



# GUI 컨트롤 II

CheckBox & RadioButton 예제

결제 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

☒ 회원

☐ 비회원

확인 (OK)

결제 내역 확인

회원 자격으로 현금 이체 결제 방법을 선택하였습니다.





# 학습 목표

■ 이 강의를 마치면 학생들은

- ❖ Checkbox 컨트롤 Function에 대하여 설명할 수 있다.
- ❖ Radio Button 컨트롤 Function에 대하여 설명할 수 있다.
- ❖ Choice와 Combo Box 컨트롤 Function에 대하여 설명할 수 있다.
- ❖ List 컨트롤 Function에 대하여 설명할 수 있다.





# Checkbox 컨트롤 (1)

## ■ Checkbox 컨트롤

### ❖ Function

◆ 다중 선택 구현

### ❖ Constructor

Constructor	Function
Checkbox()	Blank checkbox 생성
Checkbox(String text)	Text 문자열의 checkbox 생성
Checkbox(String text, Boolean state)	Text 문자열을 가지며, state 값에 따라 선택 여부 결정
Checkbox(String text, Boolean state, CheckboxGroup group)	Text 문자열을 가지며, state 값에 따라 선택 여부 결정한다. group으로 명시된 그룹의 check box 생성





# Checkbox 컨트롤 (2)

## ❖ Method

Method	Function
setText( <b>String</b> text)	Check Box의 text 지정
String getText()	Check Box의 text 반환
setSelected( <b>Boolean</b> b)	Check Box의 선택 여부 지정
Boolean isSelected()	Check Box의 선택 여부 반환





# Radio Button 컨트롤 (1)

## ■ Radio Button 컨트롤

### ❖ Function

◆ 단일 선택

### ❖ Method

Method	Function
setText( <b>String</b> text)	Radio Button의 text 지정
String getText()	Radio Button의 text 반환
setSelected( <b>Boolean</b> b)	Radio Button 의 선택 여부 지정
Boolean isSelected()	Radio Button 의 선택 여부 반환



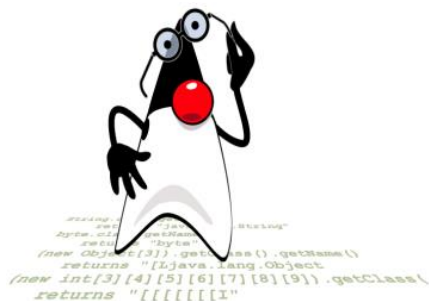


# Practice 1: CheckBox & Radio Button Control

## ❖ Project Name: CheckBoxRadioButton\_Source (Time : 20 min)

### ■ 확인(OK) button Click

- 결재 방법과 회원 구분 에서 선택한 내용을 Text Area에 추가하도록 프로그래밍하시오.



CheckBox & RadioButton 예제

결재 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

☒ 회원

☐ 비회원

확인(OK)

