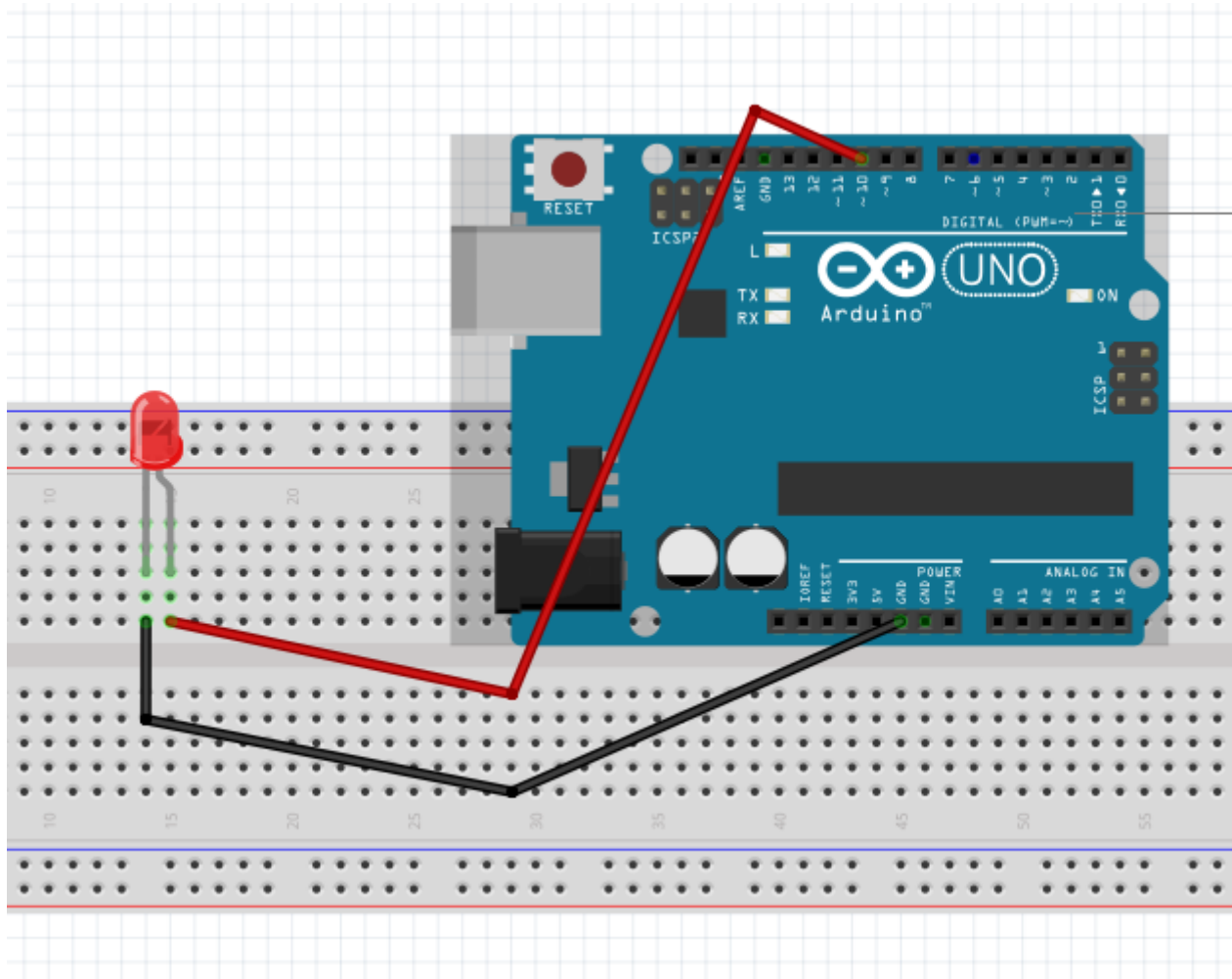
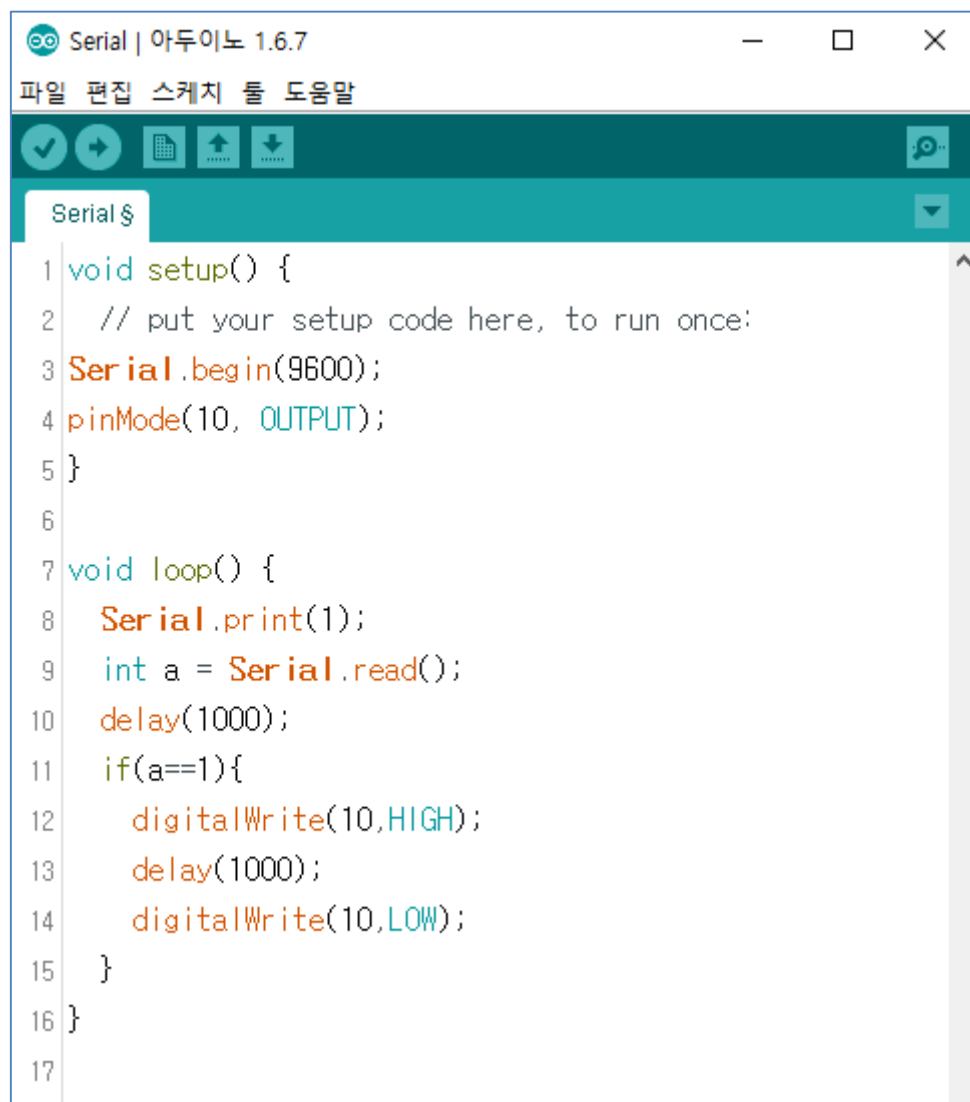


