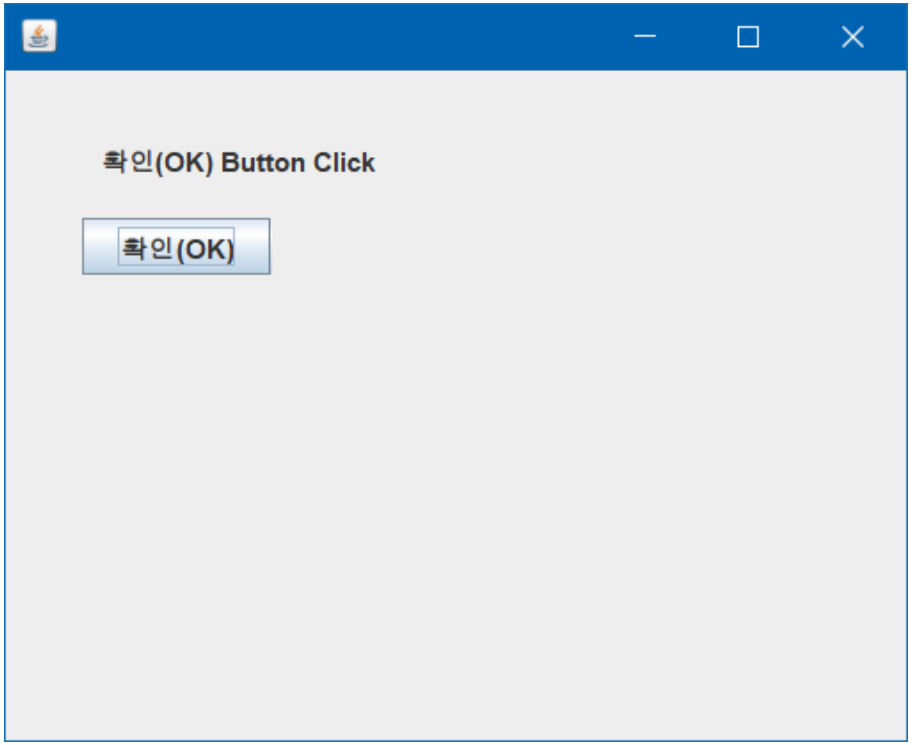
Windows 프로그래밍



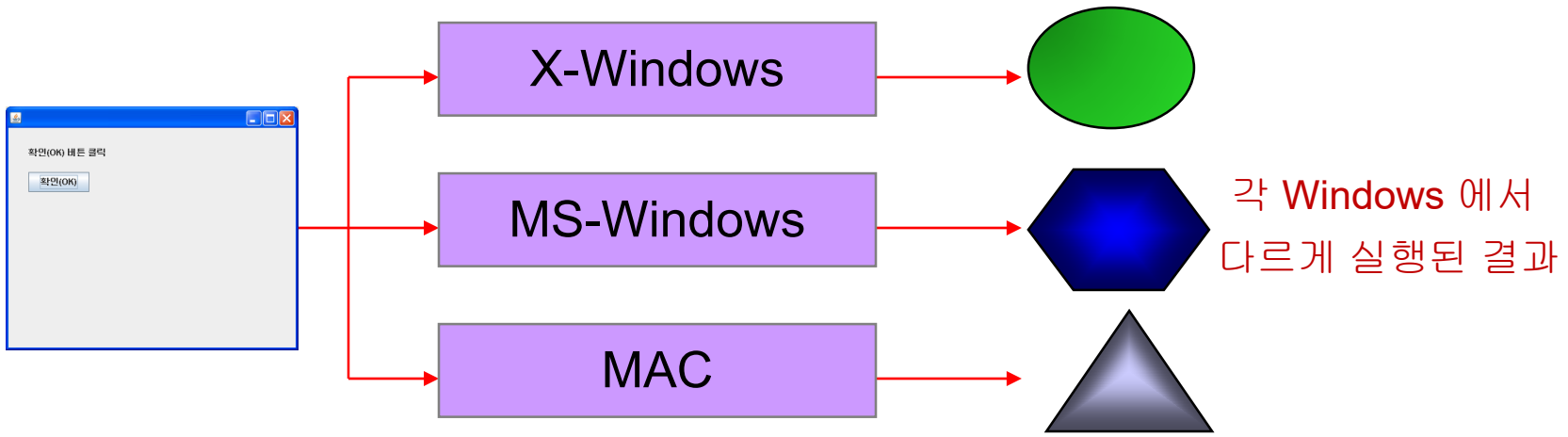
학습 목표

■ 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)
  - ❖ 정의
    - ◆ GUI를 구축하기 위한 클래스들의 모음
  - ❖ 종류
    - ◆ 사용자 Interface 클래스
    - ◆ 그래픽 처리 클래스
  - ❖ 특징

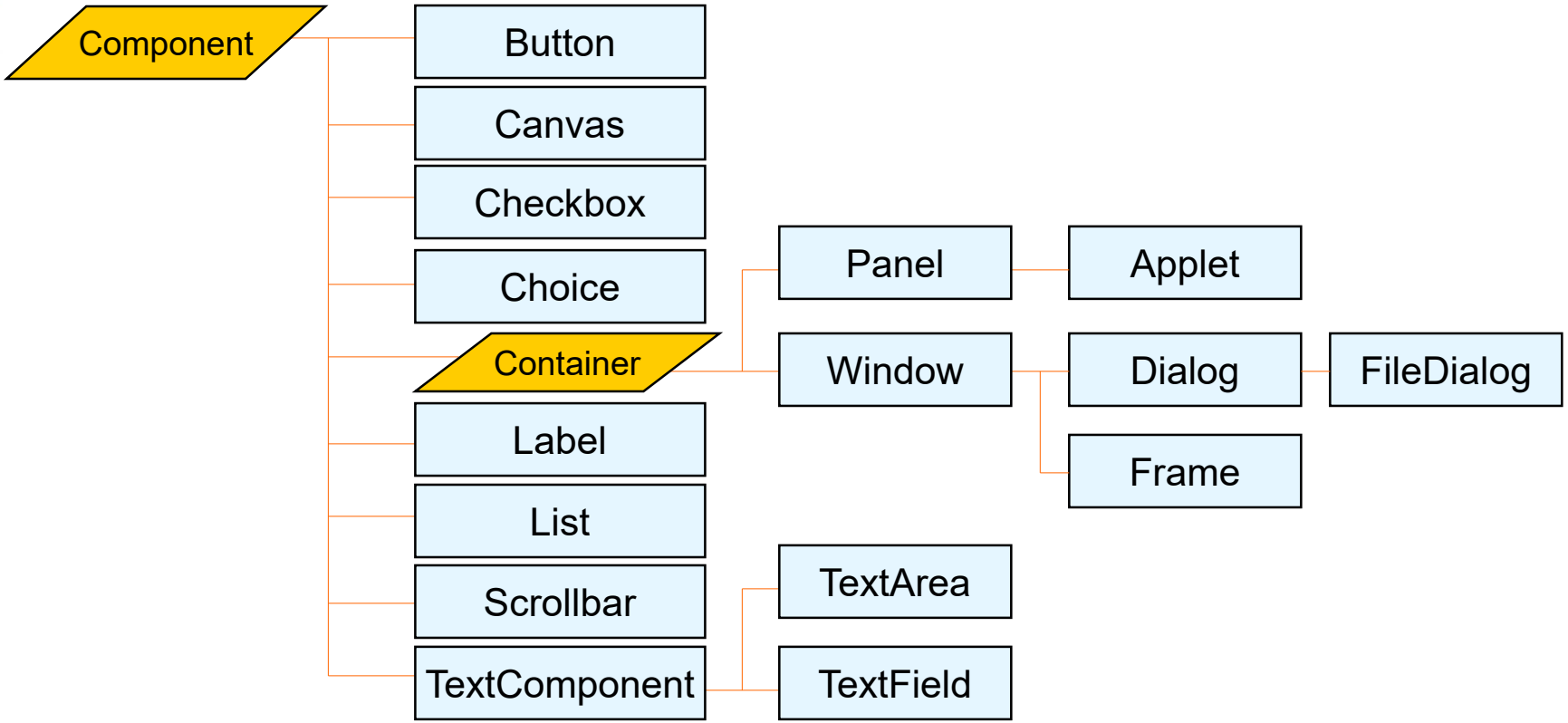


AWT 란 ? (2)

AWT의 클래스 구조

❖ Component(Component)

- ◆ Window 환경의 Component(Button, CheckBox,…)
- ◆ java.awt 클래스로 객체 모델링



AWT 란 ? (3)

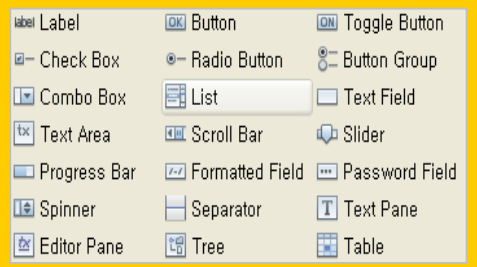
Method

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

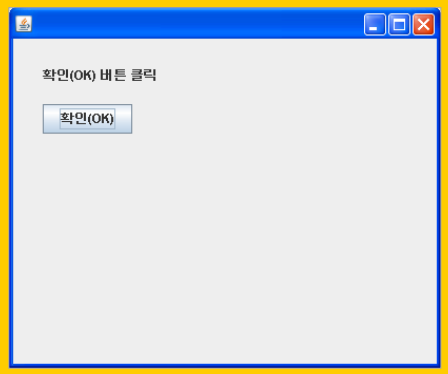
AWT 란 ? (4)