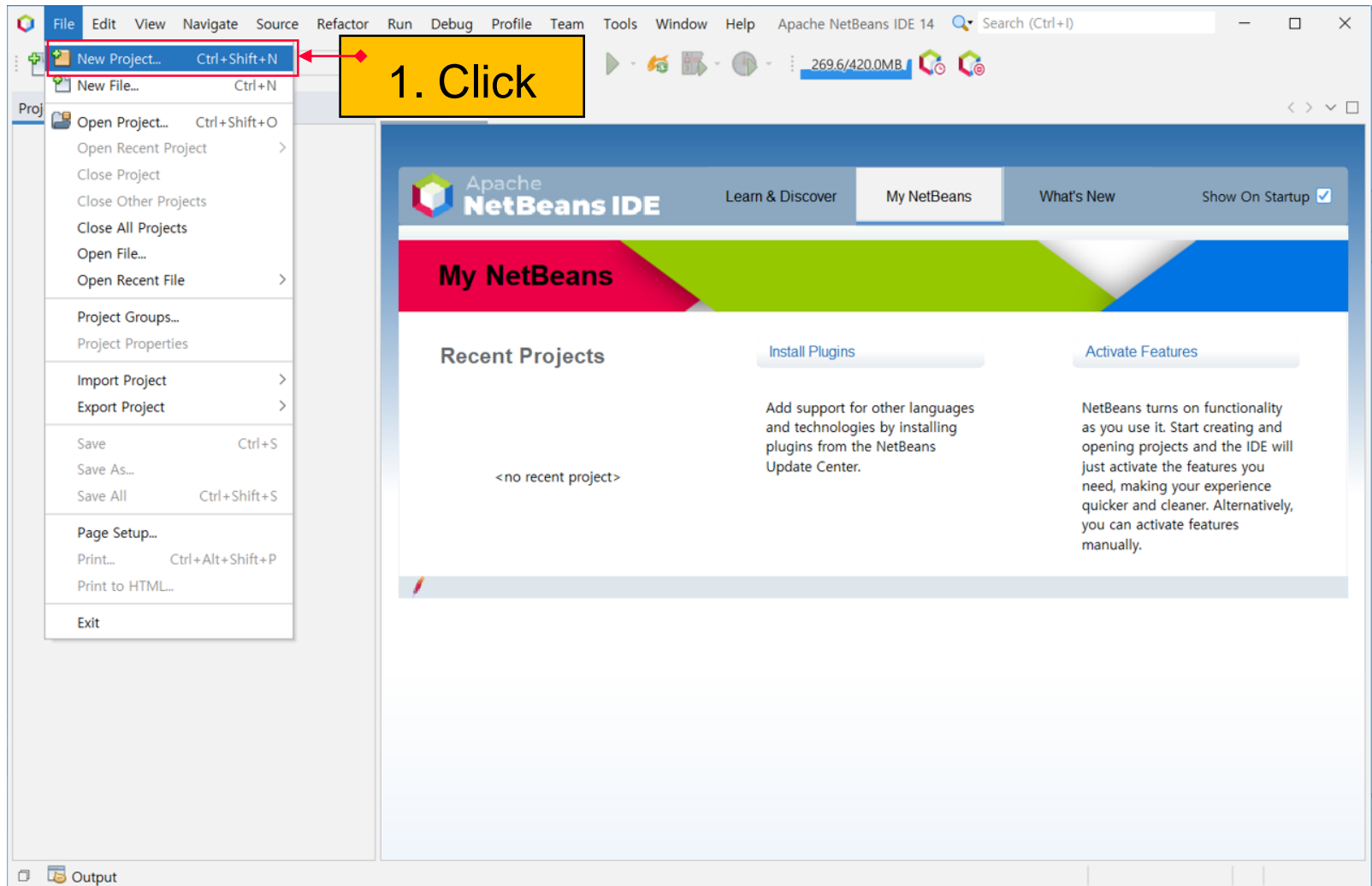
결재 내역 확인

회원 자격으로 현금 이체 결재 방법을 선택하였습니다.