아두이노와 웹서버 연동

아두이노 세팅



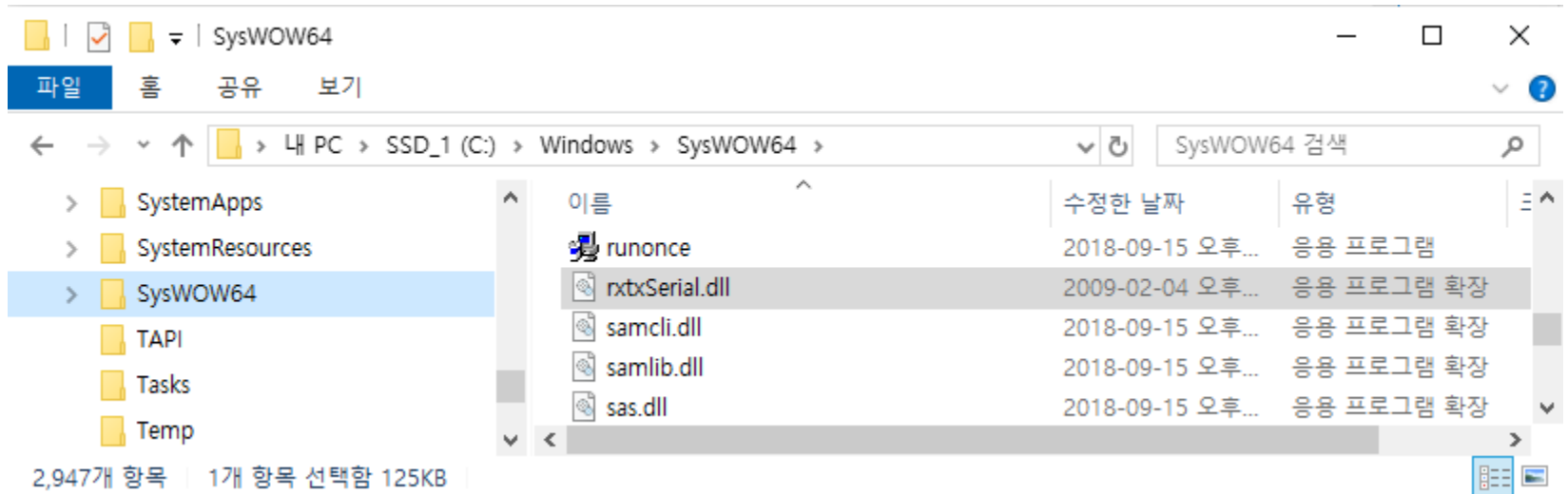
아두이노 코딩



The image shows a screenshot of the Arduino IDE's Serial Monitor window. The window title is "Serial | 아두이노 1.6.7". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for checking, running, uploading, and downloading. The Serial Monitor tab is active, showing a C++ sketch. The