Visual Component와 Container 관계

Component 생성



컨테이너 Layout



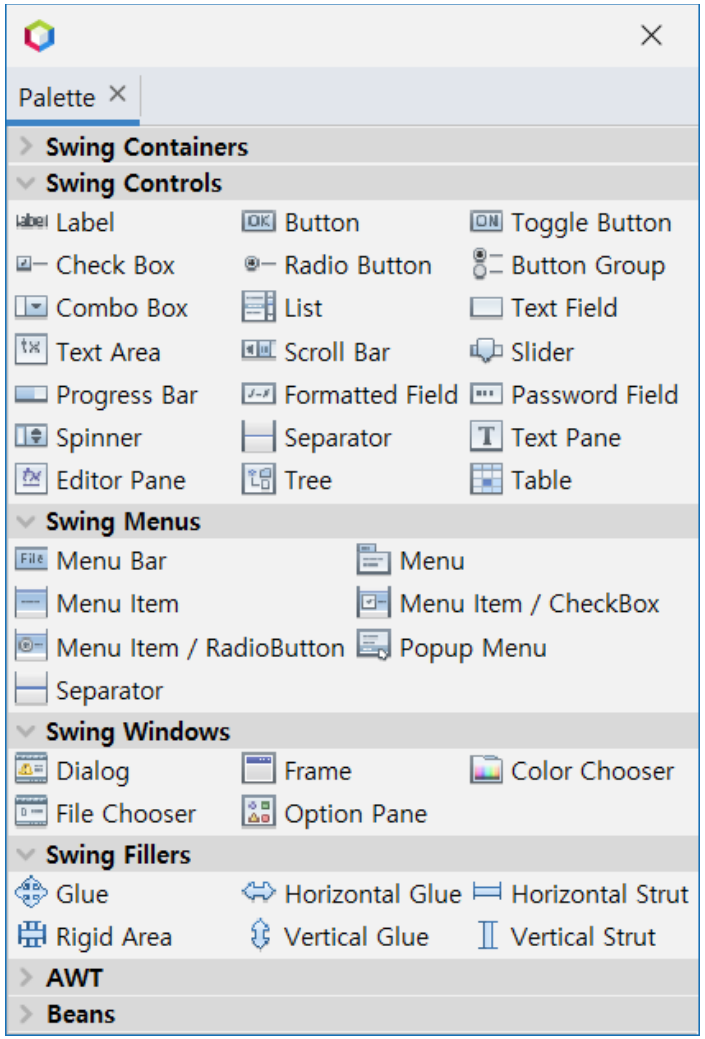
Event Handler 작성

```
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	기능
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

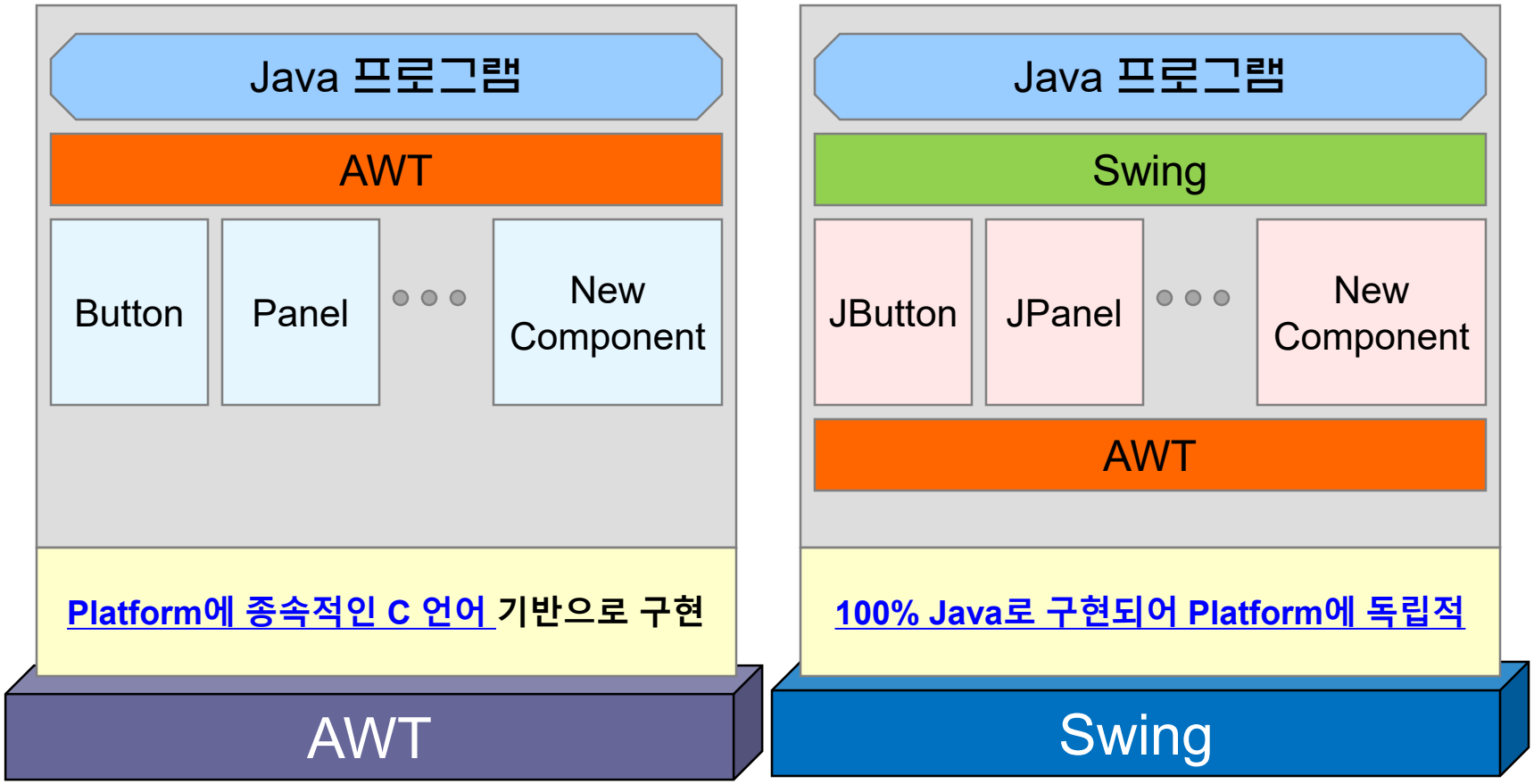


Swing 이란 ? (1)

- Swing
  - ❖ 정의
    - ◆ 순수한 자바 언어로 지원되는 GUI 개발 도구
  - ❖ 특징
    - ◆ Platform에 독립적이다.
    - ◆ 새로운 Component 제작이 쉽다.
    - ◆ 풍선도움말(ToolTip) 기능을 제공한다.
    - ◆ AWT Component를 지원한다.

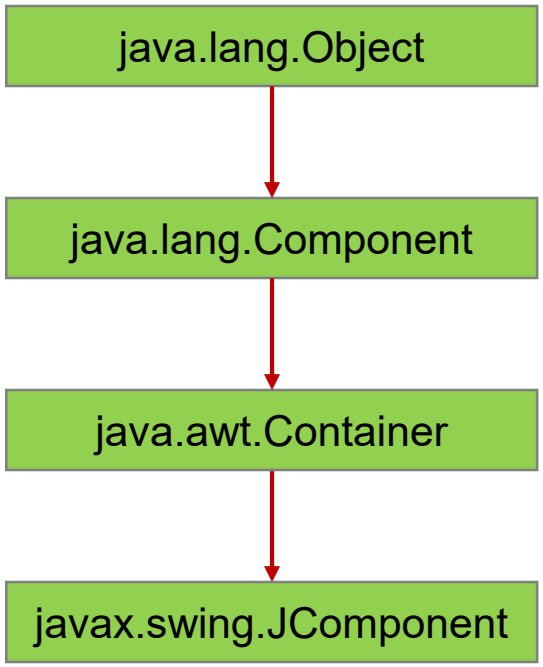
Swing 이란 ? (2)

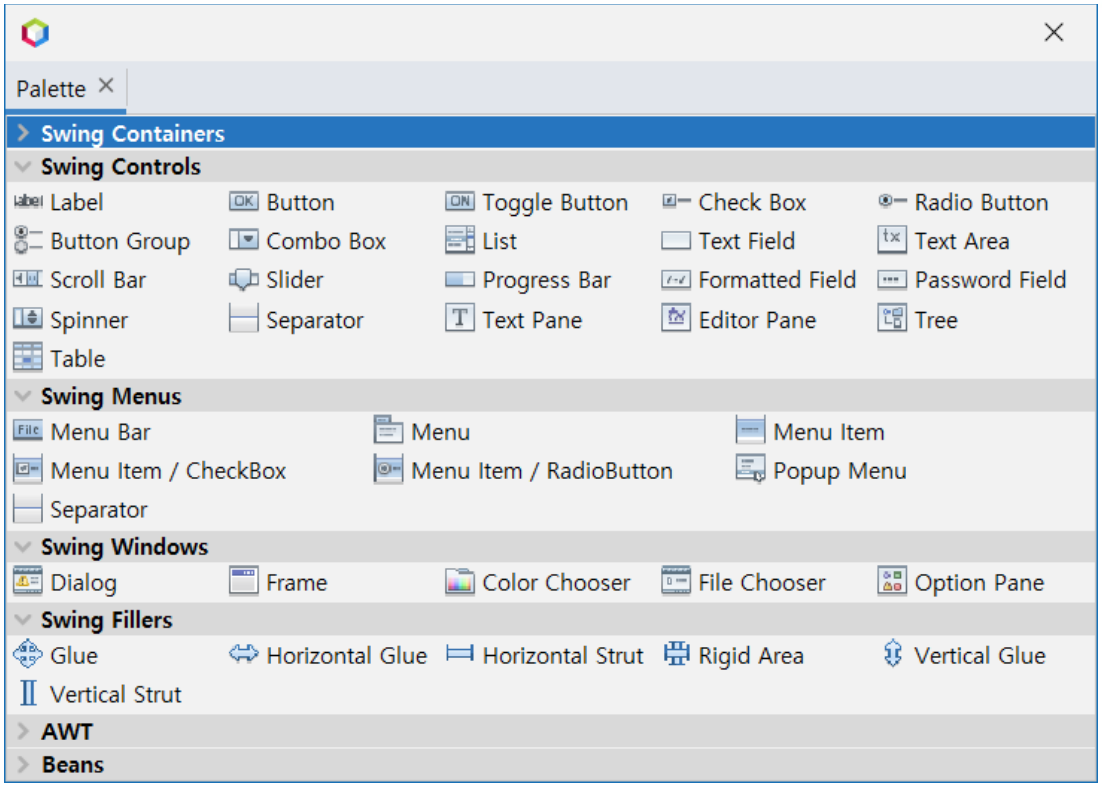
- AWT vs. Swing



Swing 이란 ? (3)

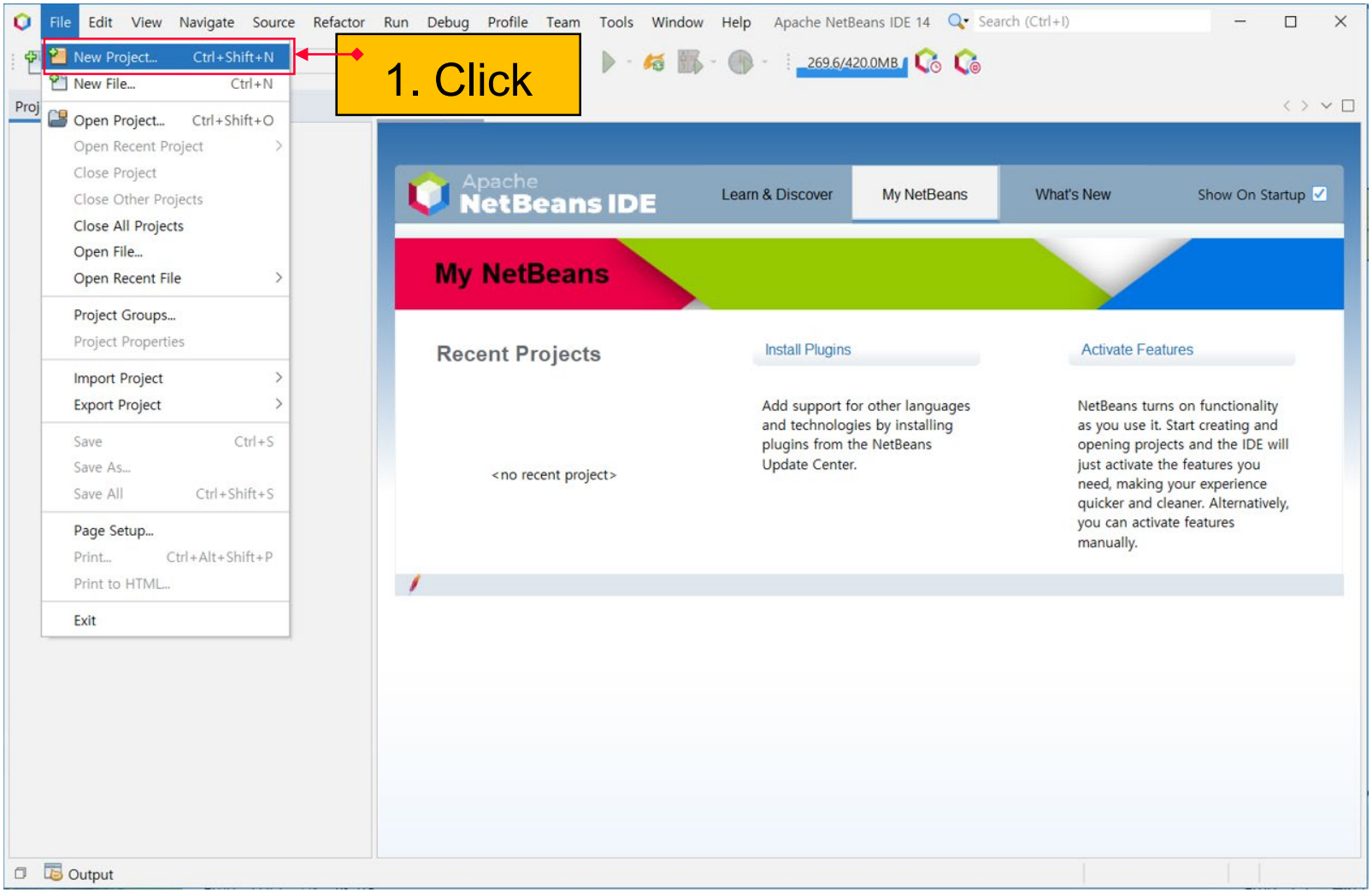
Swing의 Class 구조





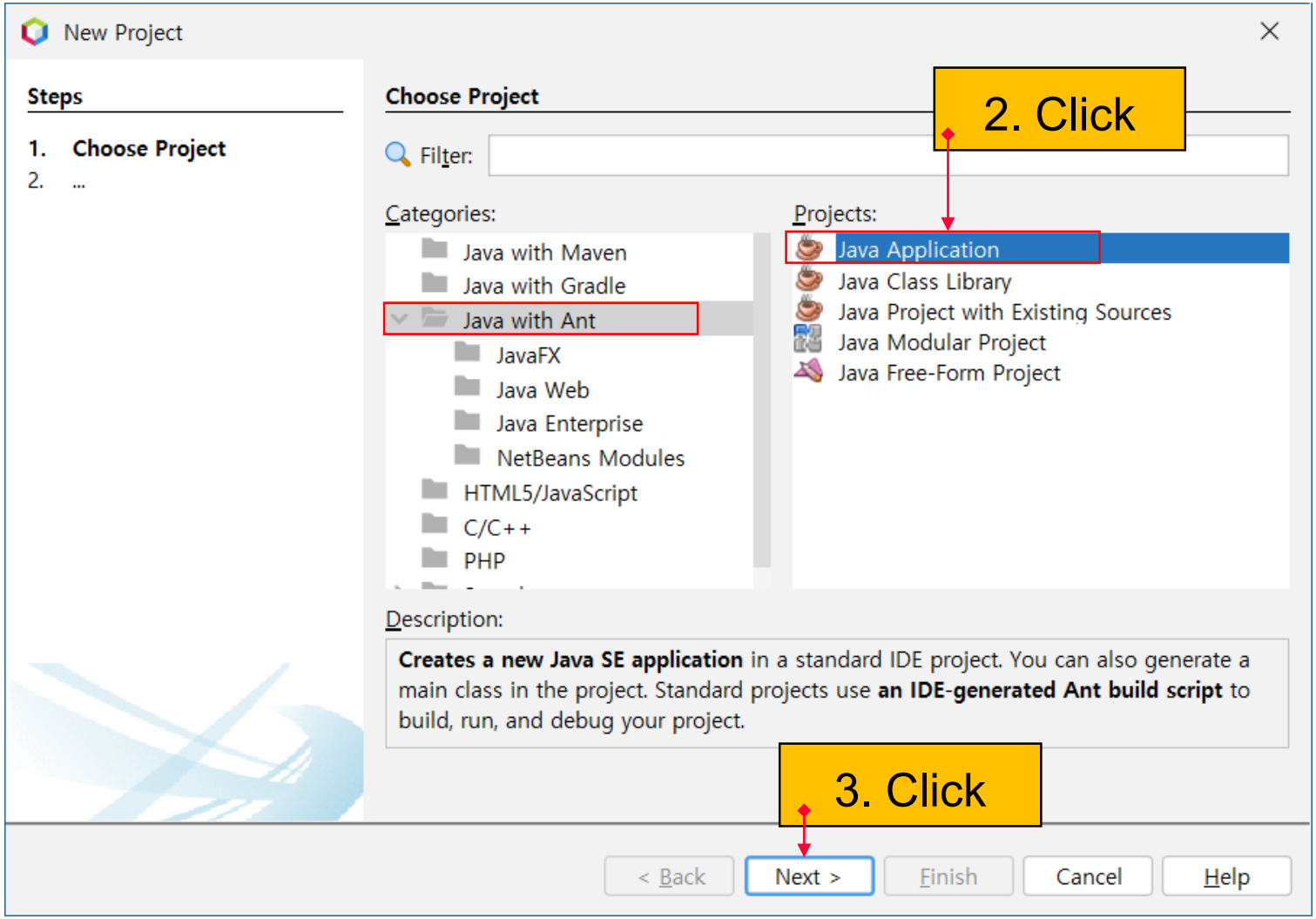
Windows Application 개발 방법 (1)

