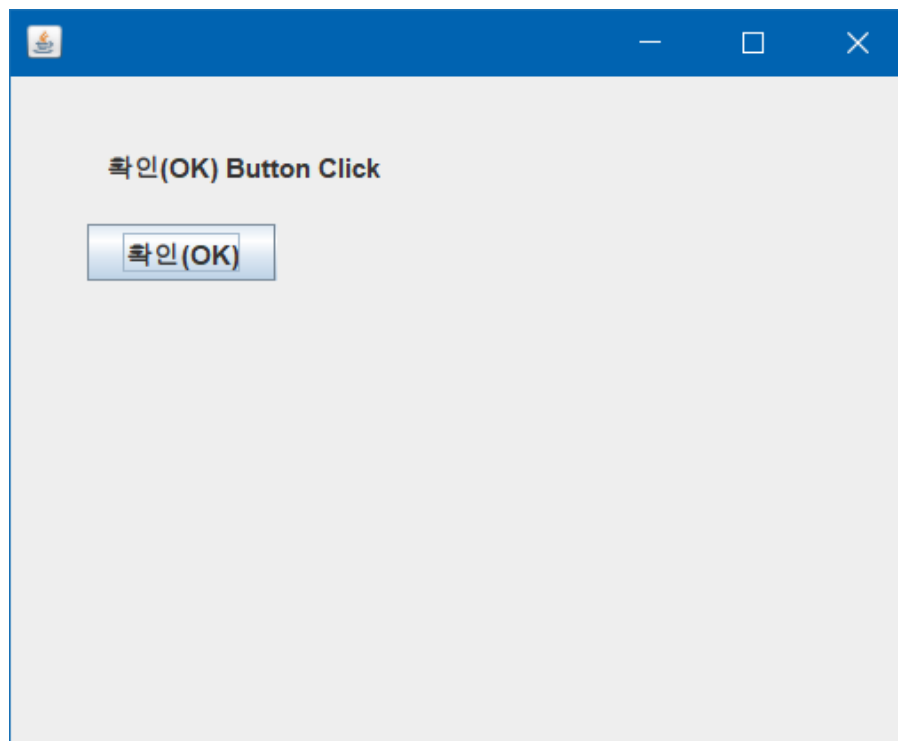




Windows 프로그래밍





학습 목표

■ 이 강의를 마치면 학생들은

- ❖ AWT(Abstract Windows Toolkit) 에 대하여 설명할 수 있다.
- ❖ Swing에 대하여 설명할 수 있다.
- ❖ Window 프로그래밍 방법에 대하여 설명할 수 있다.
- ❖ Event에 대하여 설명할 수 있다.
- ❖ Event Handler에 대하여 설명할 수 있다.
- ❖ Event Handler 구현 방법에 대하여 설명할 수 있다.





AWT 란 ? (1)

■ AWT(Abstract Windows Toolkit)

❖ Definition

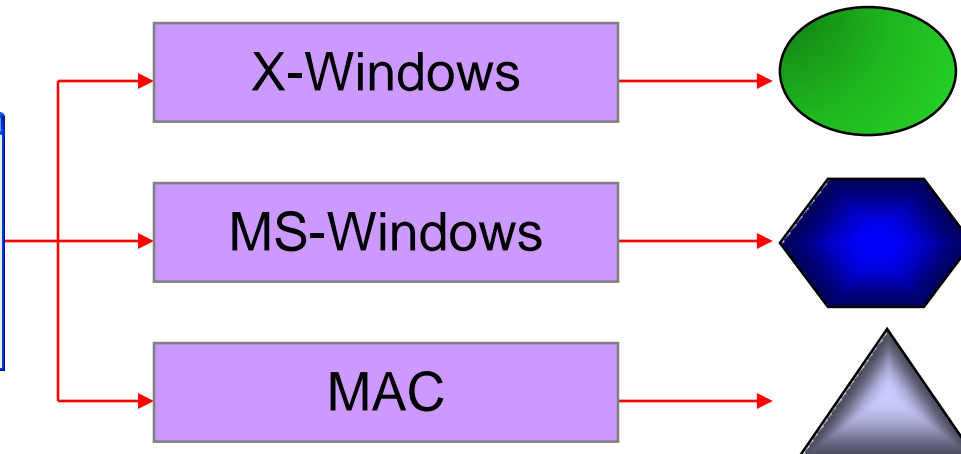
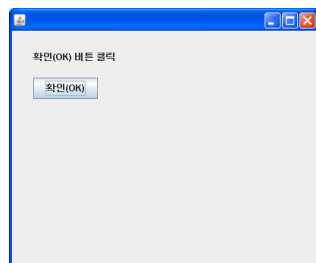
◆ GUI를 구축하기 위한 클래스들의 모음

❖ Type

◆ 사용자 Interface 클래스

◆ 그래픽 처리 클래스

❖ Characteristic



각 Windows 에서
다르게 실행된 결과





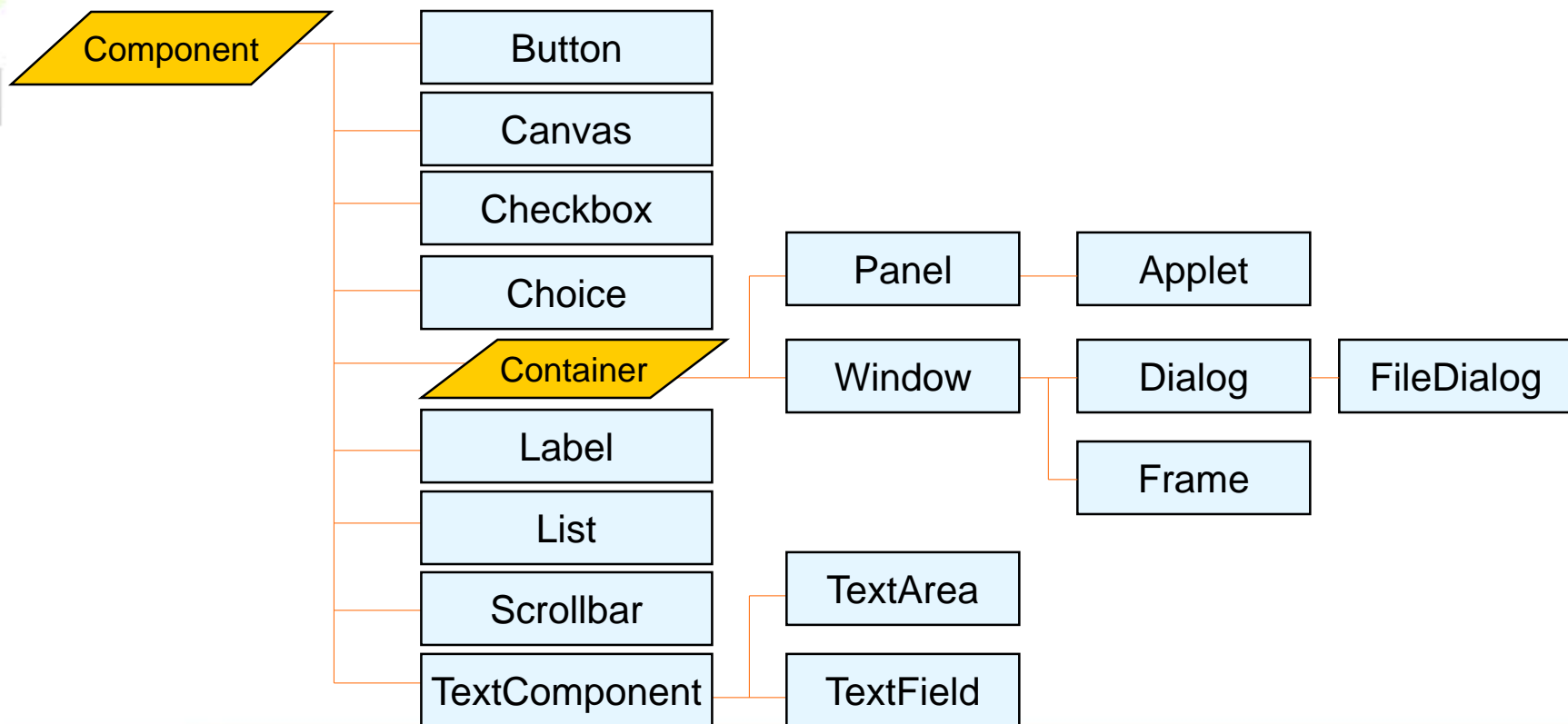
AWT 란 ? (2)

■ AWT의 클래스 구조

❖ Component(Component)

◆ Window 환경의 Component(Button, CheckBox, ...)

◆ java.awt 클래스로 객체 모델링





AWT 란 ? (3)

■ Method

Method	Function
<code>Public Dimension</code> <code>getSize()</code>	Component 현재의 크기를 Dimension 클래스 객체로 반환
<code>Public void</code> <code>setForeground(Color c)</code> <code>Public void</code> <code>setBackground(Color c)</code>	Text 색 결정 Text외의 색 결정
<code>Public void</code> <code>setFont(Font f)</code>	Font 설정
<code>Public void</code> <code>setEnabled(boolean b)</code>	<code>false</code> : inactive status
<code>void</code> <code>setBounds(int x, int y, int width, int height)</code> <code>void</code> <code>setSize(Dimension d)</code>	Component 위치 지정 Component 크기 지정
<code>void</code> <code>setVisible(boolean b)</code>	<code>true</code> : 화면에 출력 <code>false</code> : 화면에서 사라짐

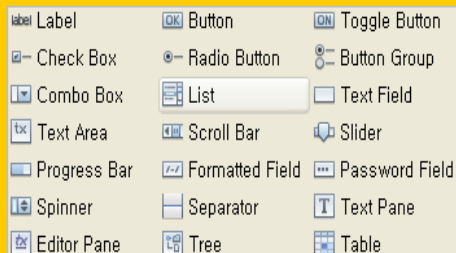




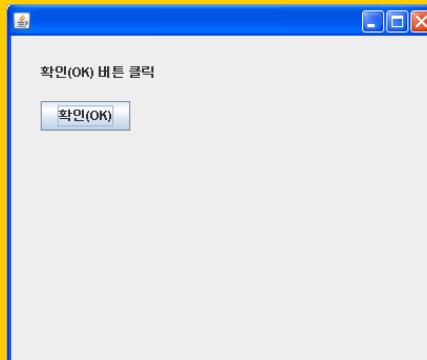
AWT 란 ? (4)

Relationship Component-Container

Component 생성



컨테이너 Layout



Event Handler 작성

```
public class MainFrame extends javax.swing.JFrame {  
    [ ... ]  
    public MainFrame() {  
        [ ... ]  
    }  
    [ ... ]  
    Generated Code  
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
        jLabel1.setText("확인(OK) 버튼 클릭");  
    }  
    [ ... ]  
    public static void main(String args[]) {  
        // Variables declaration - do not modify  
        private javax.swing.JButton jButton1;  
        private javax.swing.JLabel jLabel1;  
        // End of variables declaration  
    }  
}
```

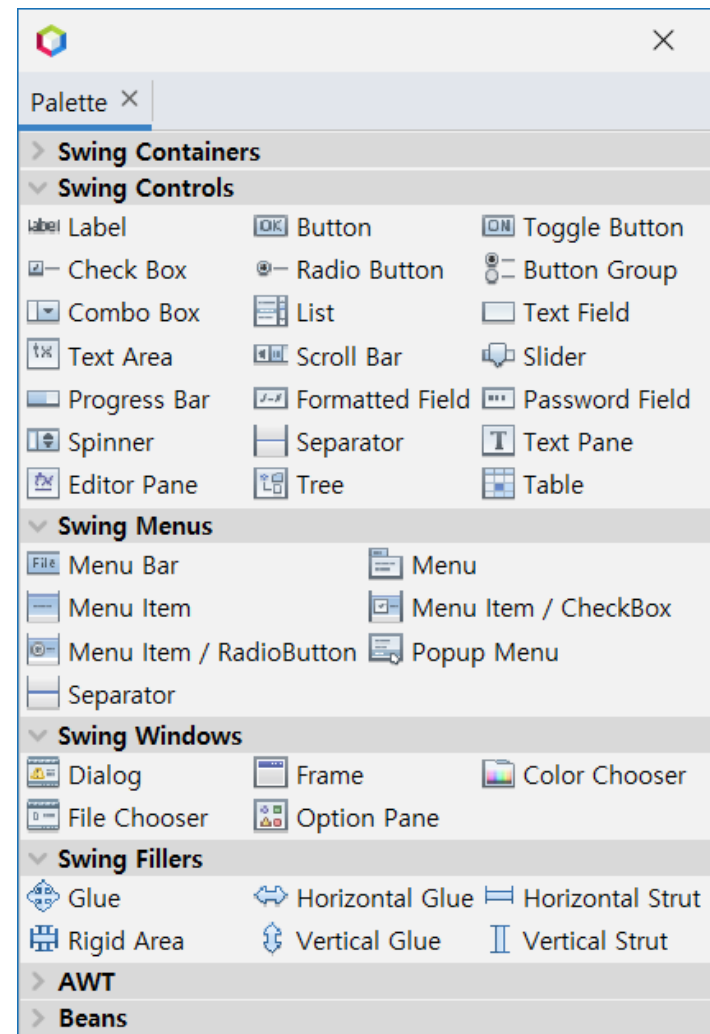




AWT 란 ? (5)

■ Component

Component	Function
Label	고정 문자열 표시
Button	버튼
TextField	1 line 문자열 입력
TextArea	여러 line 문자열 입력
Checkbox	체크박스, 옵션버튼을 작성
Choice	Drop-down 리스트를 작성
Canvas	그리기 공간 작성
List	리스트 작성
Scrollbar	스크롤바 작성





Swing 이란 ? (1)

