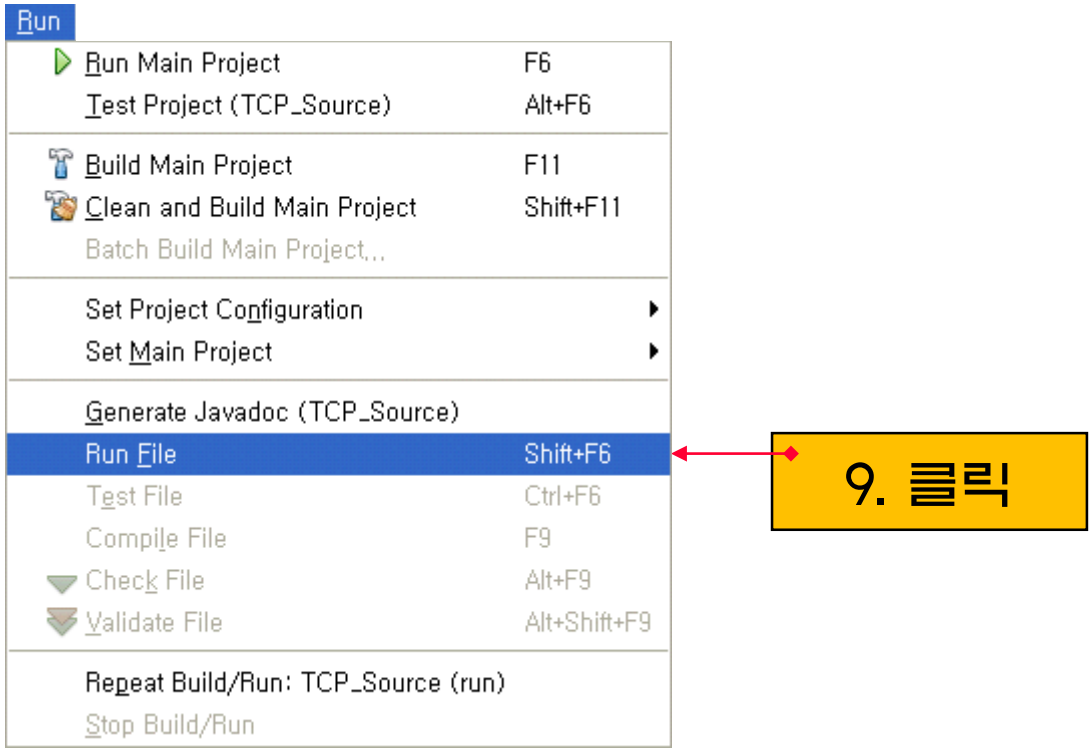
```
84  private void btnReceiverActionPerformed(java.awt.event.ActionEvent evt) {
85      try{
86          String strClientPort = txtClientPort.getText();                    // Port 번호 반환
87          int iPortNum = Integer.parseInt(strClientPort);                    // Port 번호 생성
88
89          byte    bufData[ ] = new byte[30];                                 // 수신 data 저장
90          DatagramSocket DGSocket = new DatagramSocket(iPortNum);            // Socket 생성
91          //Datagram Packet 생성
92          DatagramPacket DGPacket = new DatagramPacket(bufData, bufData.length);
93          DGSocket.receive(DGPacket);                                        // Packet 수신
94          String strOutData = new String(DGPacket.getData(), 0, DGPacket.getLength());  // Data 수신
95          jTextArea1.append(DGPacket.getAddress() + ":" + strOutData + "₩n 수신 완료... ₩n");
96
97      } catch (Exception ee) {
98          jTextArea1.setText("");
99          jTextArea1.append(" Data 수신 불가... ");
100         System.err.println(" Exception : " + ee);
101     }
102 }
```