Create Project



Windows Application 개발 방법 (2)

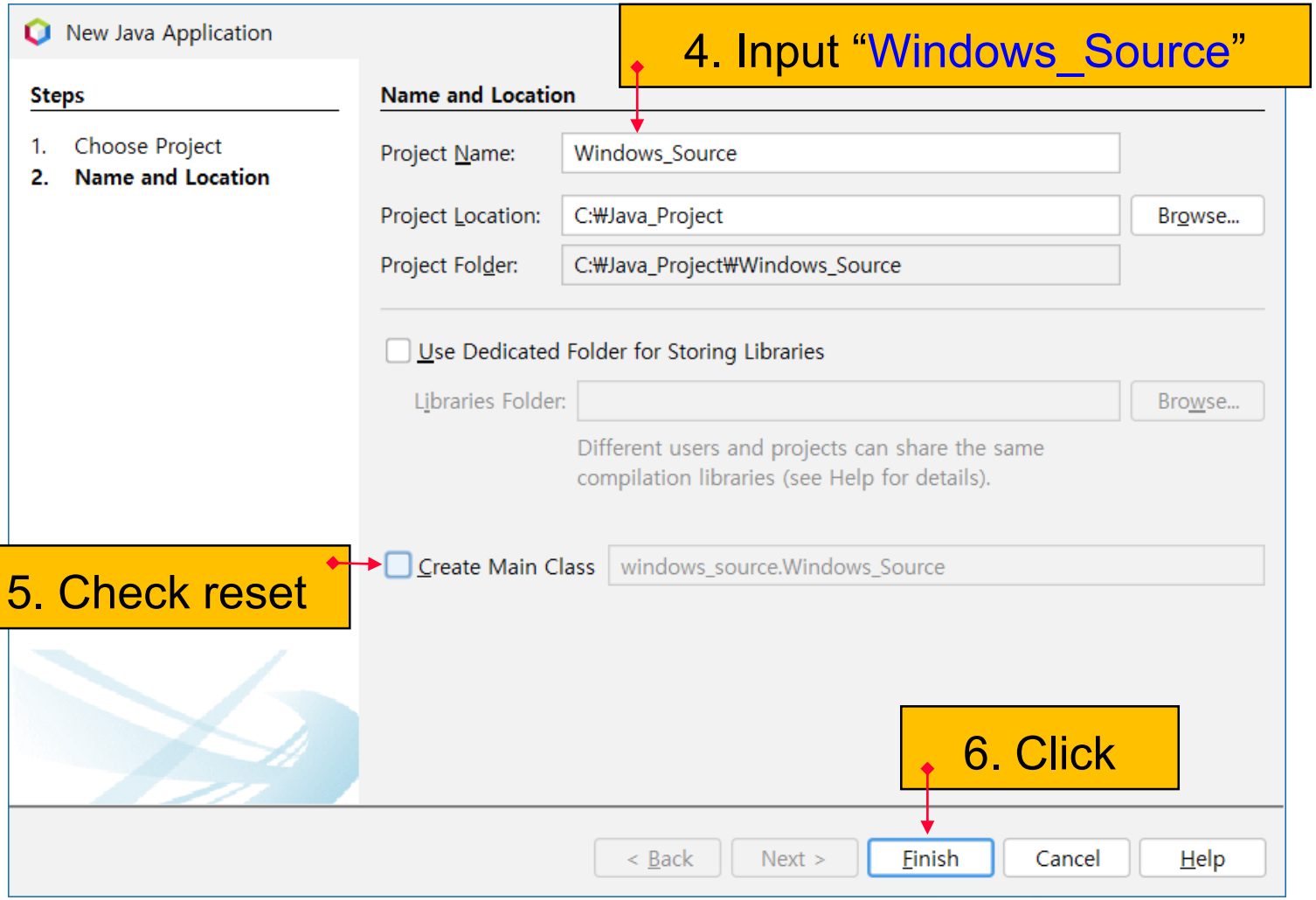
Choose Project



Windows Application 개발 방법 (3)

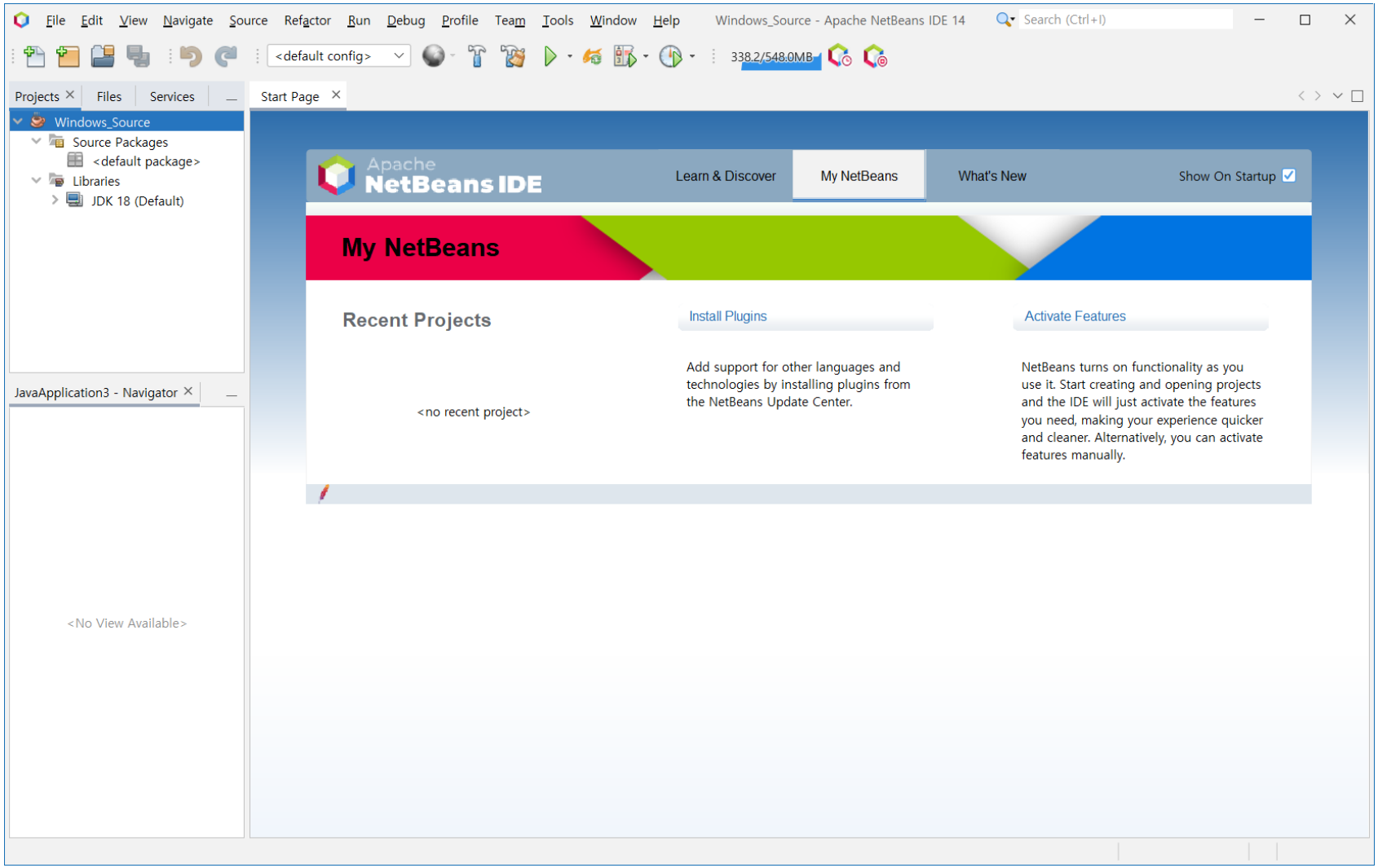
Project Name and Location

- Project name: Windows_Source



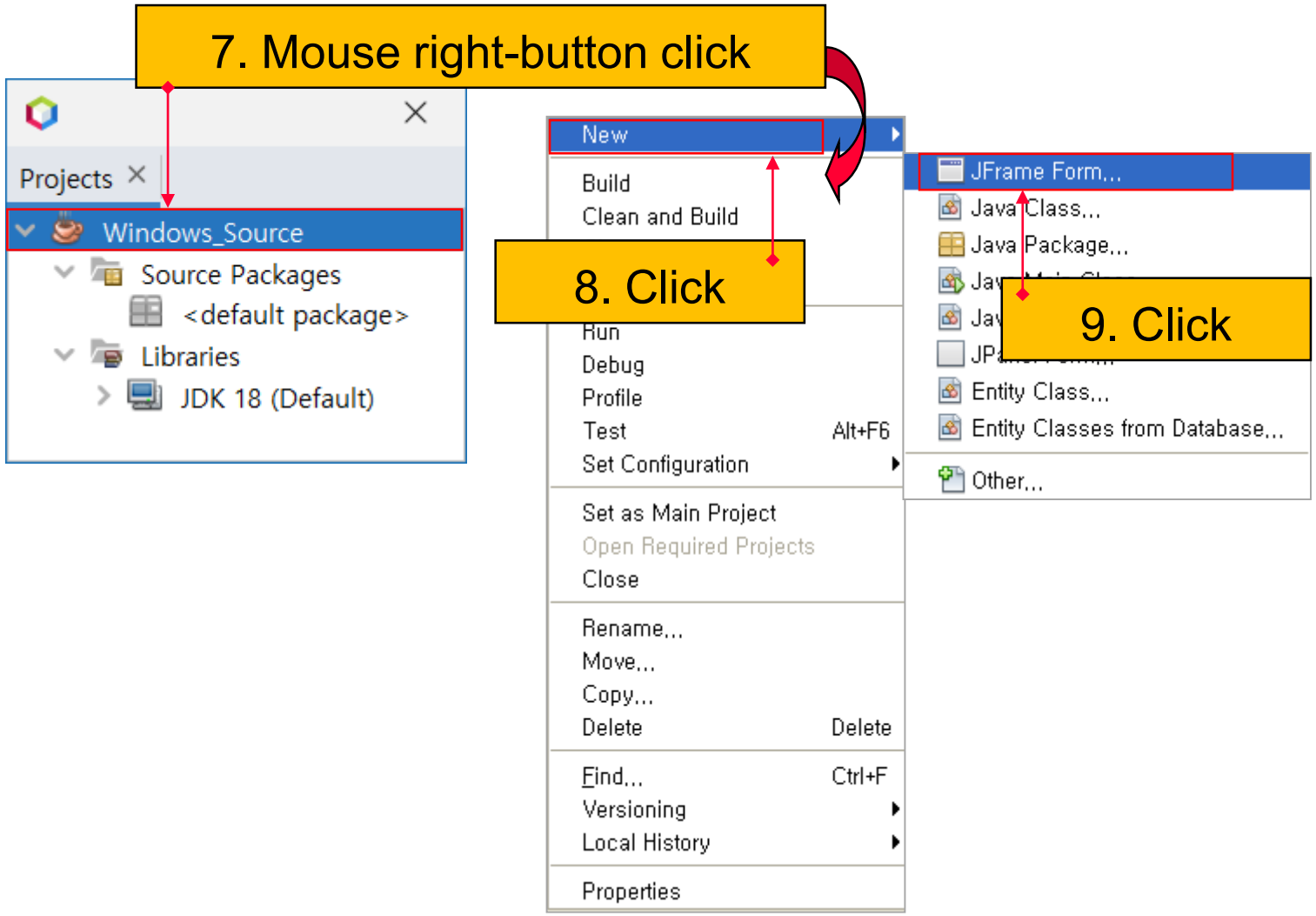
Windows Application 개발 방법 (4)

- Create project success



Windows Application 개발 방법 (5)

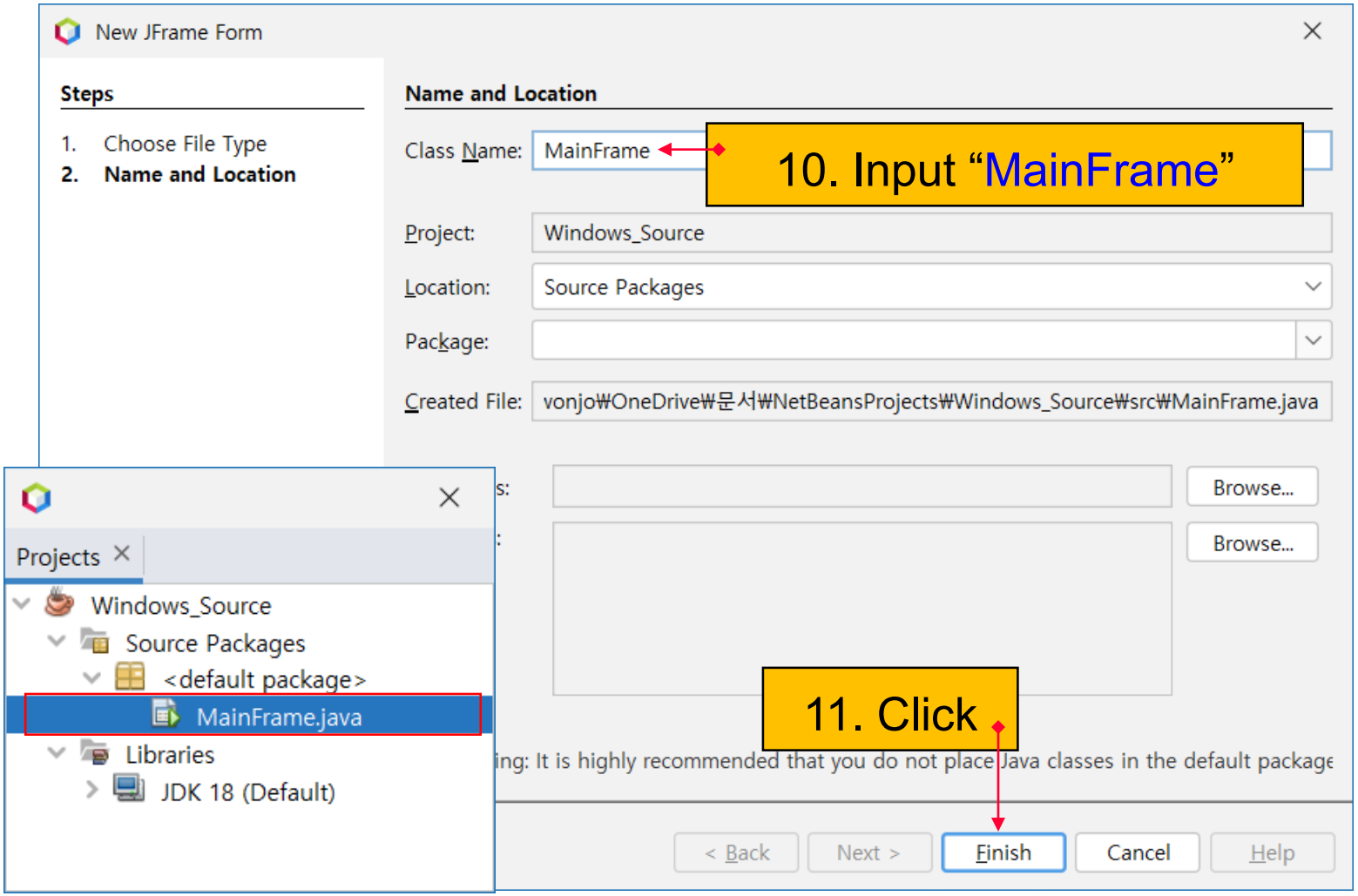
Create JFrame Form



Windows Application 개발 방법 (6)

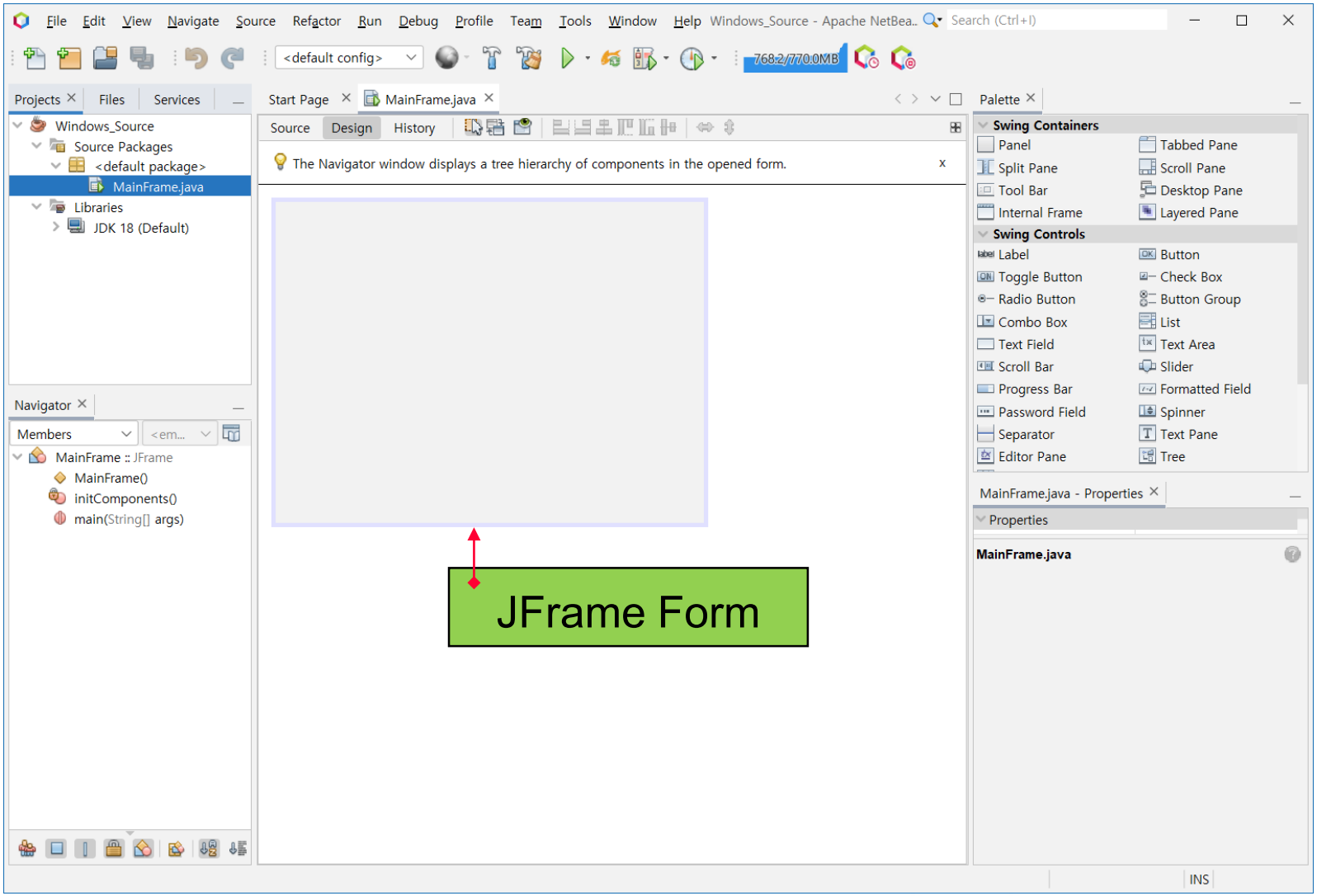
Setting Class Name

❖ Mainframe.java 생성



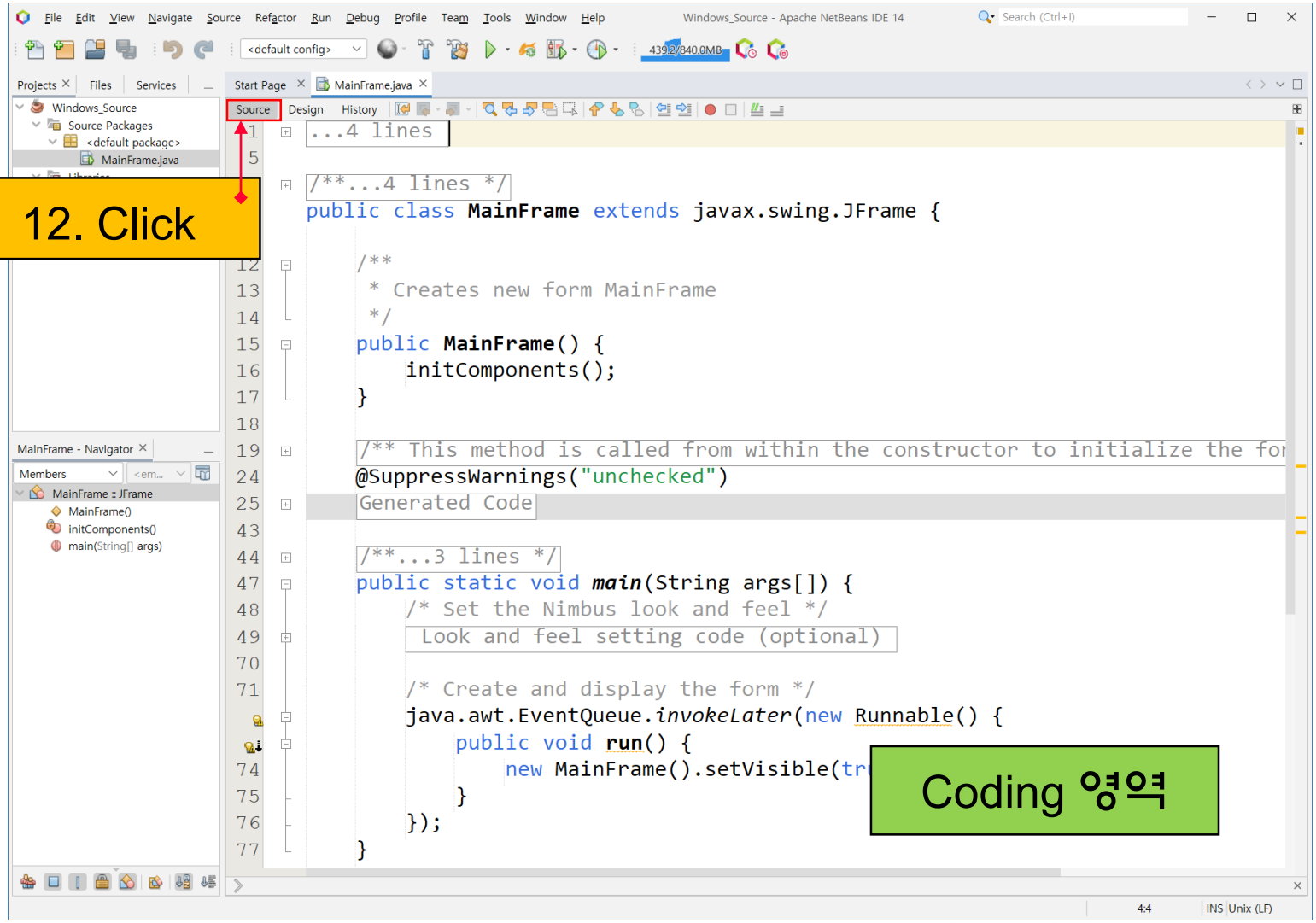
Windows Application 개발 방법 (7)

- Create JFrame form success



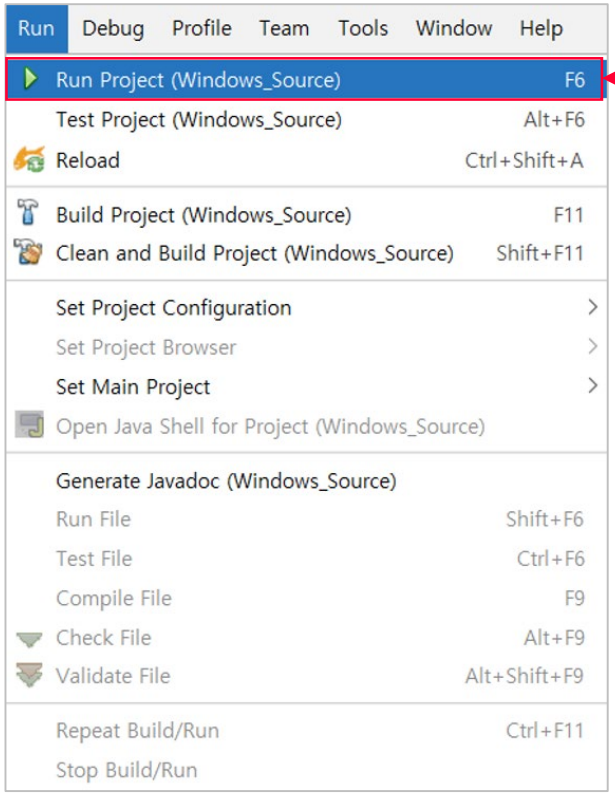
Windows Application 개발 방법 (8)

