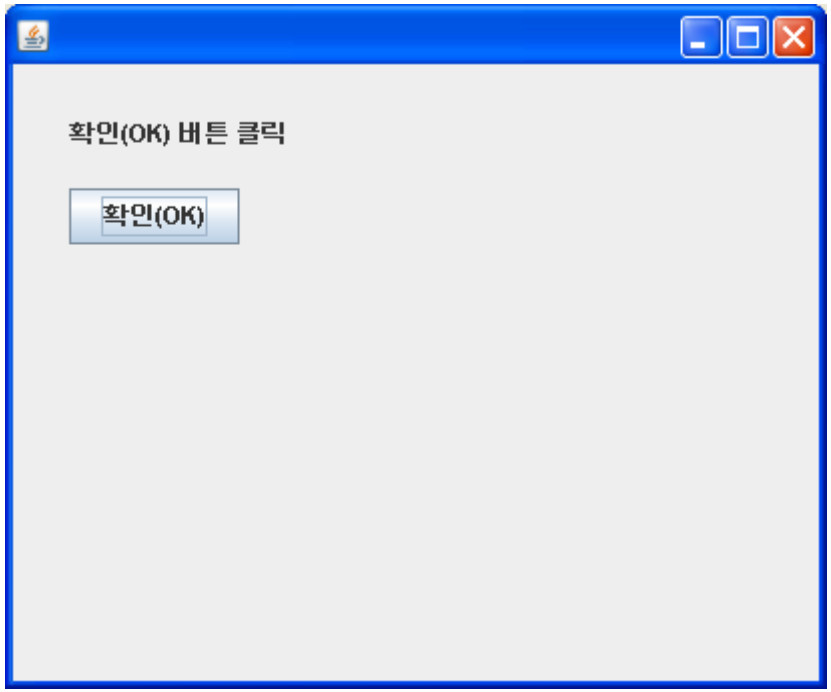
Windows 프로그래밍



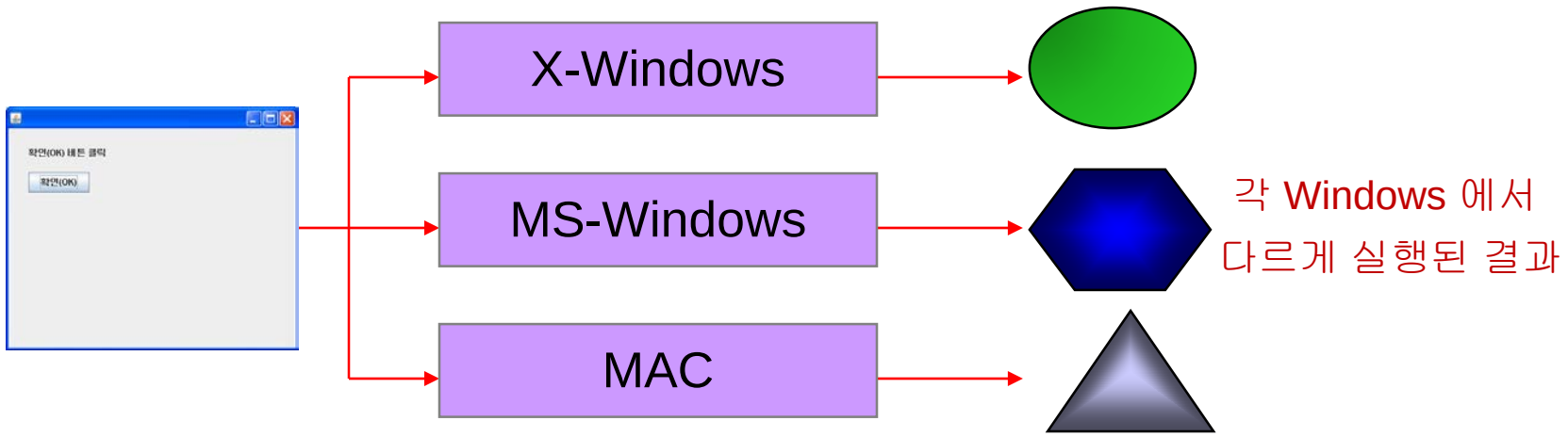
학습 목표

■ 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 인터페이스 클래스
    - ◆ 그래픽 처리 클래스
  - ❖ 특징

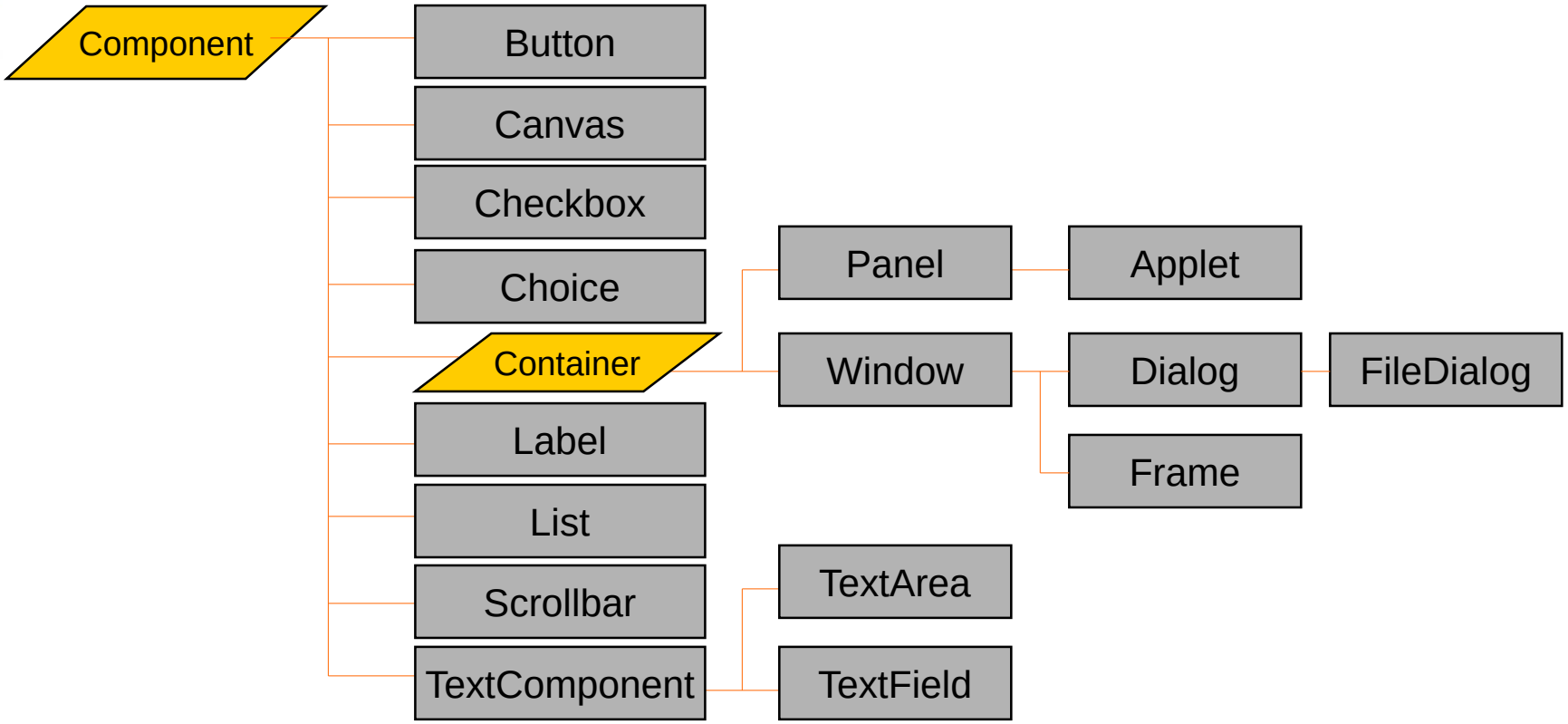


AWT 란 ? (2)

AWT의 클래스 구조

❖ 컴포넌트(Component)

- ◆ 윈도우 환경의 컴포넌트(버튼, 체크박스,…)
- ◆ java.awt 클래스로 객체 모델링



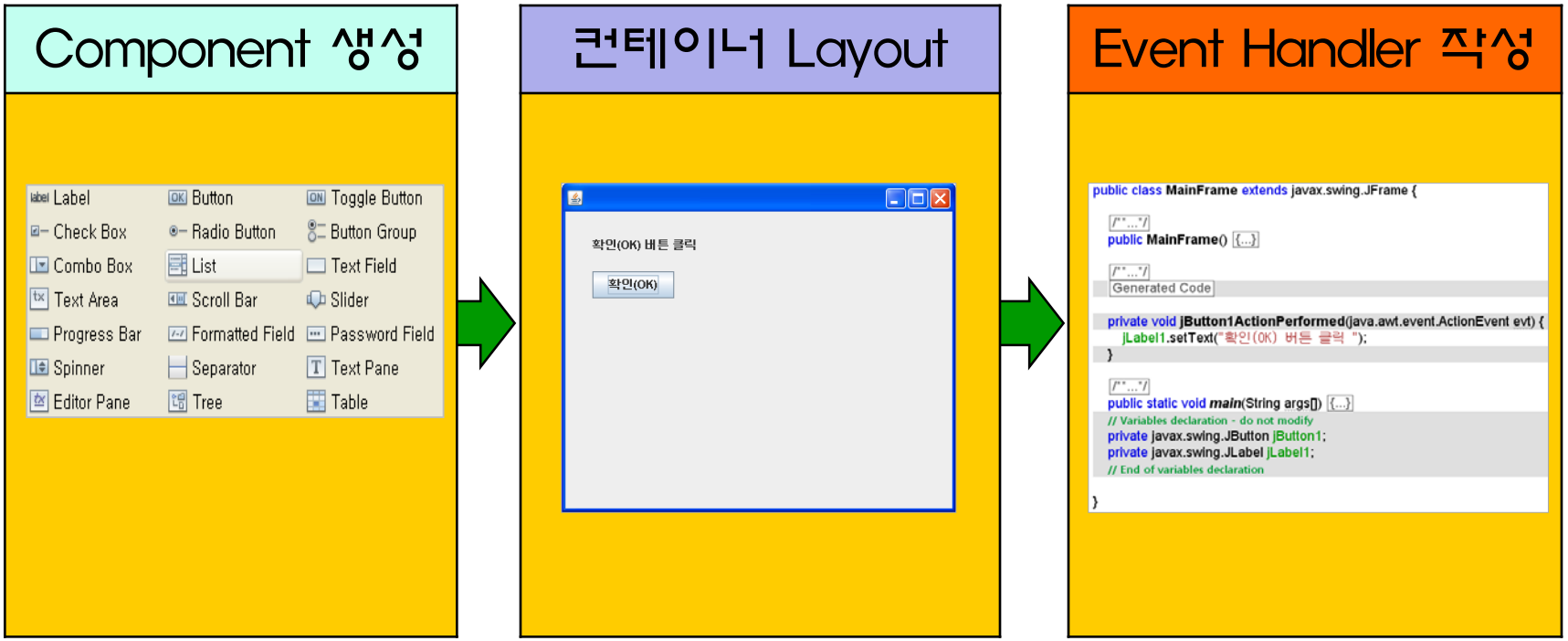
AWT 란 ? (3)

Method

Method	기능
Public Dimension getSize()	컴포넌트 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:컴포넌트 inactive 상태로 전환
void setBounds(int x, int y, int width, int height) void setSize(Dimension d)	컴포넌트 위치 지정 컴포넌트 크기 지정
void setVisible(boolean b)	True: 화면에 표현 False: 화면에서 사라짐

AWT 란 ? (4)

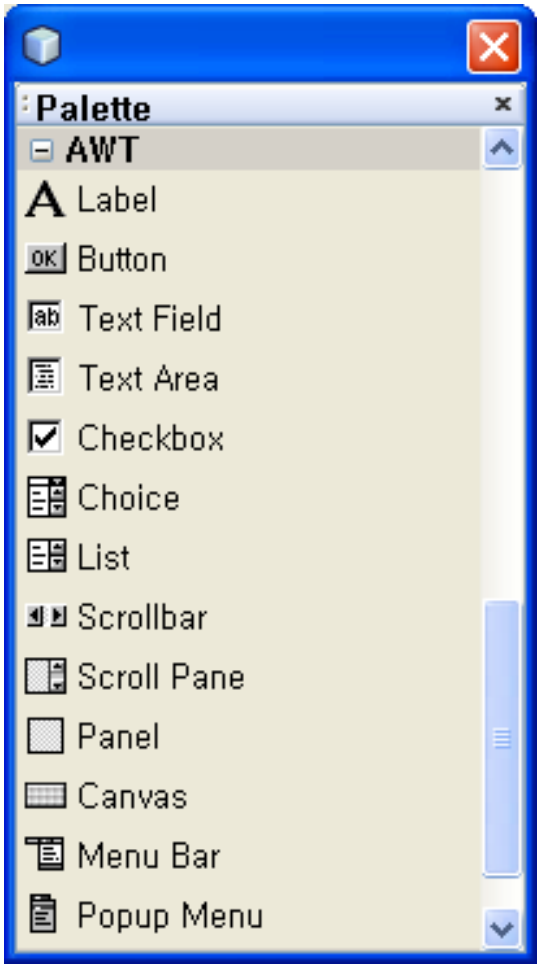
- Visual Component와 Container 관계



AWT 란 ? (5)

컴포넌트

컴포넌트	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

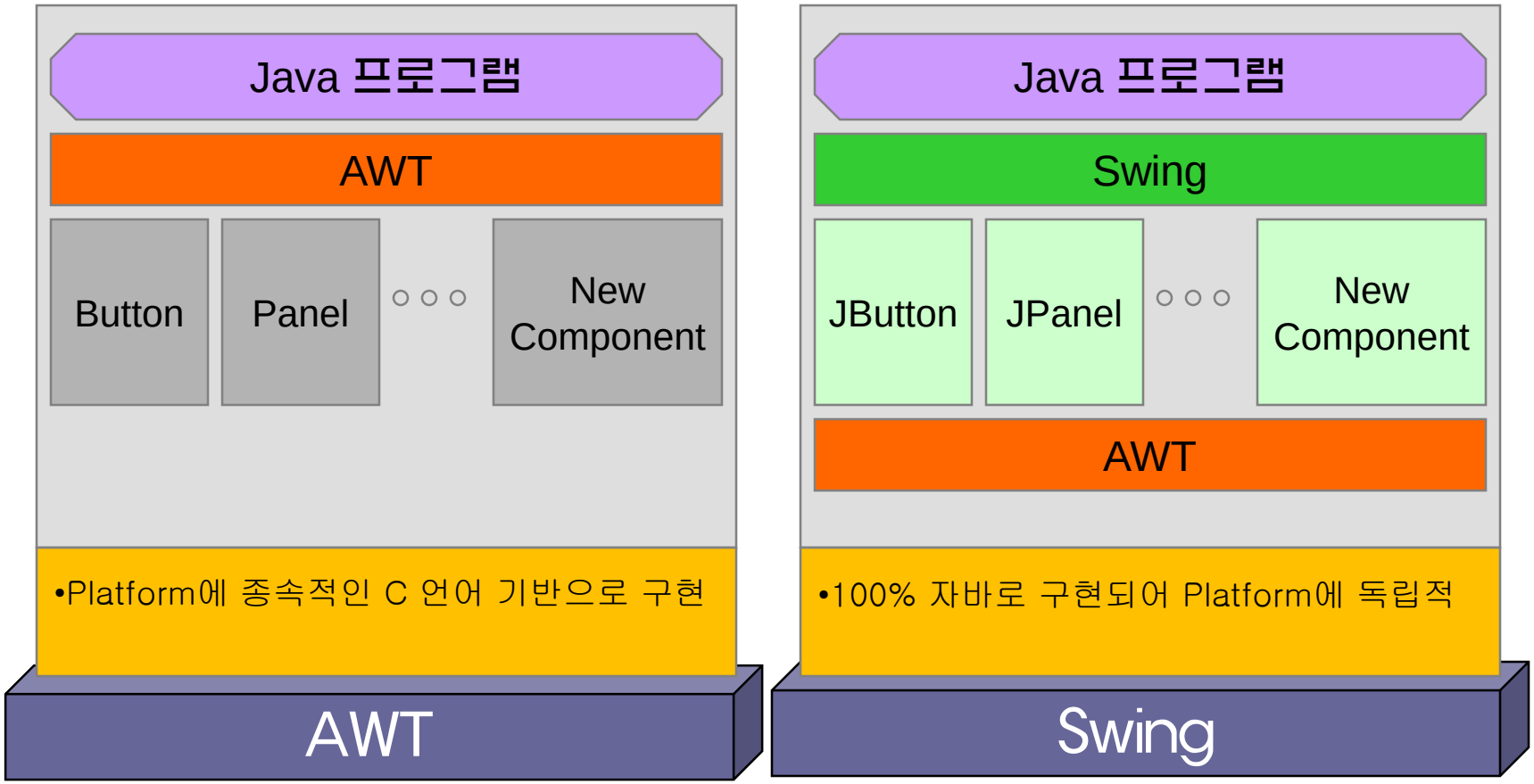
- ◆ 순수한 자바 언어로 지원되는 GUI 개발 도구

❖ 특징

- ◆ Platform에 독립적이다.
- ◆ 새로운 컴포넌트 제작이 쉽다.
- ◆ 풍선도움말(ToolTip) 기능을 제공한다.
- ◆ AWT 컴포넌트를 지원한다.