■ Swing

❖ Definition

- ◆ 순수한 자바 언어로 지원되는 GUI 개발 도구

❖ Characteristic

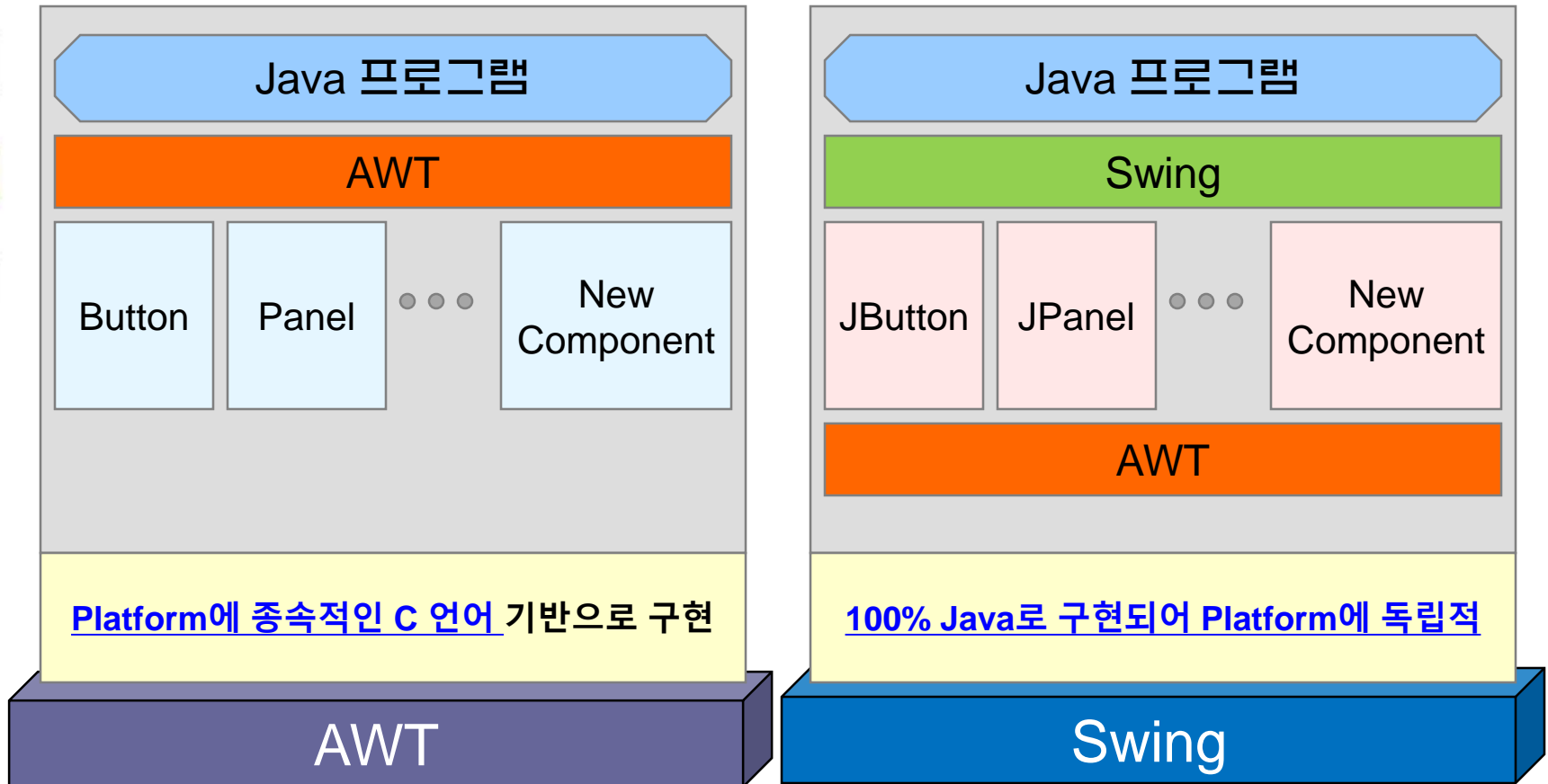
- ◆ Platform에 독립적이다.
- ◆ 새로운 Component 제작이 쉽다.
- ◆ 풍선도움말(ToolTip) Function을 제공한다.
- ◆ AWT Component를 지원한다.





Swing 이란 ? (2)

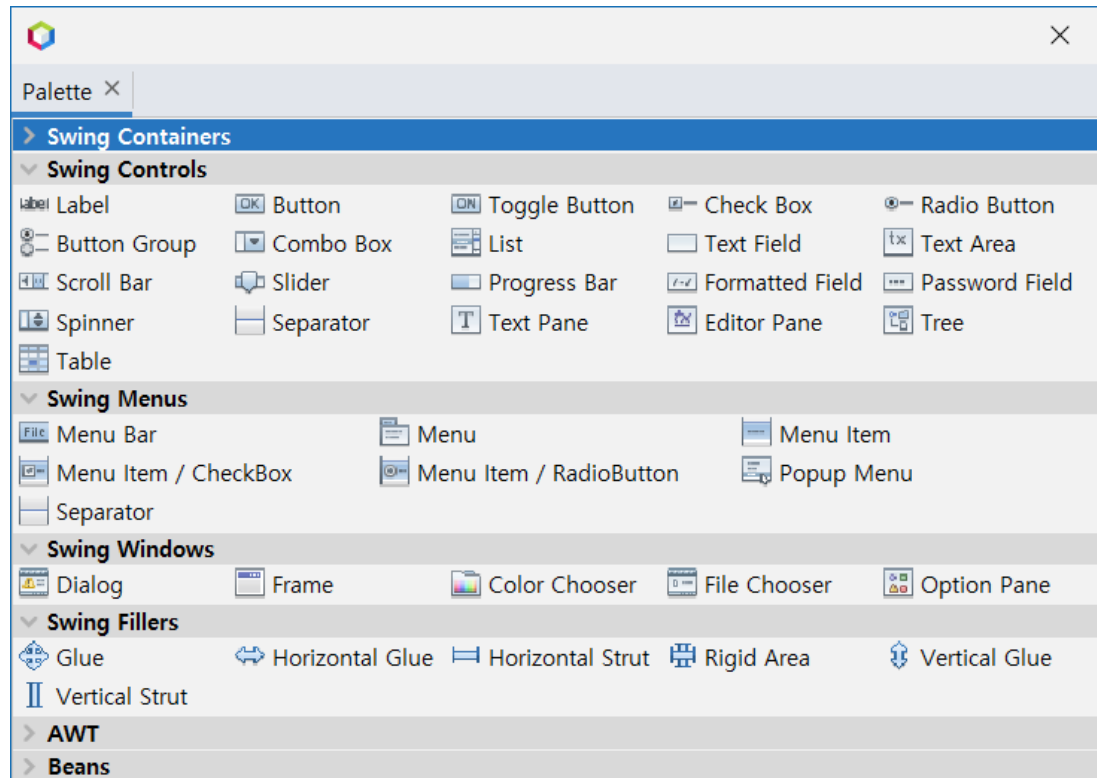
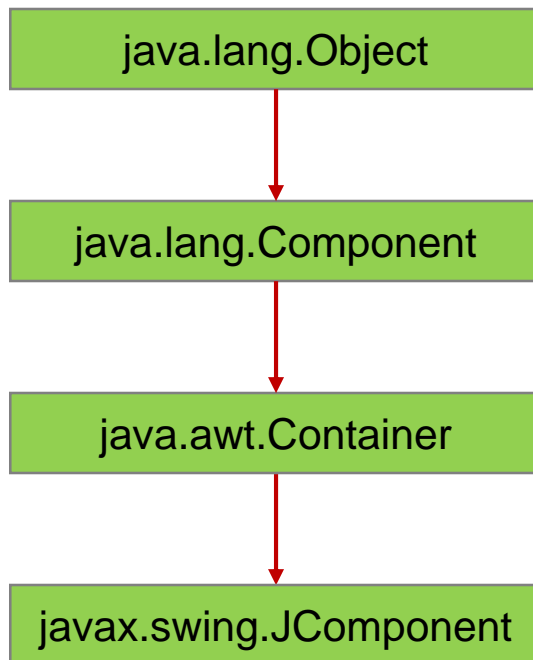
■ AWT vs. Swing





Swing 이란 ? (3)

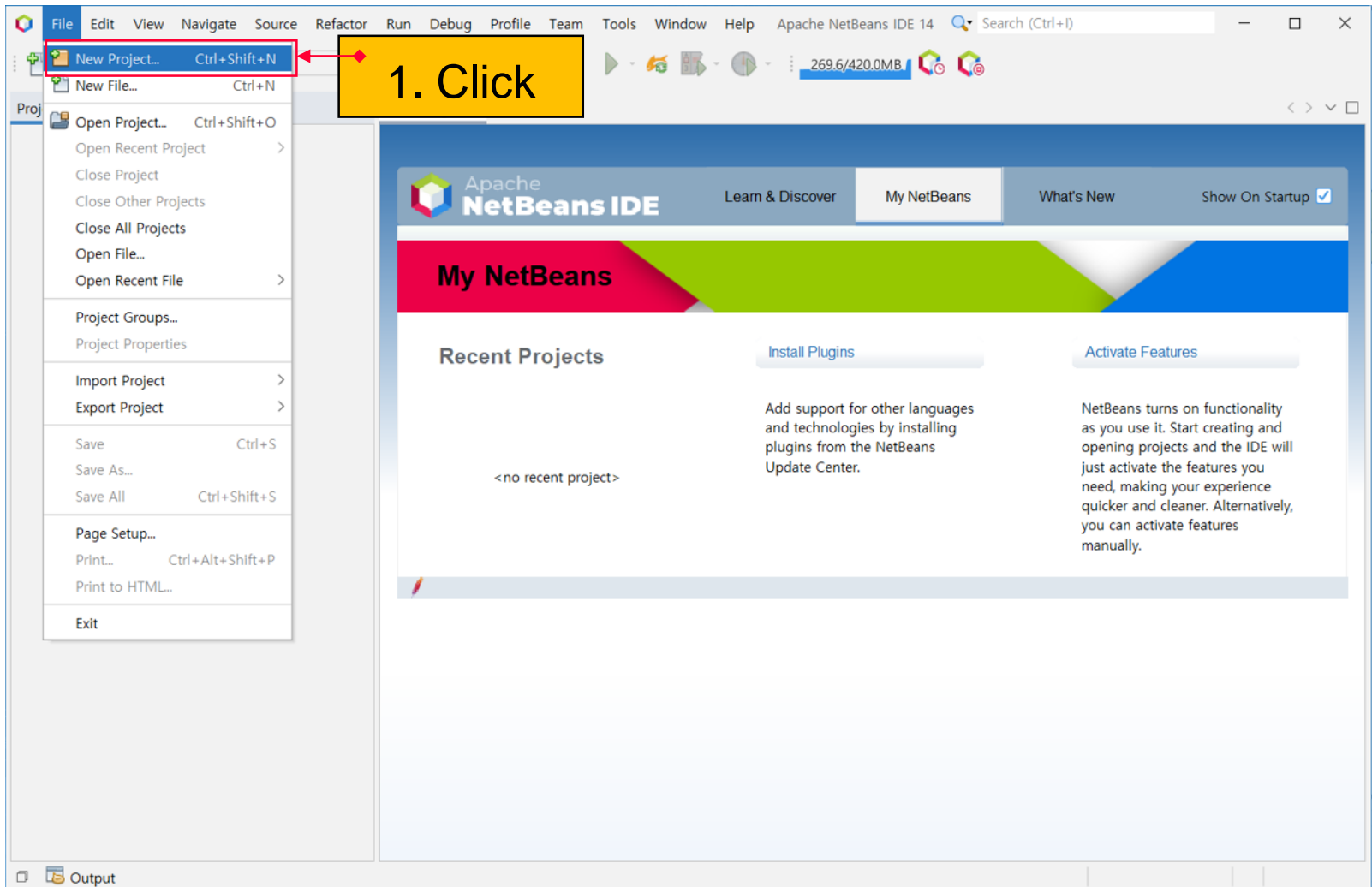
■ Swing의 Class 구조





Windows Application 개발 방법 (1)

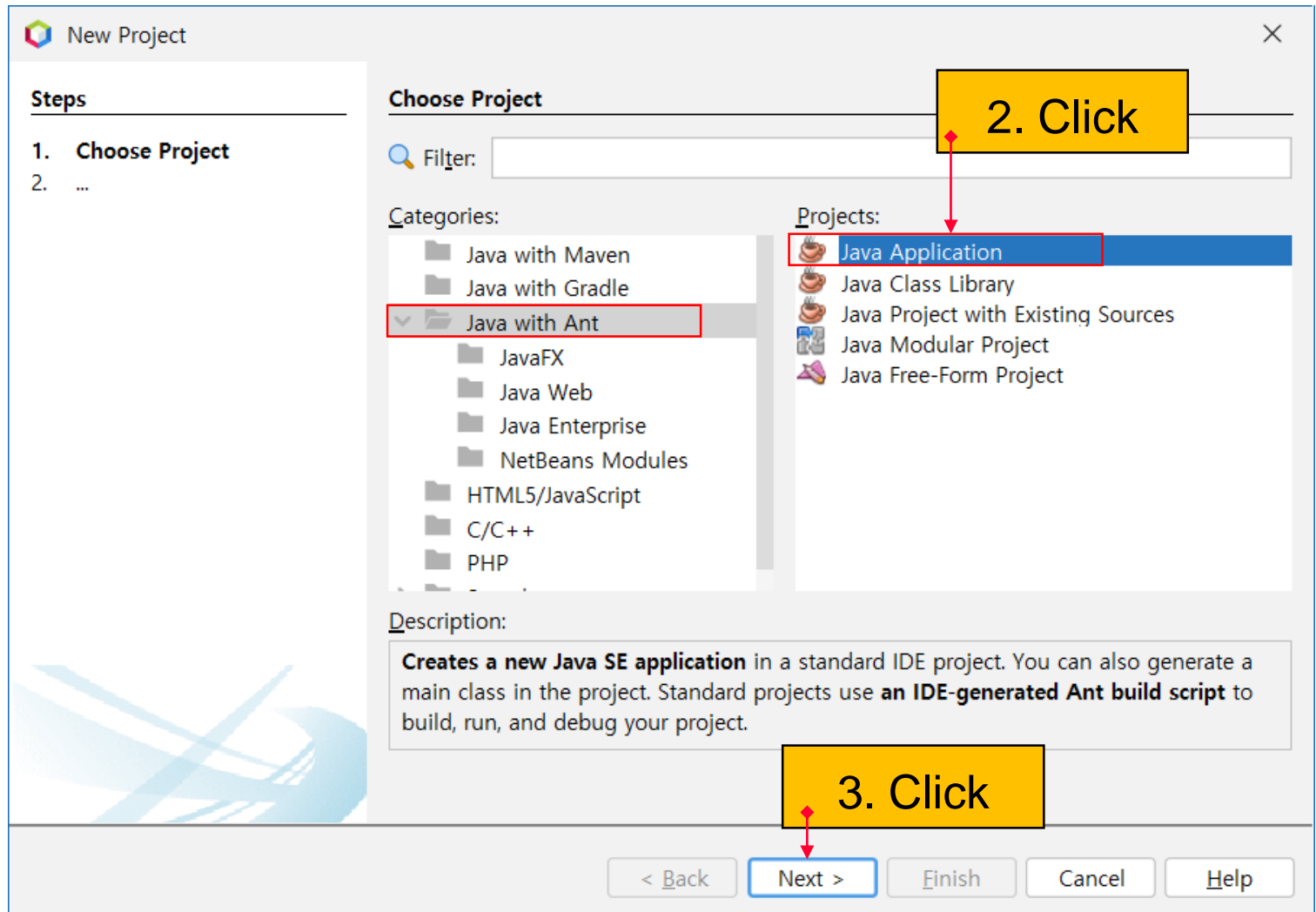
Create Project





Windows Application 개발 방법 (2)

Choose Project





Windows Application 개발 방법 (3)

■ Project Name and Location

❖ Project name: Windows_Source

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: Windows_Source

Project Location: C:\Java_Project Browse...