code is as follows:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3   Serial.begin(9600);  
4   pinMode(10, OUTPUT);  
5 }  
6  
7 void loop() {  
8   Serial.print(1);  
9   int a = Serial.read();  
10  delay(1000);  
11  if(a==1){  
12    digitalWrite(10,HIGH);  
13    delay(1000);  
14    digitalWrite(10,LOW);  
15  }  
16 }  
17
```

rxtxSerial.dll 추가



경로 →

C:\Windows\SysWOW64\rxtxSerial.dll

윈도우 10 64bit 경로

PC - 이클립스

프로젝트 생성
(JAVA Project)

Main.java coding

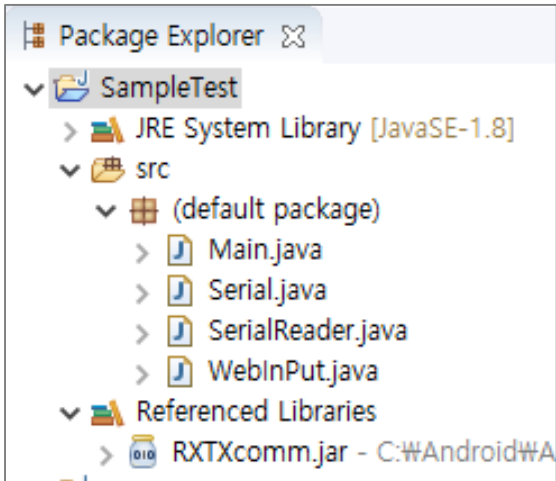
server.jsp coding

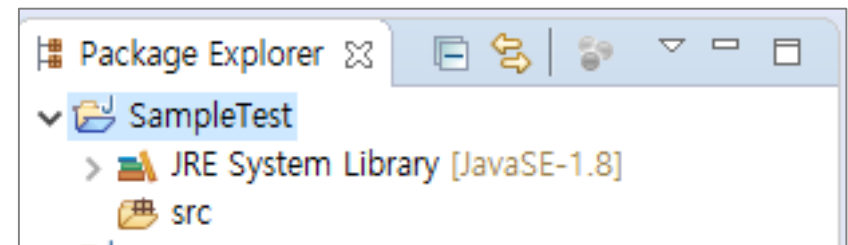
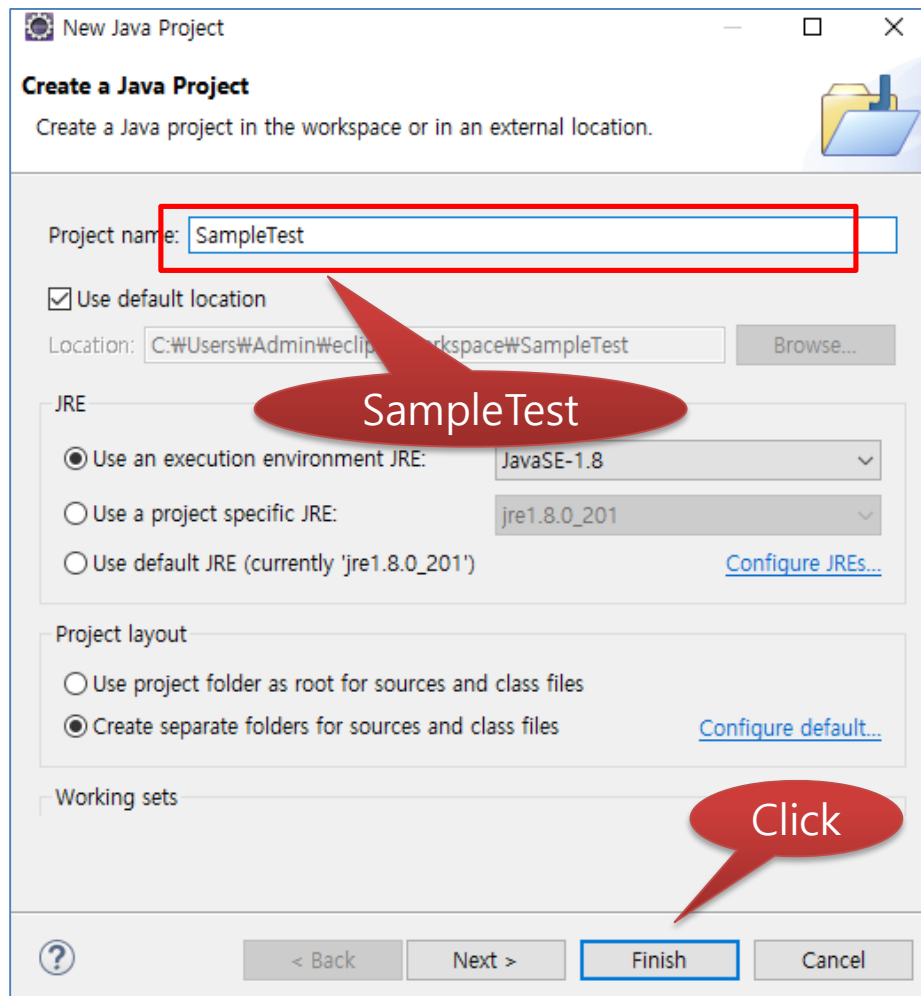
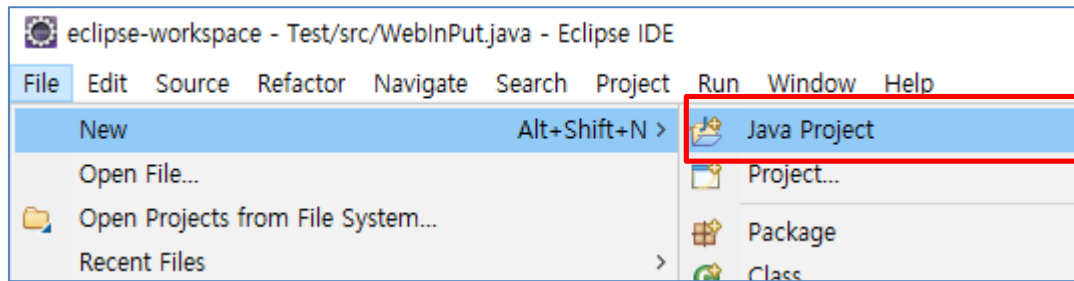
Serial.java coding

SerialReader.java
coding