1:1 INS

8. Coding

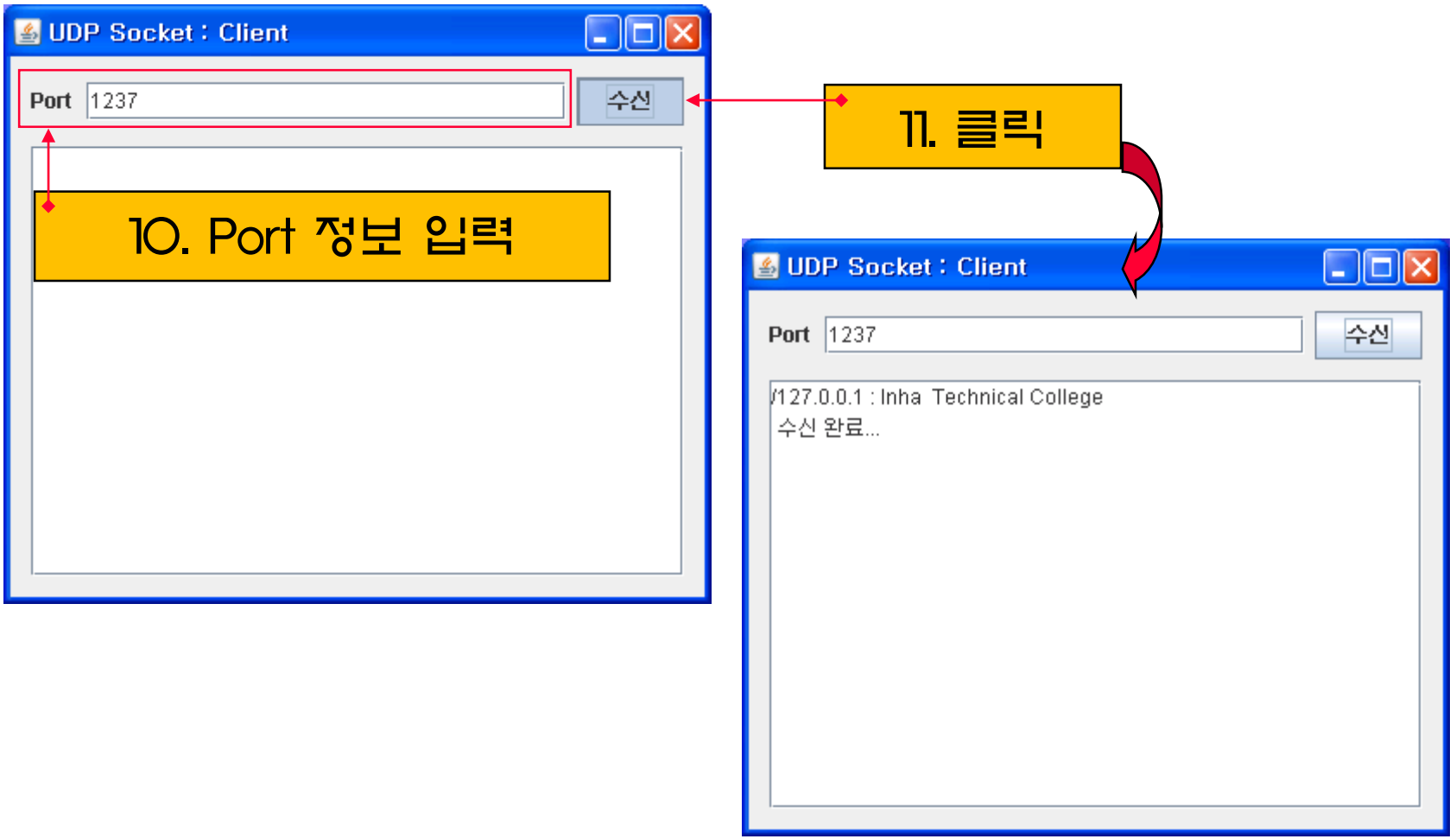
실습 4 : UDP Socket Client 프로그래밍 (6)

실행



실습 4 : UDP Socket Client 프로그래밍 (7)

실행 결과



학습 요약

- Network 개요
- InetAddress 클래스
- URL 클래스
- URLConnection 클래스
- TCP Socket 프로그래밍
- UDP Socket 프로그래밍