Project Folder: C:\Java_Project\Windows_Source

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☐ Create Main Class windows_source.Windows_Source

4. Input "Windows_Source"

5. Check reset

6. Click

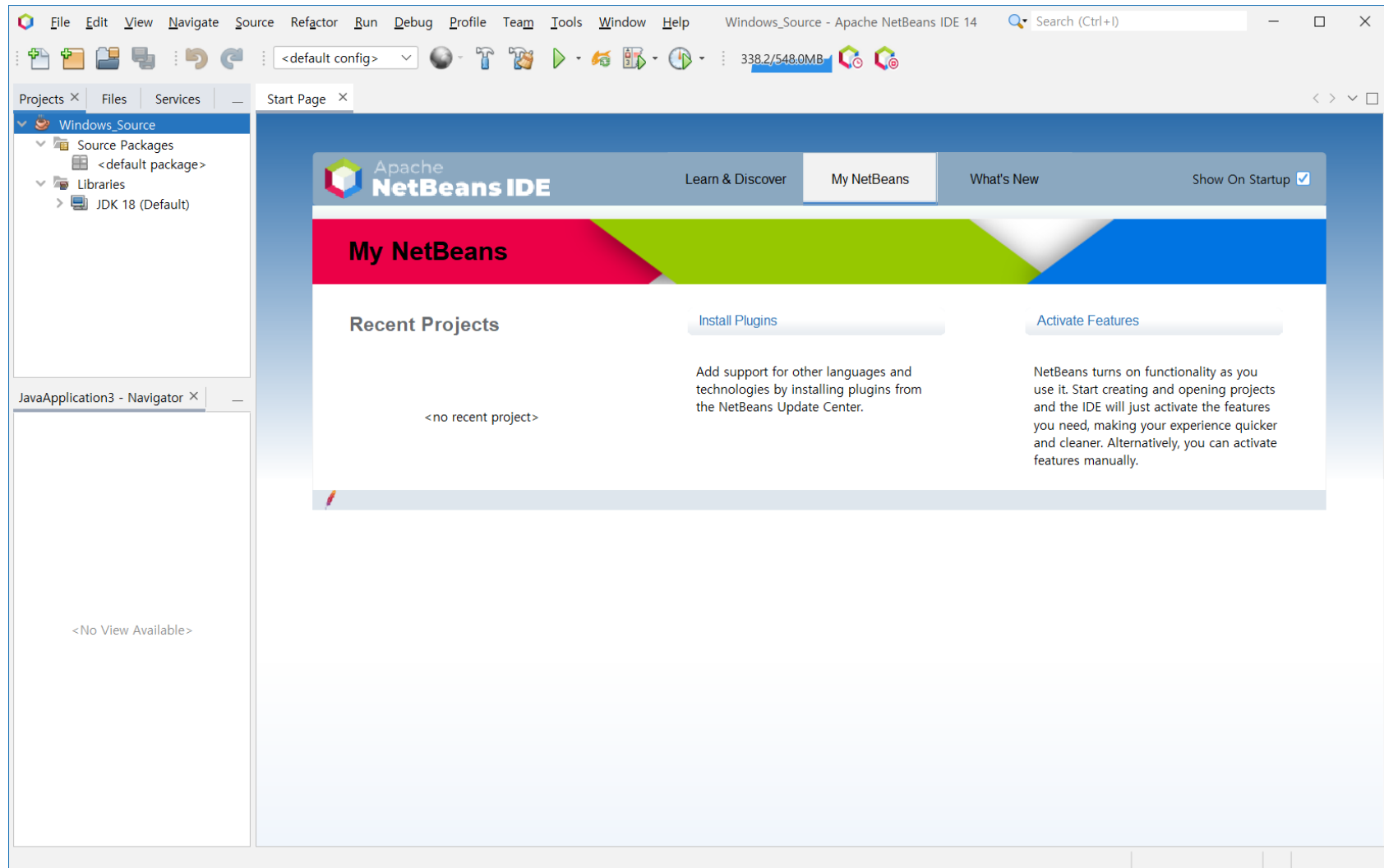
< Back Next > Finish Cancel Help





Windows Application 개발 방법 (4)

Create project success

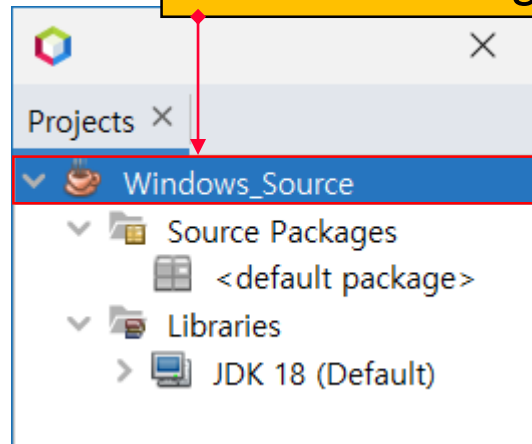




Windows Application 개발 방법 (5)

Create JFrame Form

7. Mouse right-button click



New

Build
Clean and Build

8. Click

Run
Debug
Profile
Test
Set Configuration

Set as Main Project
Open Required Projects
Close

Rename...
Move...
Copy...
Delete

Find...
Versioning
Local History

Properties

9. Click

JFrame Form...

Java Class...

Java Package...

Java Module...

Java Project...

JP...

Entity Class...

Entity Classes from Database...

Other...





Windows Application 개발 방법 (6)

Setting Class Name

❖ Mainframe.java 생성

The screenshot shows the NetBeans IDE interface. The 'New JFrame Form' dialog is open, and the 'Steps' pane on the left indicates that the user is at step 2, 'Name and Location'. The 'Name and Location' section of the dialog has the following fields:

- Class Name:** MainFrame (highlighted with a yellow box and the text '10. Input "MainFrame"')
- Project:** Windows_Source
- Location:** Source Packages
- Package:** (empty)
- Created File:** vonjo\OneDrive\문서\NetBeansProjects\Windows_Source\src\MainFrame.java

The 'Projects' tree on the left shows the project structure:

- Windows_Source
 - Source Packages
 - <default package>
 - MainFrame.java (highlighted with a red box)
- Libraries
 - JDK 18 (Default)

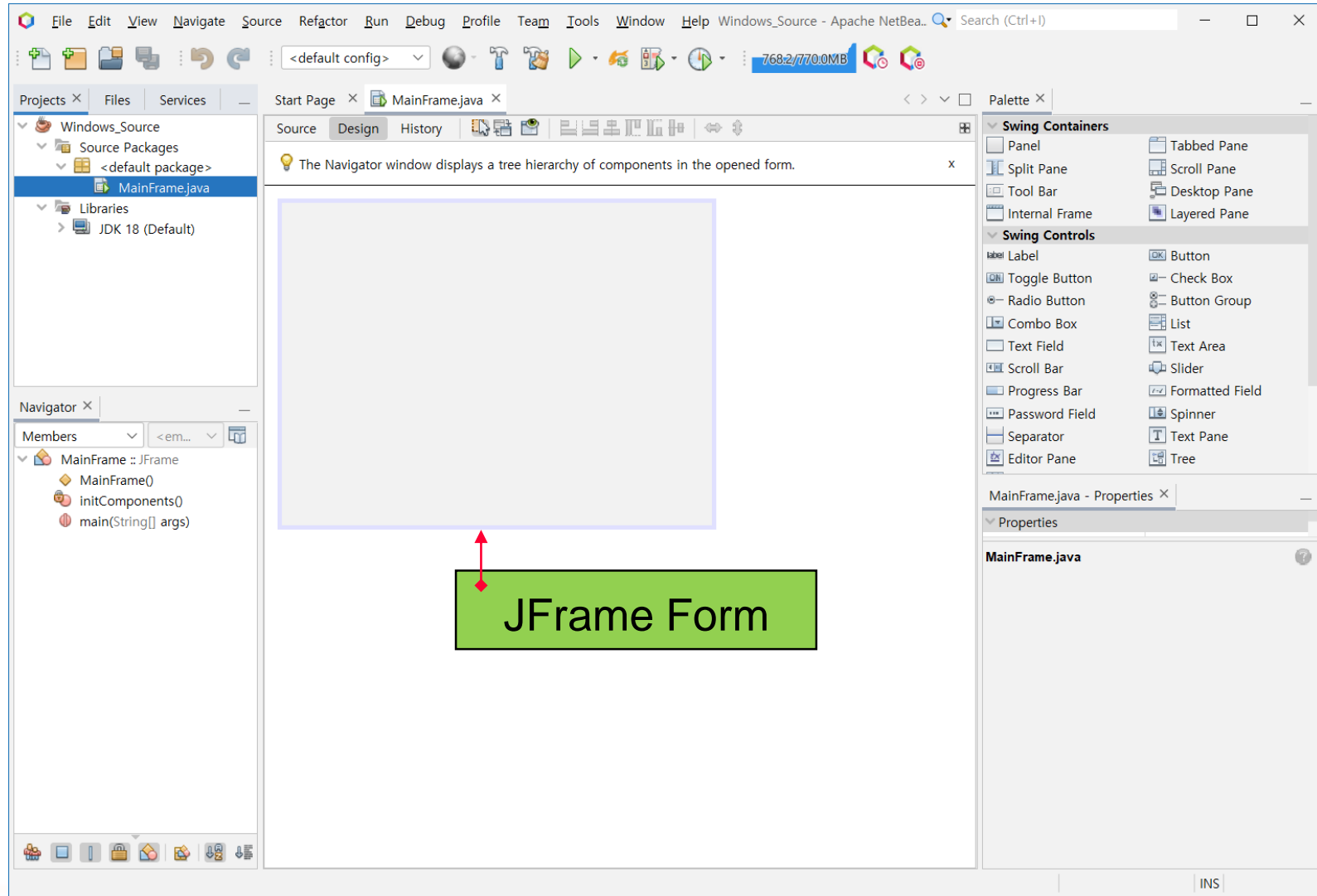
At the bottom of the dialog, the 'Finish' button is highlighted with a yellow box and the text '11. Click', with a red arrow pointing to it. The 'Finish' button is part of a row of buttons that also includes '< Back', 'Next >', 'Cancel', and 'Help'.





Windows Application 개발 방법 (7)

Create JFrame form success

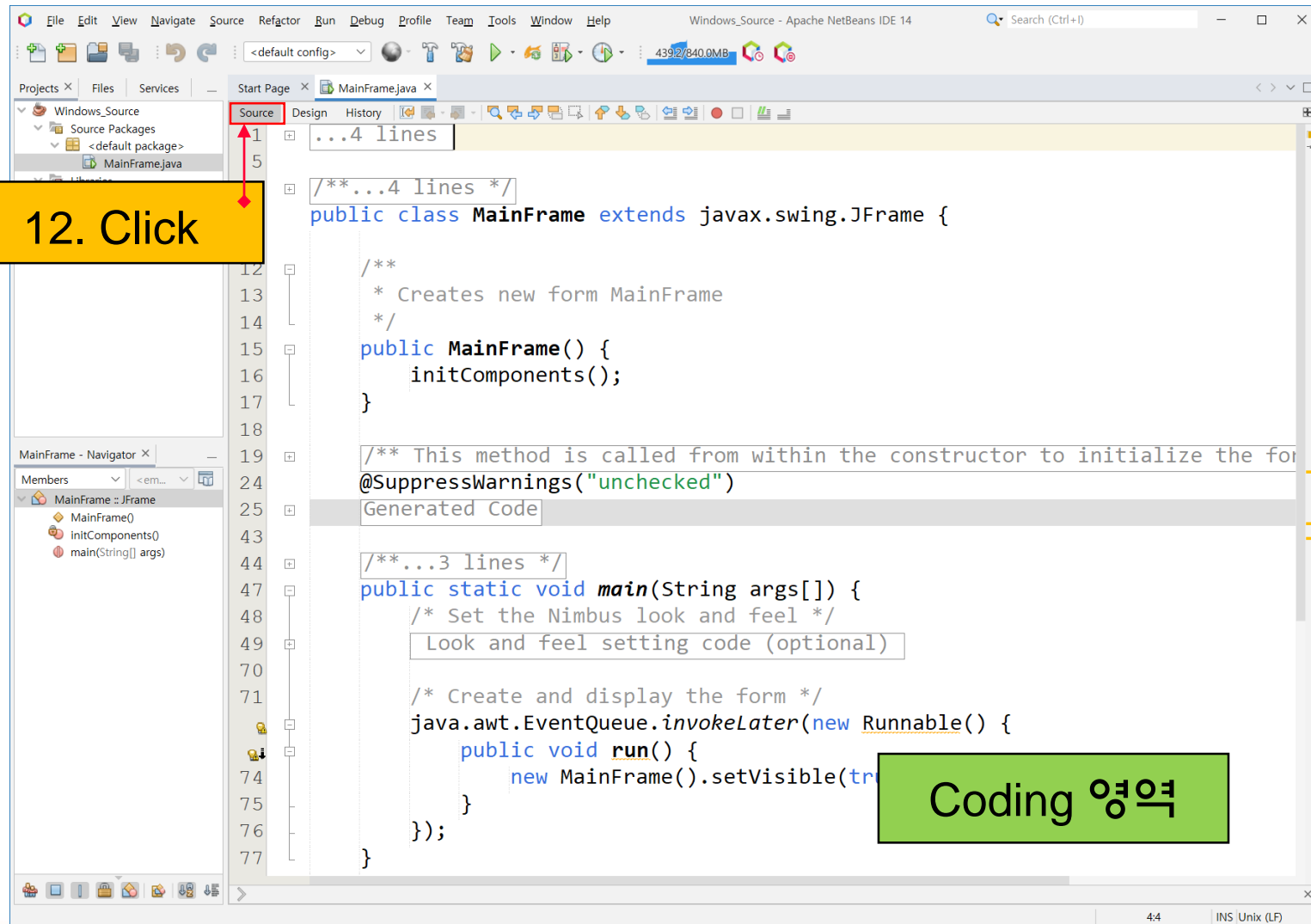




Windows Application 개발 방법 (8)