Source Code

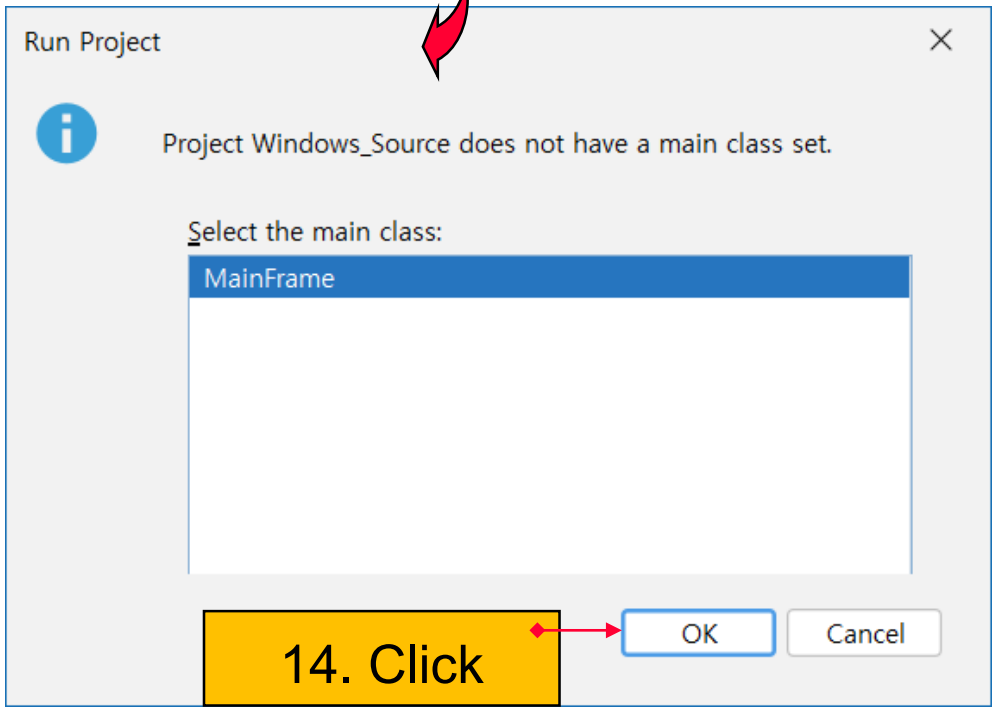


Windows Application 개발 방법 (9)

Run



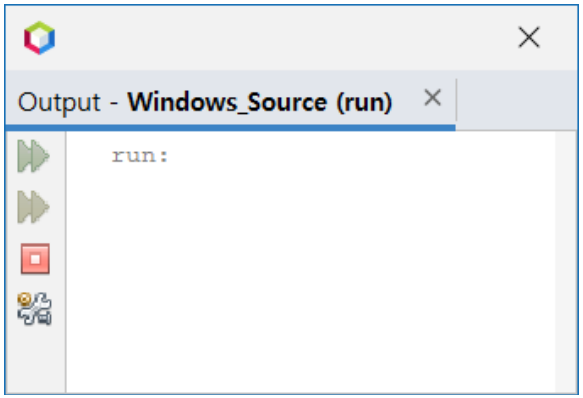
13. Click

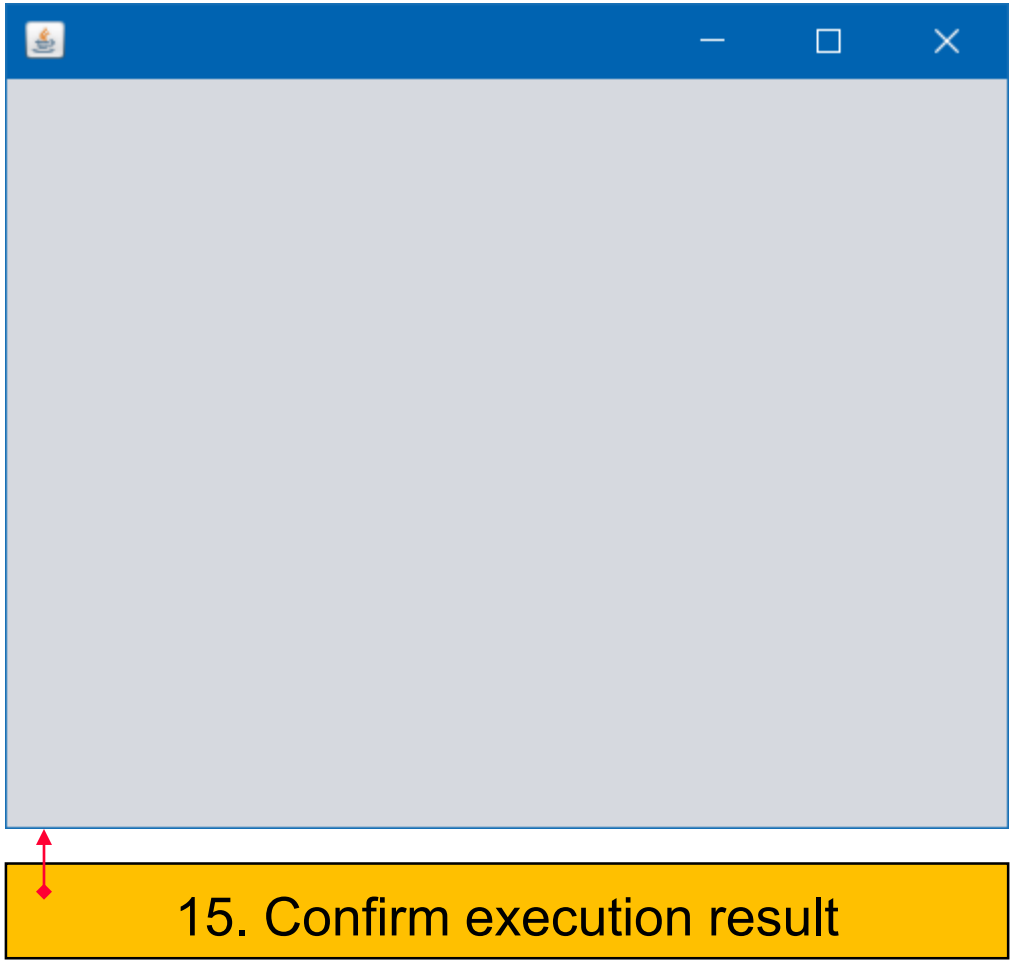


14. Click

Windows Application 개발 방법 (10)

Run result



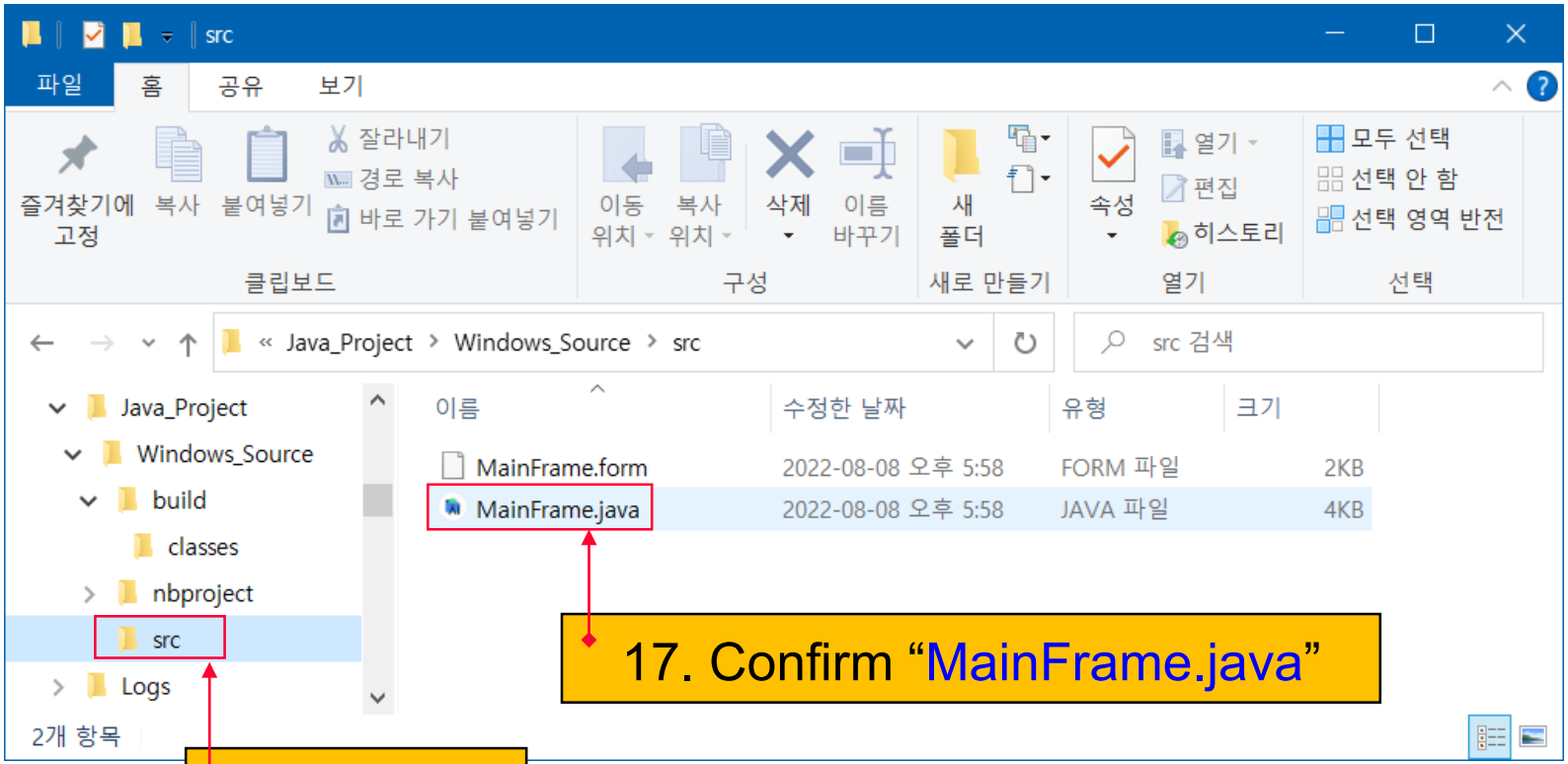


15. Confirm execution result

Windows Application 개발 방법 (11)

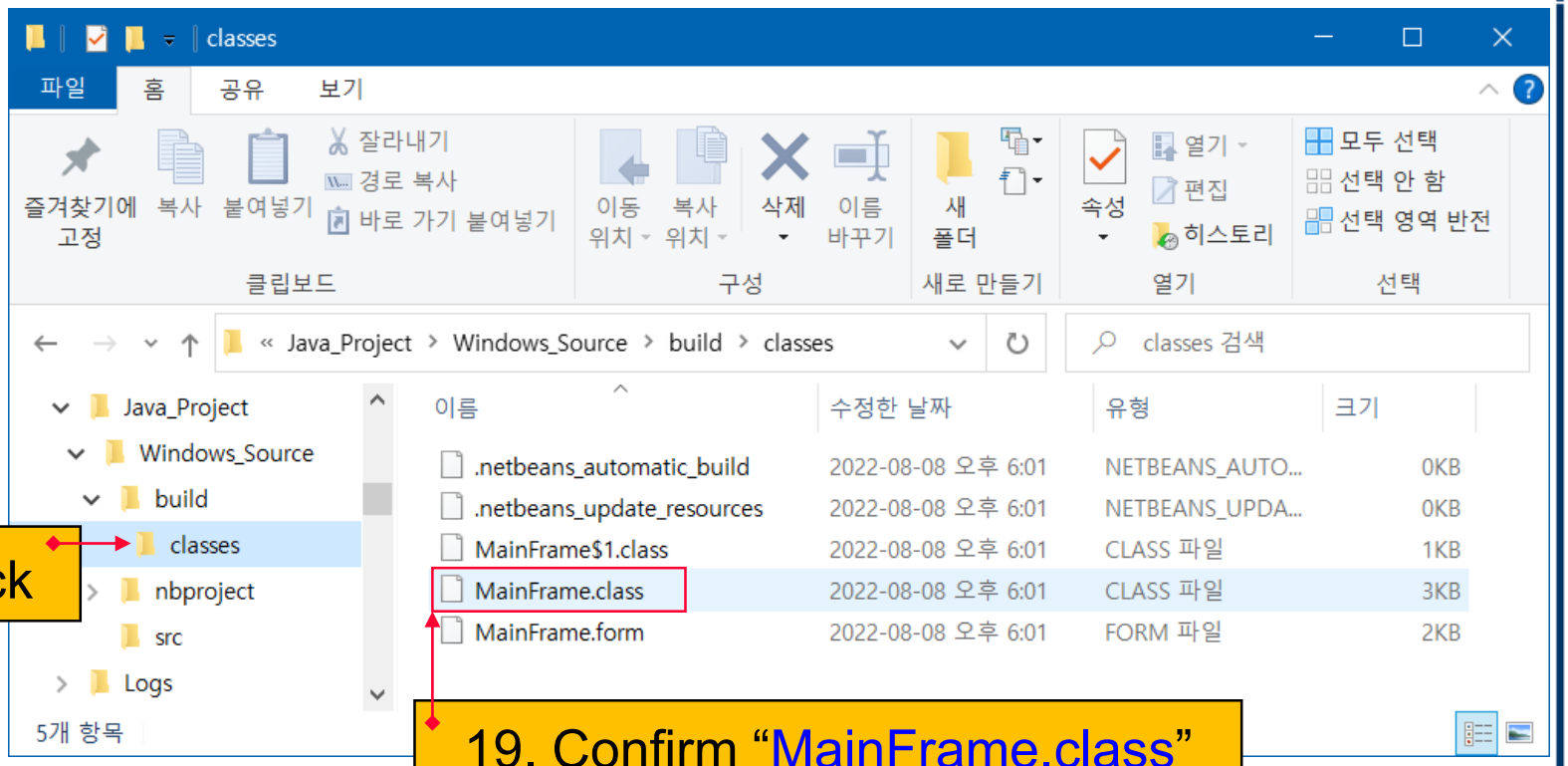
Project 구조

❖ Java source file



Windows Application 개발 방법 (12)

❖ Class file

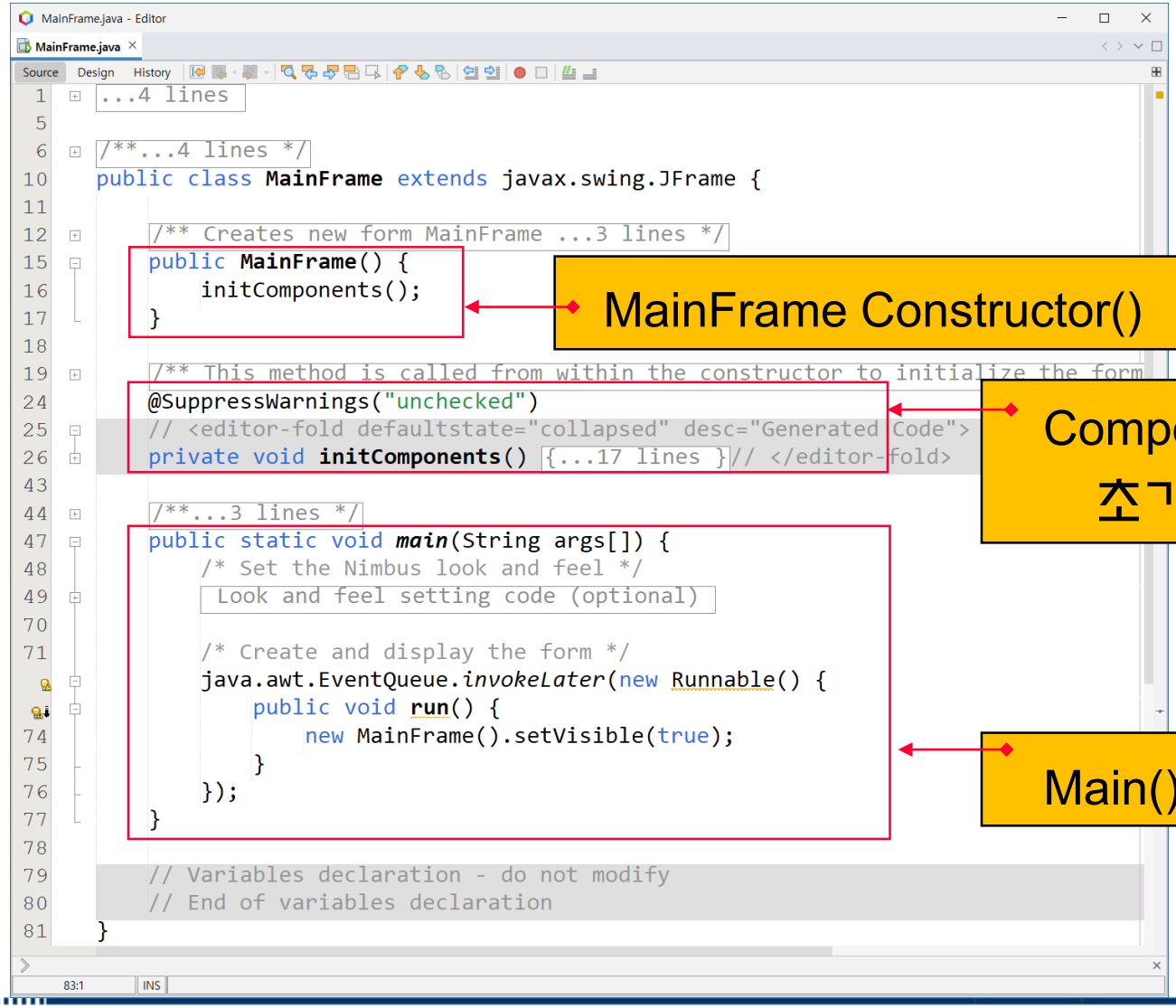


18. Click

19. Confirm "MainFrame.class"

Windows Application 소스 분석 (1)

MainFrame.java Source



Windows Application 소스 분석 (2)

❖ InitComponents() 메소드

MainFrame.java - Editor

MainFrame.java

Source Design History