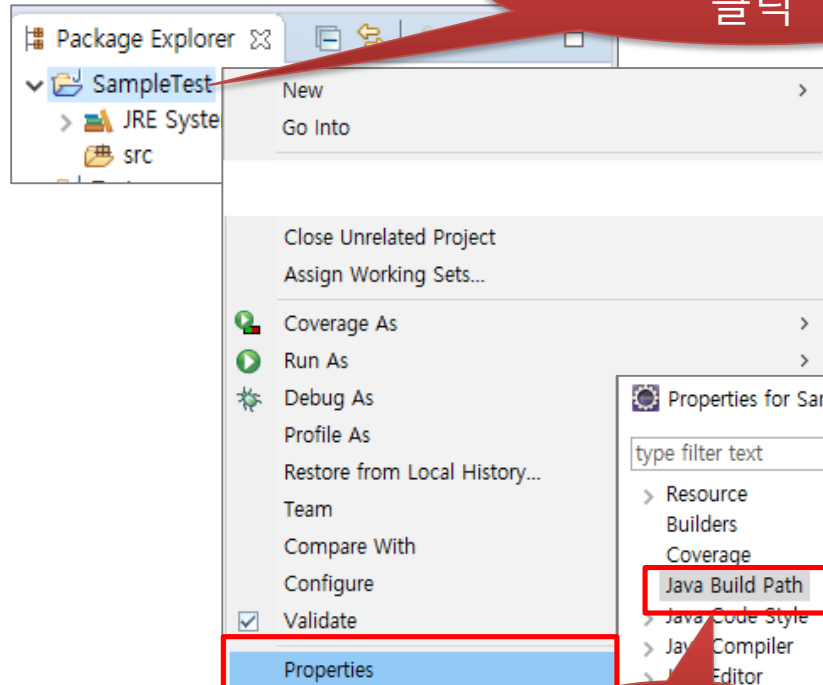
WebInPut.java
coding

168.126.146.39 서버
학번/server.jsp
에 저장



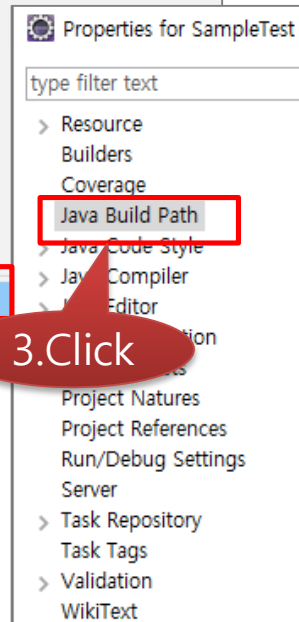


1.오른쪽 마우스
클릭

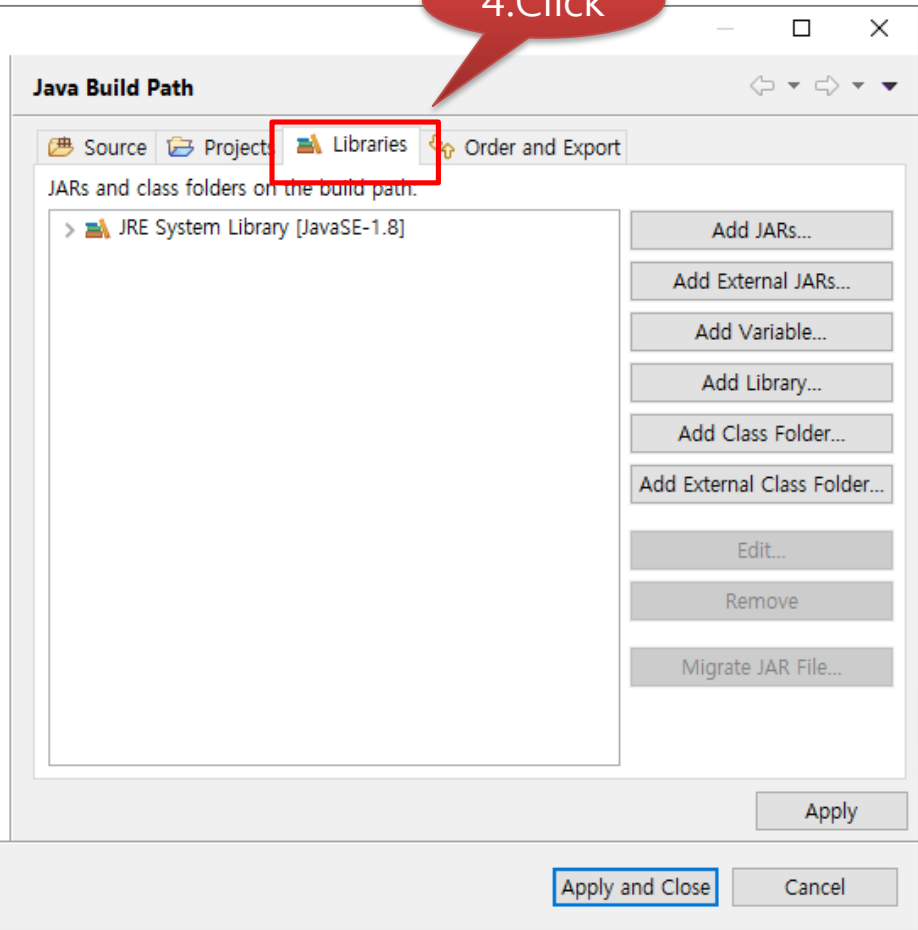


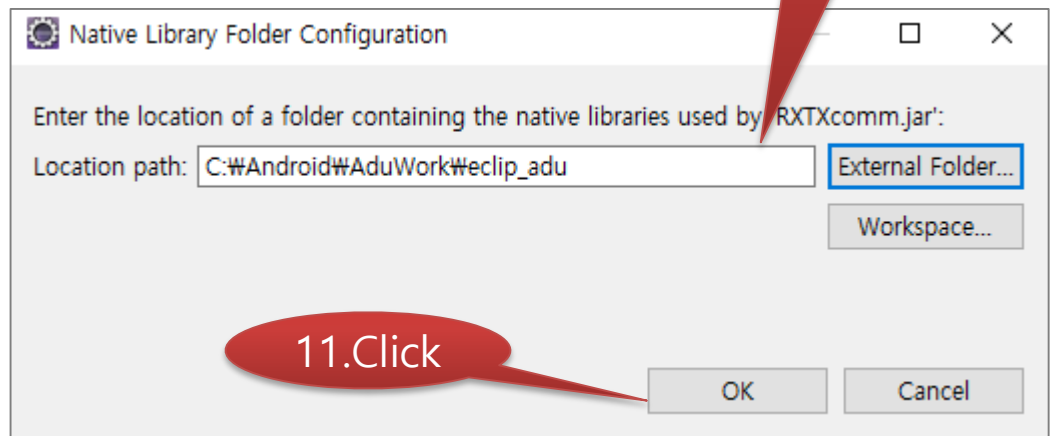
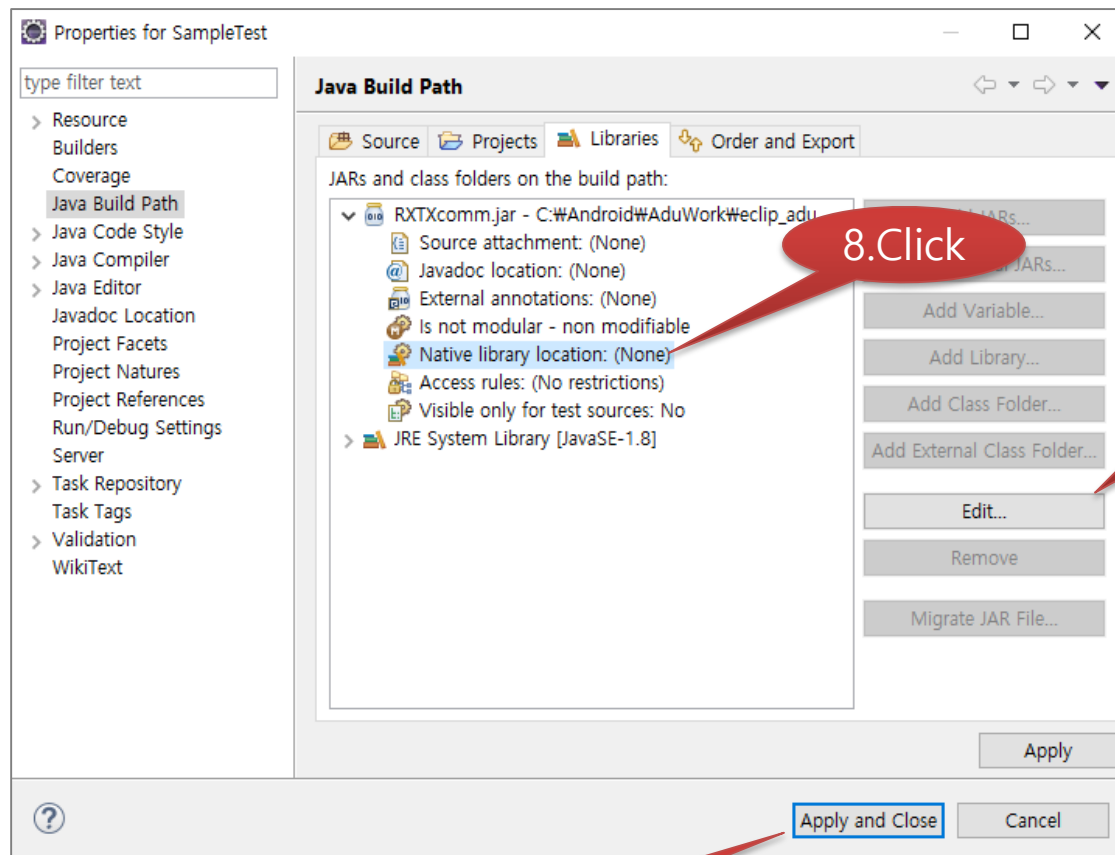
2.Click

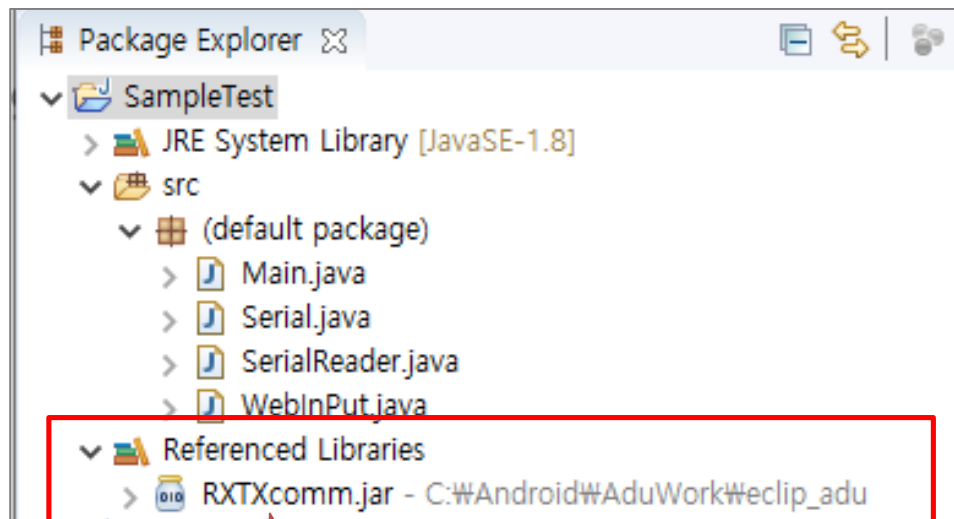
3.Click