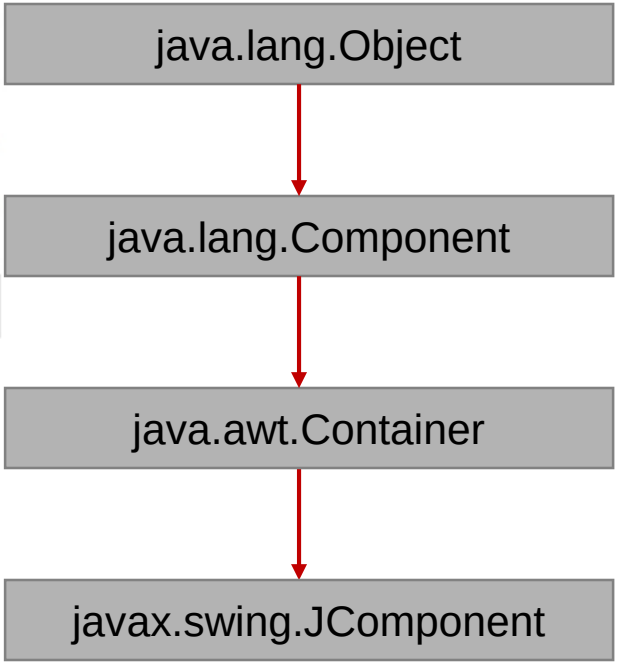
Swing 이란 ? (2)

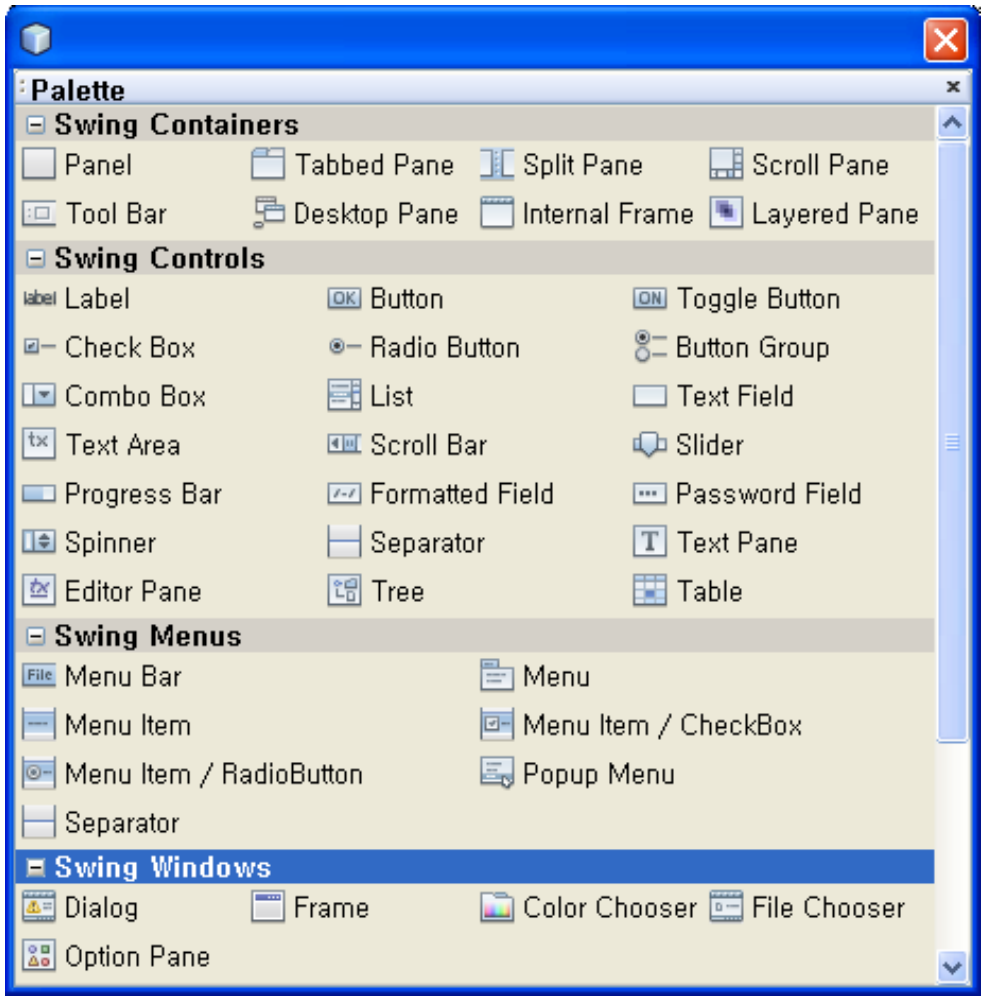
- AWT vs. Swing



Swing 이란 ? (3)

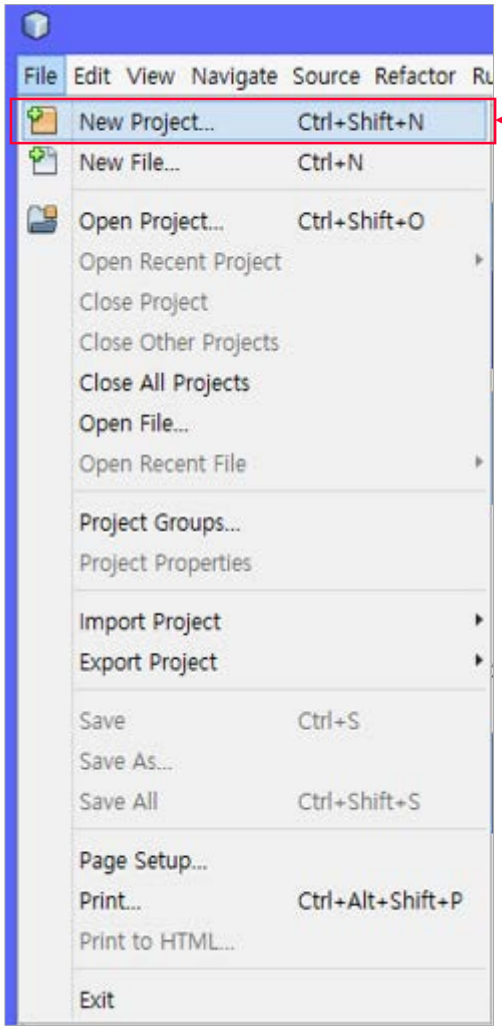
Swing의 Class 구조





Windows Application 개발 방법 (1)

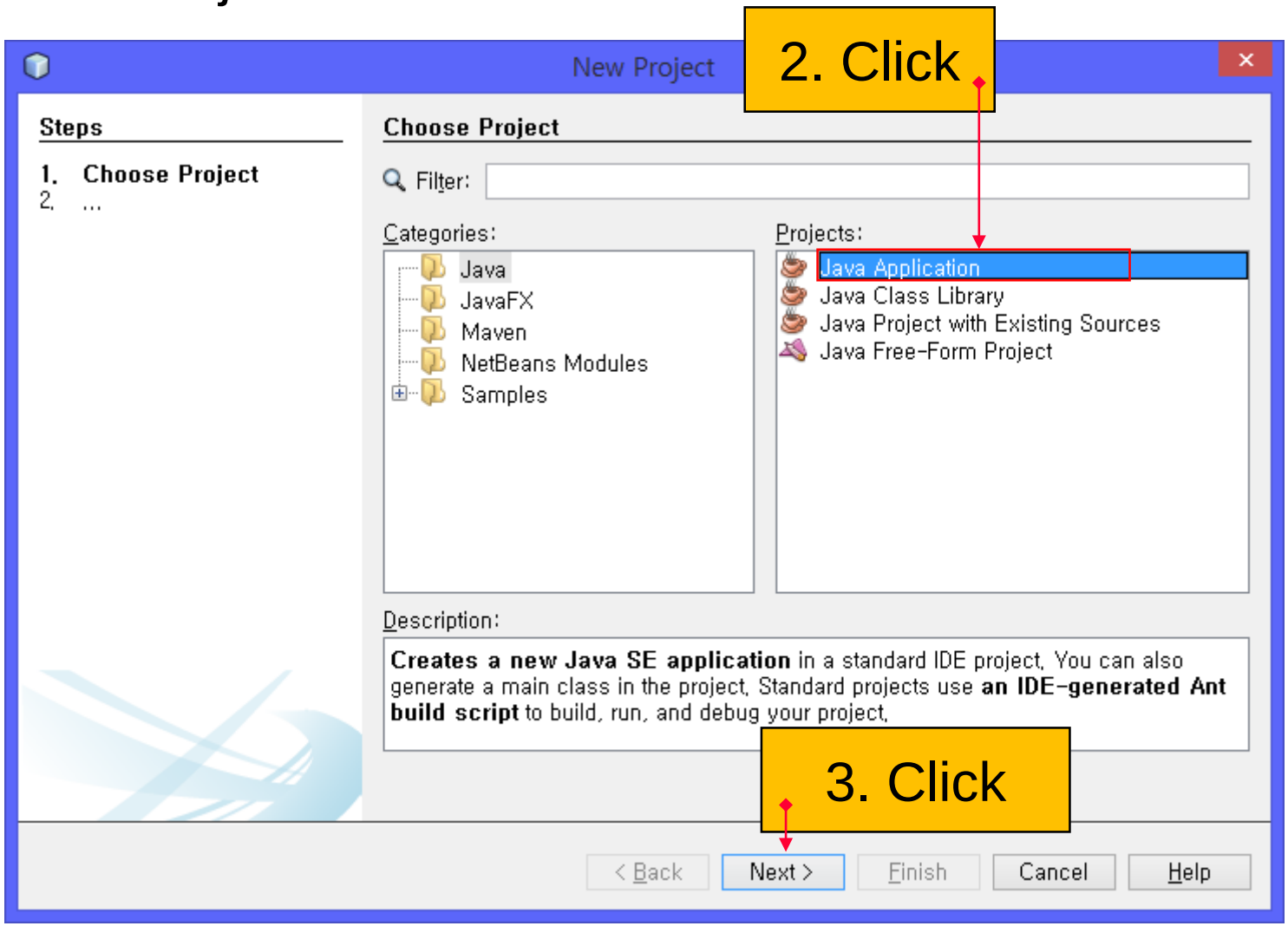
- Application 생성



1. Click

Windows Application 개발 방법 (2)

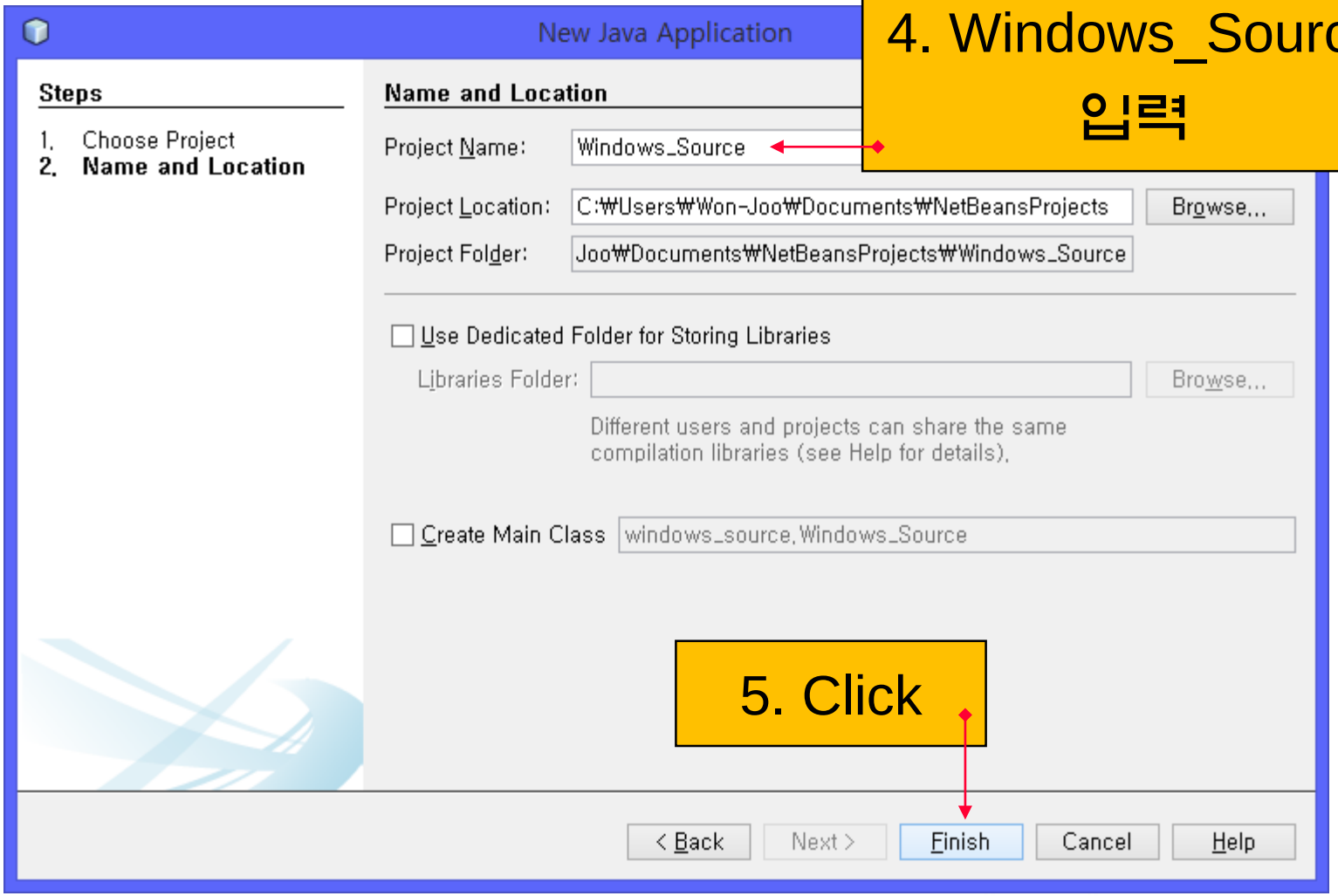
New Project



Windows Application 개발 방법 (3)