# Practice 1: CheckBox & Radio Button Control (1)

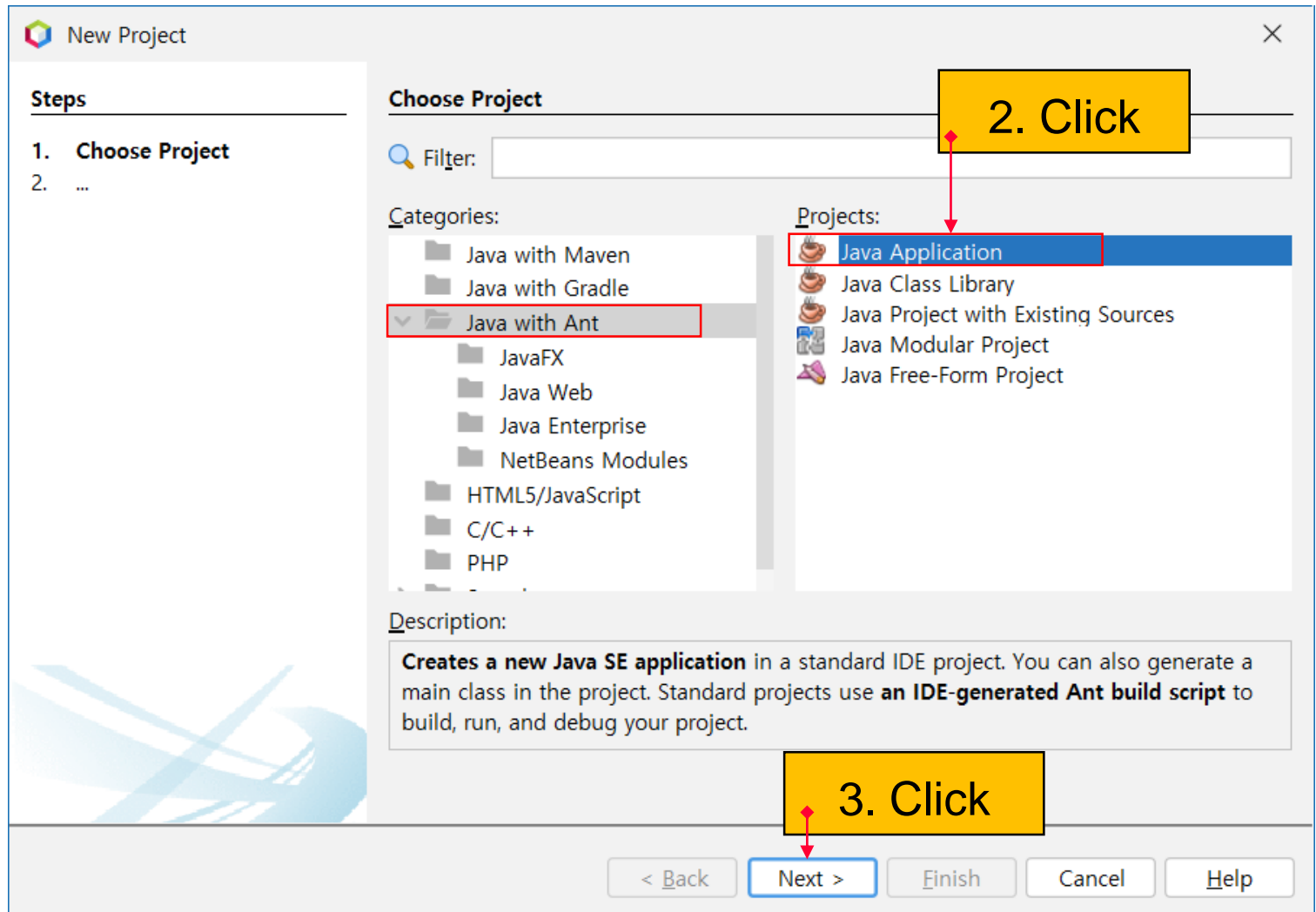
## Create Project





# Practice 1: CheckBox & Radio Button Control (2)

## Choose Project







# Practice 1: CheckBox & Radio Button Control (3)

## ■ Project Name and Location

❖ Project name : CheckBoxRadioButton\_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:

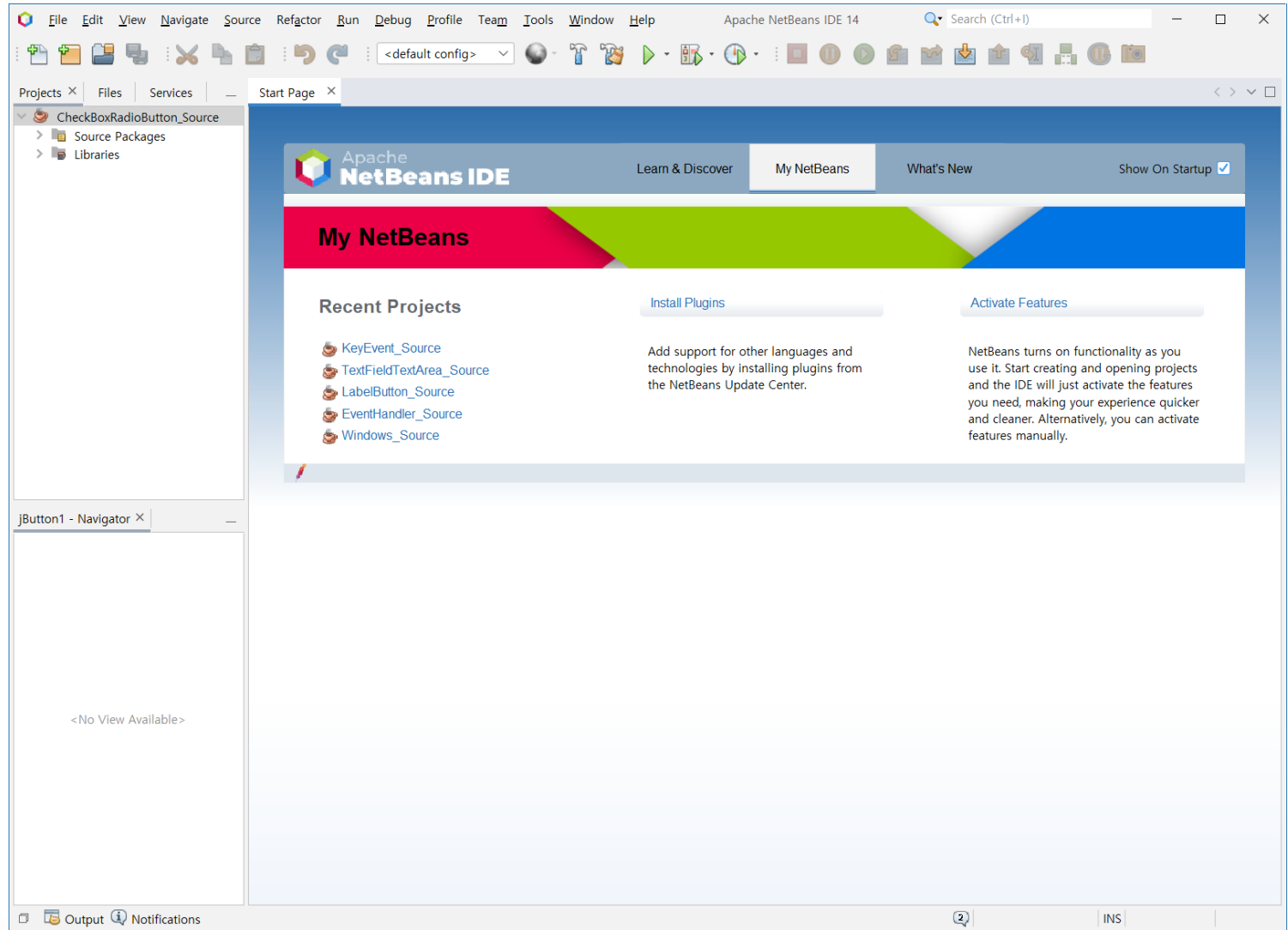
Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class



# Practice 1: CheckBox & Radio Button Control (4)

## Create project success





# Practice 1: CheckBox & Radio Button Control (5)

## ■ Create JFrame Form

7. Mouse right-button Click

8. Click

9. Click

The screenshot illustrates the process of creating a new JFrame Form in an IDE. The 'Projects' window on the left shows the project 'CheckBoxRadioButton\_Source' selected. A right-click context menu is open over the project, with the 'New' option highlighted. The 'New' submenu is also open, showing various options, with 'JFrame Form...' selected. Red arrows and yellow boxes with numbers 7, 8, and 9 indicate the sequence of actions: 7. Mouse right-button Click, 8. Click, and 9. Click.