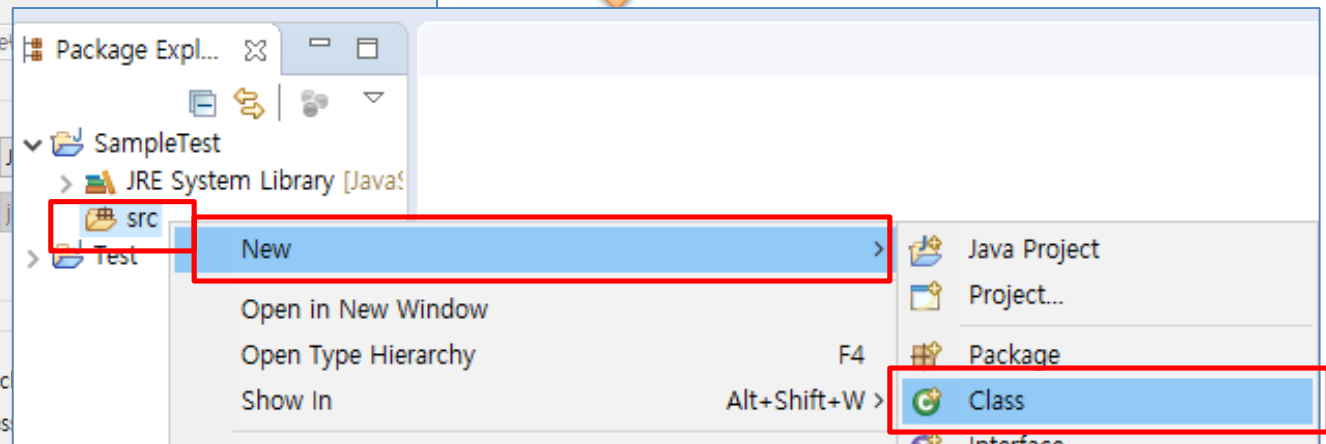
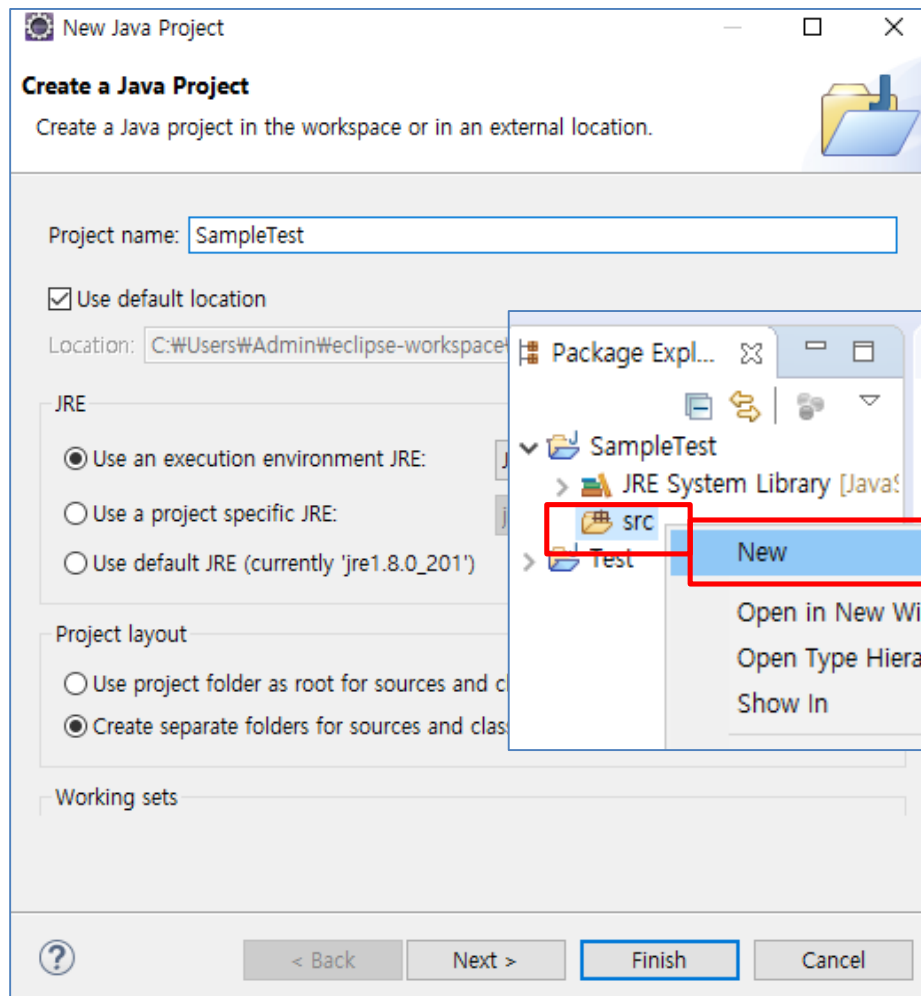
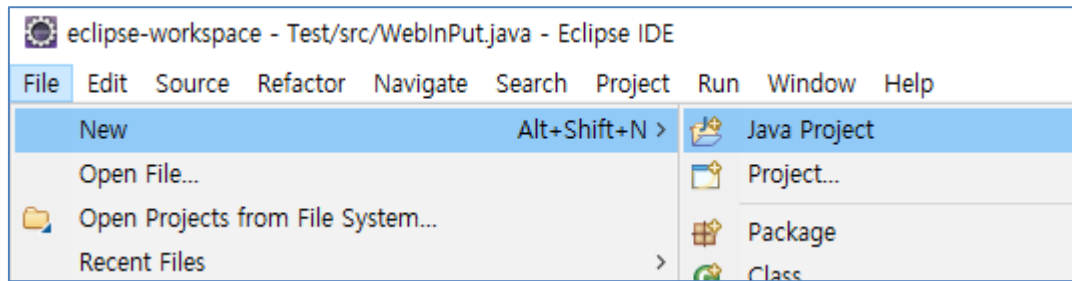
4.Click







13.추가 확인



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: SampleTest/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: Main

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Click

Finish Cancel

Main.java