Source Code

12. Click



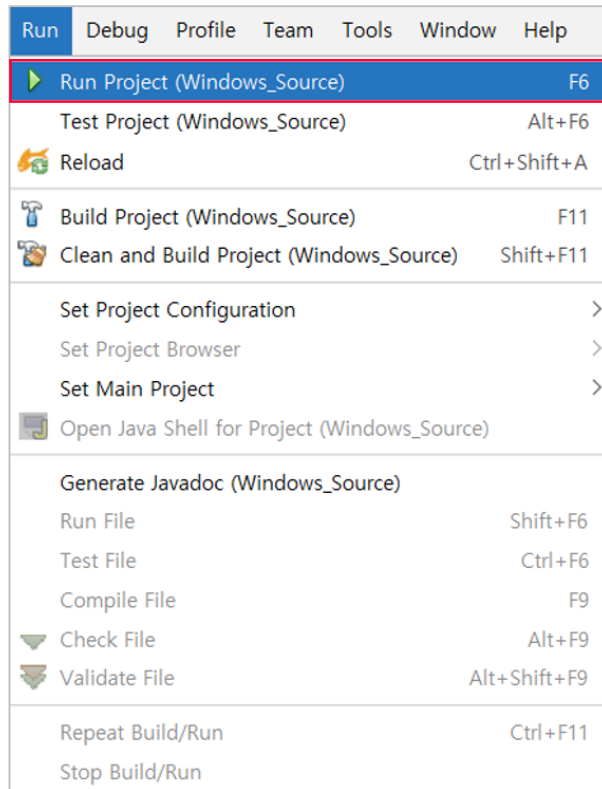
Coding 영역



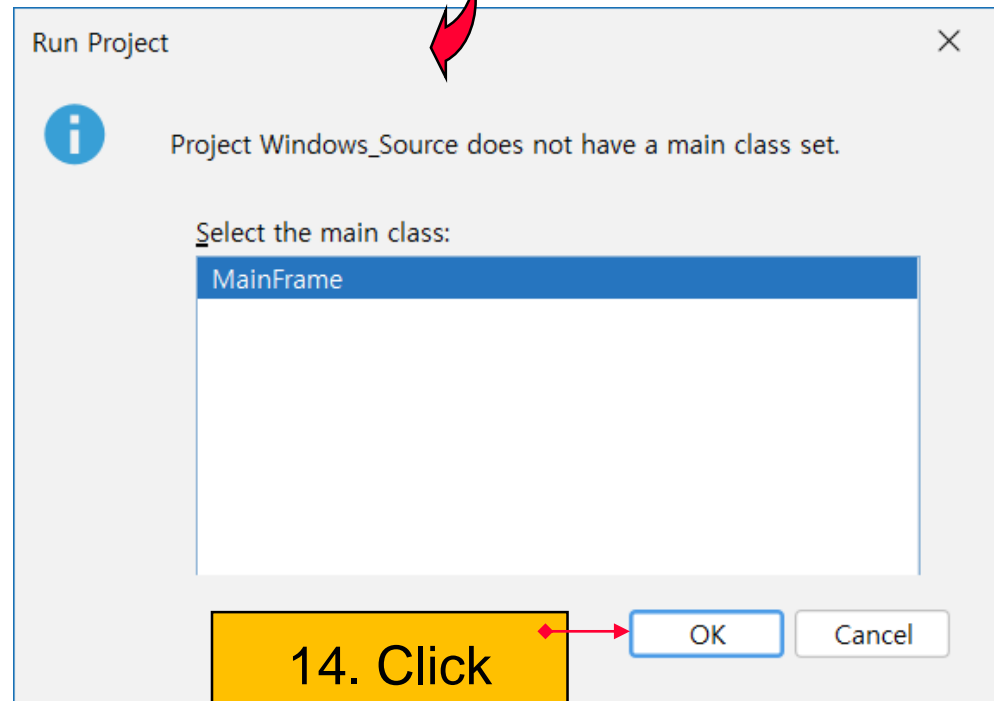


Windows Application 개발 방법 (9)

Run



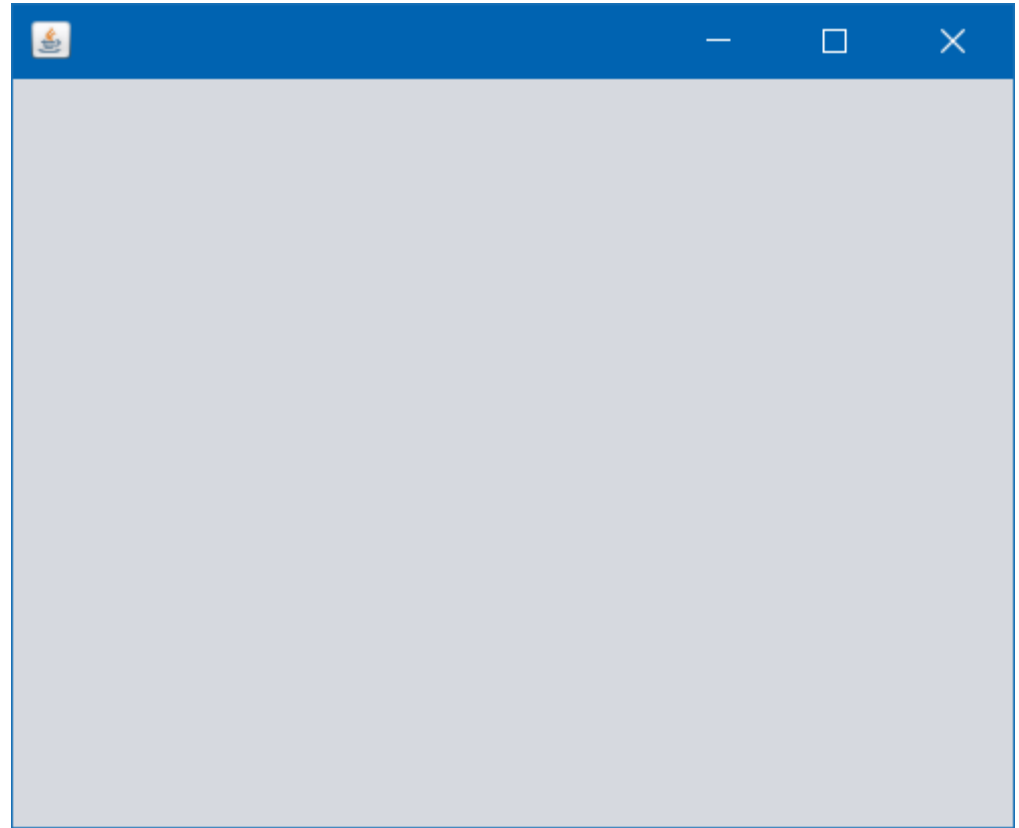
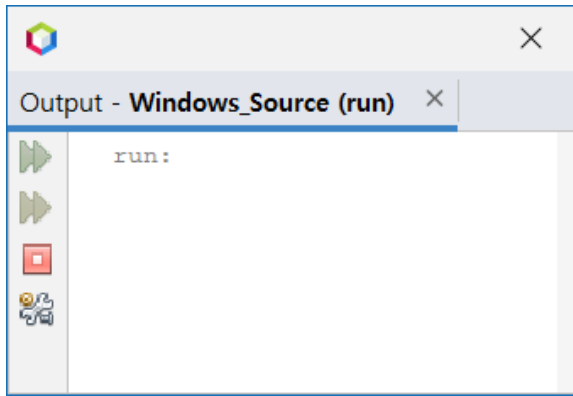
13. Click





Windows Application 개발 방법 (10)

Run result



15. Confirm execution result

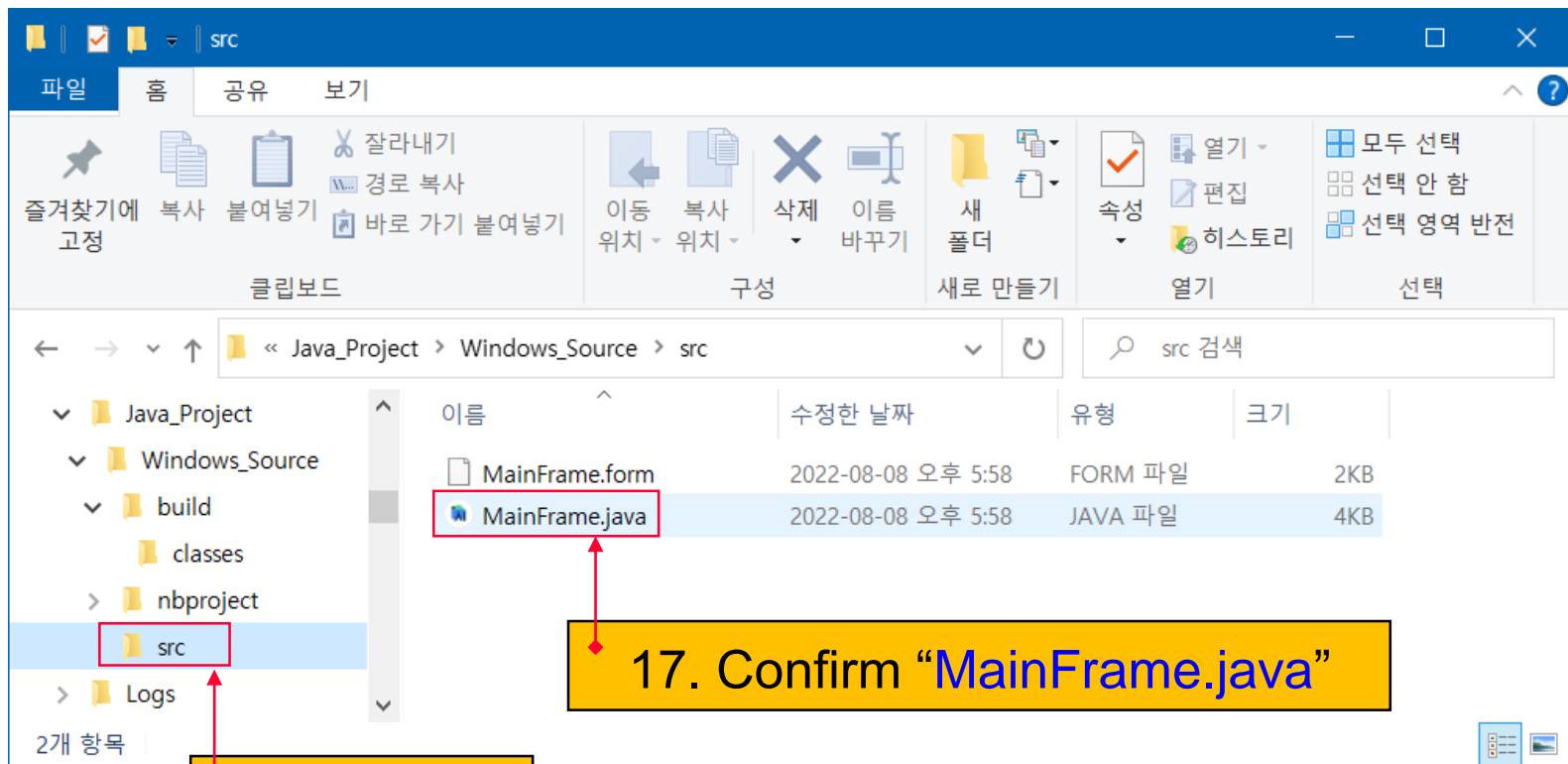




Windows Application 개발 방법 (11)

■ Project 구조

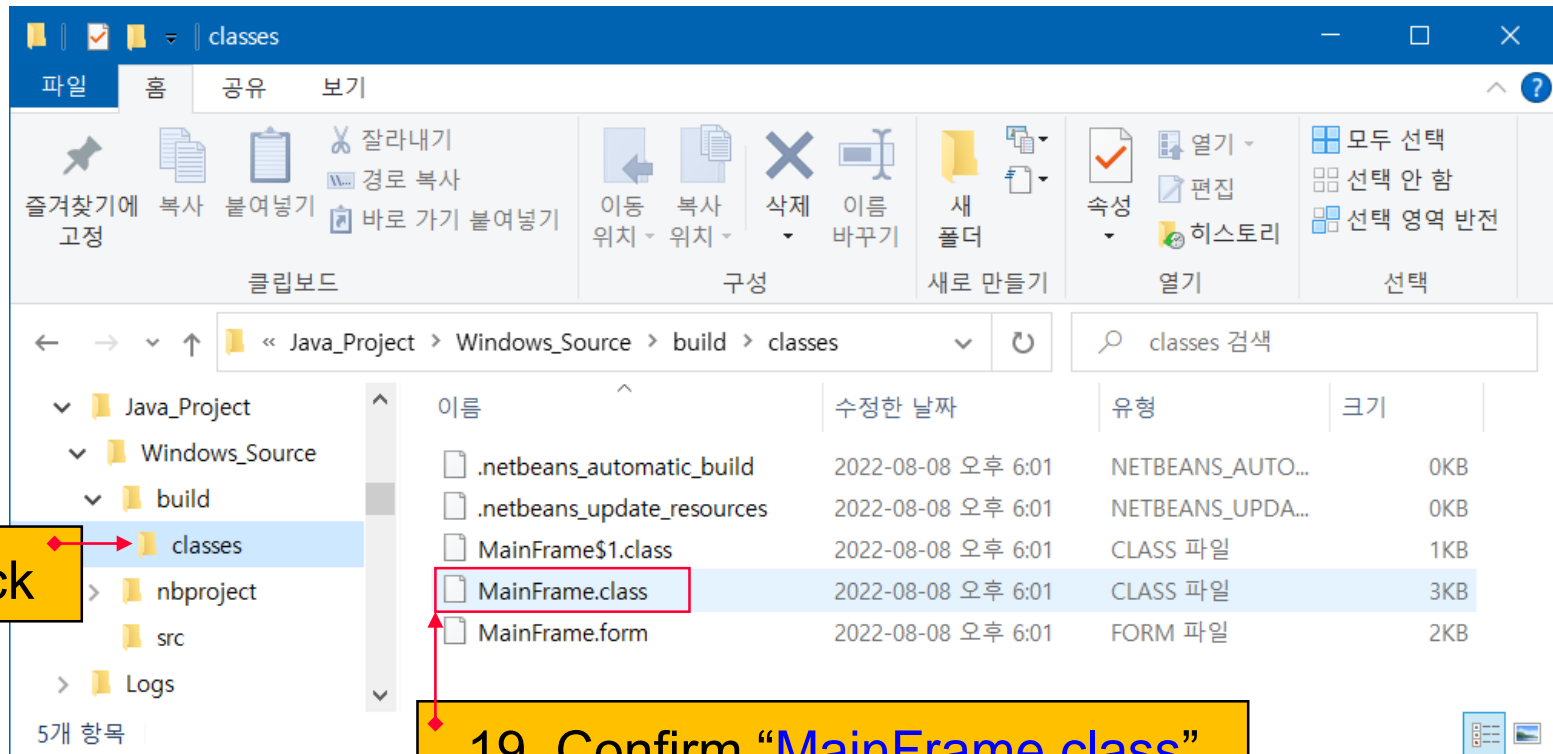
❖ Java source file





Windows Application 개발 방법 (12)

❖ Class file





Windows Application 소스 분석 (1)

MainFrame.java Source

```
MainFrame.java - Editor
MainFrame.java
Source Design History
1 ...4 lines
5
6 /**...4 lines */
10 public class MainFrame extends javax.swing.JFrame {
11
12     /** Creates new form MainFrame ...3 lines */
15     public MainFrame() {
16         initComponents();
17     }
18
19     /** This method is called from within the constructor to initialize the form
24     @SuppressWarnings("unchecked")
25     // <editor-fold defaultstate="collapsed" desc="Generated Code">
26     private void initComponents() { ...17 lines } // </editor-fold>
43
44     /**...3 lines */
47     public static void main(String args[]) {
48         /* Set the Nimbus look and feel */
49         Look and feel setting code (optional)
50
51         /* Create and display the form */
52         java.awt.EventQueue.invokeLater(new Runnable() {
53             public void run() {
54                 new MainFrame().setVisible(true);
55             }
56         });
57     }
58
59     // Variables declaration - do not modify
60     // End of variables declaration
61 }
```

MainFrame Constructor()