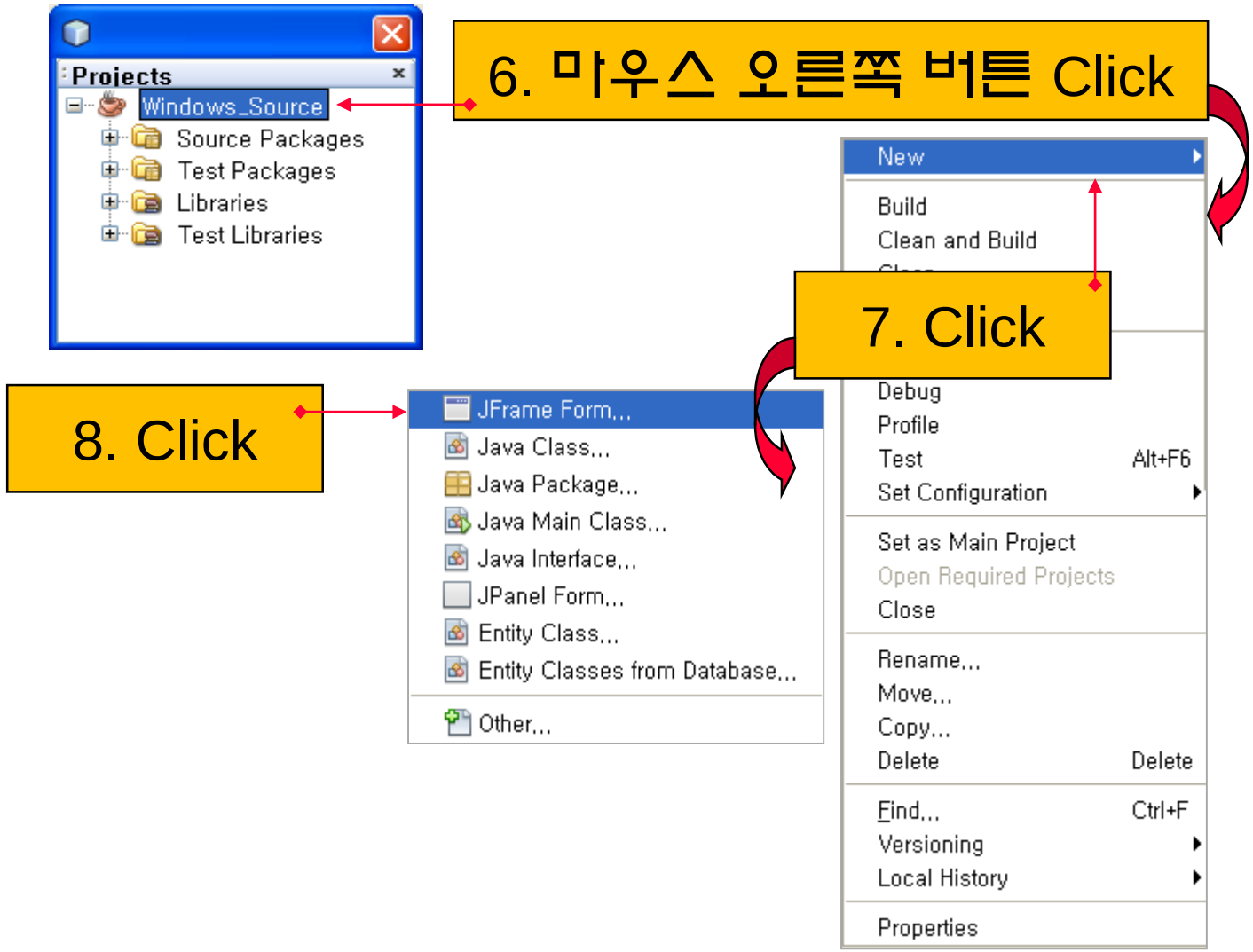
Project Name and Location

- Project name: Windows_Source



Windows Application 개발 방법 (4)

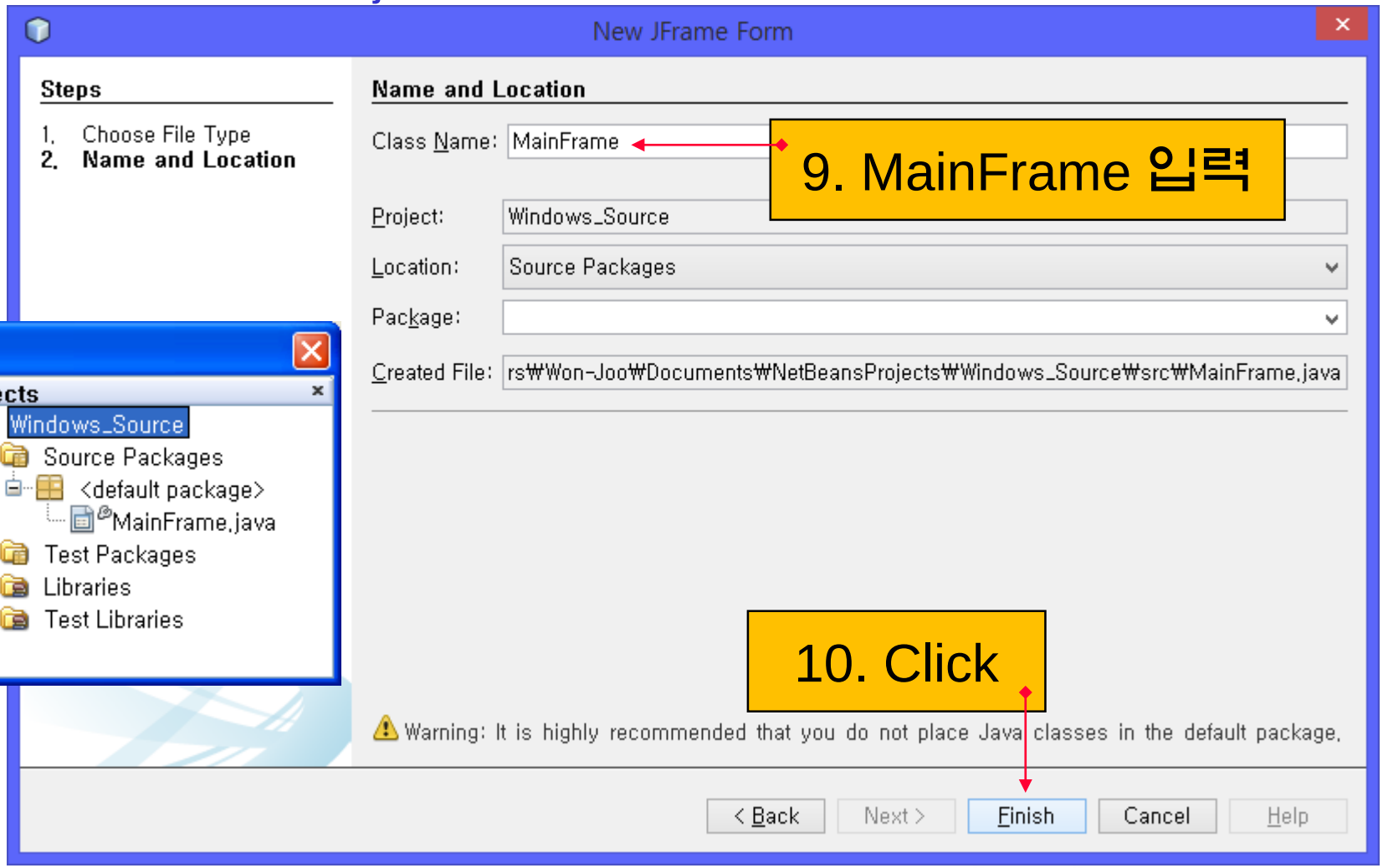
JFrame Form 생성



Windows Application 개발 방법 (5)

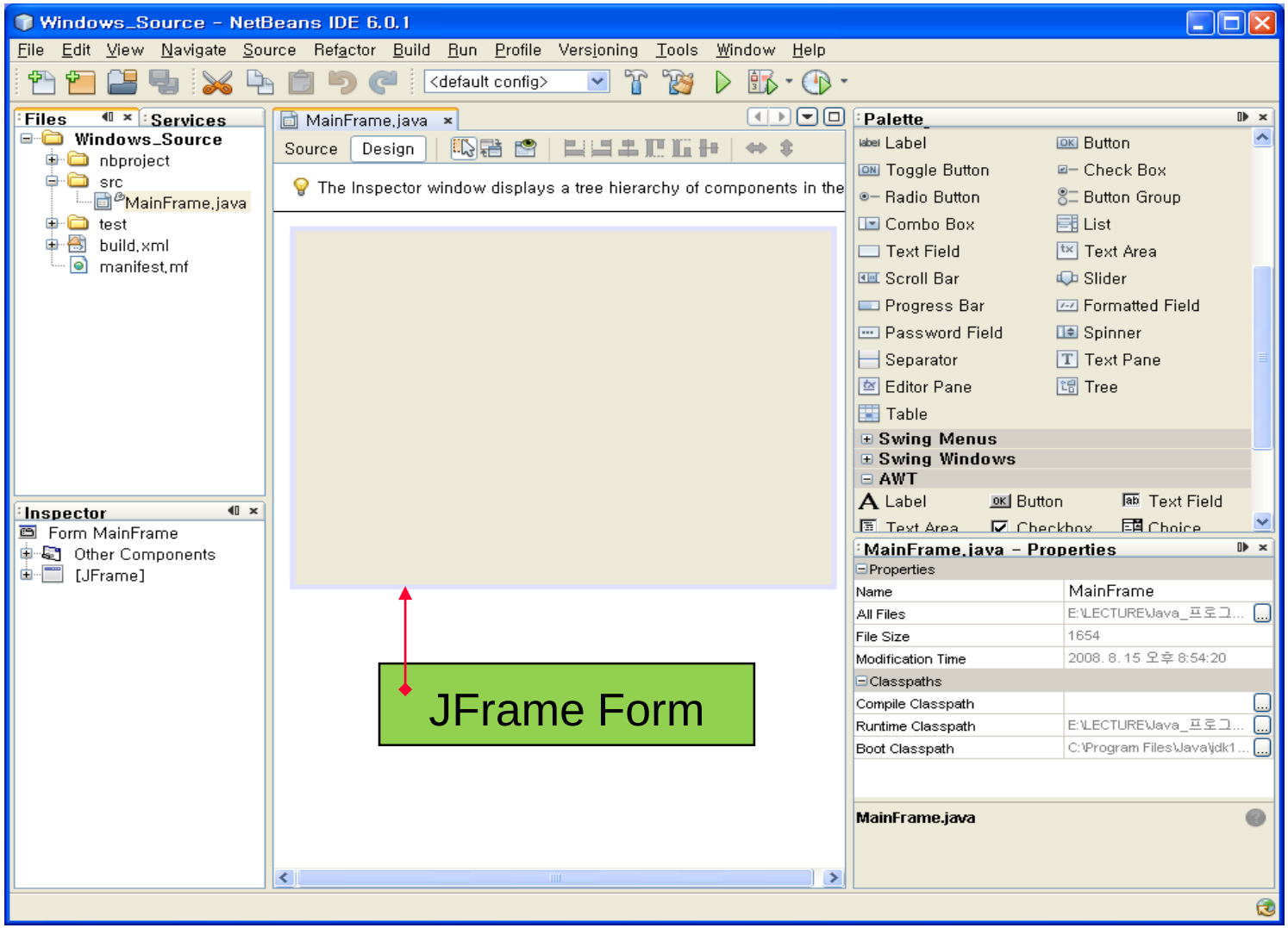
JFrame Form Name 지정

❖ Mainframe.java 생성



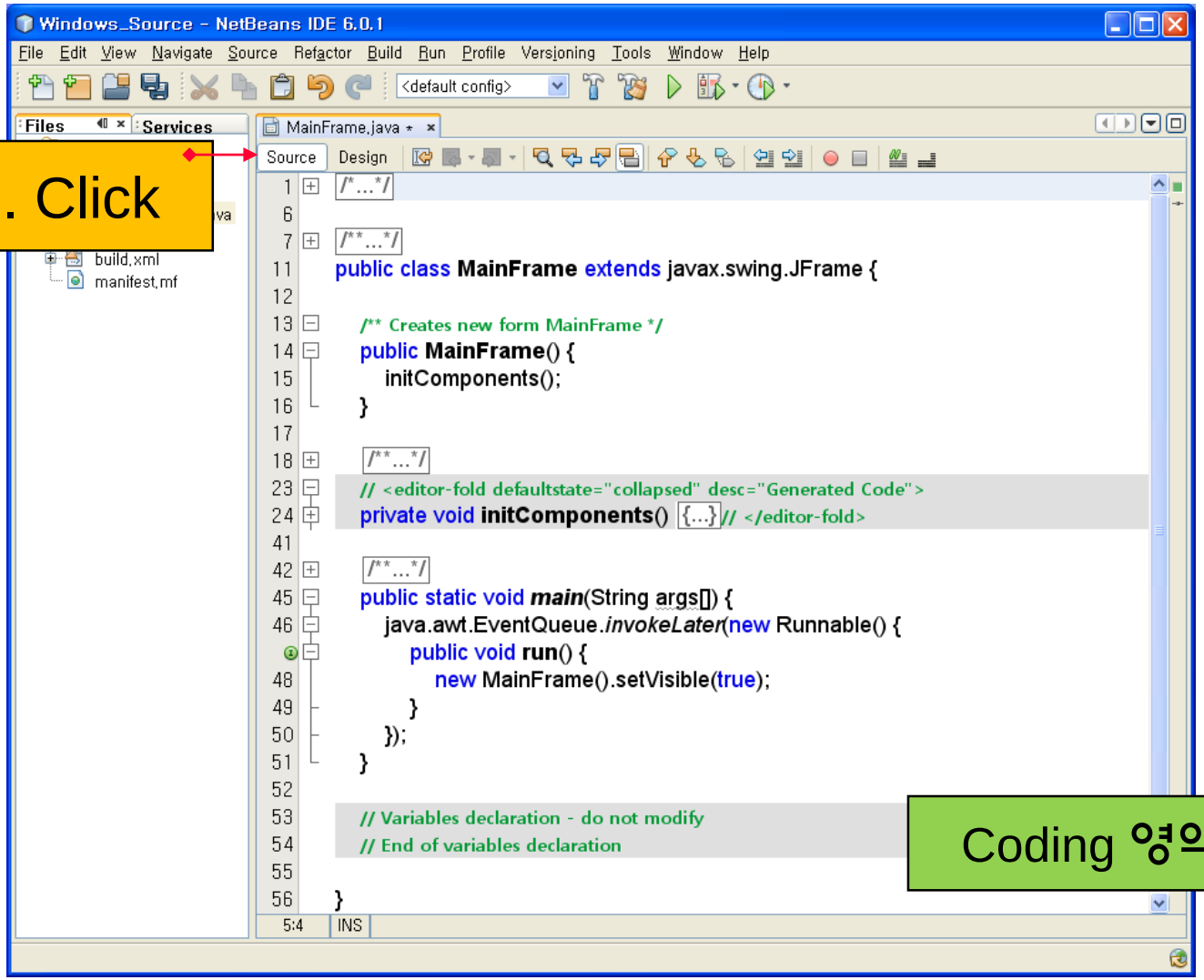
Windows Application 개발 방법 (6)

- Windows_Source Project 생성 완료



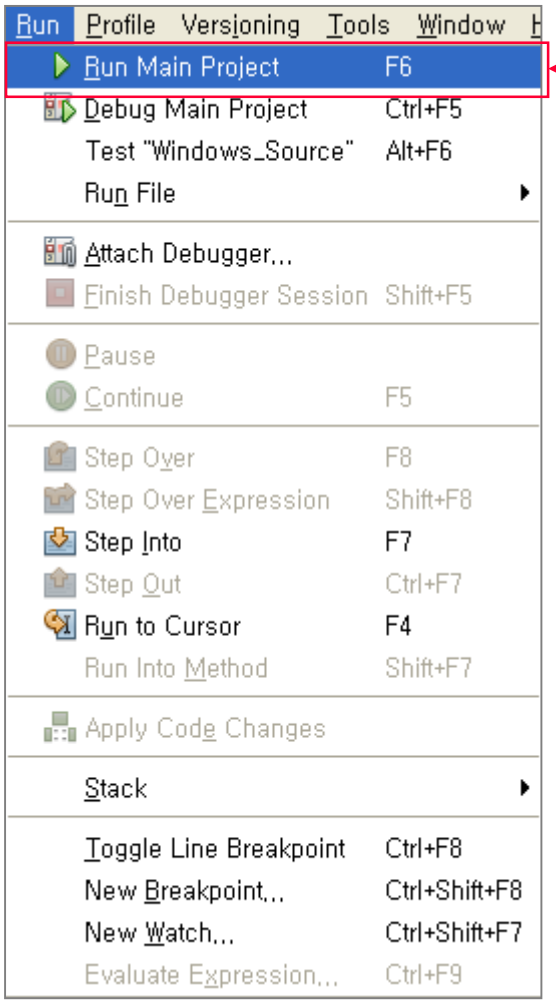
Windows Application 개발 방법 (7)