```
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 with
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>
```

initComponents() method

83:1 INS

Event 란 ? (1)

Event

- Definition
  - **Message** that occurs **when a window component is clicked**
- Event 처리 과정

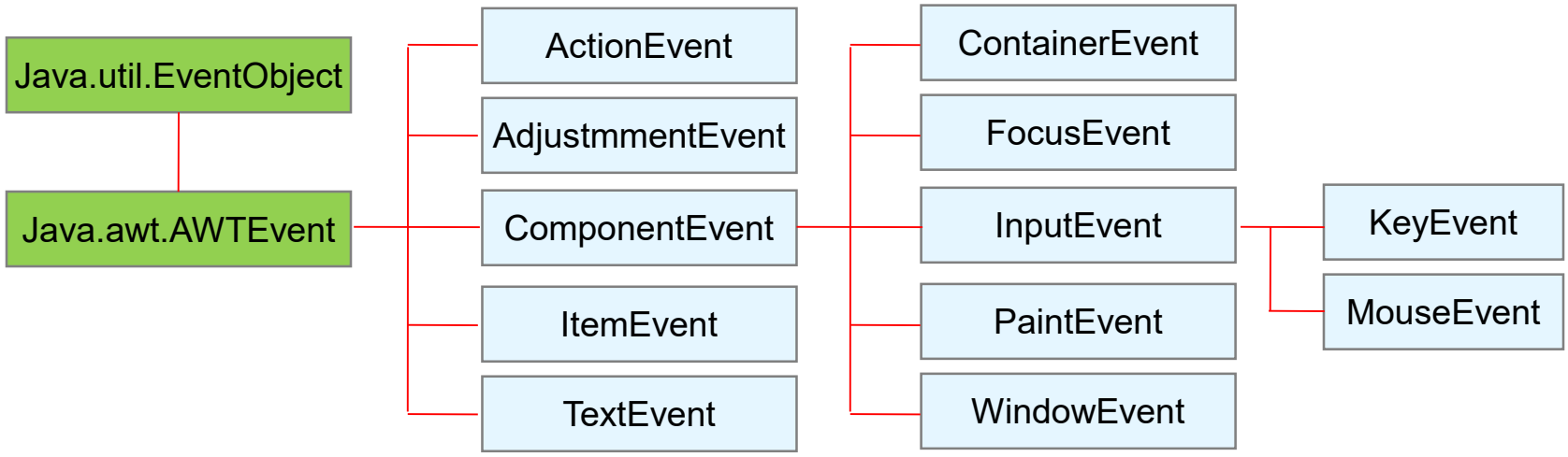
Event 란 ? (2)

Event 구성

- Event Source
  - Event를 발생시키는 Button, Scrollbar, Mouse, Keyboard,.. 등의 Component
- Event Class
  - 특정 Component에 따라 발생하는 Event를 분류한 것을 의미한다.
- Event Handler
  - Event 처리를 위한 클래스

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

Event 란 ? (5)

Component-Event 관계

Component	Component에서 발생하는 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 란 ? (6)

Component	Component에서 발생하는 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)

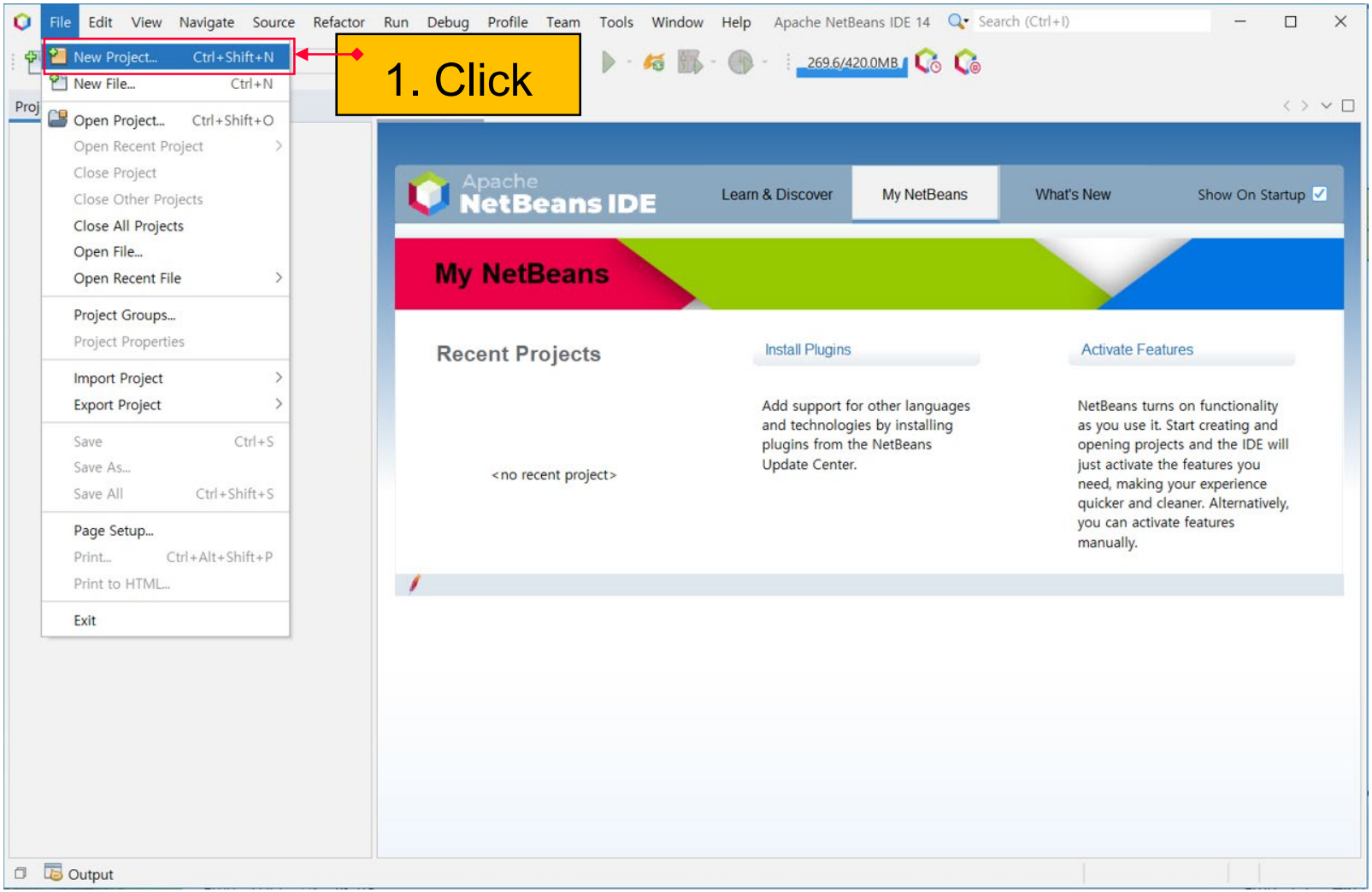
Listener Interface Adapter	Method
ItemListener	ItemStateChanged(ItemEvent)
KeyListener KeyAdapter	KeyPressed(KeyEvent) KeyReleased(KeyEvent) KeyTyped(KeyEvent)