Component 초기화

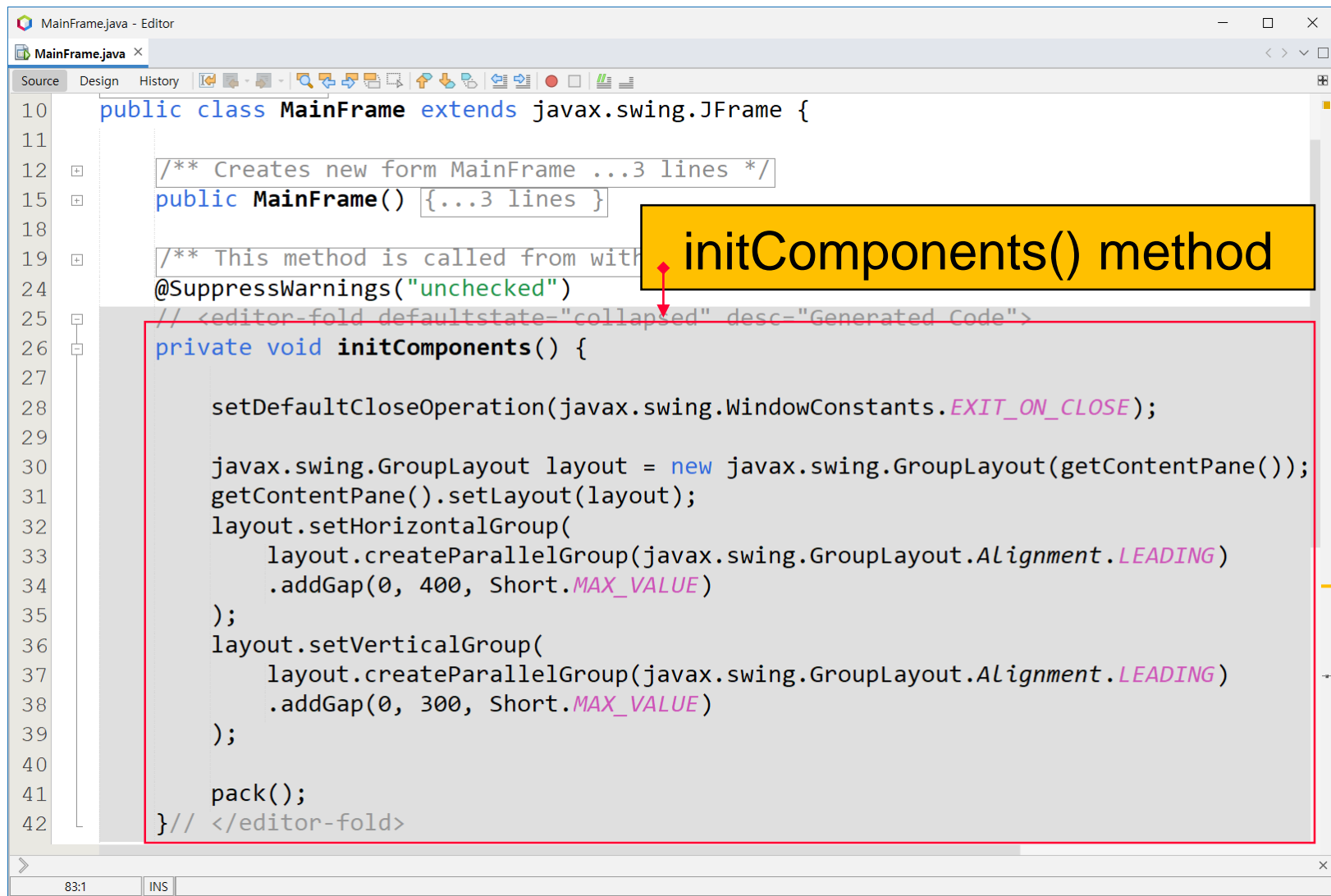
Main() method





Windows Application 소스 분석 (2)

❖ initComponents() 메소드



```
10 public class MainFrame extends javax.swing.JFrame {
11
12     /** Creates new form MainFrame ...3 lines */
15     public MainFrame() {...3 lines }
18
19     /** This method is called from within the IDE to initialize the form.
24     @SuppressWarnings("unchecked")
25     // <editor-fold defaultstate="collapsed" desc="Generated Code">
26     private void initComponents() {
27
28         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
29
30         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
31         getContentPane().setLayout(layout);
32         layout.setHorizontalGroup(
33             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
34                 .addGap(0, 400, Short.MAX_VALUE)
35         );
36         layout.setVerticalGroup(
37             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
38                 .addGap(0, 300, Short.MAX_VALUE)
39         );
40
41         pack();
42     } // </editor-fold>
```





Event란 ? (1)

■ Event

❖ Definition

◆ Message that occurs when a window component is clicked

❖ Event 처리 과정





Event 란 ? (2)

■ Event 구성

❖ Event Source

- ◆ Event를 발생시키는 Button, Scrollbar, Mouse, Keyboard,.. 등의 Component

❖ Event Class

- ◆ 특정 Component에 따라 발생하는 Event를 분류한 것

❖ Event Handler

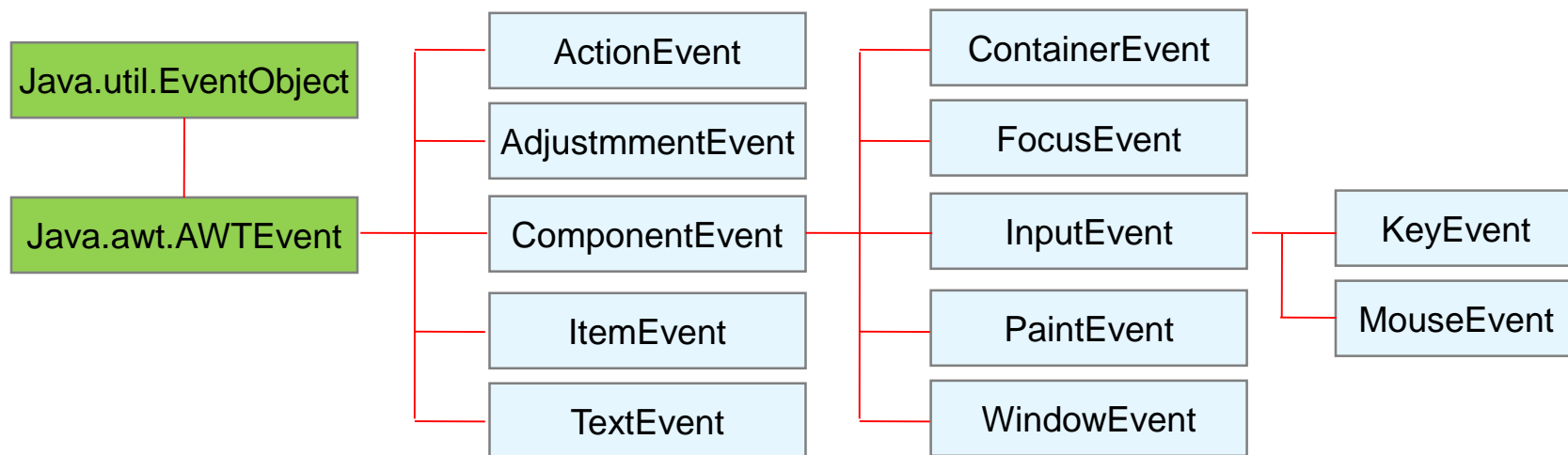
- ◆ Event 처리를 위한 Class





Event 란 ? (3)

■ Event Class 구조도





Event 란 ? (4)

■ Event Class

Event Class	Comment
ActionEvent	Component가 활성화될 때 발생
AdjustmentEvent	스크롤바와 같이 조정 가능한 Component에서 조정이 있을 때 발생
ContainerEvent	Container에 Component가 추가/삭제되는 경우 발생
FocusEvent	Component에 focus가 들어왔을 때 발생
ItemEvent	List, choice, .. 등의 Component에서 선택항목이 선택될 때 발생
KeyEvent	키보드 입력에 의해서 발생
MouseEvent	Mouse 움직임에 의해서 발생
PaintEvent	Component가 그려져야 할 때 발생
TextEvent	Text Component의 내용이 변화할 때 발생
WindowEvent	Window 활성화, 또는 종료할 때 발생





Event란 ? (5)

■ Relationship Component-Event

Component	Event
Adjustable	AdjustmentEvent
Applet	ContainerEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Button	ActionEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Canvas	FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Checkbox	ItemEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
CheckboxMenuItem	ActionEvent, ItemEvent
Choice	ItemEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Component	FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Container	ContainerEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Dialog	ContainerEvent, WindowEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent





Event 관 ? (6)

Component	Event
FileDialog	ContainerEvent, WindowEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Frame	
Label	FocusEvent, KeyEvent, MouseEvent, ComponentEvent
List	ItemEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Menu	ActionEvent
MenuItem	
PopupMenu	
Panel	ContainerEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Scrollbar	AdjustmentEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
ScrollPane	ContainerEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent





Event 란 ? (7)

Component	Component에서 발생하는 Event
TextArea	ContainerEvent, WindowEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
TextComponent	
TextField	FocusEvent, TextEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent
Window	ContainerEvent, WindowEvent, FocusEvent, KeyEvent, MouseEvent, ComponentEvent





Event Handler (1)

■ Event Handler 구성

❖ Listener Interface

◆ 각 Event를 처리하기 위해, 준비된 method를 선언한 Interface

◆ Event + Listener

- ActionEvent = ActionListener, FocusEvent=FocusListener

❖ 예제

```
class MyActionListener implements ActionListener
{
    public void actionPerformed(ActionEvent ae)
    {
        System.out.println("Action Event가 발생했습니다.");
    }
}
```





Event Handler (2)

Listener Interface Adapter	Method
ActionListener	actionPerformed(ActionEvent)
AdjustmentListener	adjustmentValueChanged(AdjustmentEvent)
ComponentListener ComponentAdapter	componentHidden(ComponentEvent) componentShown(ComponentEvent) componentMoved(ComponentEvent) componentResized(ComponentEvent)
ContainerListener ContainerAdapter	componentAdded(ContainerEvent) componentRemoved(ContainerEvent)
FocusListener FocusAdapter	focusGained(FocusEvent) focusLost(FocusEvent)





Event Handler (3)

Listener Interface Adapter	Method
ItemListener	ItemStateChanged(ItemEvent)
KeyListener KeyAdapter	KeyPressed(KeyEvent) KeyReleased(KeyEvent) KeyTyped(KeyEvent)
MouseListener MouseAdapter	mouseClicked(MouseEvent) mouseEntered(MouseEvent) mouseExited(MouseEvent) mousePressed(MouseEvent) mouseReleased(MouseEvent)
MouseMotionListener MouseMotionAdapter	mouseDragged(MouseEvent) mouseMoved(MouseEvent)
TextListener	textValueChanged(TextEvent)





Event Handler (4)

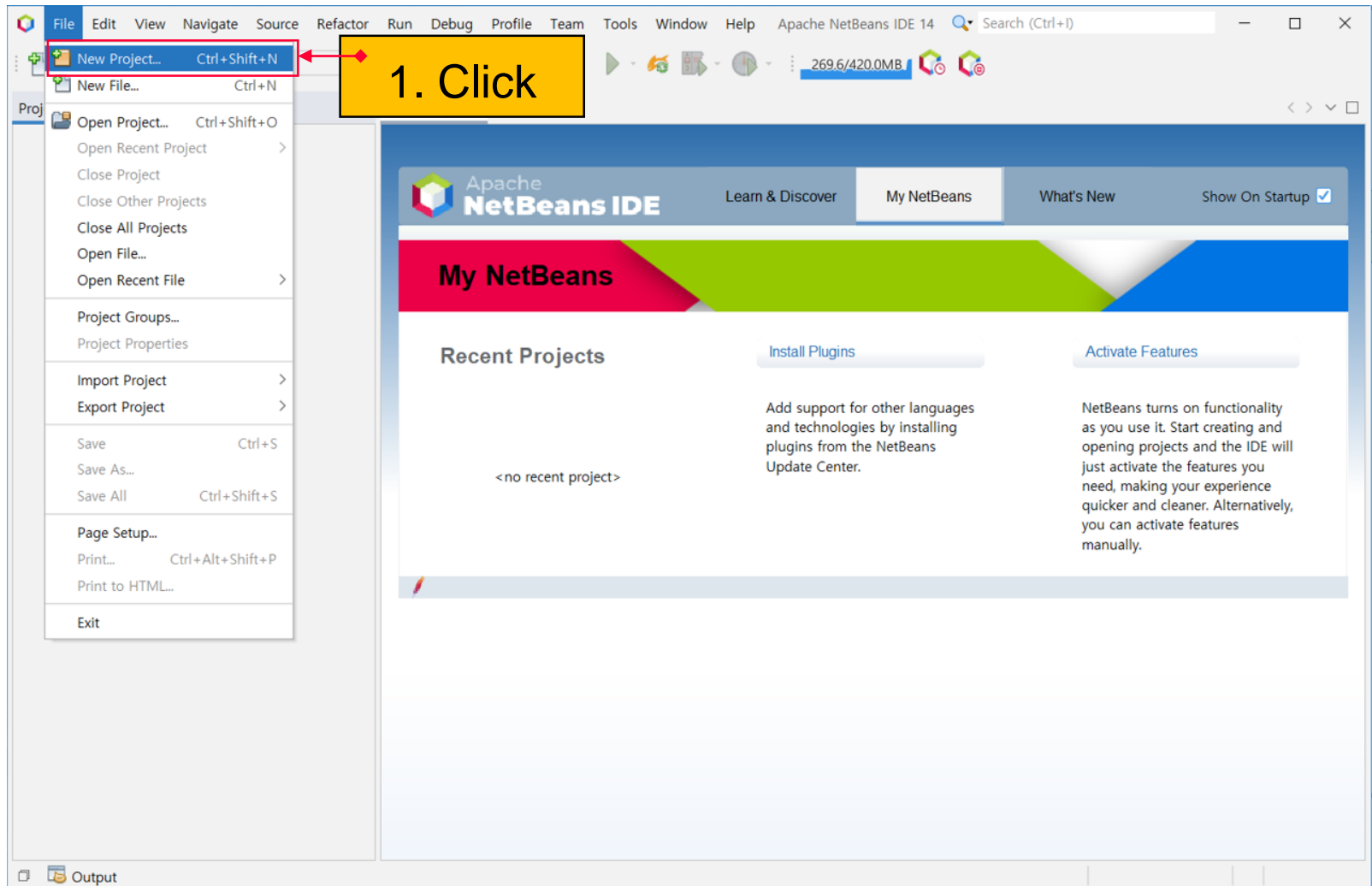
Listener Interface Adapter	Method
WindowListener WindowAdapter	windowOpened(WindowEvent) windowClosing(WindowEvent) windowClosed(WindowEvent) windowActivated(WindowEvent) windowDeactivated(WindowEvent) windowConified(WindowEvent) windowDeconified(WindowEvent)





Practice : Event Handler 구현 (1)