Source Code

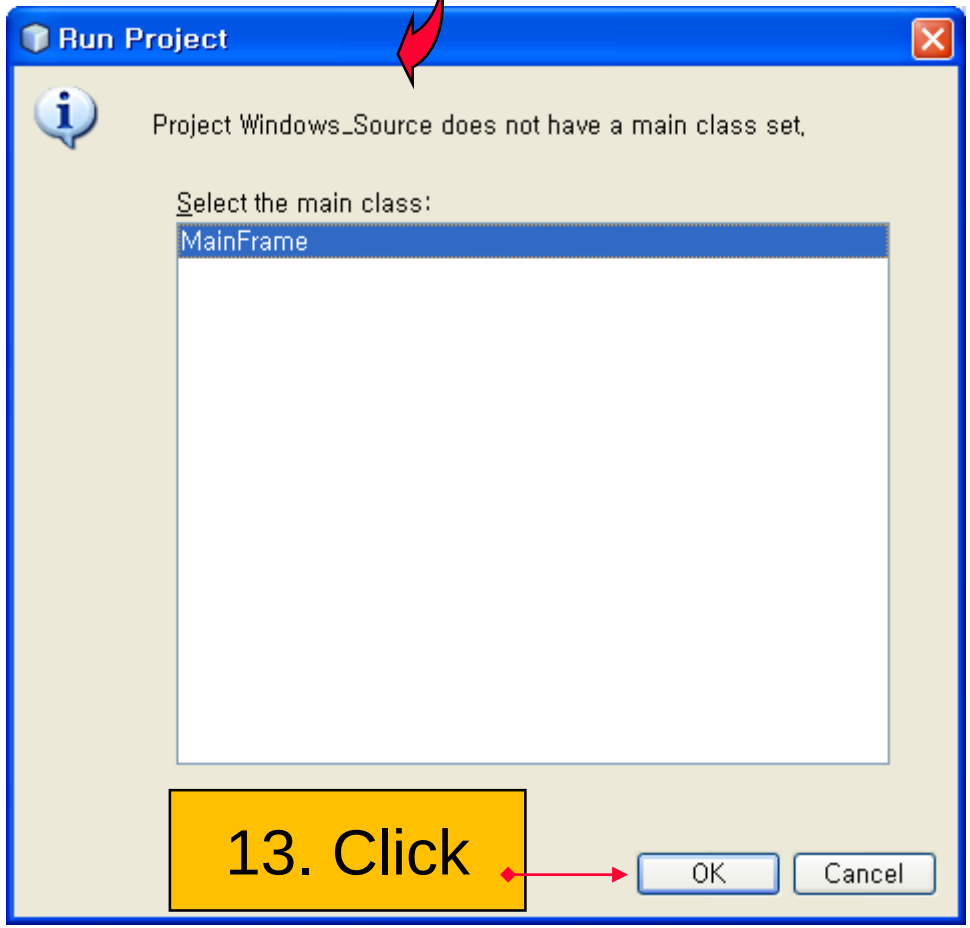


Windows Application 개발 방법 (8)

실행



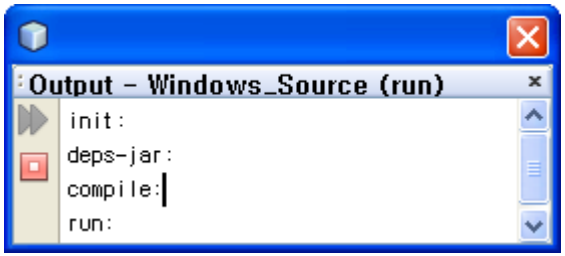
12. Click

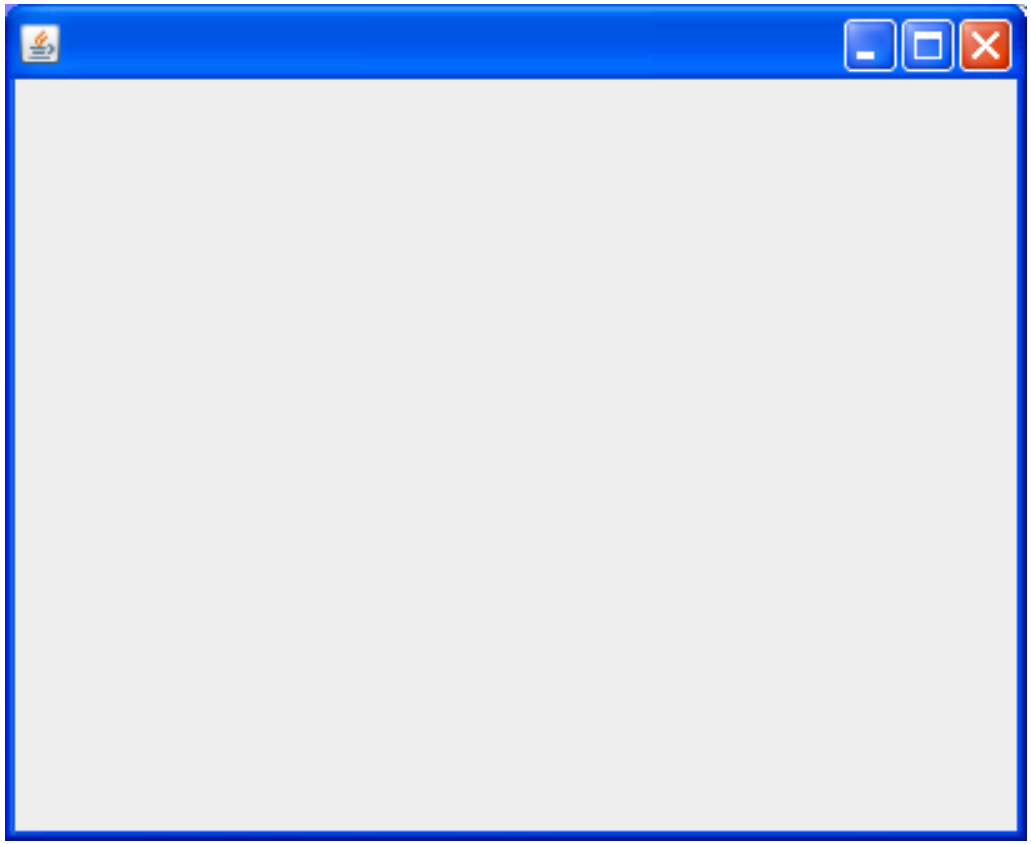


13. Click

Windows Application 개발 방법 (9)

실행 결과



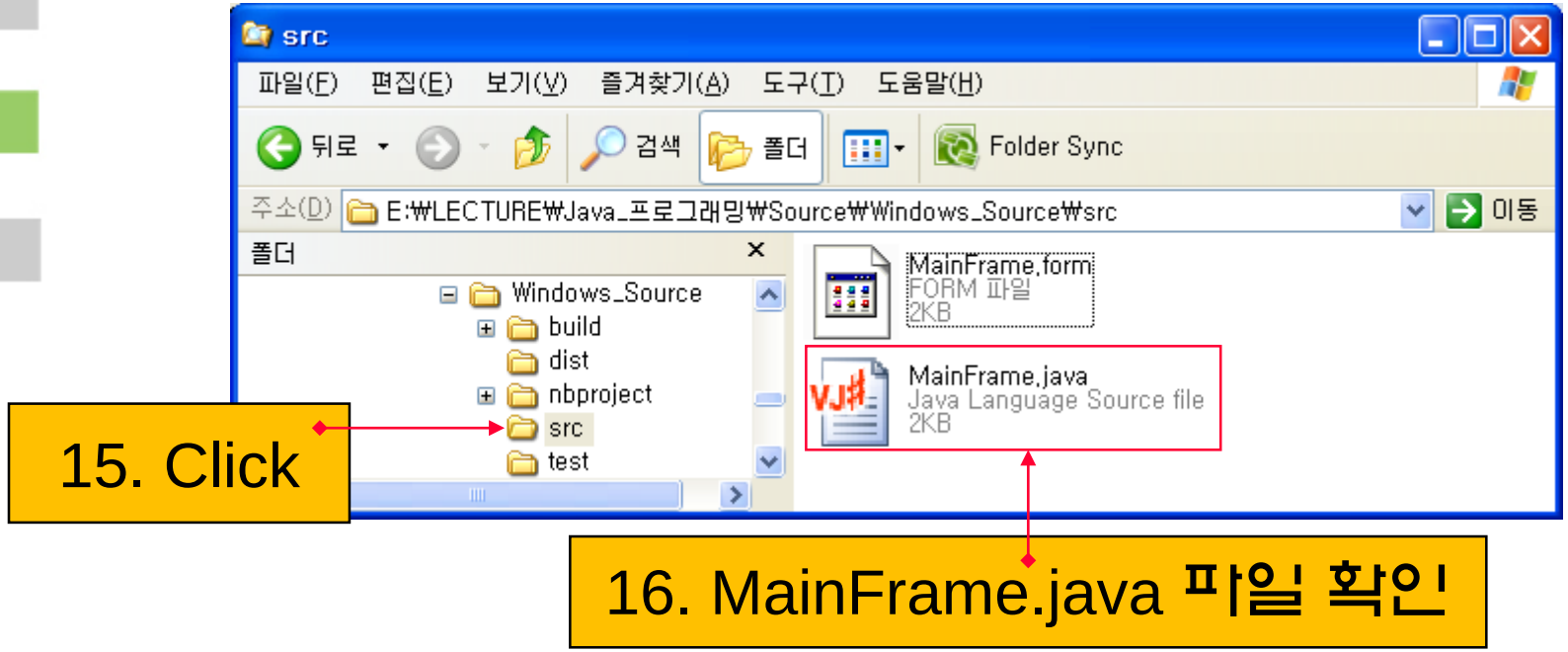


14. 실행 결과 확인

Windows Application 개발 방법 (10)

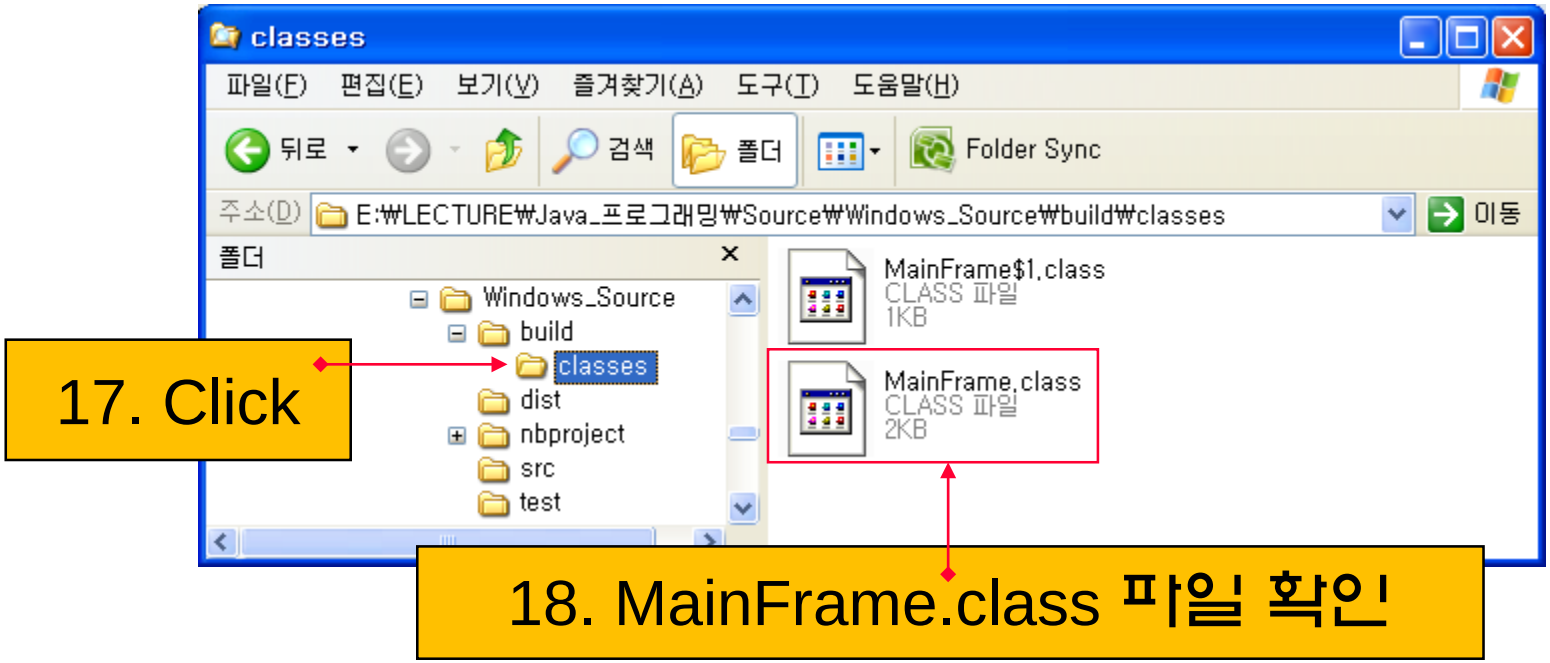
Project 구조

- Java source file



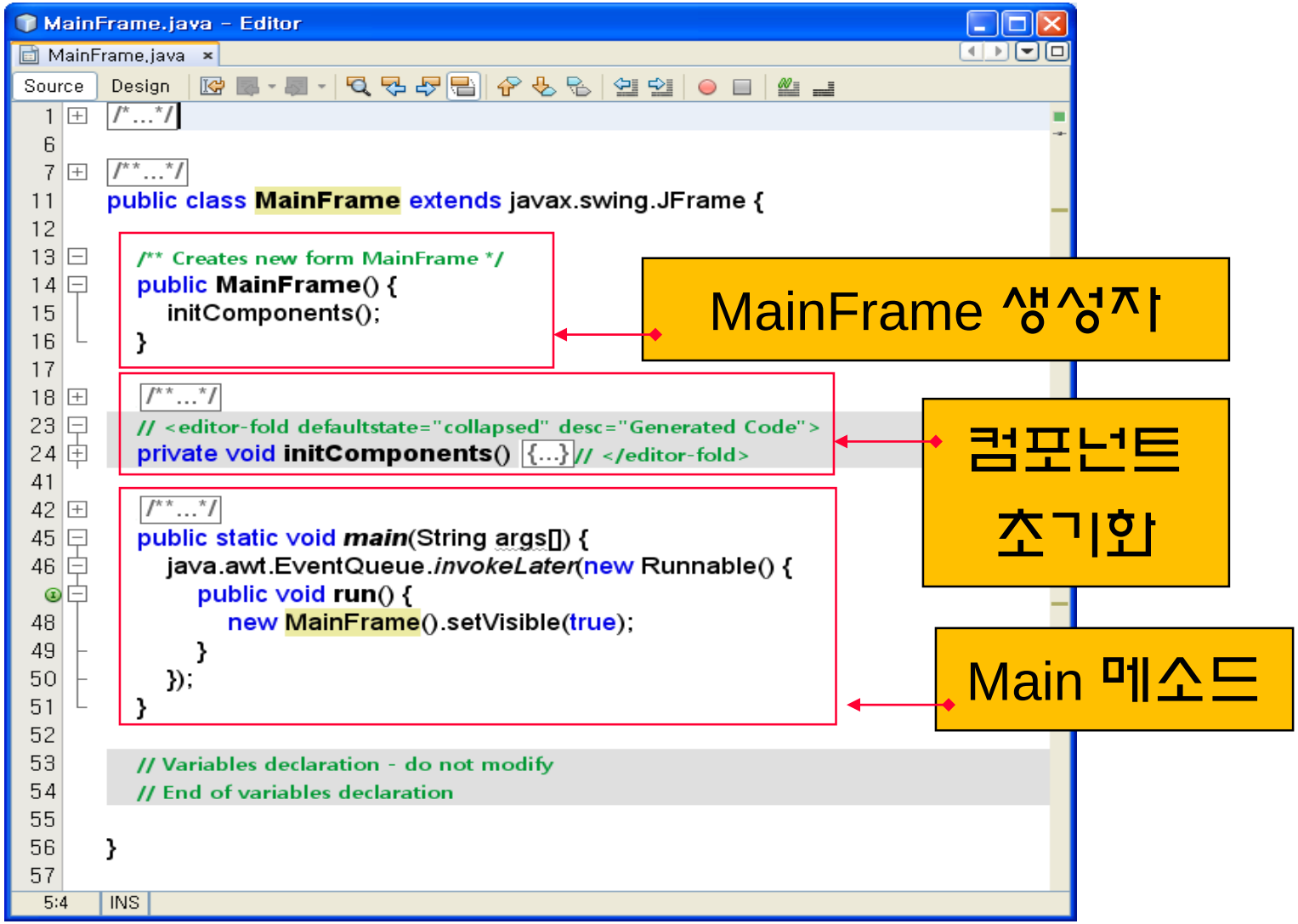
Windows Application 개발 방법 (11)