MouseListener MouseAdapter	mouseClicked(MouseEvent) mouseEntered(MouseEvent) mouseExited(MouseEvent) mousePressed(MouseEvent) mouseReleased(MouseEvent)
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)

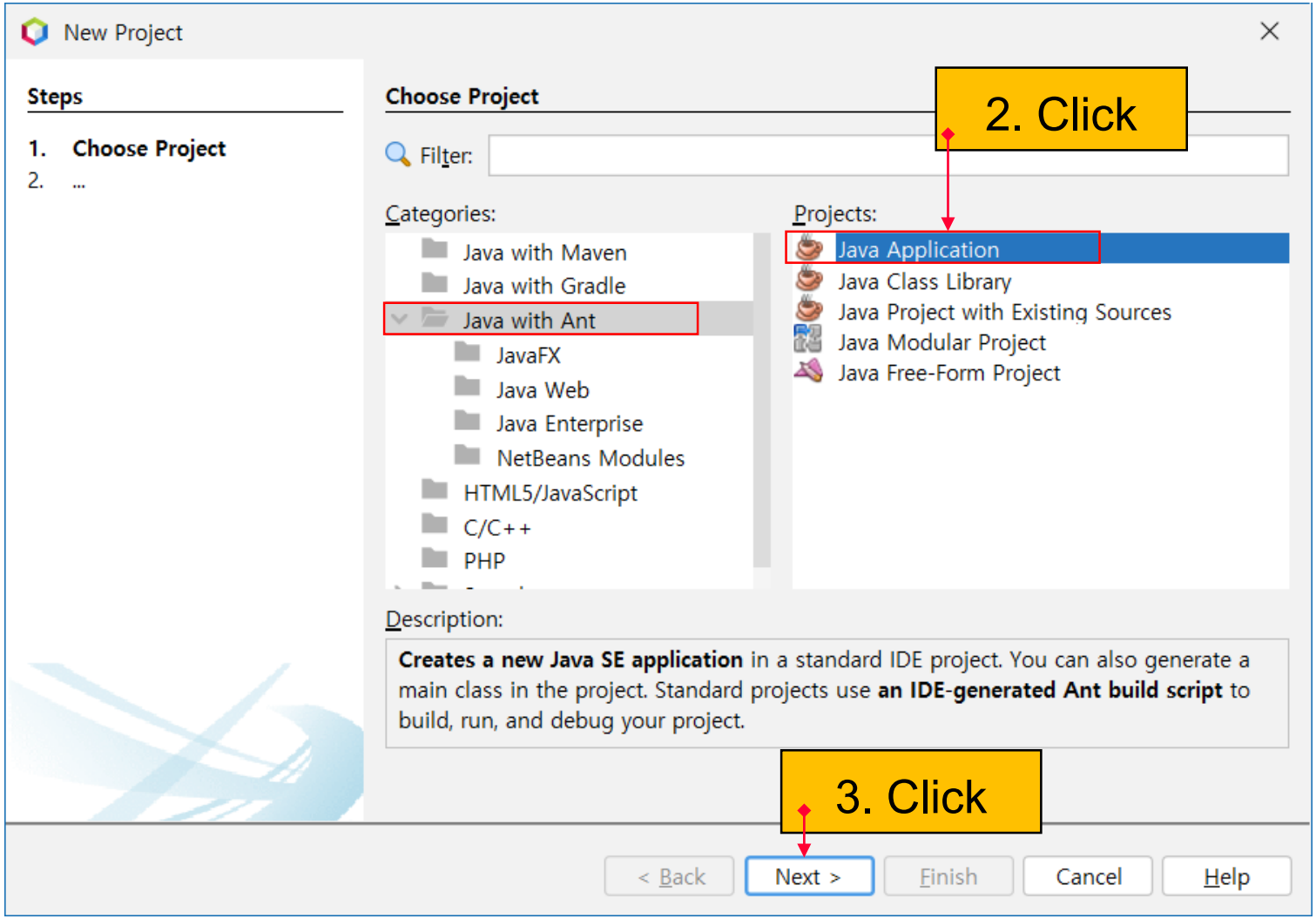
Practice : Event Handler 구현 (1)

Create Project



Practice : Event Handler 구현 (2)

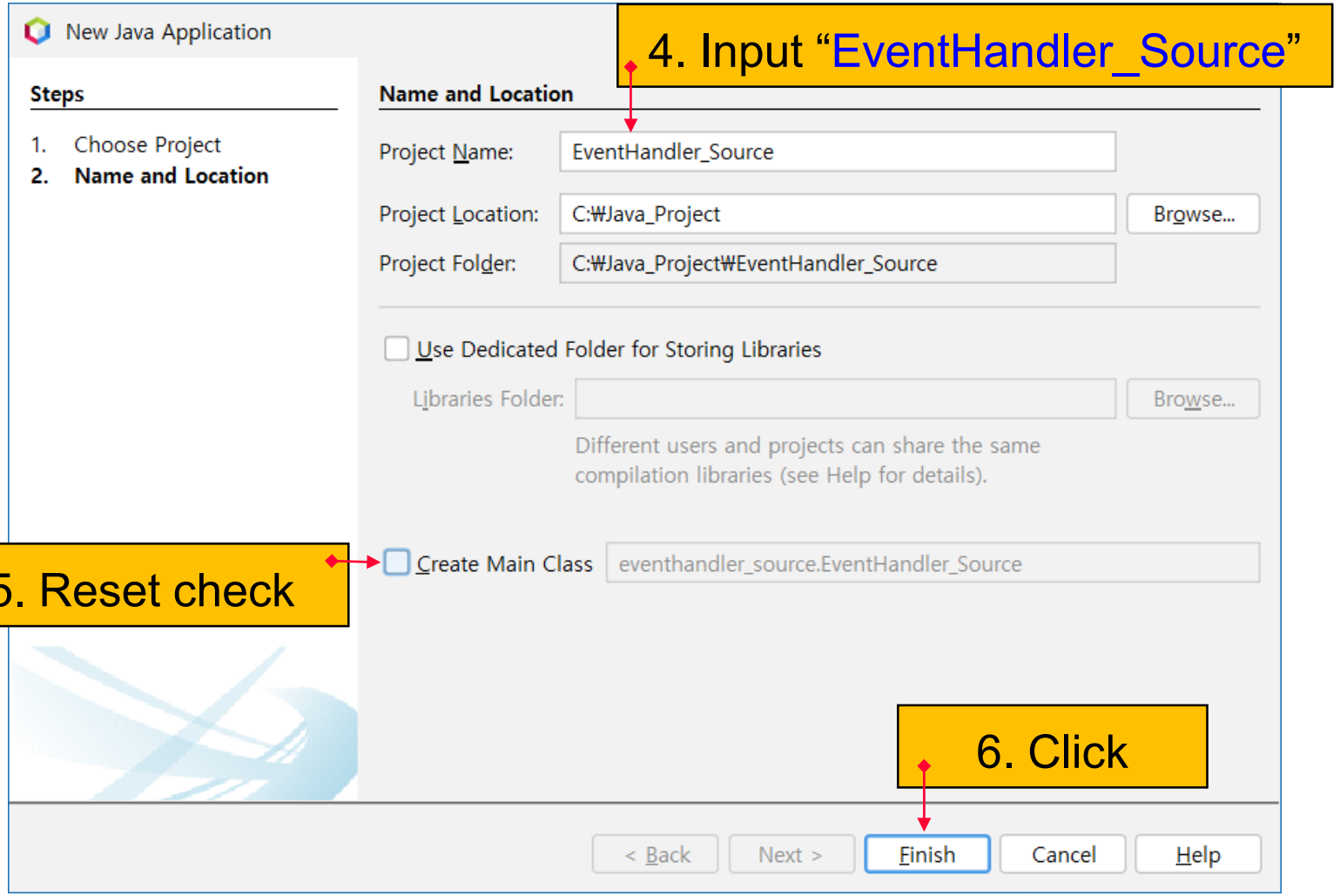
Choose Project



Practice : Event Handler 구현 (3)

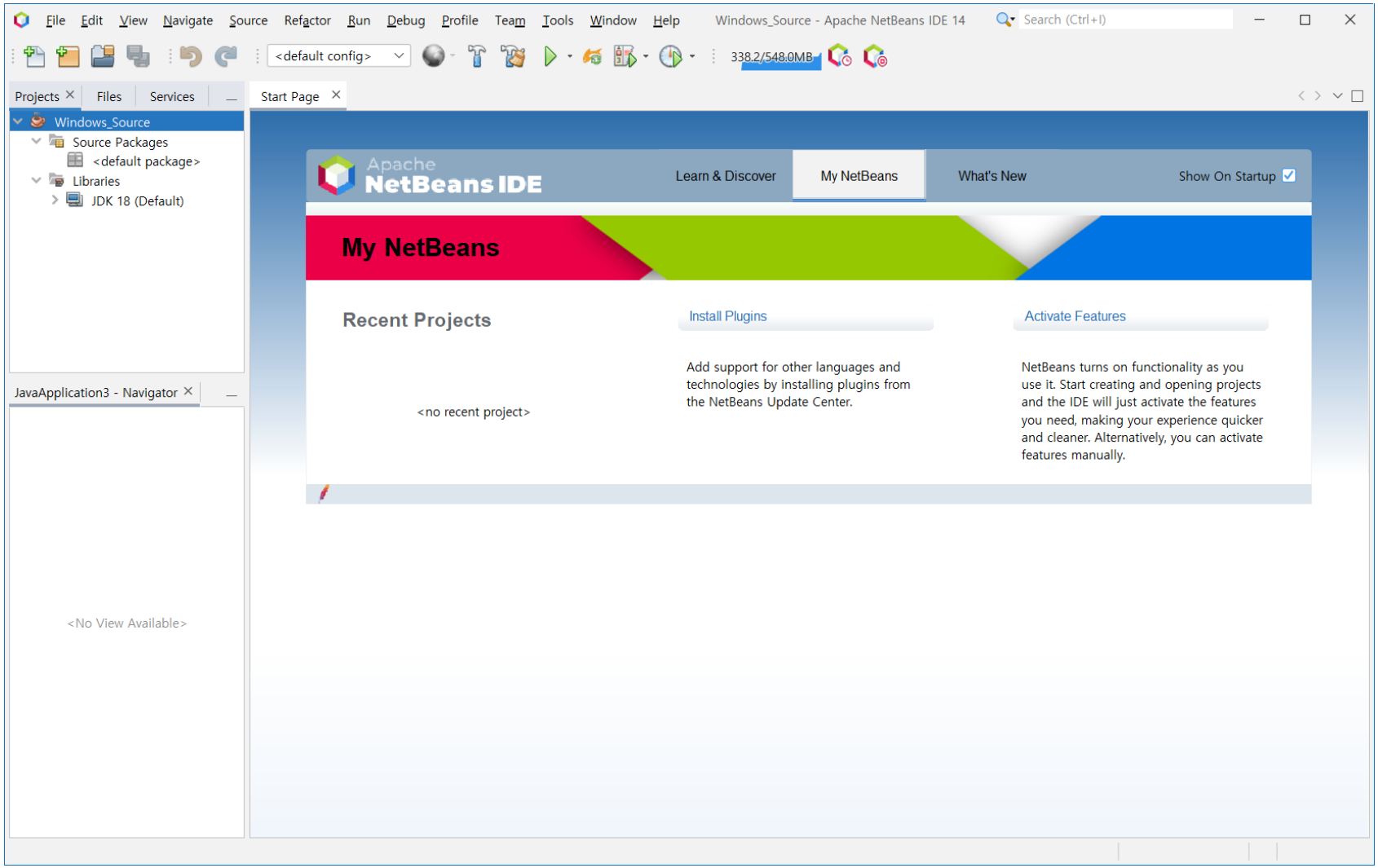
Project Name and Location

- Project name: EventHandler_Source



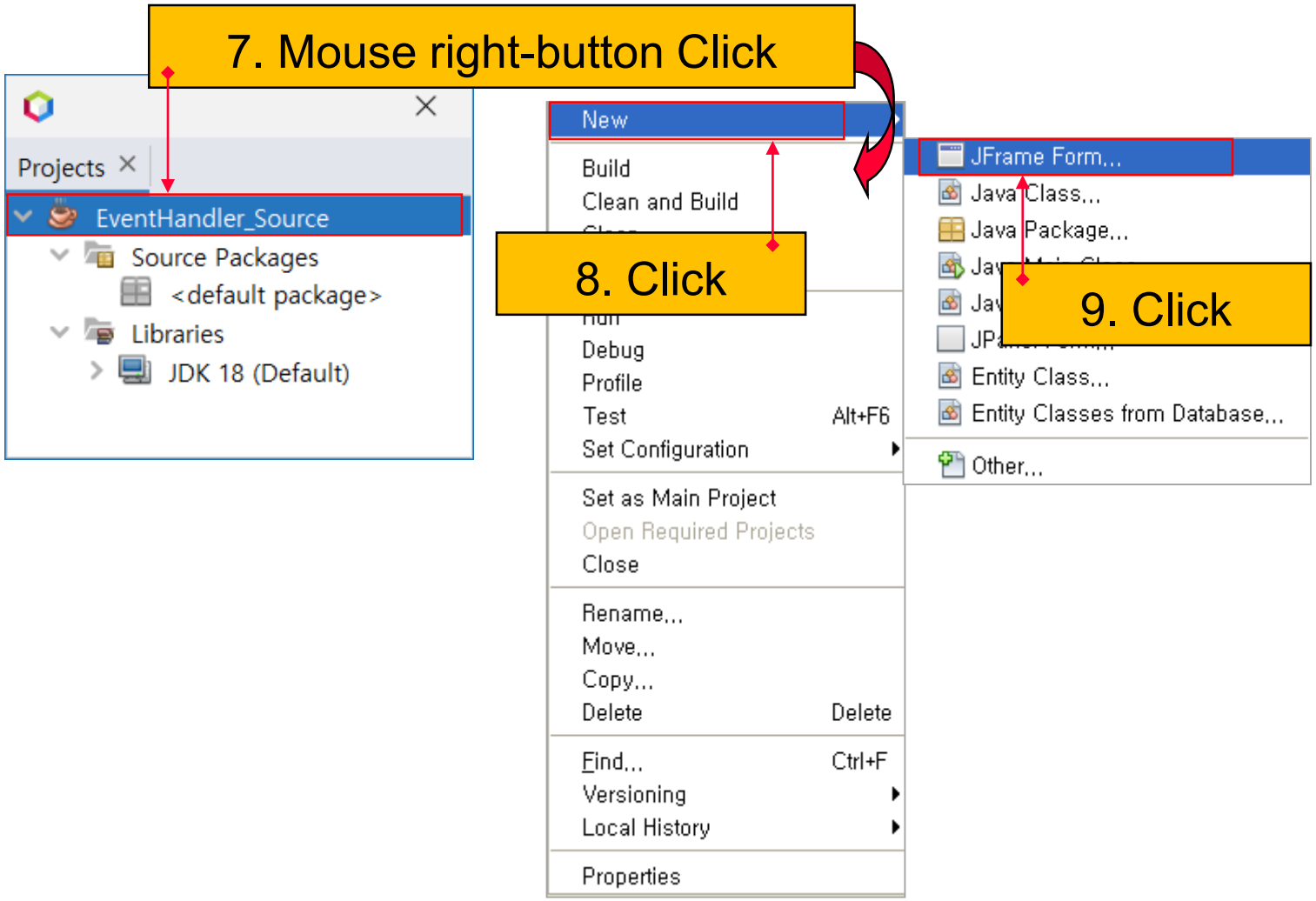
Practice : Event Handler 구현 (4)

- Create project success



Practice : Event Handler 구현 (5)

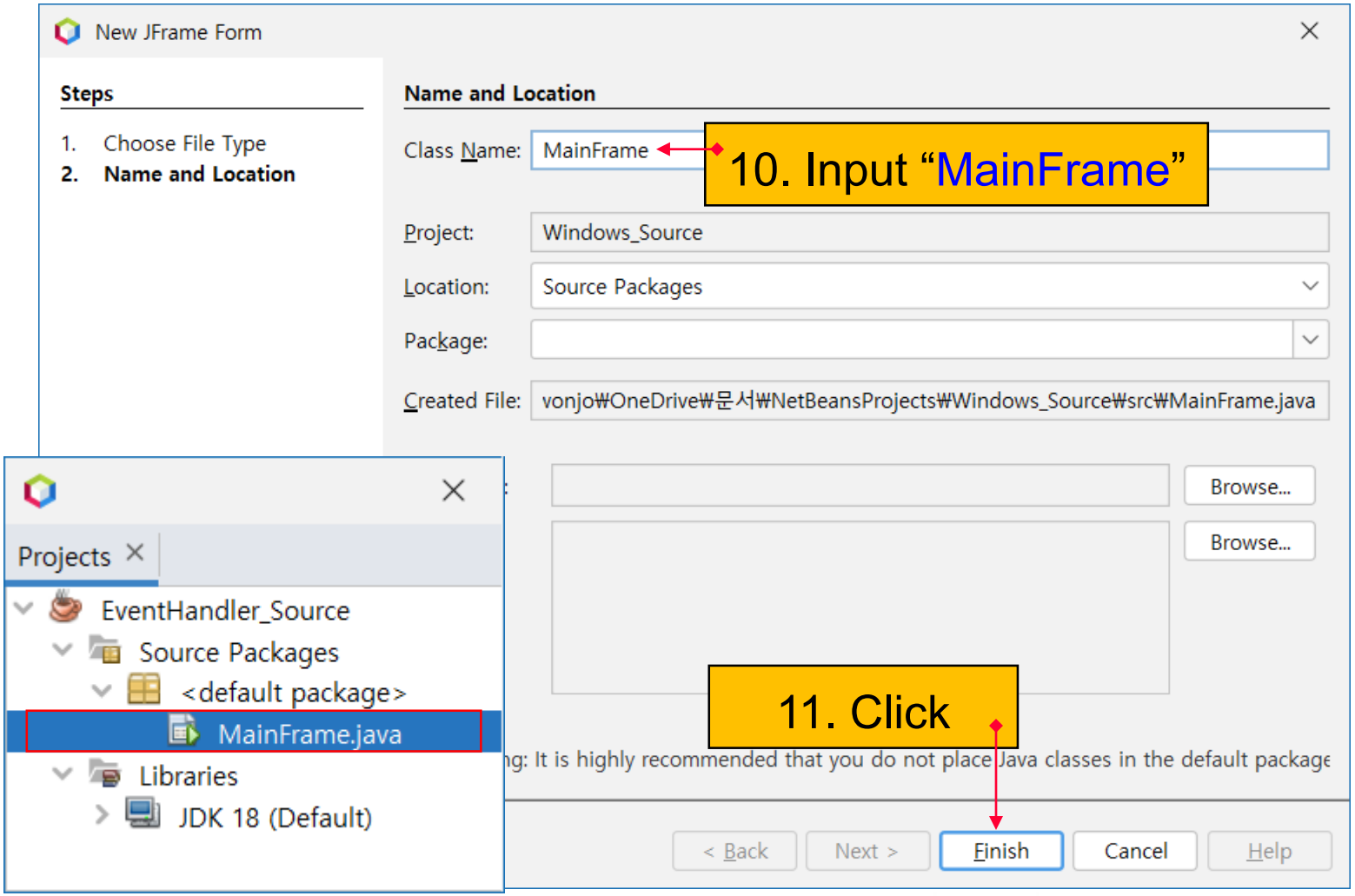
Create JFrame Form



Practice : Event Handler 구현 (6)

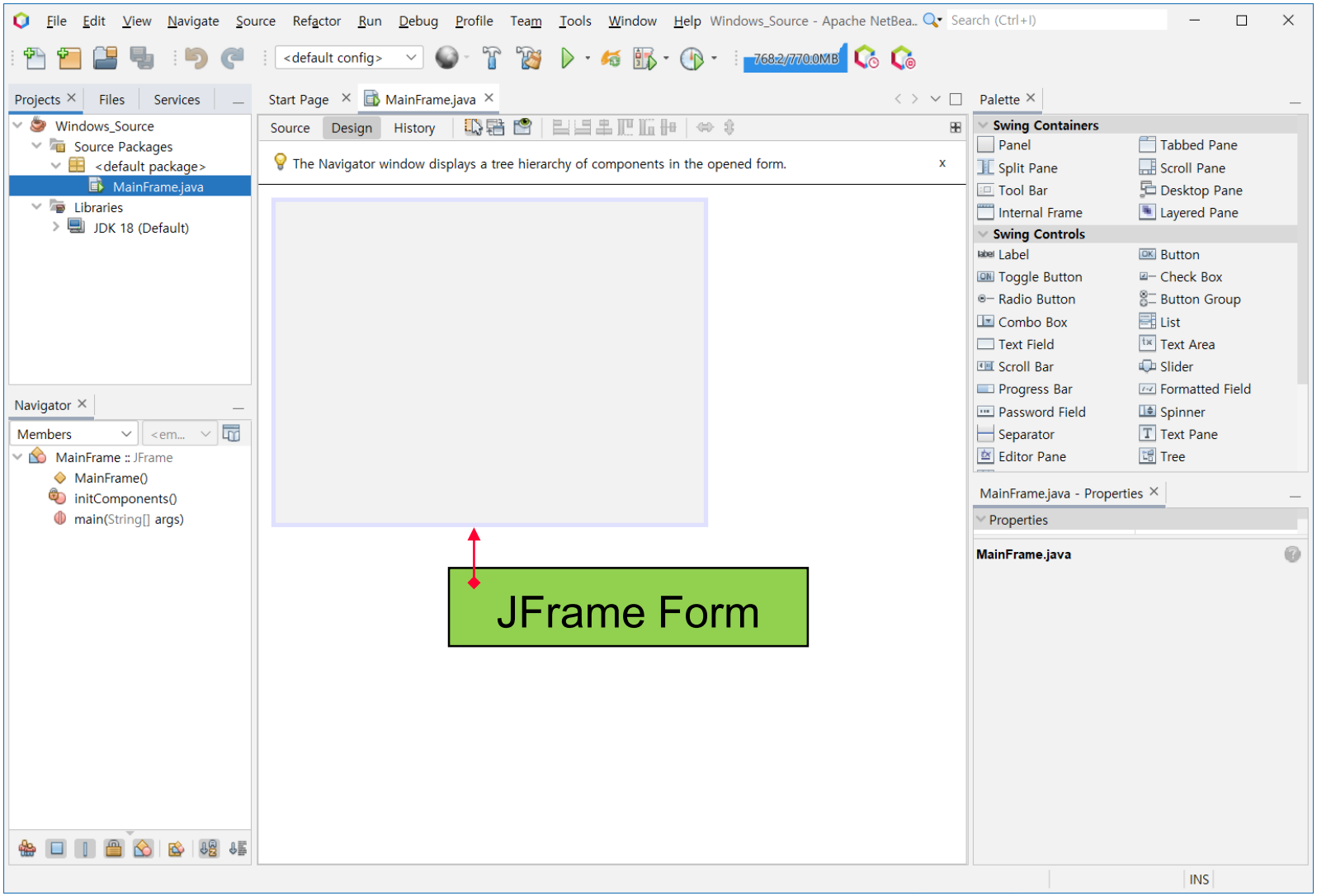
Setting Class Name

❖ Mainframe.java 생성



Practice : Event Handler 구현 (7)

- Create JFrame form success

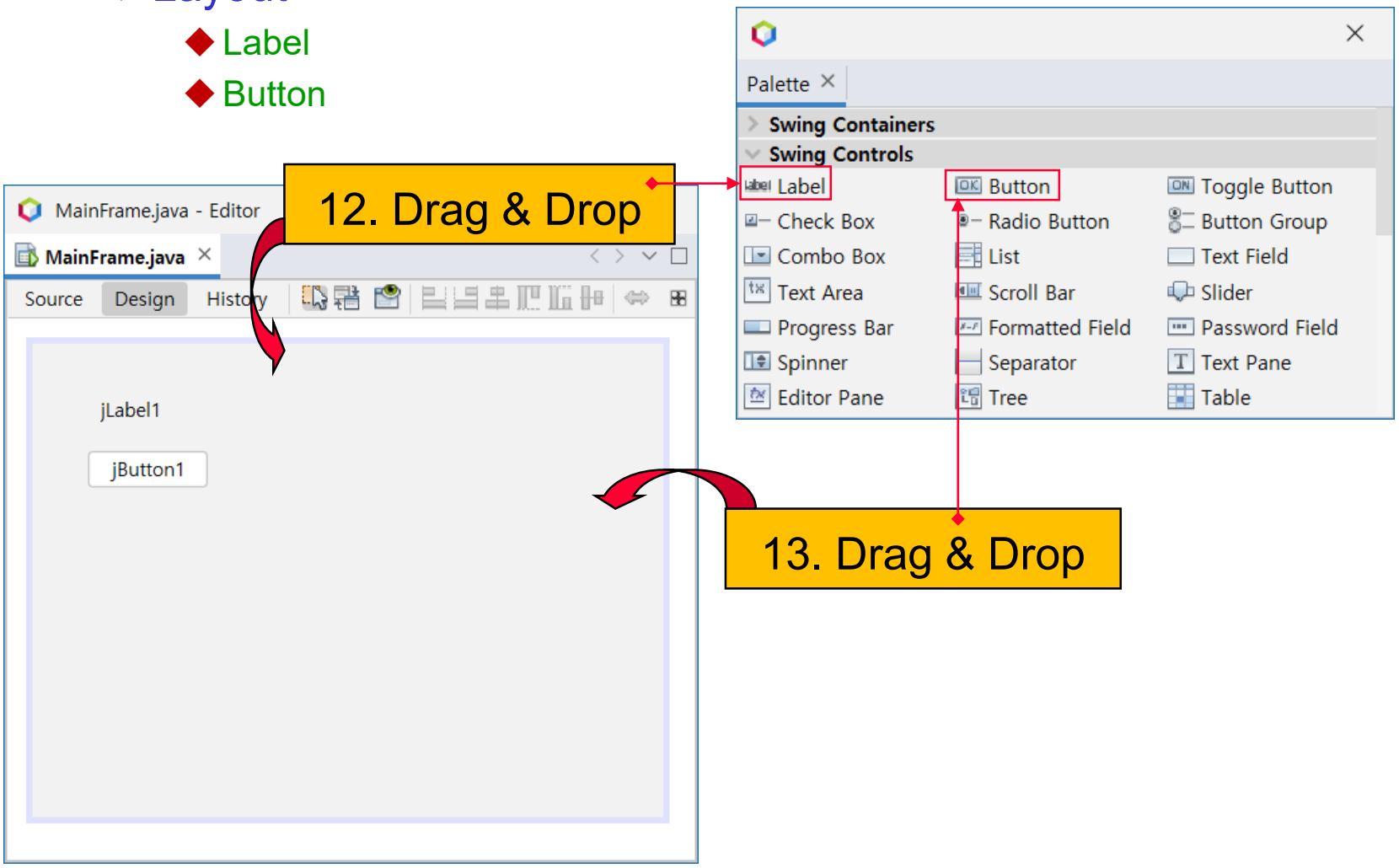


Practice : Event Handler 구현 (8)

GUI 구현

Layout

- Label
- Button



Practice : Event Handler 구현 (9)

❖ MainFrame.java Source Code

MainFrame.java - Editor