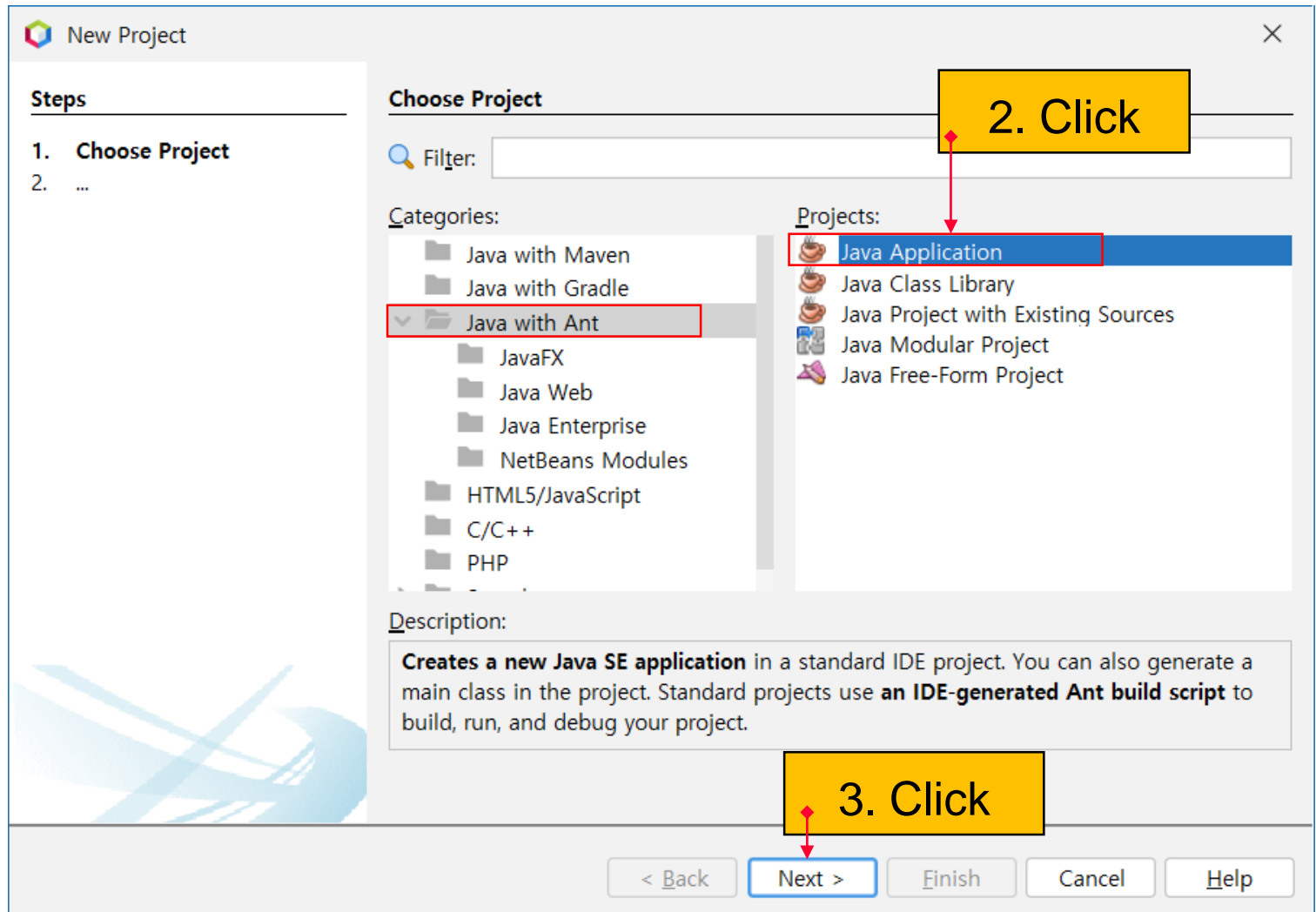
Create Project





Practice : Event Handler 구현 (2)

Choose Project





Practice : Event Handler 구현 (3)

■ Project Name and Location

❖ Project name: `EventHandler_Source`

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

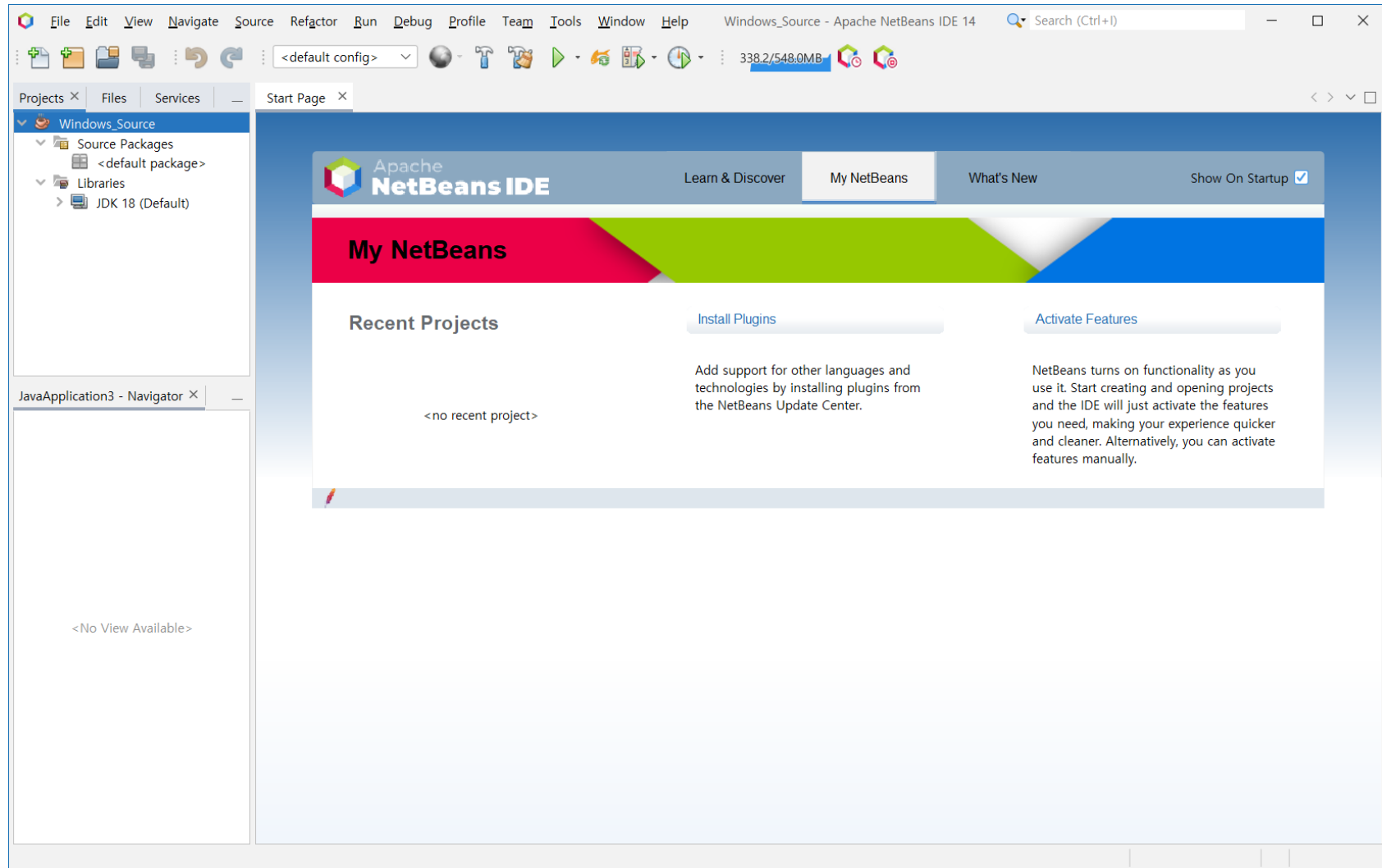
☒ Create Main Class





Practice : Event Handler 구현 (4)

Create project success





Practice : Event Handler 구현 (5)

■ Create JFrame Form

7. Mouse right-button Click

8. Click

9. Click

Projects ×

- ▼ **EventHandler_Source**
- ▼ Source Packages
 - <default package>
- ▼ Libraries
 - > JDK 18 (Default)

New

- Build
- Clean and Build
- Run
- Debug
- Profile
- Test Alt+F6
- Set Configuration
- Set as Main Project
- Open Required Projects
- Close
- Rename...
- Move...
- Copy...
- Delete Delete
- Find... Ctrl+F
- Versioning
- Local History
- Properties

JFrame Form...

- Java Class...
- Java Package...
- Java Module...
- Java Project...
- JSP...
- Entity Class...
- Entity Classes from Database...
- Other...





Practice : Event Handler 구현 (6)

■ Setting Class Name

❖ Mainframe.java 생성

New JFrame Form

Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name: MainFrame

Project: EventHandler_Source

Location: Source Packages

Package:

Created File: C:\Java_Project\EventHandler_Source\src\MainFrame.java

Projects

- ▼ EventHandler_Source
 - ▼ Source Packages
 - ▼ <default package>
 - ▼ MainFrame.java
 - ▼ Libraries
 - > JDK 18 (Default)

10. Input "MainFrame"

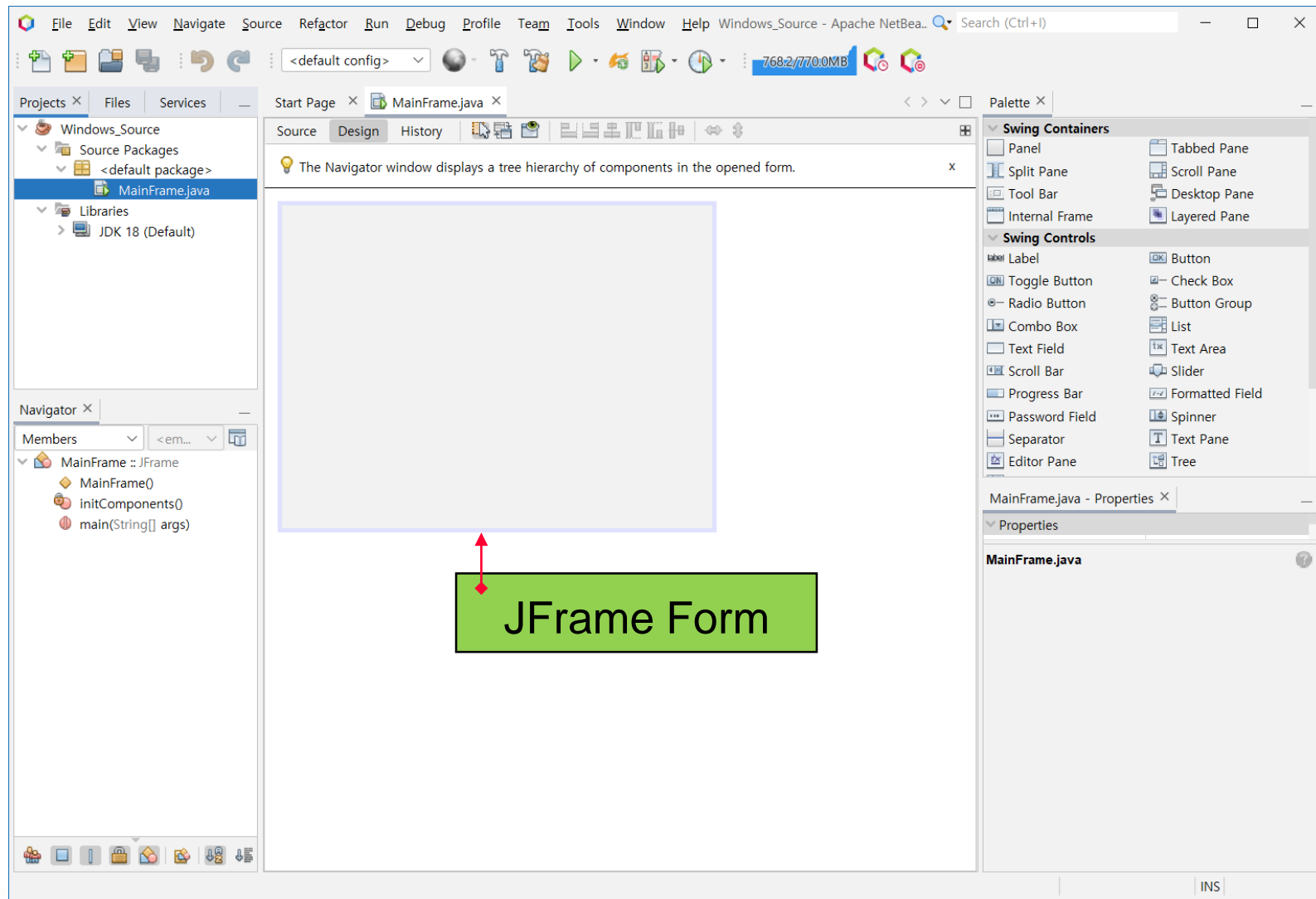
11. Click

Finish



Practice : Event Handler 구현 (7)

Create JFrame form success





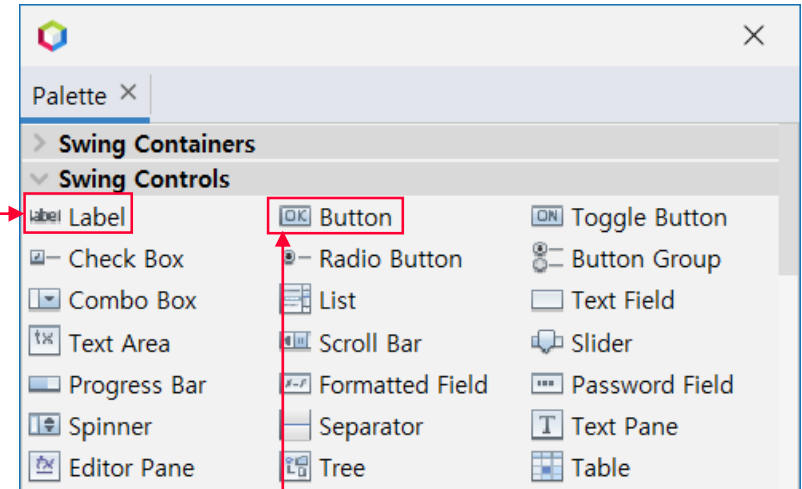
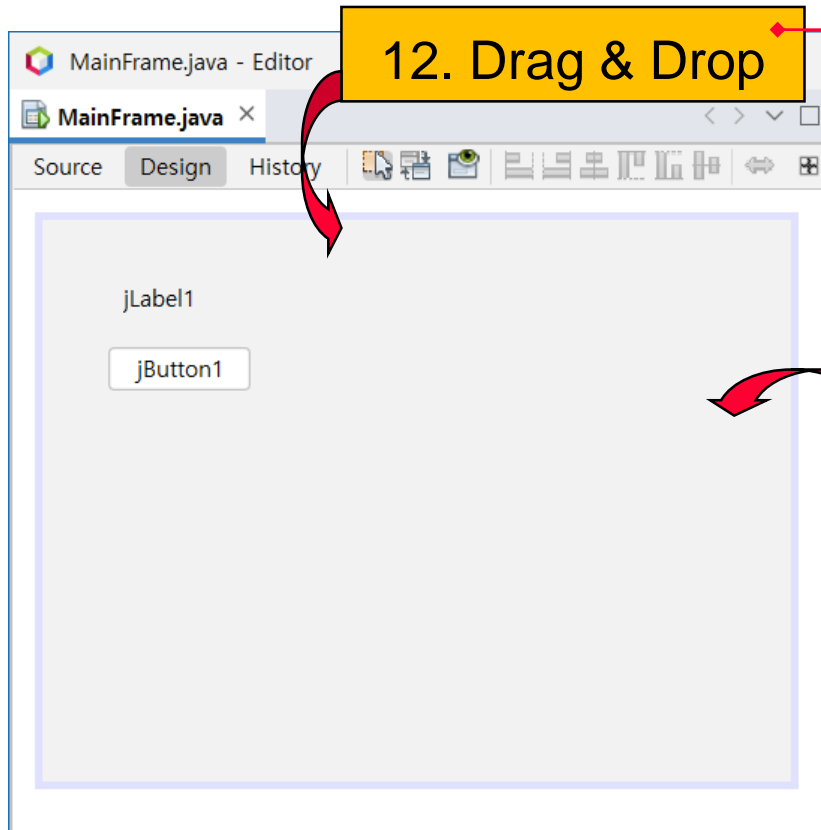
Practice : Event Handler 구현 (8)