❖ Class file



Windows Application 소스 분석 (1)

- MainFrame 클래스(Class) Source



Windows Application 소스 분석 (2)

❖ InitComponents() 메소드

MainFrame.java - Editor

MainFrame.java ×

Source | Design

```
18  /**...*/
23  // <editor-fold defaultstate="collapsed" desc="Generated Code">
24  private void initComponents() {

26      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

28      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
29      getContentPane().setLayout(layout);
30      layout.setHorizontalGroup(
31          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
32          .addGap(0, 400, Short.MAX_VALUE)
33      );
34      layout.setVerticalGroup(
35          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
36          .addGap(0, 300, Short.MAX_VALUE)
37      );

39      pack();
40  }// </editor-fold>
```

44:8 INS

Event 란 ? (1)

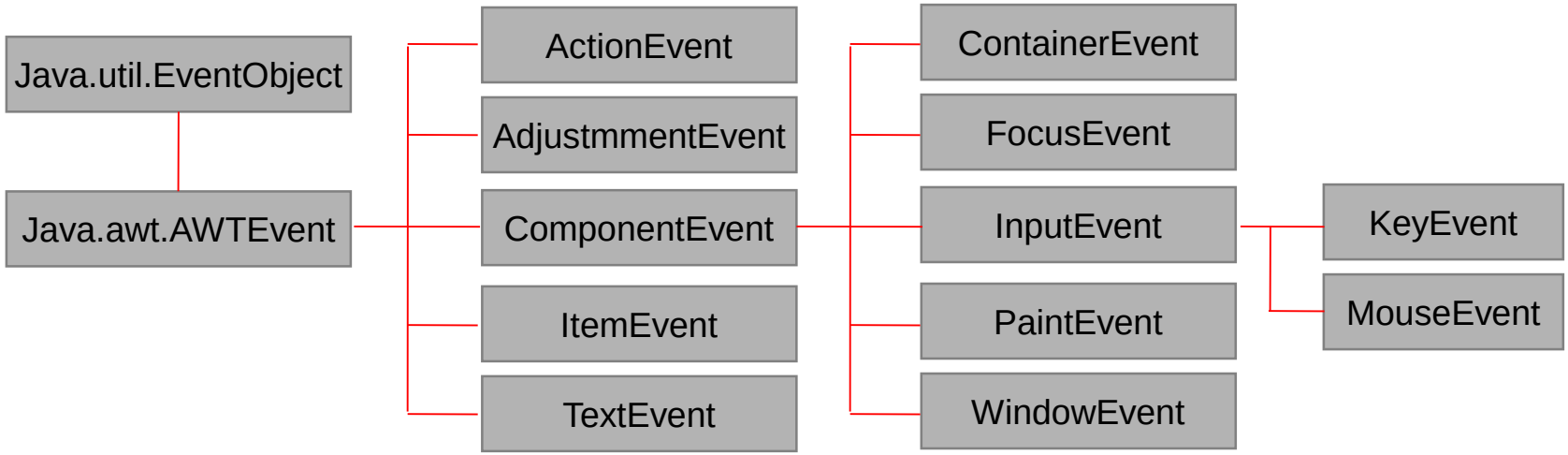
- Event
  - ❖ 정의
    - ◆ 윈도우의 컴포넌트를 Click할 때 발생하는 메세지
  - ❖ Event 처리 과정

Event 란 ? (2)

- Event 구성
  - Event Source
    - Event를 발생시키는 Button, Scrollbar, Mouse, Keyboard,.. 등의 컴포넌트
  - Event Class
    - 특정 컴포넌트에 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	내 용
ActionEvent	컴포넌트가 활성화될 때 발생
AdjustmentEvent	스크롤바와 같이 조정 가능한 컴포넌트에서 조정이 있을 때 발생
ContainerEvent	Container에 컴포넌트가 추가/삭제되는 경우 발생
FocusEvent	컴포넌트에 focus가 들어왔을 때 발생
ItemEvent	List, choice, .. 등의 컴포넌트에서 선택항목이 선택될 대 발생
KeyEvent	키보드 입력에 의해서 발생
MouseEvent	Mouse 움직임에 의해서 발생
PaintEvent	컴포넌트가 그려져야 할 때 발생
TextEvent	Text 컴포넌트의 내용이 변화할 때 발생
WindowEvent	Window 활성화, 또는 종료할 때 발생

Event 란 ? (5)

컴포넌트-Event 관계

컴포넌트	컴포넌트에서 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)

컴포넌트	컴포넌트에서 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)