```
Main.java ✖
1
2 public class Main {
3
4     public static void main(String[] args) {
5
6         try {
7             (new Serial()).connect("COM4");
8         } catch (Exception e) {
9             e.printStackTrace();
10        }
11
12    }
13 }
```

Serial port

Serial.java

Serial.java

```
1 import java.io.InputStream;
2 import java.io.OutputStream;
3 import java.net.URL;
4
5 import gnu.io.CommPort;
6 import gnu.io.CommPortIdentifier;
7 import gnu.io.SerialPort;
8
9 public class Serial {
10     WebInPut client = null;
11     public Serial() {
12         super();
13     }
14
15     void connect(String portName) throws Exception {
16         CommPortIdentifier portIdentifier = CommPortIdentifier.getPortIdentifier(portName);
17         if (portIdentifier.isCurrentlyOwned()) {
18             System.out.println("Error: Port is currently in use");
19         } else {
20             CommPort commPort = portIdentifier.open(this.getClass().getName(), 2000);
21
22             if (commPort instanceof SerialPort) {
23                 SerialPort serialPort = (SerialPort) commPort;
24                 serialPort.setSerialPortParams(9600, SerialPort.DATABITS_8, SerialPort.STOPBITS_1,
25                     SerialPort.PARITY_NONE);
26             }
27         }
28     }
29 }
```

Web Server
Address

```
26      InputStream in = serialPort.getInputStream();
27      OutputStream out = serialPort.getOutputStream();
29
30      client = new WebInPut();
31      URL url = new URL("http://168.126.146.39:8080/egu/server.jsp");
32      client.init(url, out);
33
34      (new Thread(new SerialReader(in, client))).start();
35
36  } else {
37      System.out.println("Error: Only serial ports are handled by this example.");
38  }
39  }
40  }
41  }
42  }
```