MainFrame.java

Source | Design | History

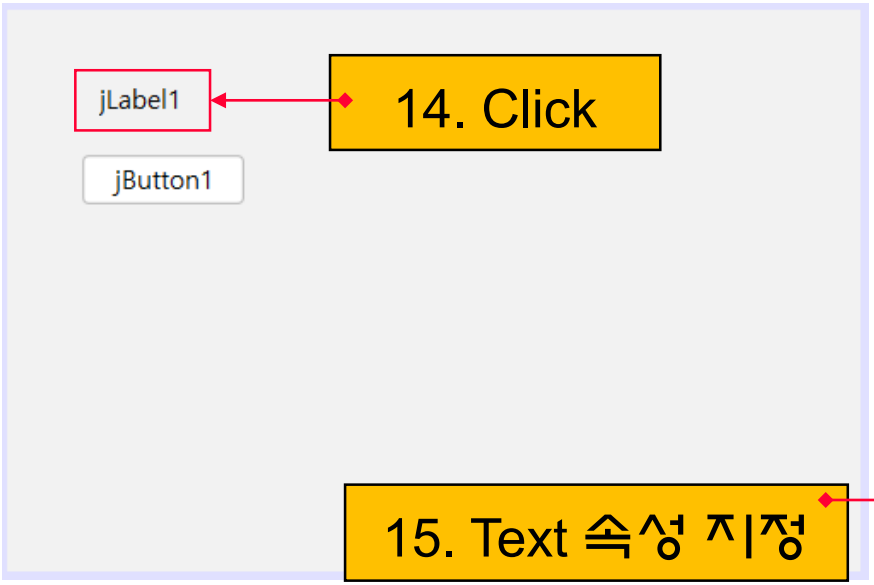
```
1
2   public class MainFrame extends javax.swing.JFrame {
3
4       public MainFrame() {
5           initComponents();
6       }
7
8       @SuppressWarnings("unchecked")
9       Generated Code
47
48      public static void main(String args[]) {
49
50          /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                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  }
```

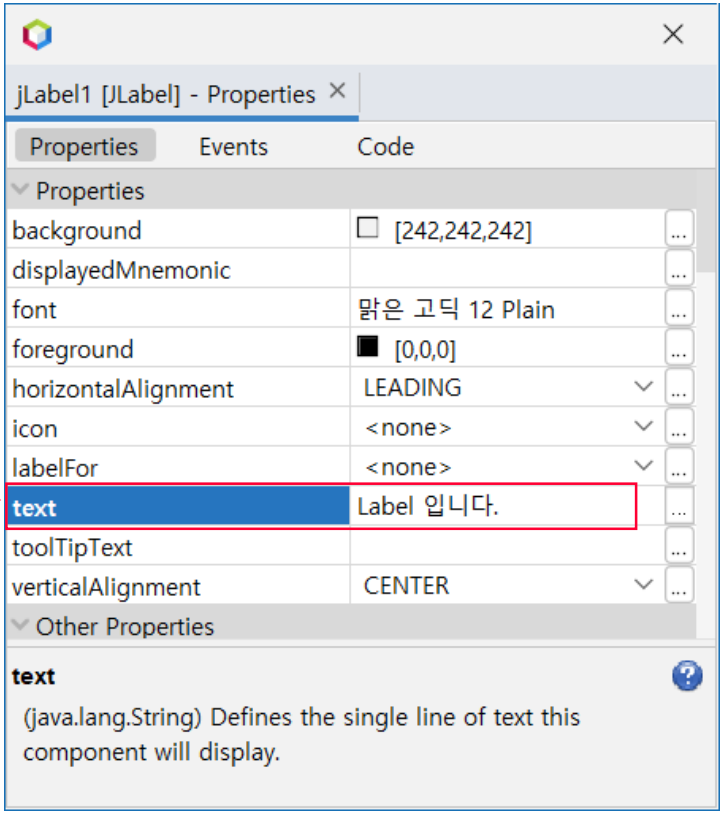
Label , Button 변수 선언

64:1 INS

Practice : Event Handler 구현 (10)

Control Layout & Property Setting

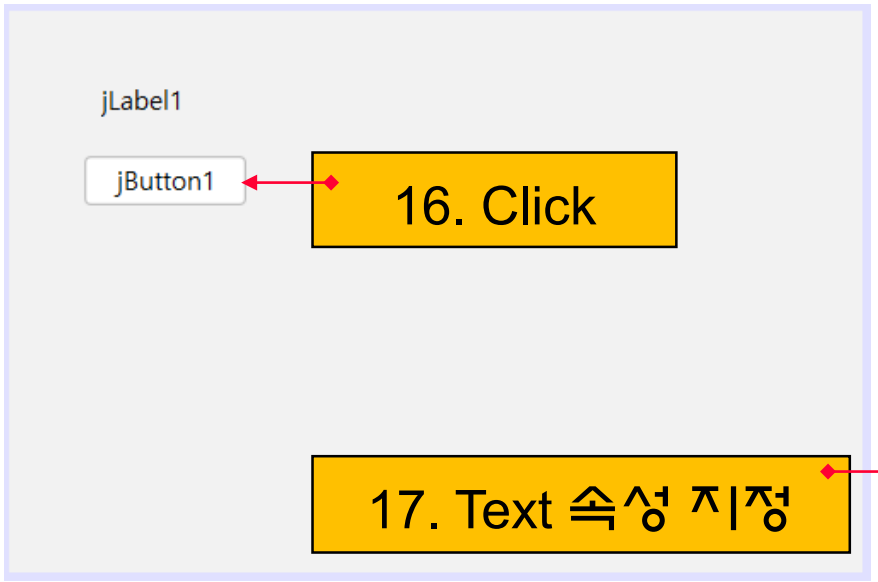


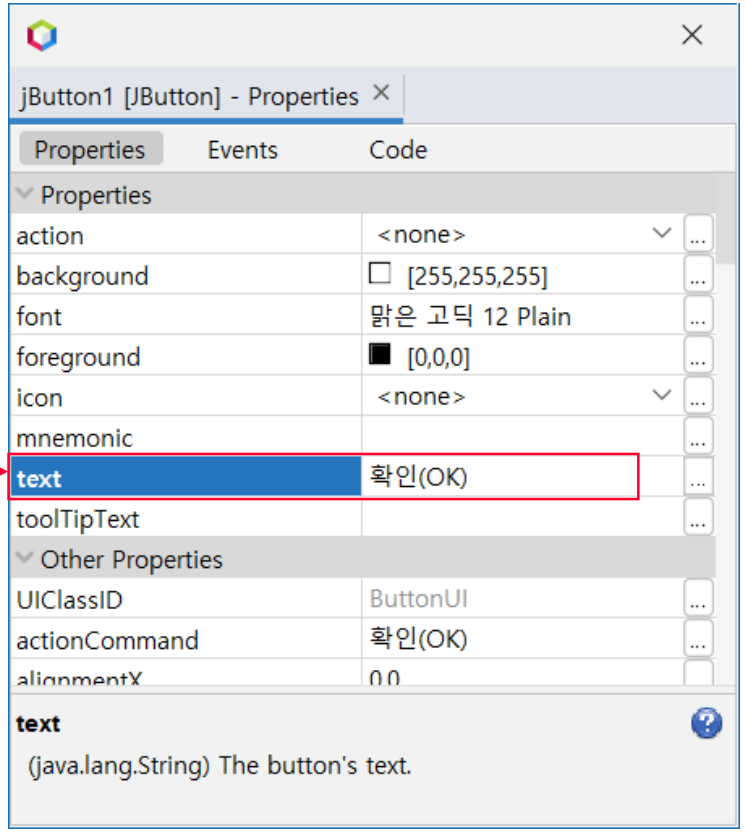


컨트롤	속성지정
Label	•Text: Label 입니다.

Practice : Event Handler 구현 (11)

❖ Button Control Property Setting

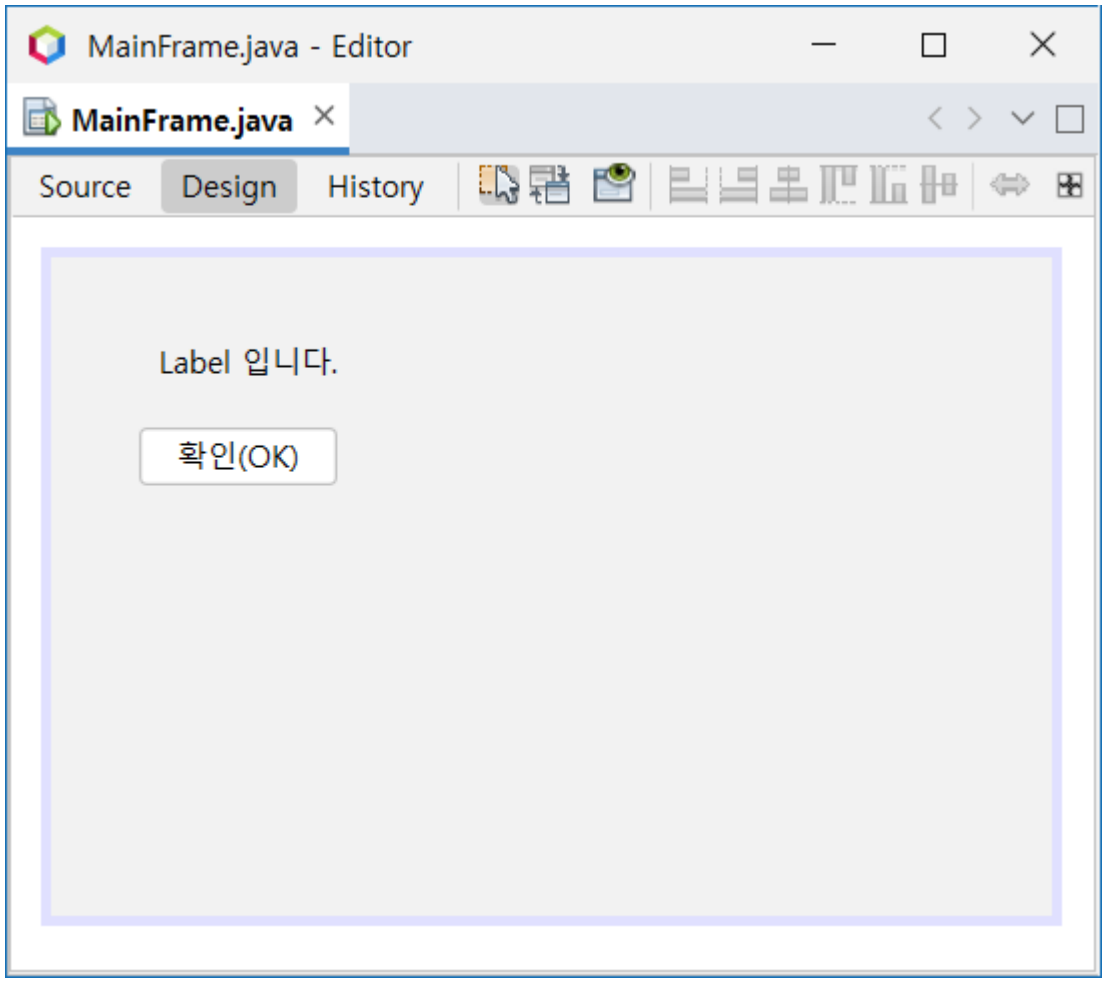




컨트롤	속성지정
Button	•Text: 확인(OK)

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

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
19          jButton1.setText("확인(OK)");
```