컴포넌트	컴포넌트에서 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 인터페이스
  - 각 이벤트를 처리하기 위해, 준비된 메서드를 선언한 인터페이스
  - 이벤트 이름 + Listener
    - ActionEvent = ActionListener, FocusEvent=FocusListener
- 예제

```
class MyActionListener implements ActionListener
{
    public void actionPerformed(ActionEvent ae)
    {
        System.out.println("Action 이벤트가 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)

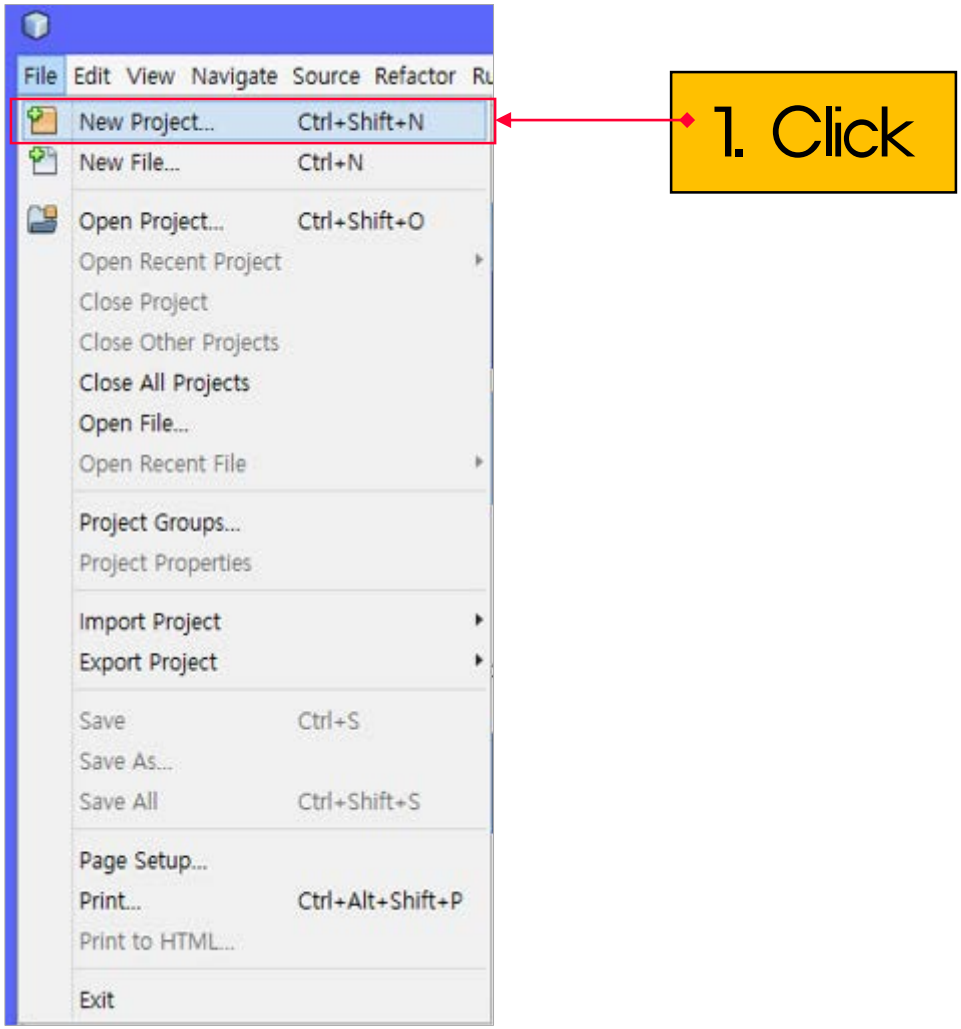
Listener Interface Adapter	Method
ItemListener	ItemStateChanged(ItemEvent)
KeyListener KeyAdapter	KeyPressed(KeyEvent) KeyReleased(KeyEvent) KeyTyped(KeyEvent)
MouseListener MouseAdapter	mouseClicked(MouseEvent) mouseEntered(MouseEvent) mouseExited(MouseEvent) mousePressed(MouseEvent) mouseReleased(MouseEvent)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

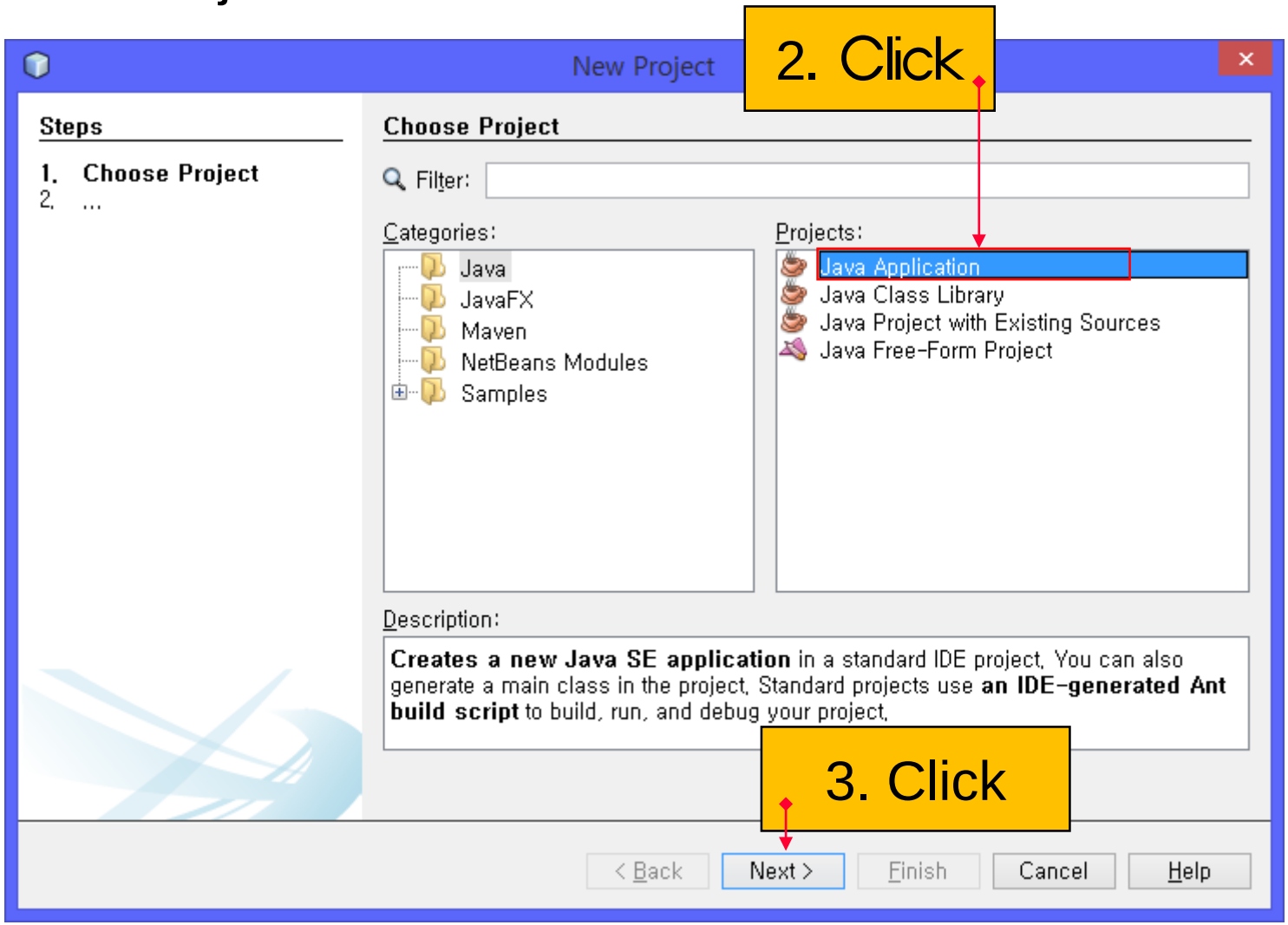
실습: Event Handler 구현 (1)

- Application 생성