# Practice 1: CheckBox & Radio Button Control (6)

## ■ Setting Class Name

### ❖ Create Mainframe.java

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

Class Name: MainFrame

Project: CheckBoxRadioButton\_Source

Location: Source Packages

Package:

Created File: C:\Java\_Project\CheckBoxRadioButton\_Source\src\MainFrame.java

**Projects**

- ✓ ☕ CheckBoxRadioButton\_Source
  - ✓ 📁 Source Packages
    - ✓ 📁 <default package>
      - 📄 MainFrame.java
  - ✓ 📁 Libraries
    - > 🖨️ JDK 18 (Default)

10. Input "MainFrame"

11. Click

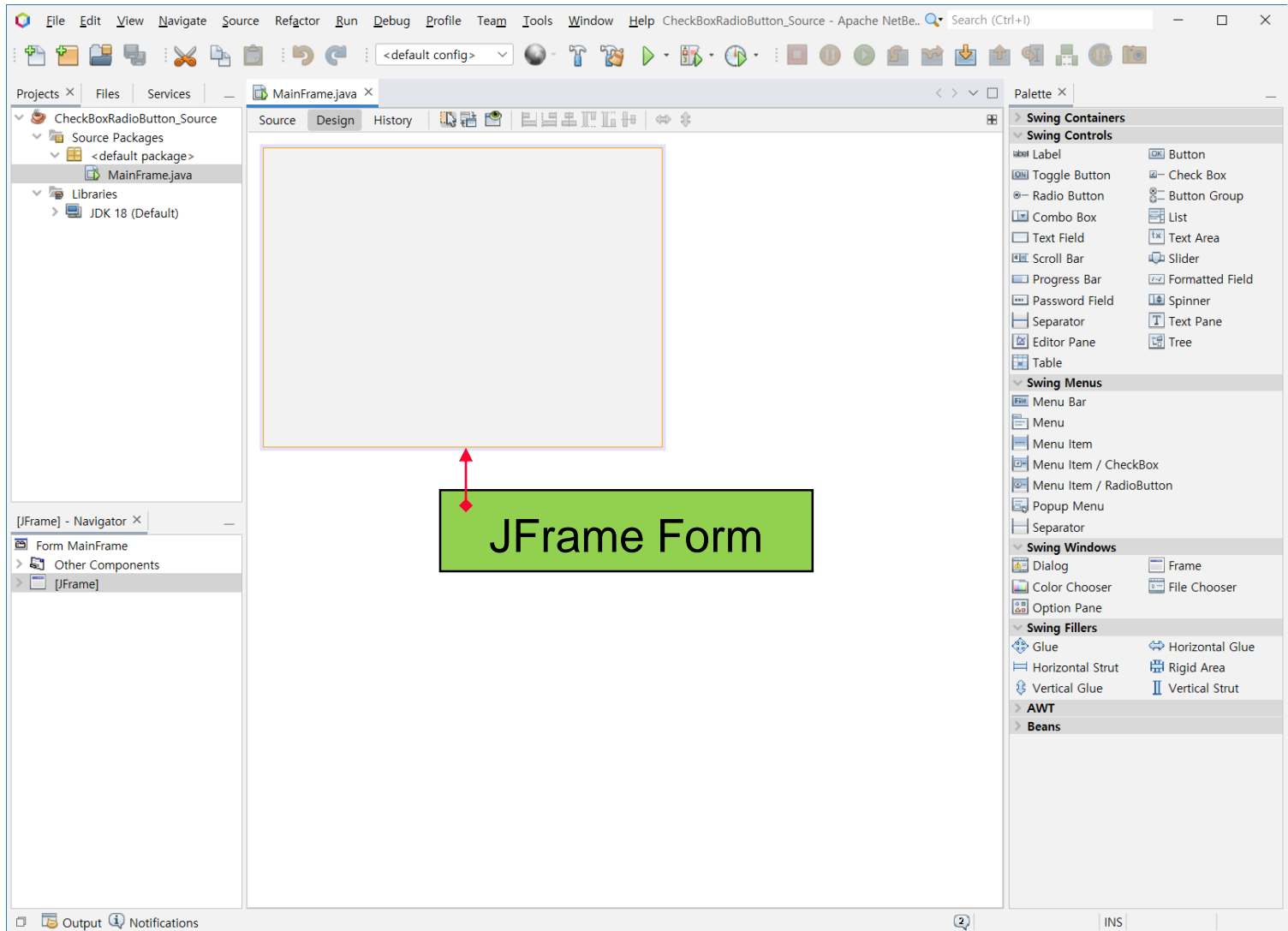
It is highly recommended that you do not place Java classes in the default package