52:32 INS

Create Label , Button instance

Setting Label , Button property

Practice : Event Handler 구현 (14)

MainFrame.java - Editor

MainFrame.java

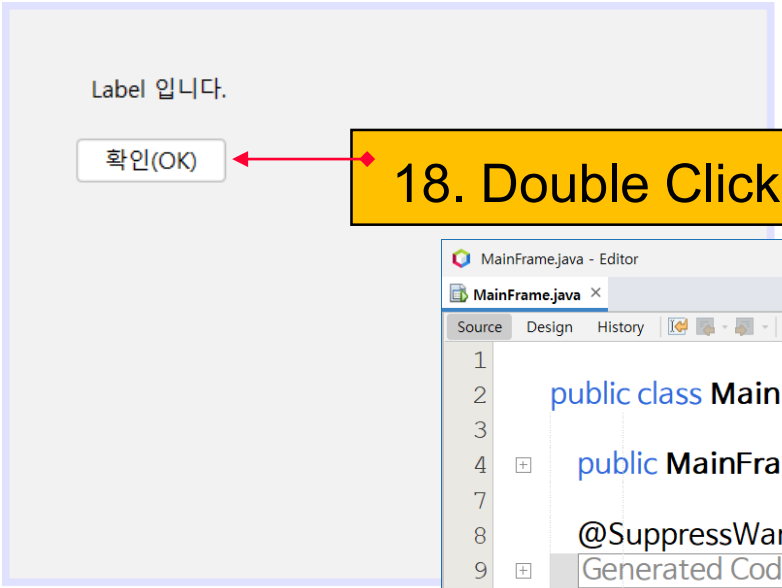
Source | Design | History

```
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>
```

52:32 INS

Practice : Event Handler 구현 (15)

[확인(OK)] Button Event Handler 구현



MainFrame.java - Editor

MainFrame.java

Source Design History

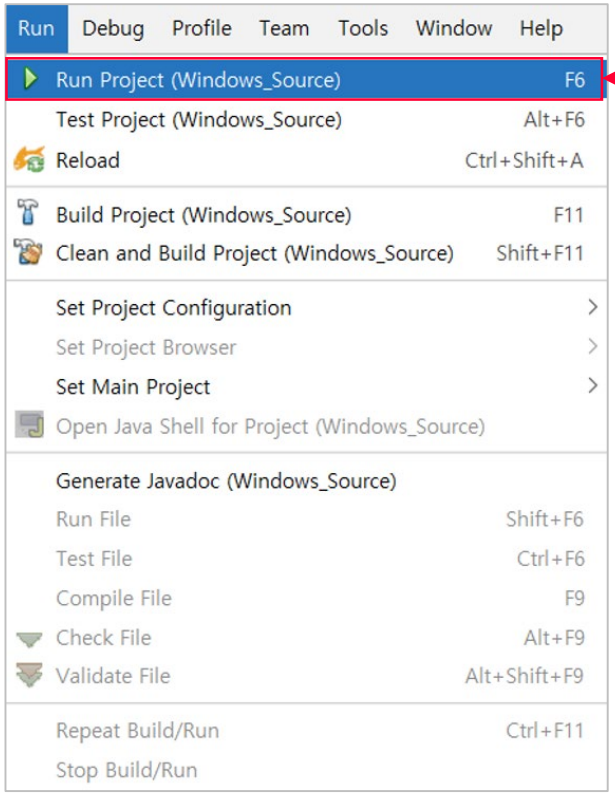
```
1
2   public class MainFrame extends javax.swing.JFrame {
3
4       public MainFrame() {...3 lines }
7
8       @SuppressWarnings("unchecked")
9       Generated Code
52
        private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
54          // TODO add your handling code here:
55          jLabel1.setText("확인(OK) Button Click");
56      }
57
58      public static void main(String args[]) {...9 lines }
67
68      // Variables declaration - do not modify
69      private javax.swing.JButton jButton1;
70      private javax.swing.JLabel jLabel1;
71      // End of variables declaration
72  }
```

19. Coding

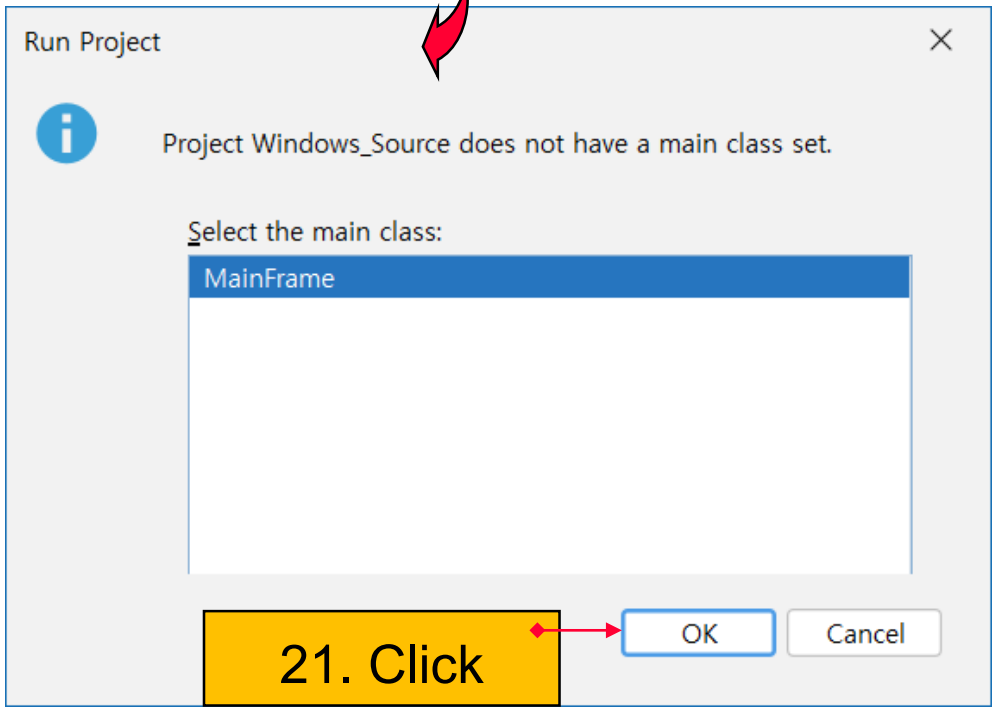
74:1 INS

Practice : Event Handler 구현 (16)

Run



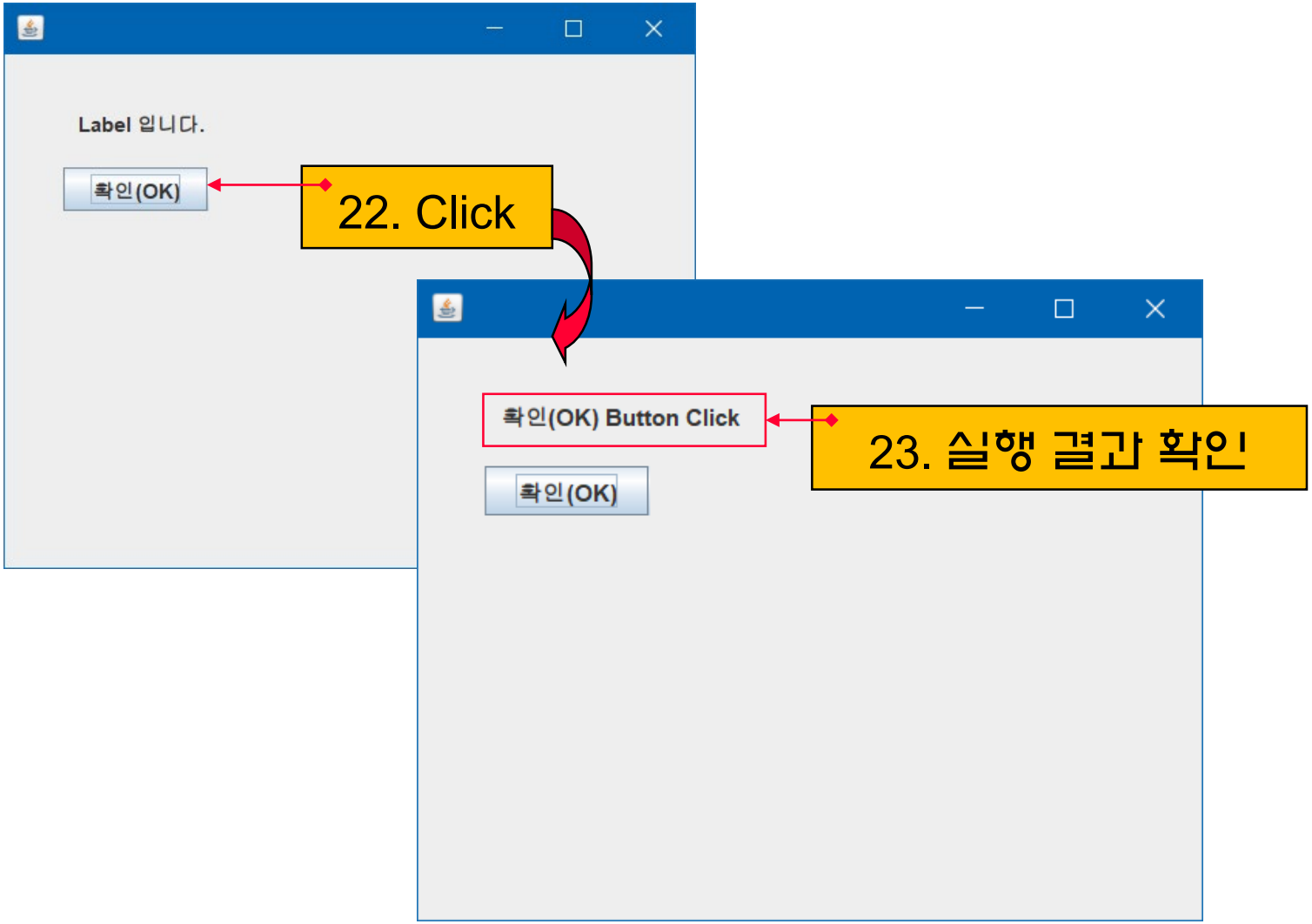
20. Click



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