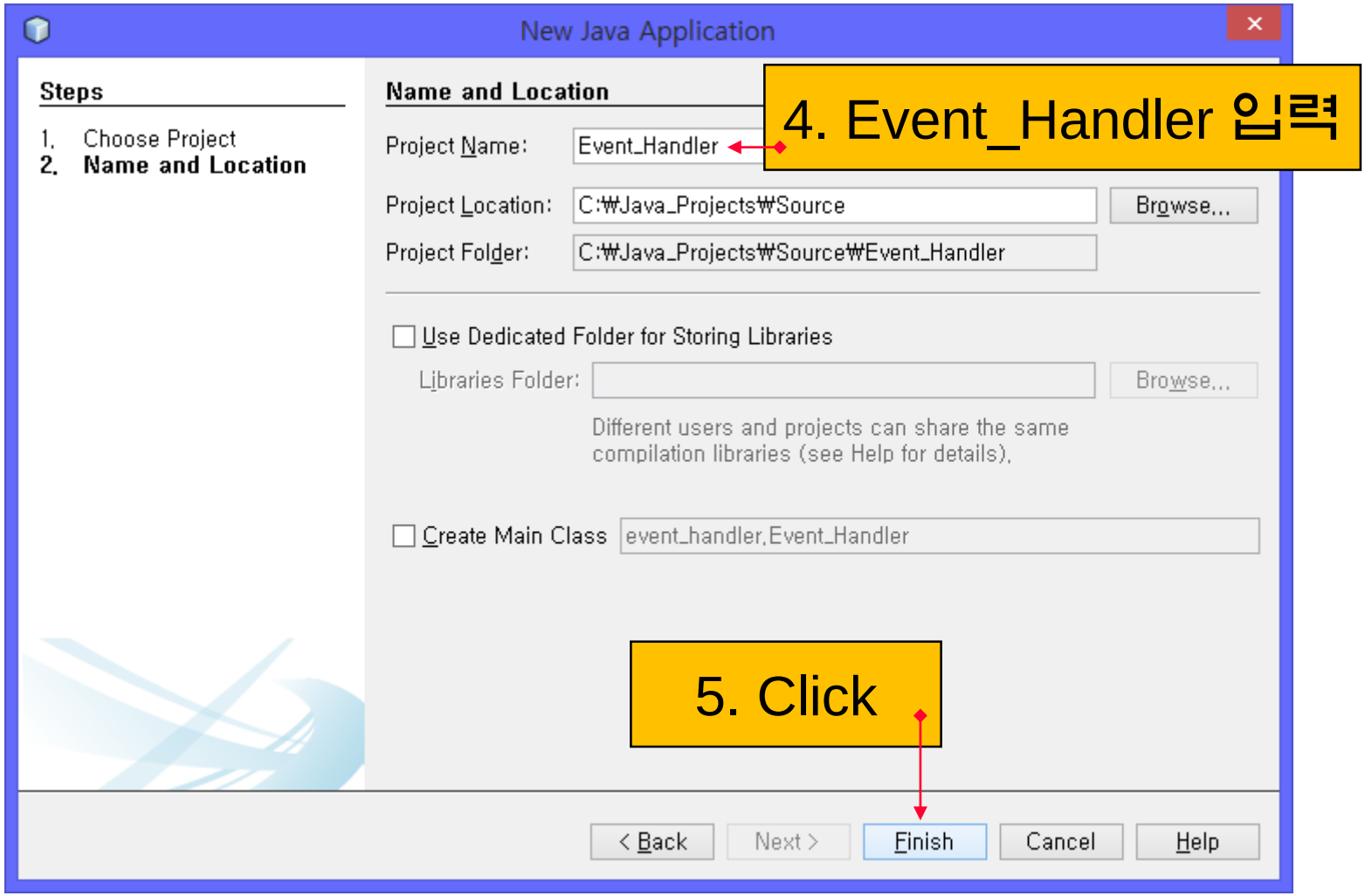
실습: Event Handler 구현 (2)

New Project



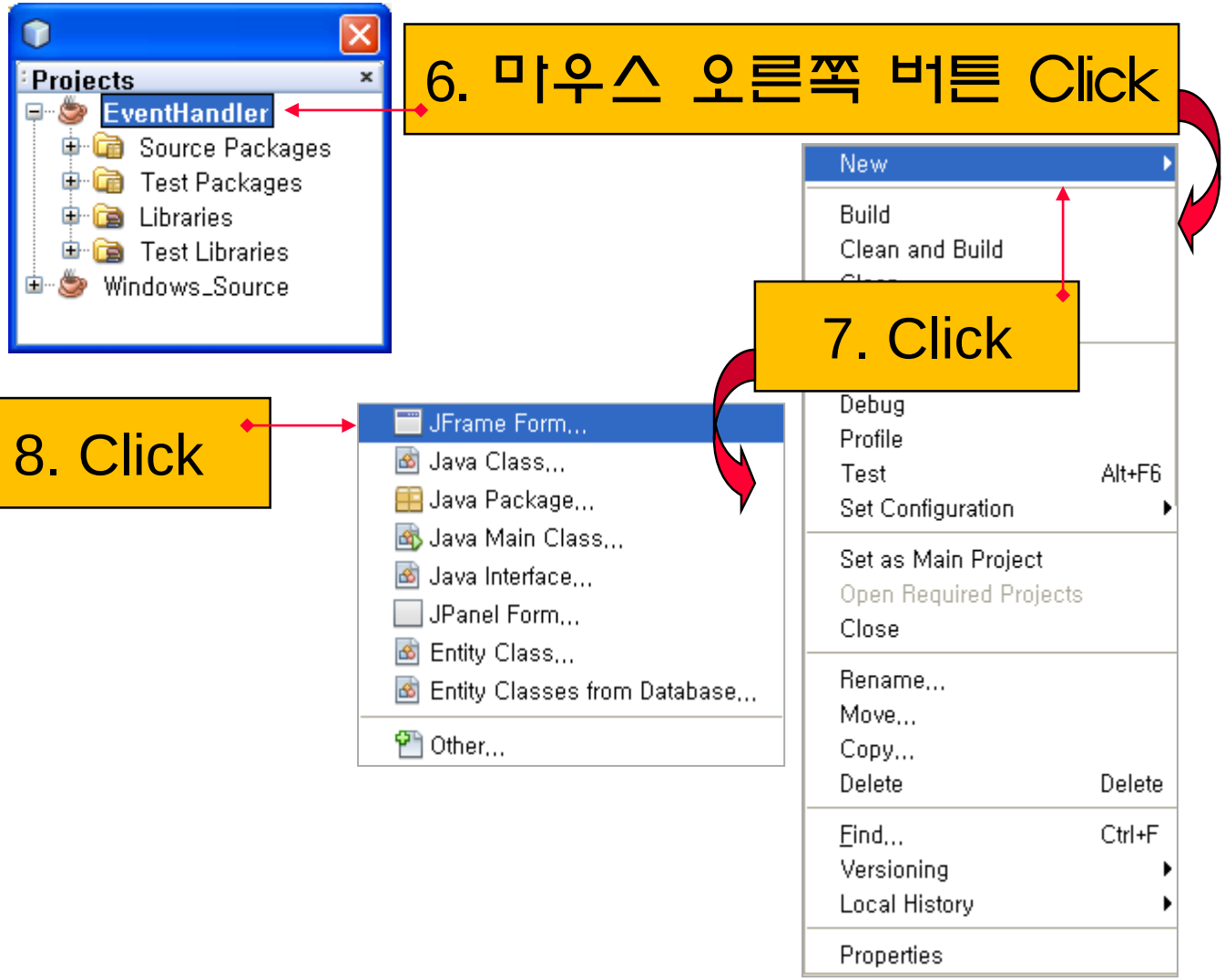
실습: Event Handler 구현 (3)

- Project Name and Location
  - Project name: Event_Handler



실습: Event Handler 구현 (4)

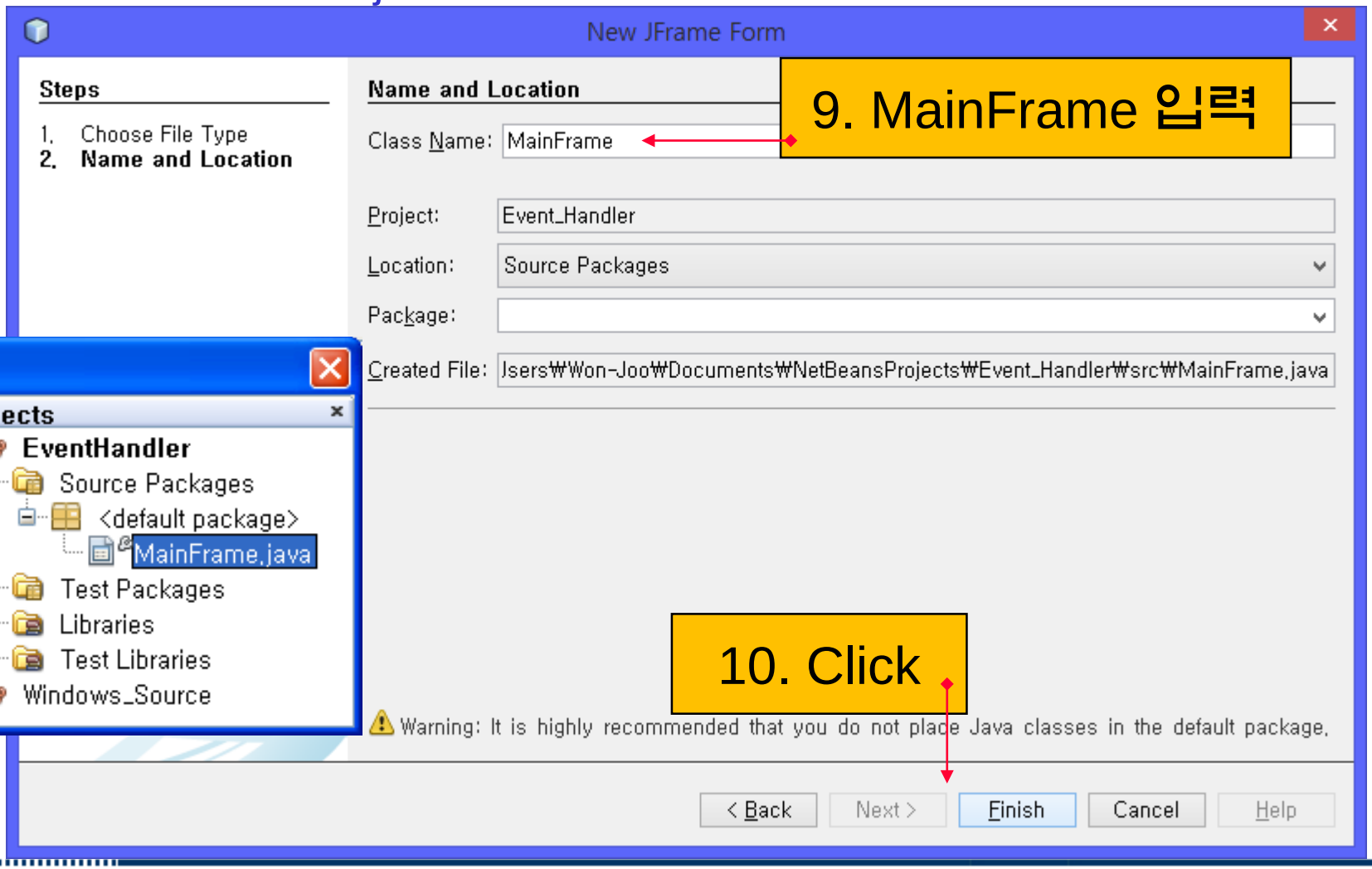
- JFrame Form 생성



실습: Event Handler 구현 (5)

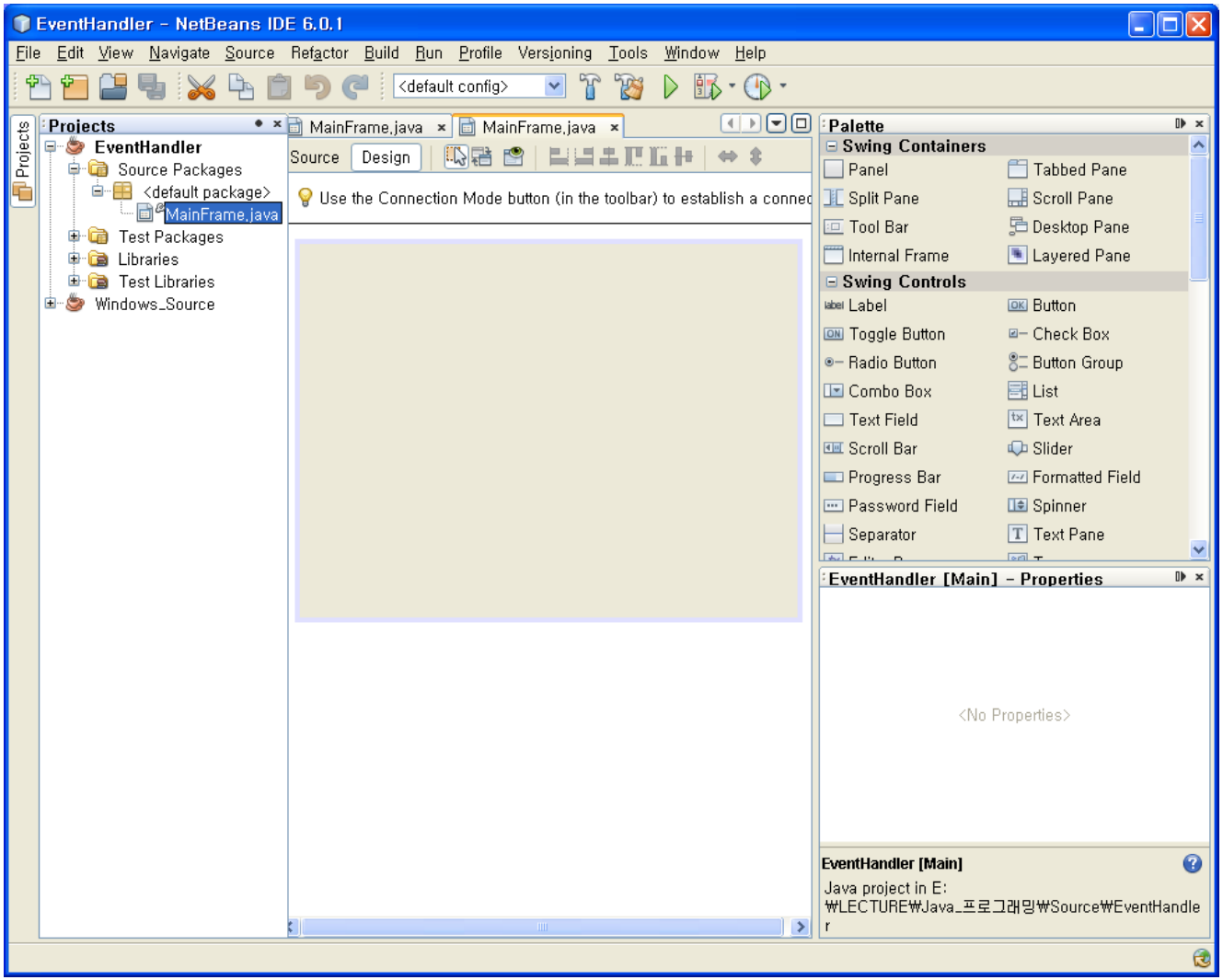
JFrame Form Name 지정

- Mainframe.java 생성



실습: Event Handler 구현 (6)

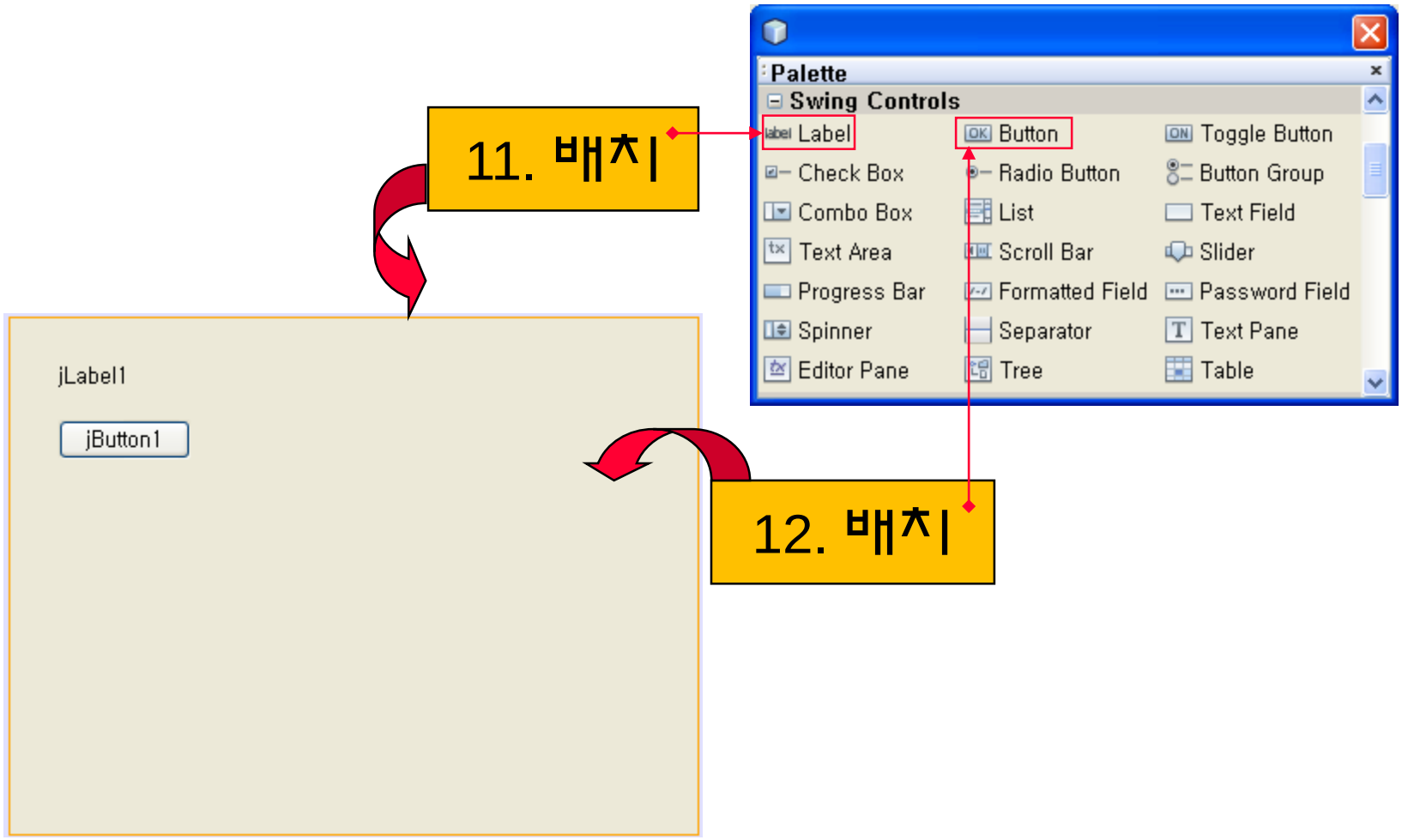
- Event_Handler Project 생성 완료



실습: Event Handler 구현 (7)

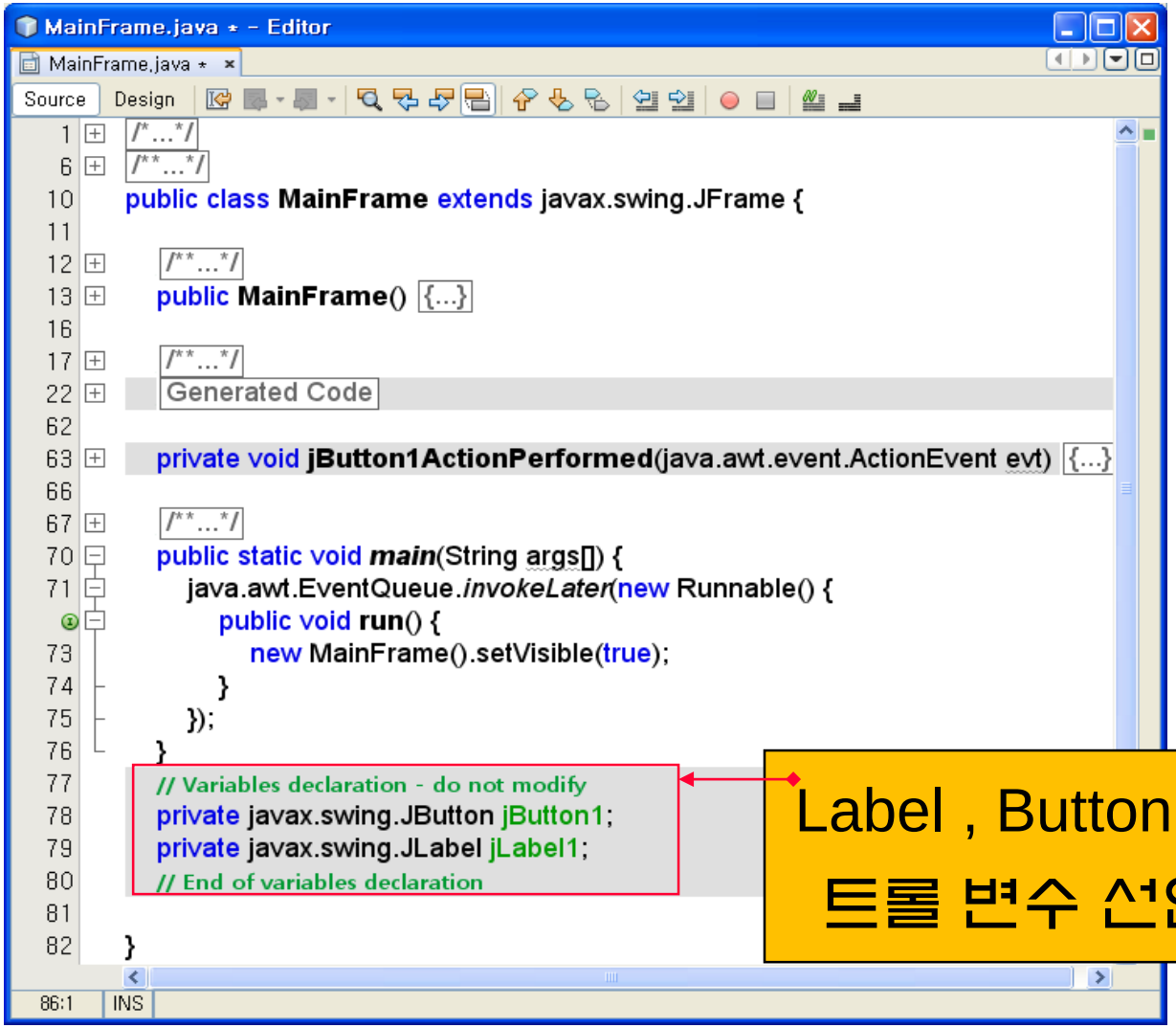
GUI 구현

- Label 및 Button 배치



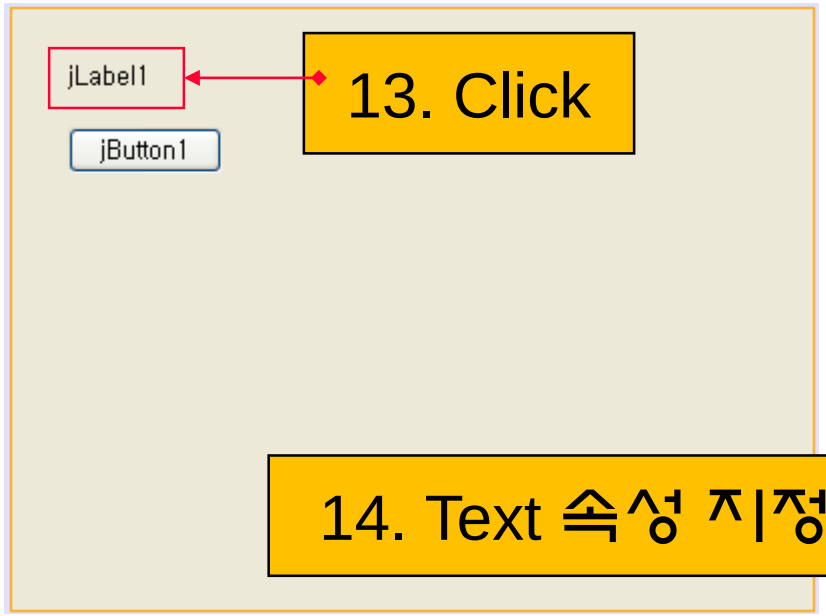
실습: Event Handler 구현 (8)

- Label 및 Button 배치에 따른 Source Coding

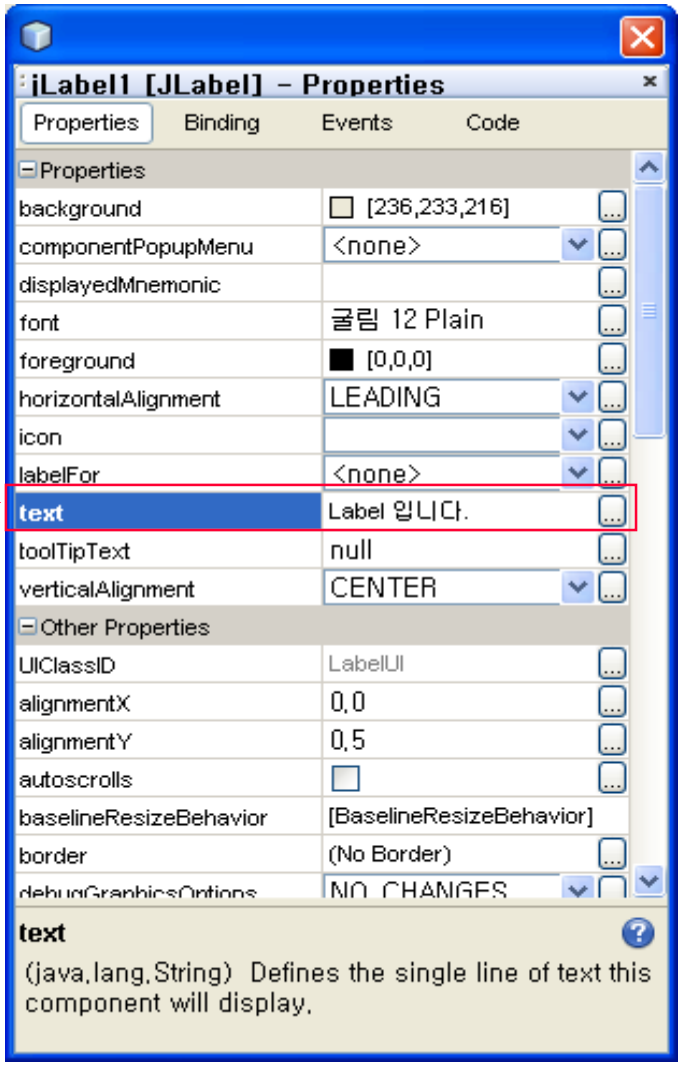


실습: Event Handler 구현 (9)

❖ Label 속성 지정

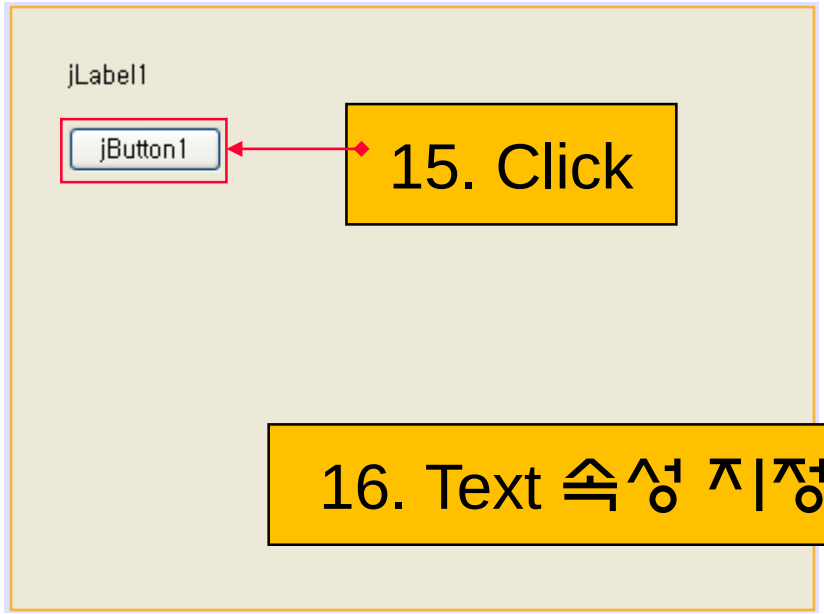


컨트롤	속성지정
Label	•Text: Label 입니다.