< Back Next > Finish Cancel Help



# Practice 1: CheckBox & Radio Button Control (7)

## Create JFrame form success





# Practice 1: CheckBox & Radio Button Control (8)

## Control Layout & Property Setting

jLabel1

jLabel2

☐ jCheckBox1

☐ jCheckBox2

☐ jCheckBox3

jLabel3

☐ jRadioButton1

☐ jRadioButton2

jButton1

jLabel4

Control	Properties Setting
jLabel1	• Text: CheckBox & RadioButton 예제
jLabel2	• Text: 결제 방법
jCheckBox1	• Text: 현금 이체 • Selected: 체크
jCheckBox2	• Text: 카드
jCheckBox3	• Text: 포인트
jLabel3	• Text: 회원 구분
jRadiobutton1	• Text: 회원 • Selected: 체크
jRadiobutton2	• Text: 비회원
jLabel4	• Text: 결제 내역 확인
jTextArea1	• Text:
jButton1	• Text: 확인(OK)

Control	Properties Setting
jButton1	• Variable Name: btnOK





# Practice 1: CheckBox & Radio Button Control (9)

## ■ Button Group 컨트롤 추가

CheckBox & RadioButton 예제

결제 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

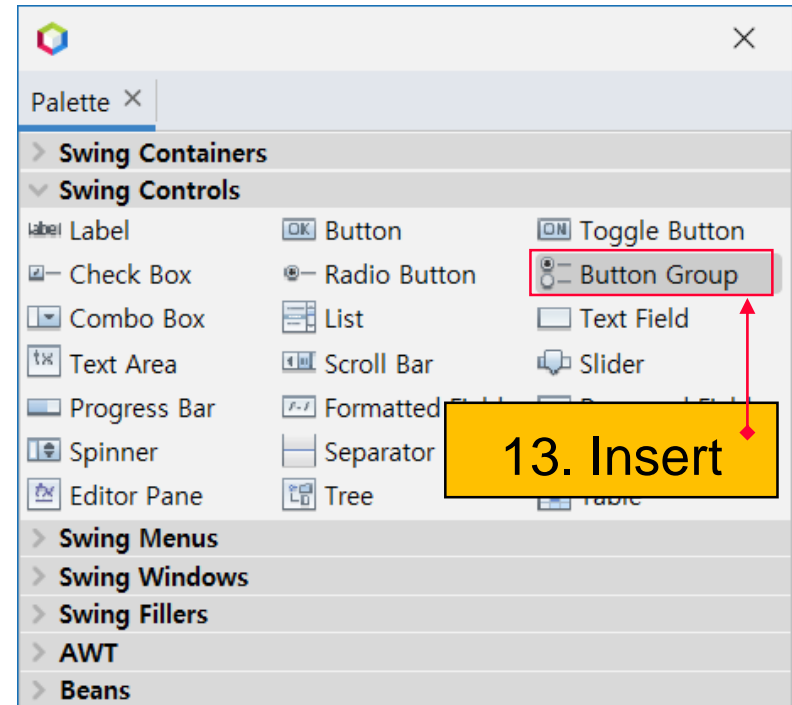
☒ 회원

☐ 비회원

확인(OK)