```
SerialReader.java ✖
1 import java.io.IOException;
2 import java.io.InputStream;
3
4 public class SerialReader implements Runnable {
5     InputStream in;
6     public static String Data = "";
7     WebInPut client = null;
8     public SerialReader(InputStream in, WebInPut _client) {
9         this.in = in;
10        this.client = _client;
11    }
12    public void run() {
13        byte[] buffer = new byte[1024];
14        int len = -1;
15    }
```

SerialReader.java

```
*SerialReader.java
15
16     try {
17
18         while ((len = this.in.read(buffer)) > -1) {
19             System.out.print(new String(buffer, 0, len));
20             String iaa = new String(buffer, 0, len);
21             if(iaa.equals("1")) {
22                 System.out.print("1--");
23                 Data = iaa;
24                 System.out.println(Data);
25                 client.sendData( Data);
26             }
27         }
28     } catch (IOException e) {
29         e.printStackTrace();
30     }
31 }
32 }
```

Arduino Read
data

Arduino Read
Data를 Web
Server로 전송

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.io.OutputStream;
5 import java.io.OutputStreamWriter;
6 import java.net.MalformedURLException;
7 import java.net.URL;
8 import java.net.URLConnection;
9 import java.net.URLEncoder;
10
11
12 public class WebInPut{
13
14     public int inData = 0;
15     public int outData = 0;
16     public URL url;
17     public boolean isInit = false;
18     public OutputStream out = null;
19
20     public void init(URL _url, OutputStream _out){
21         this.isInit = true;
22         this.url = _url;
23         this.out = _out;
24     }
```

```
25 public void sendData( String reqData)throws MalformedURLException {
26
27
28     try {
29         if(!isInit)
30         {
31             return;
32         }
33
34         URLConnection urlConn = null;
35         OutputStreamWriter osw = null;
36         BufferedReader br = null;
37         String resData = null;
38
39         urlConn = url.openConnection();
40         urlConn.setDoOutput(true);
41
42         osw = new OutputStreamWriter(urlConn.getOutputStream());
43         System.out.println("indata-"+reqData);
44         reqData = URLEncoder.encode("msg","UTF-8") + "="
45             + URLEncoder.encode(reqData,"UTF-8");
46         osw.write(reqData);
47         osw.flush();
48         String result = "";
49         br = new BufferedReader(new InputStreamReader(urlConn.getInputStream()));
50         while((resData = br.readLine()) != null) {
51             result += resData;
52         }
53     }
```

WebInPut.java

Web Server
로 전송Web Server
data 받기

WebInPut.java

WebInPut.java

```
52         }
53
54         System.out.println("web_output:"+result);
55
56         osw.close();
57         br.close();
58
59         out.write(Integer.parseInt(result));
60
61     }
62     }catch (IOException e) {
63         e.printStackTrace();
64     }
65 }
66
67 }
68
```

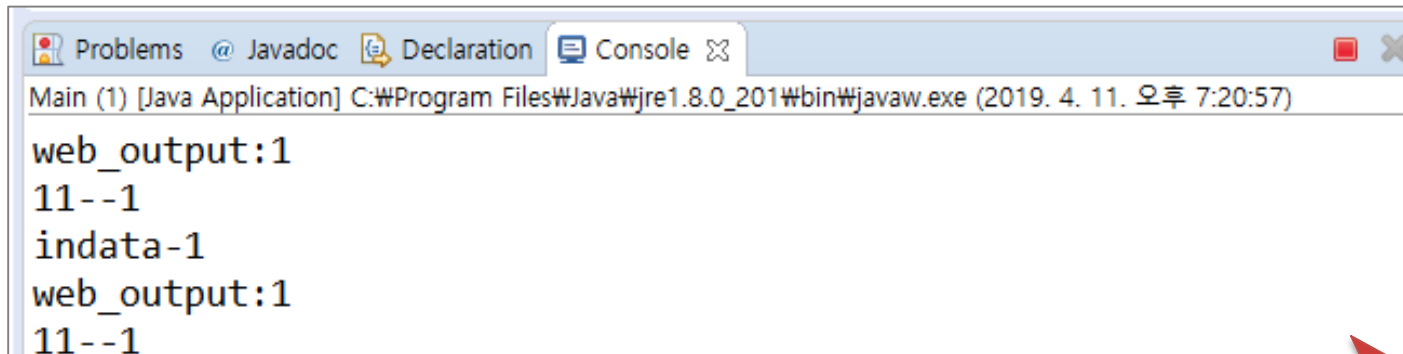
Web Server
data를 Arduino
로 보내기

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"  
    pageEncoding="EUC-KR"%>  
<% String recvMessage = request.getParameter("msg");  
    out.println(recvMessage);  
%>
```



Web Server
jsp 파일

실행



The screenshot shows an IDE window with a 'Console' tab. The title bar indicates the application is 'Main (1) [Java Application]' running 'C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe' at '2019. 4. 11. 오후 7:20:57'. The console output consists of five lines: 'web_output:1', '11--1', 'indata-1', 'web_output:1', and '11--1'.

```
Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (2019. 4. 11. 오후 7:20:57)
web_output:1
11--1
indata-1
web_output:1
11--1
```

이클립스