■ GUI 구현

❖ Layout

◆ Label

◆ Button



13. Drag & Drop





Practice : Event Handler 구현 (9)

❖ MainFrame.java Source Code

```
MainFrame.java - Editor
MainFrame.java
Source Design History
1
2 public class MainFrame extends javax.swing.JFrame {
3
4     public MainFrame() {
5         initComponents();
6     }
7
8     @SuppressWarnings("unchecked")
9     Generated Code
47
48 public static void main(String args[]) {
49
50     /* Create and display the form */
51     java.awt.EventQueue.invokeLater(new Runnable() {
52         public void run() {
53             new MainFrame().setVisible(true);
54         }
55     });
56 }
57
58 // Variables declaration - do not modify
59 private javax.swing.JButton jButton1;
60 private javax.swing.JLabel jLabel1;
61 // End of variables declaration
62 }
```

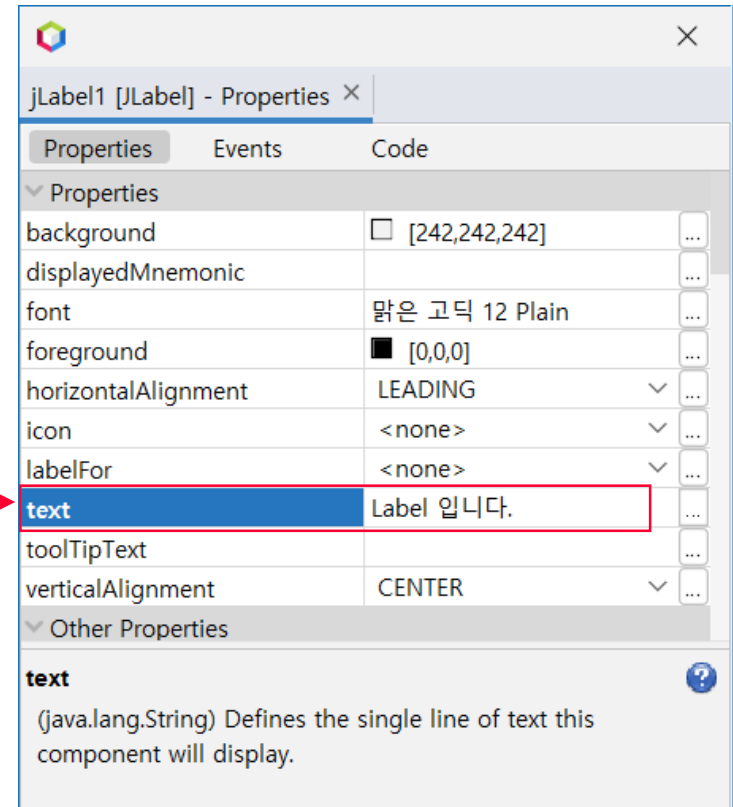
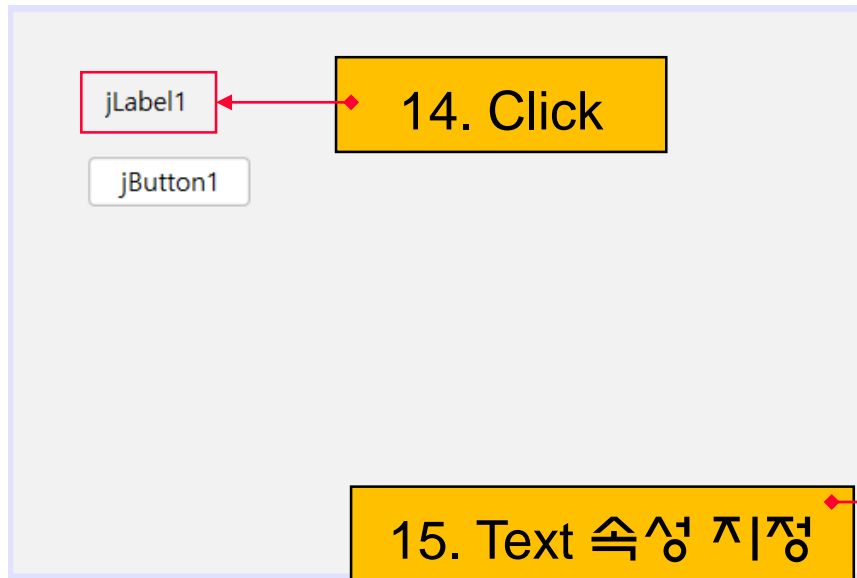
Label , Button 변수 선언





Practice : Event Handler 구현 (10)

Control Layout & Property Setting



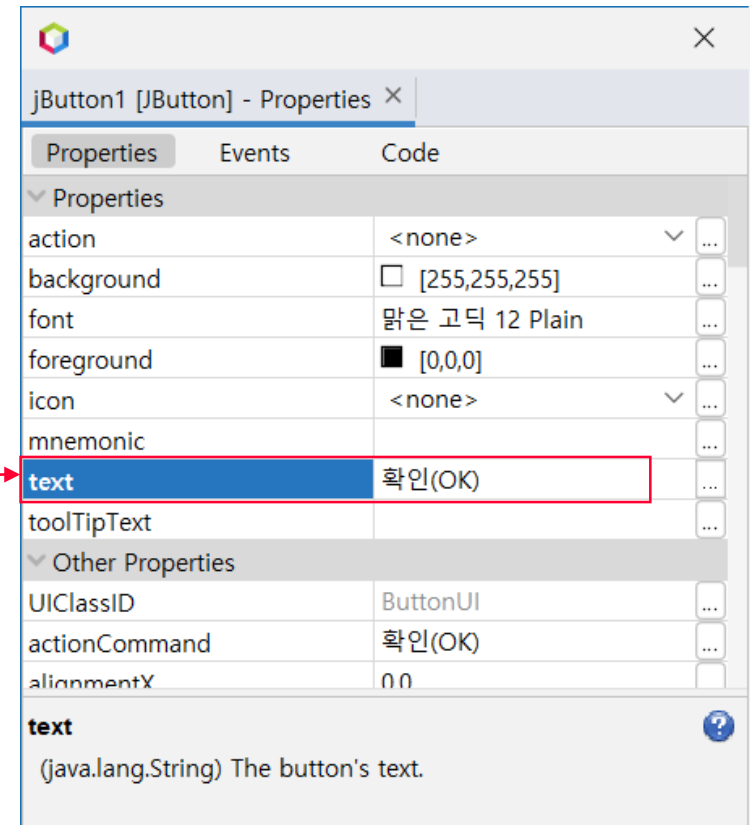
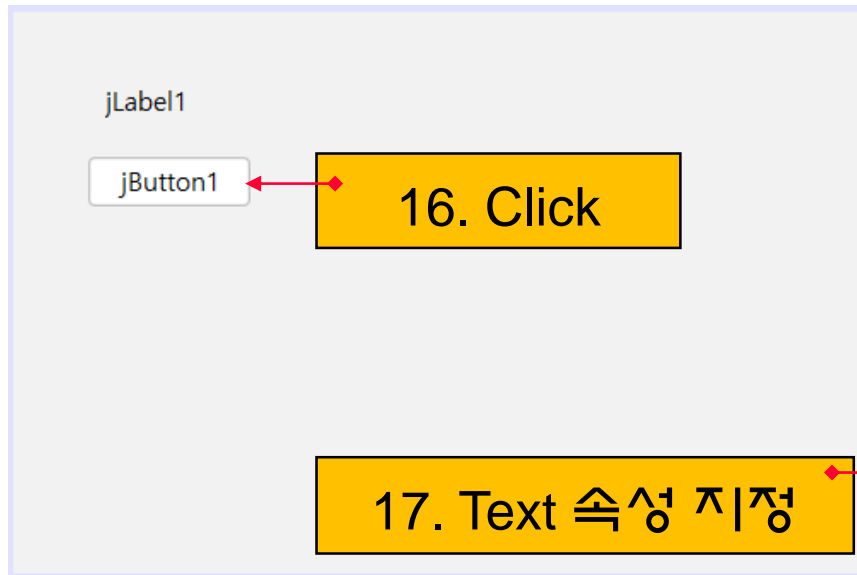
Control	Property Setting
Label	•Text: Label 입니다.





Practice : Event Handler 구현 (11)

❖ Button Control Property Setting



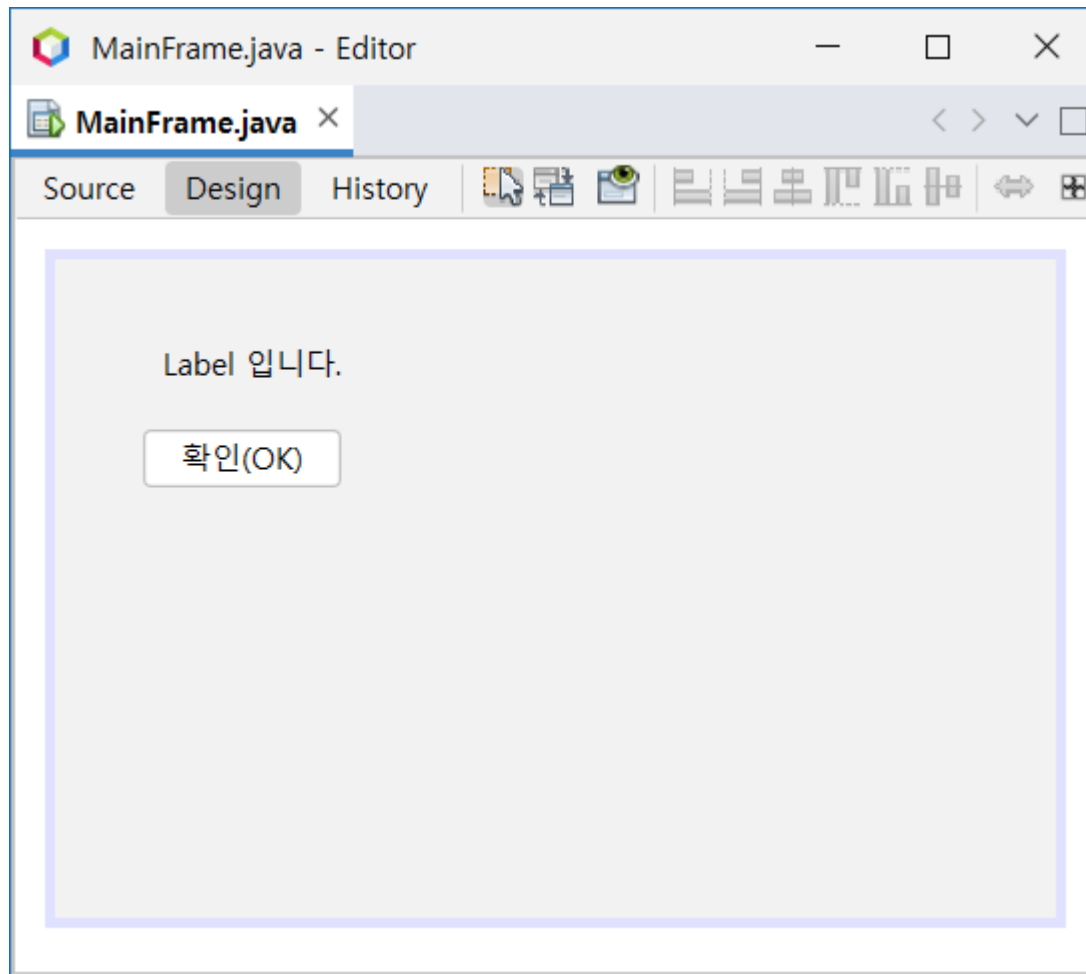
Control	Property Setting
Button	•Text: 확인(OK)





Practice : Event Handler 구현 (12)

■ GUI 구현 완료





Practice : Event Handler 구현 (13)

❖ Label 및 Button 속성 지정에 따른 Source Coding

```
MainFrame.java - Editor
MainFrame.java x
Source Design History
1
2 public class MainFrame extends javax.swing.JFrame {
3
4     public MainFrame() {
5         initComponents();
6     }
7
8     @SuppressWarnings("unchecked")
9     // <editor-fold defaultstate="collapsed" desc="Generated Code">
10    private void initComponents() {
11
12        jLabel1 = new javax.swing.JLabel();
13        jButton1 = new javax.swing.JButton();
14
15        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
16
17        jLabel1.setText("Label 입니다.");
18        jButton1.setText("확인(OK)");
19    }
20 }
```

Create Label , Button instance

Setting Label , Button property





Practice : Event Handler 구현 (14)

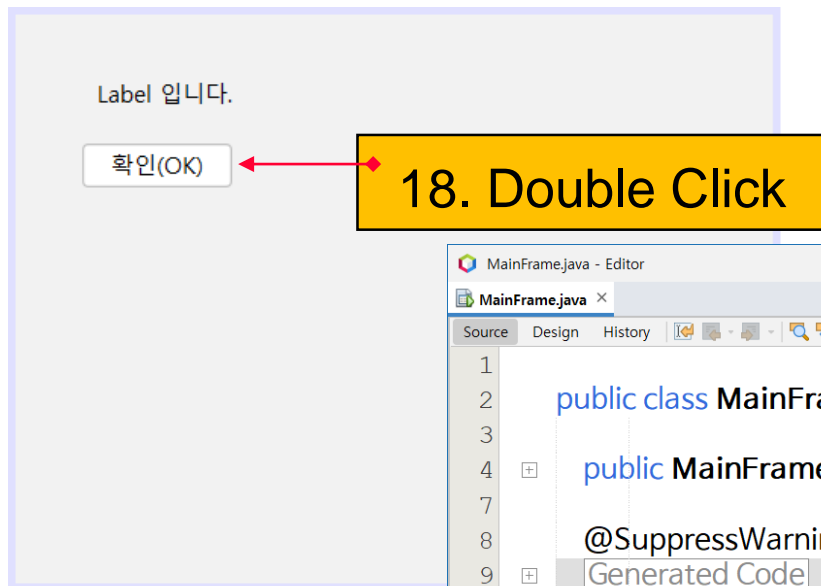
```
MainFrame.java - Editor
MainFrame.java
Source Design History
20
21 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
22 getContentPane().setLayout(layout);
23 layout.setHorizontalGroup(
24     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
25     .addGroup(layout.createSequentialGroup()
26         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
27             .addGroup(layout.createSequentialGroup()
28                 .addGap(43, 43, 43)
29                 .addComponent(jLabel1))
30             .addGroup(layout.createSequentialGroup()
31                 .addGap(35, 35, 35)
32                 .addComponent(jButton1)))
33         .addContainerGap(286, Short.MAX_VALUE))
34     );
35 layout.setVerticalGroup(
36     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
37     .addGroup(layout.createSequentialGroup()
38         .addGap(33, 33, 33)
39         .addComponent(jLabel1)
40         .addGap(18, 18, 18)
41         .addComponent(jButton1)
42         .addContainerGap(209, Short.MAX_VALUE))
43     );
44
45 pack();
46 }// </editor-fold>
```