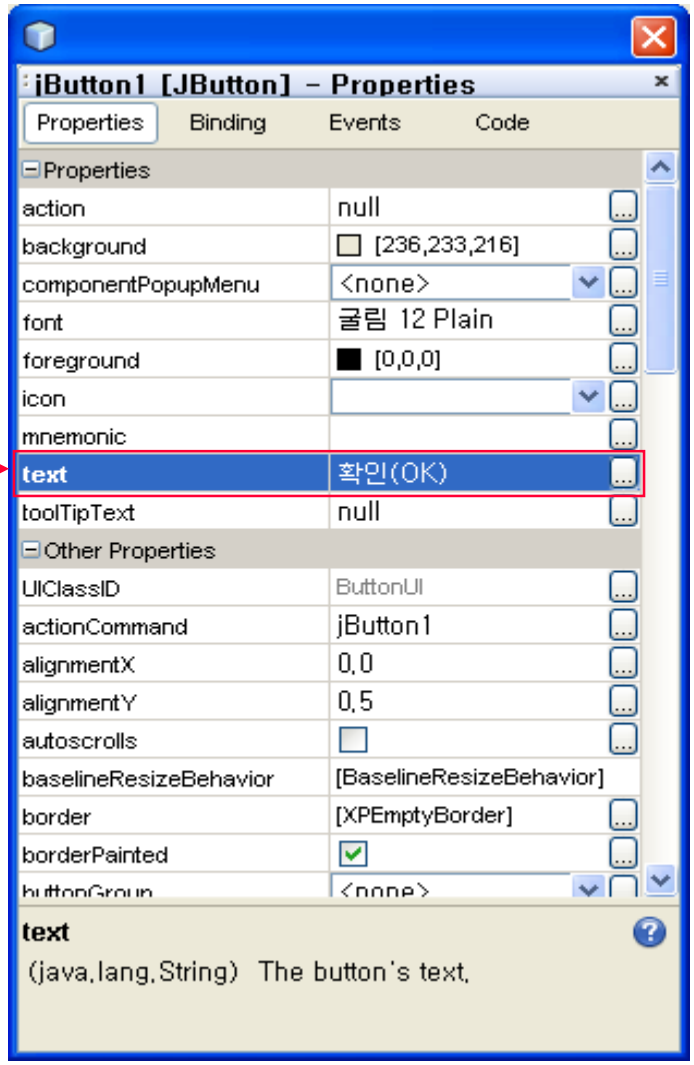


실습: Event Handler 구현 (10)

❖ Button 속성 지정

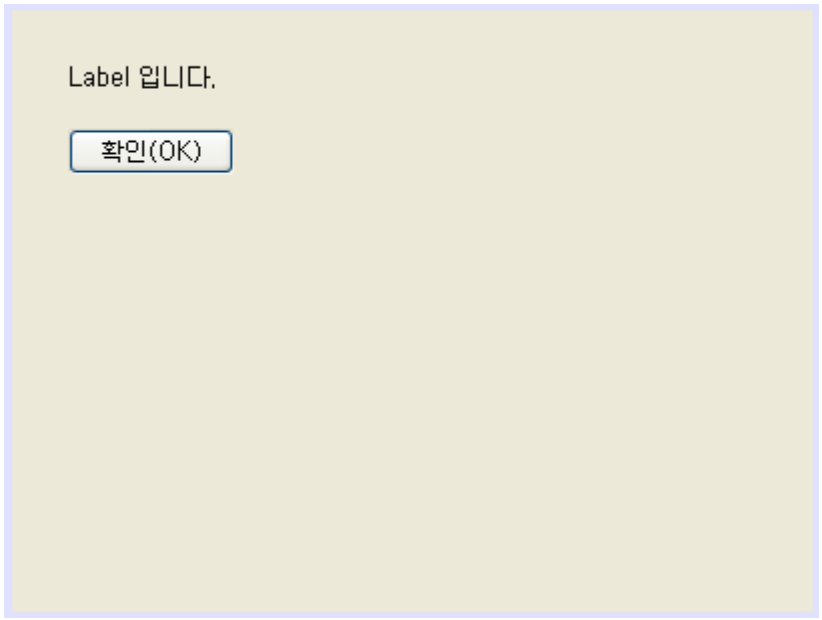


컨트롤	속성지정
Button	•Text: 확인(OK)



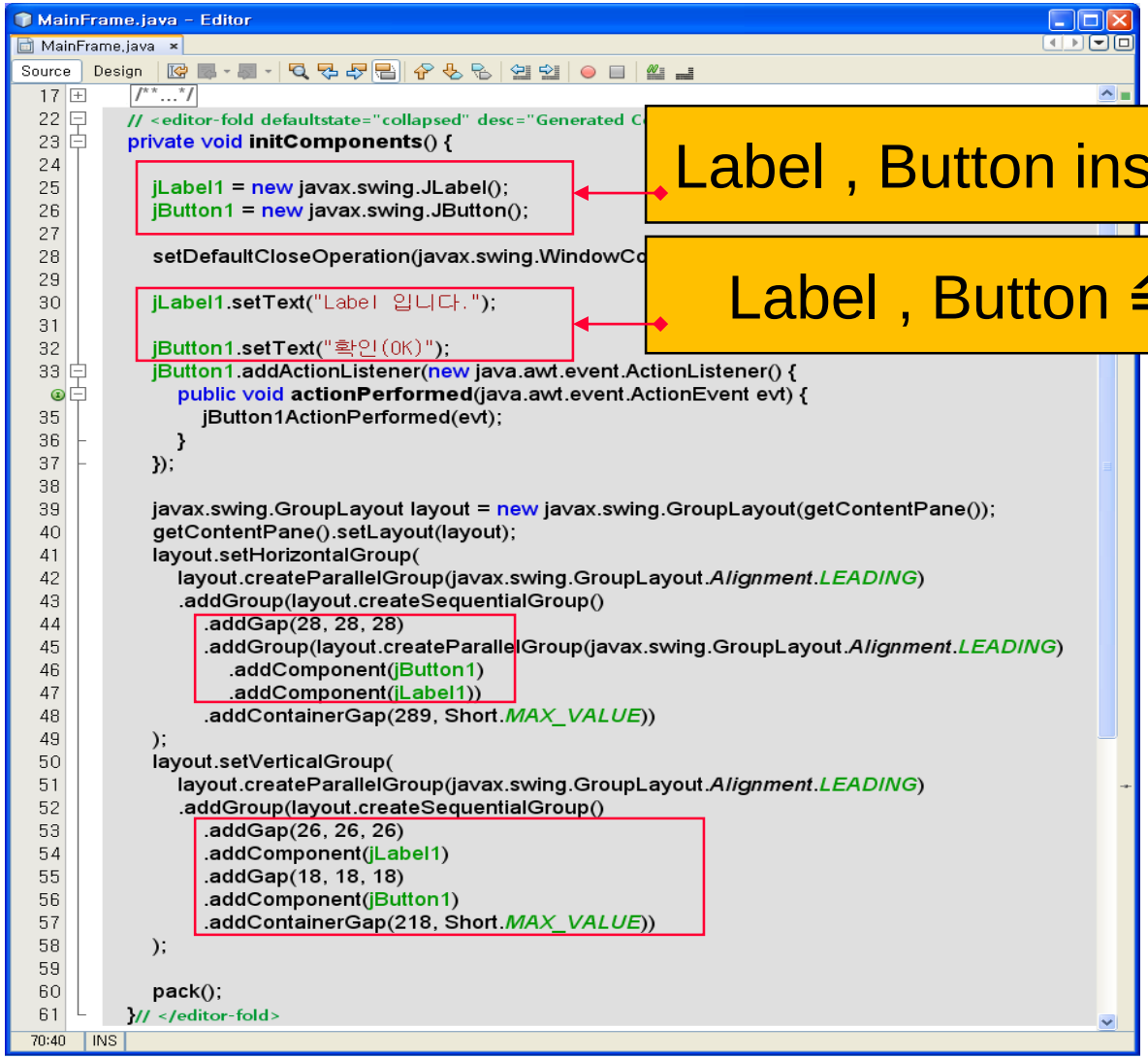
실습: Event Handler 구현 (11)

- GUI 구현 완료



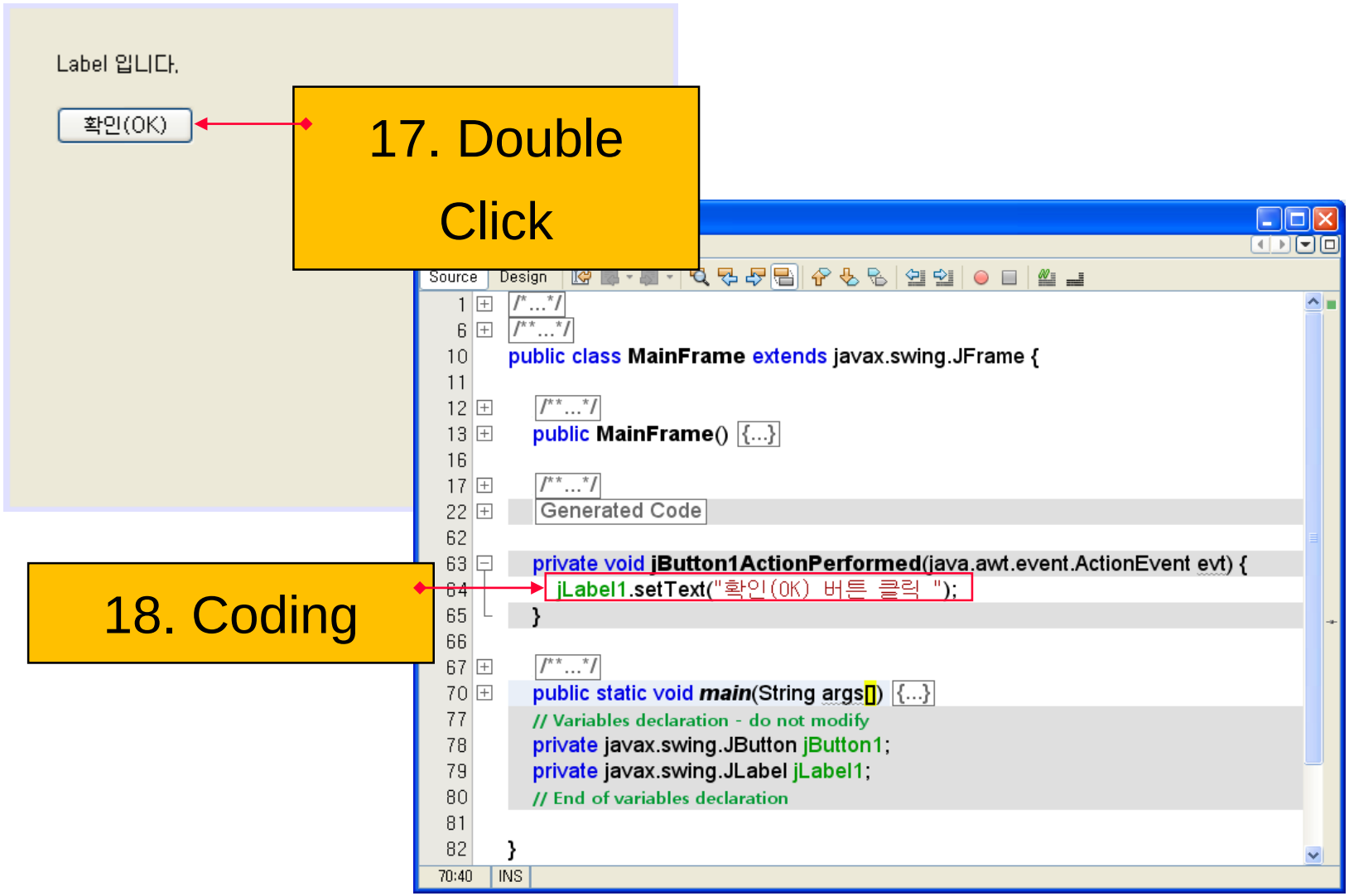
실습: Event Handler 구현 (12)

❖ Label 및 Button 속성 지정에 따른 Source Coding



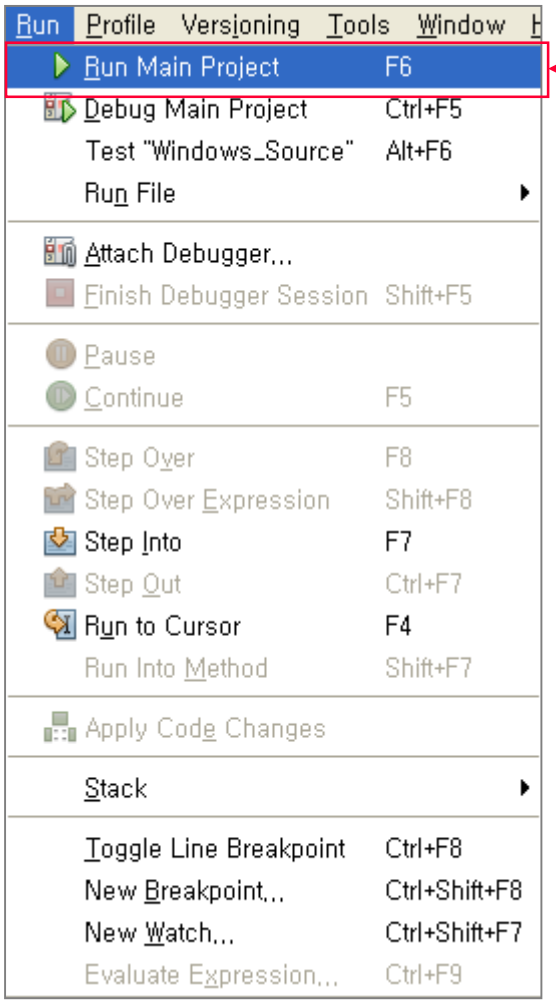
실습: Event Handler 구현 (13)

[확인(OK)] Button Event Handler 구현

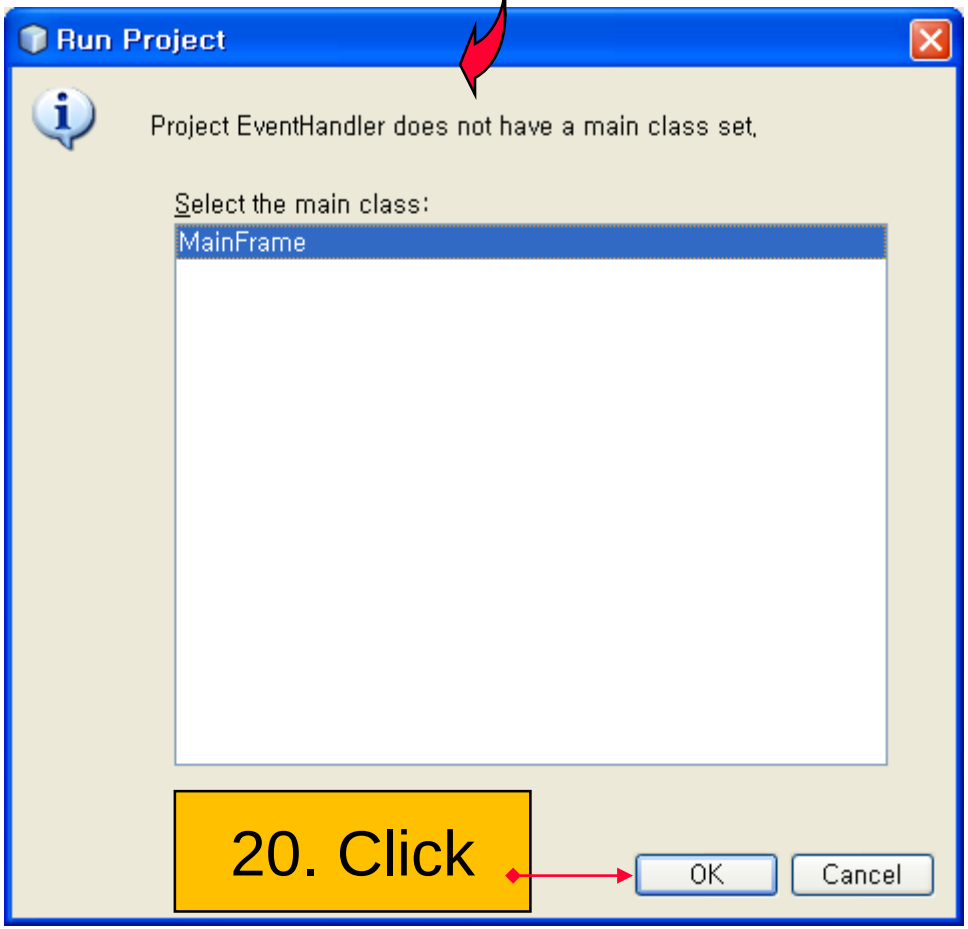


실습: Event Handler 구현 (14)

실행

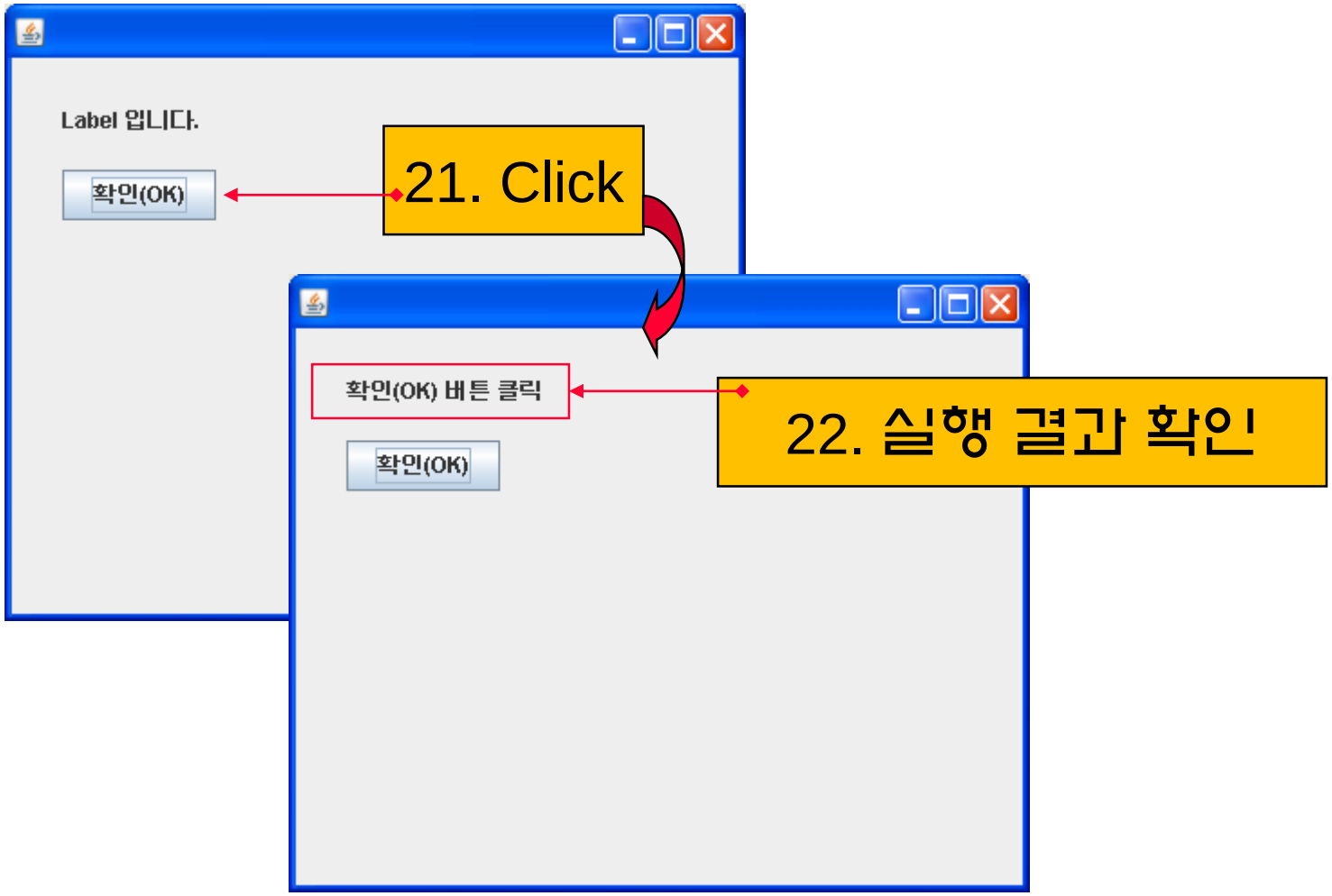


19. Click



실습: Event Handler 구현 (15)

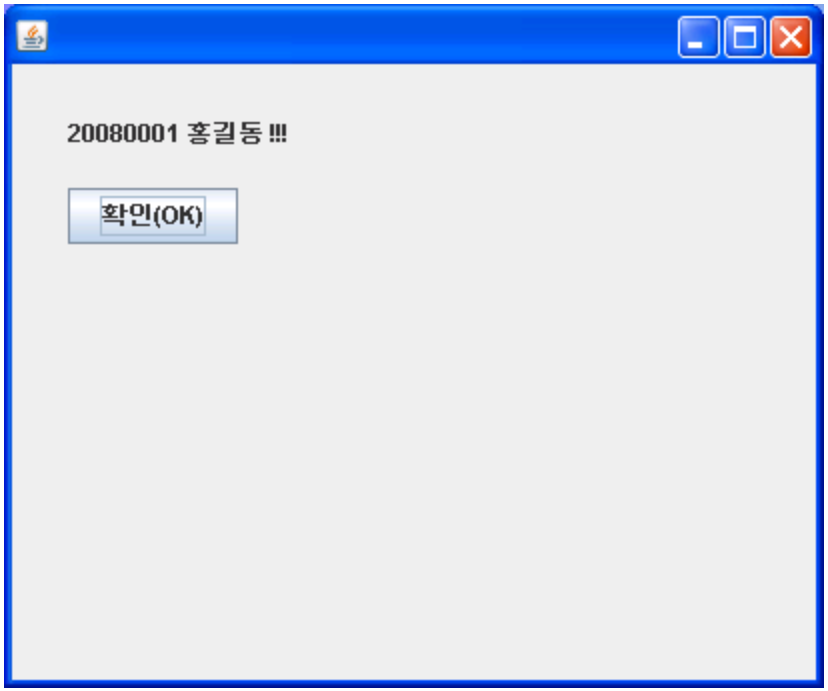
실행 결과

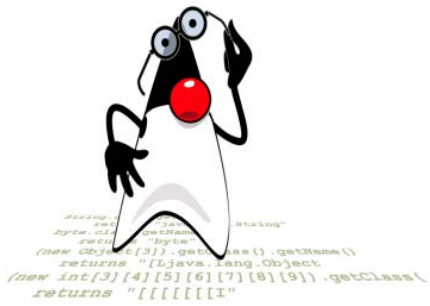


실습: Event Handler 구현

❖ Event Handler 구현(실습^시간 : 20분)

- Project Name : Student_Source
- 확인(OK) 버튼 Click^시 "학번 + 이름" 이 출력되도록 프로그래밍 하^시오.





학습 요약

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