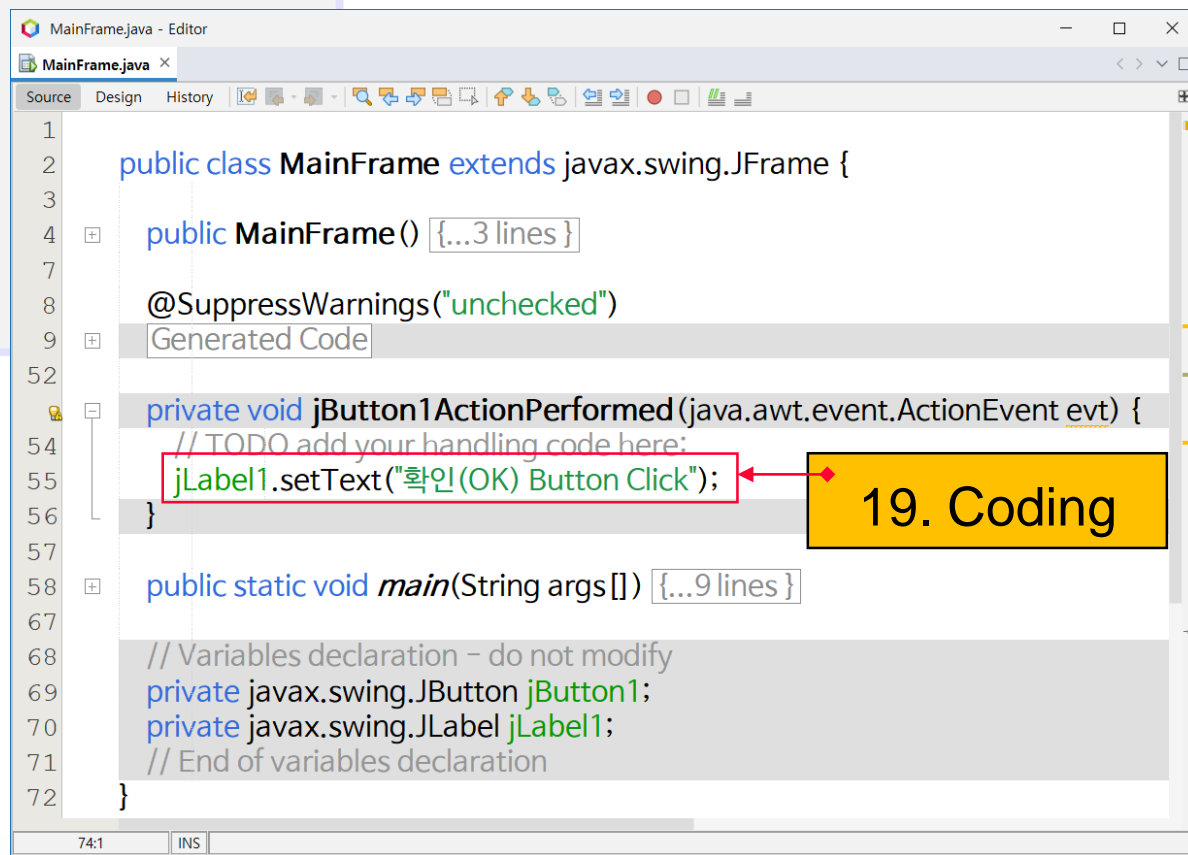


Practice : Event Handler 구현 (15)

■ [확인(OK)] Button Event Handler 구현



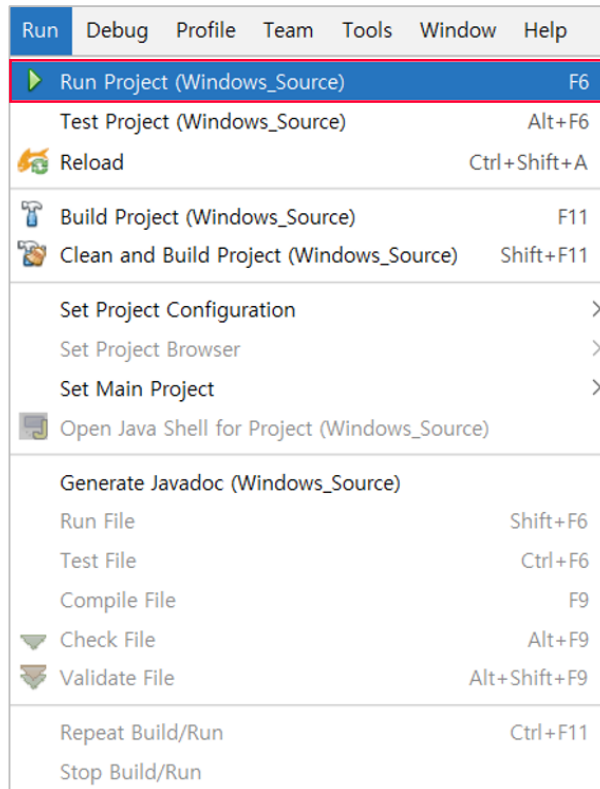
18. Double Click



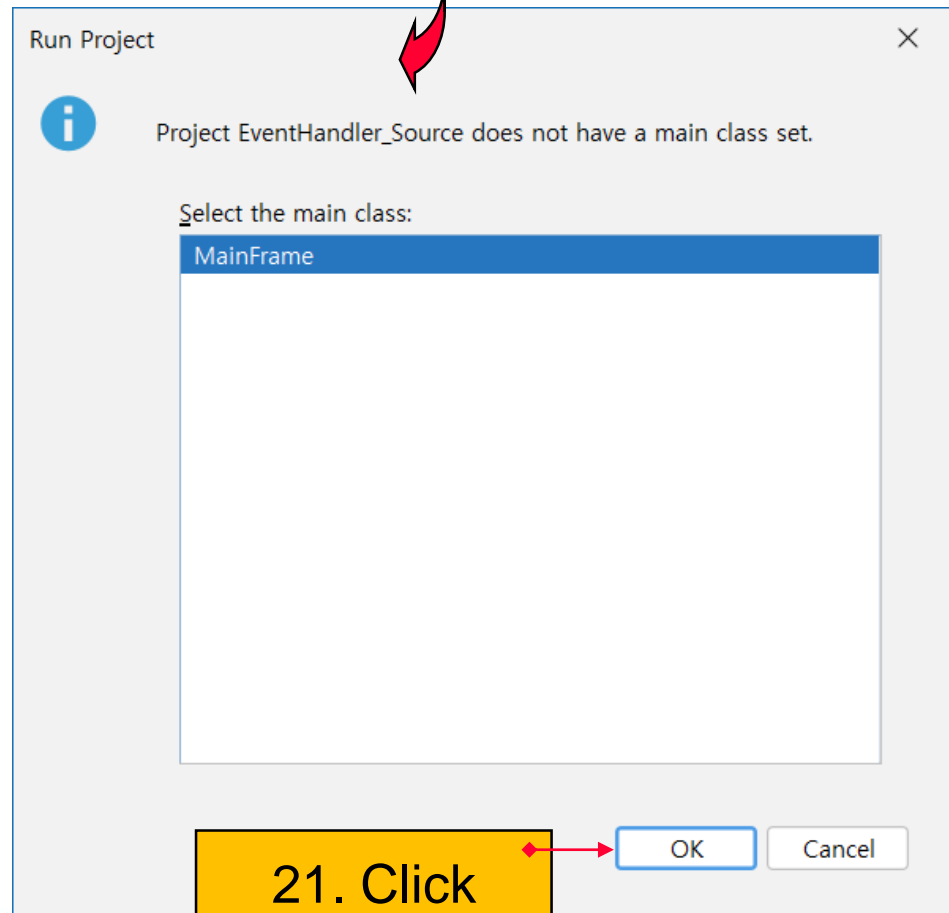


Practice : Event Handler 구현 (16)

Run



20. Click



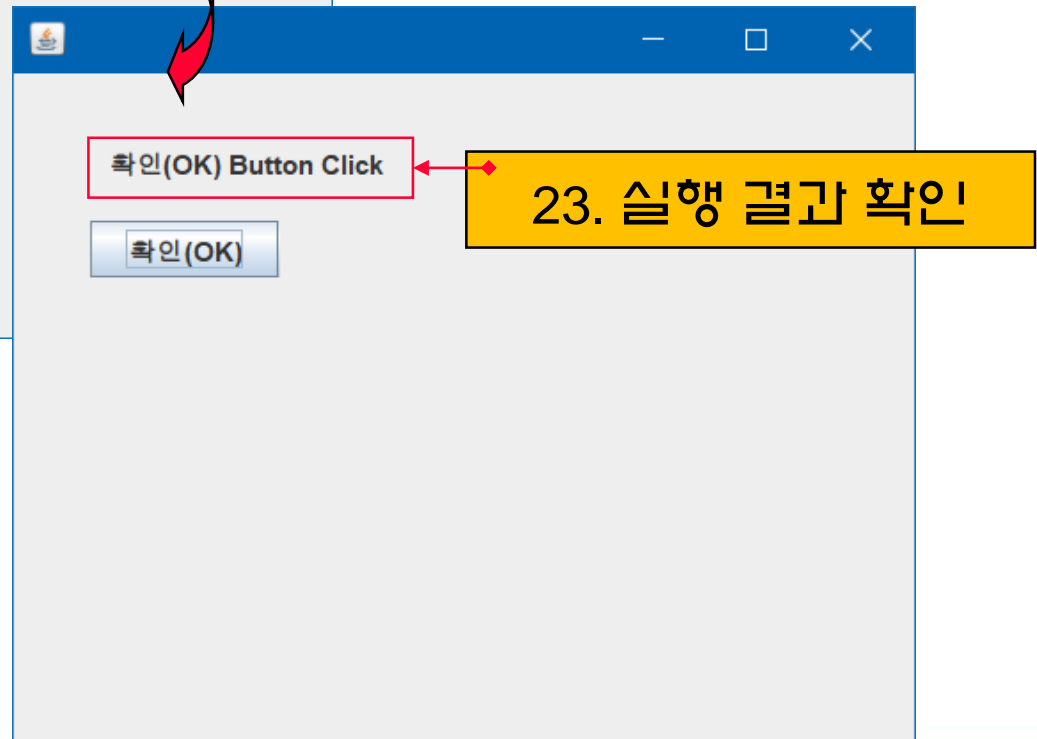
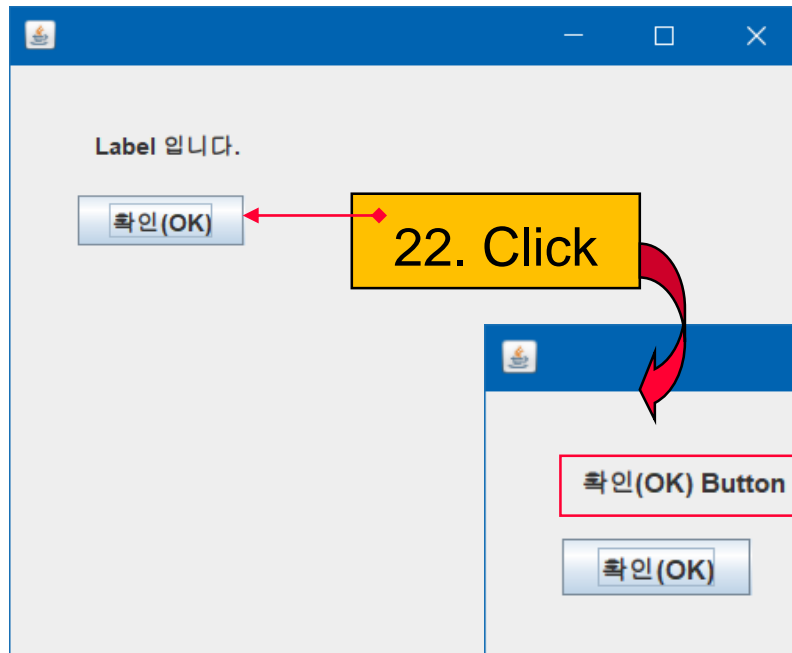
21. Click





Practice : Event Handler 구현 (17)

■ Run

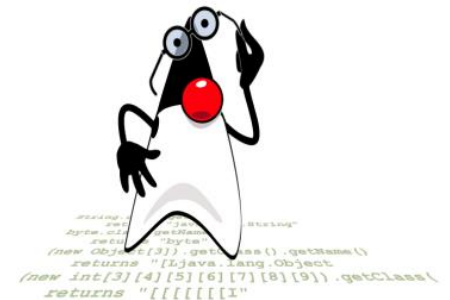
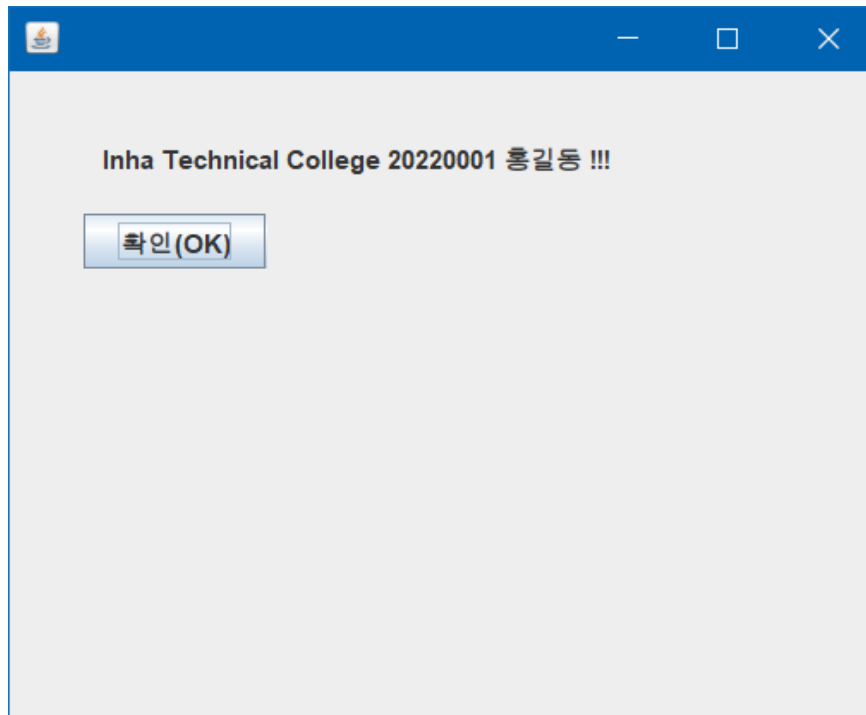




Practice : Event Handler 구현

❖ Event Handler 구현(Time: 20 min)

- Project Name : Student_Source
- 확인(OK) 버튼 Click시 아래 그림과 같이 출력하도록 프로그래밍 하시오.





학습 요약

- ❖ AWT(Abstract Windows Toolkit)
- ❖ Swing
- ❖ Window 프로그래밍 방법
- ❖ Event
- ❖ Event Handler
- ❖ Event Handler 구현 방법