결제 내역 확인

12. UI Design



13. Insert





# Practice 1: CheckBox & Radio Button Control (10)

## ❖ Button Group 추가 및 지정

CheckBox & RadioButton 예제

결제 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

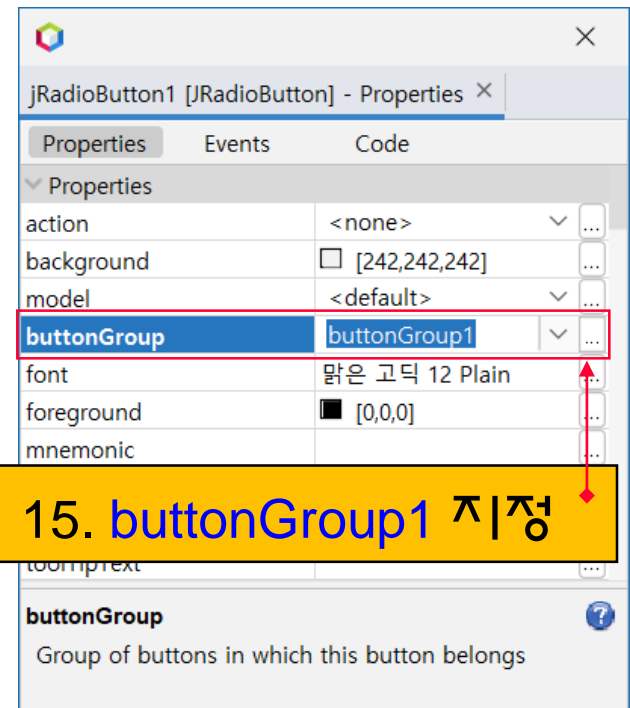
☒ 회원

☐ 비회원

결제 내역 확인

확인(OK)

14. Click



Control	Properties Setting
JRadioButton1	• buttonGroup: buttonGroup1
JRadioButton2	• buttonGroup: buttonGroup1







# Practice 1: CheckBox & Radio Button Control (11)

## ■ 컨트롤 배치 및 속성 지정 완료

CheckBox & RadioButton 예제

결제 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

☒ 회원

☐ 비회원

확인(OK)

결제 내역 확인





# Practice 1: CheckBox & Radio Button Control (12)

## ❖ 확인(OK) Button Event Handler

```
MainFrame.java - Editor
MainFrame.java x
Source Design History
13 public class MainFrame extends javax.swing.JFrame {
14
15     /** Creates new form MainFrame */
16     public MainFrame() {
17         initComponents();
18     }
19
20     /** This method is called from within the constructor to ...5 lines */
25     Generated Code
143
146 private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {
147     String strData = "회원님은 "; //문자열 변수 strData 선언
148     if(jRadioButton1.isSelected()) //RadioButton1 선택 지정
149         strData = "회원 자격으로 ";
150     else
151         strData = "비회원 자격으로 ";
152     if(jCheckBox1.isSelected()) strData += jCheckBox1.getText(); //jCheckBox1 선택 지정
153     if(jCheckBox2.isSelected()) strData += jCheckBox2.getText(); //jCheckBox2 선택 지정
154     if(jCheckBox3.isSelected()) strData += jCheckBox3.getText(); //jCheckBox3 선택 지정
155     strData += " 결재 방법을 선택하였습니다. \n";
156     jTextArea1.setText(strData); // jTextArea1에 문자열 지정
157     strData = null;
158 }
```

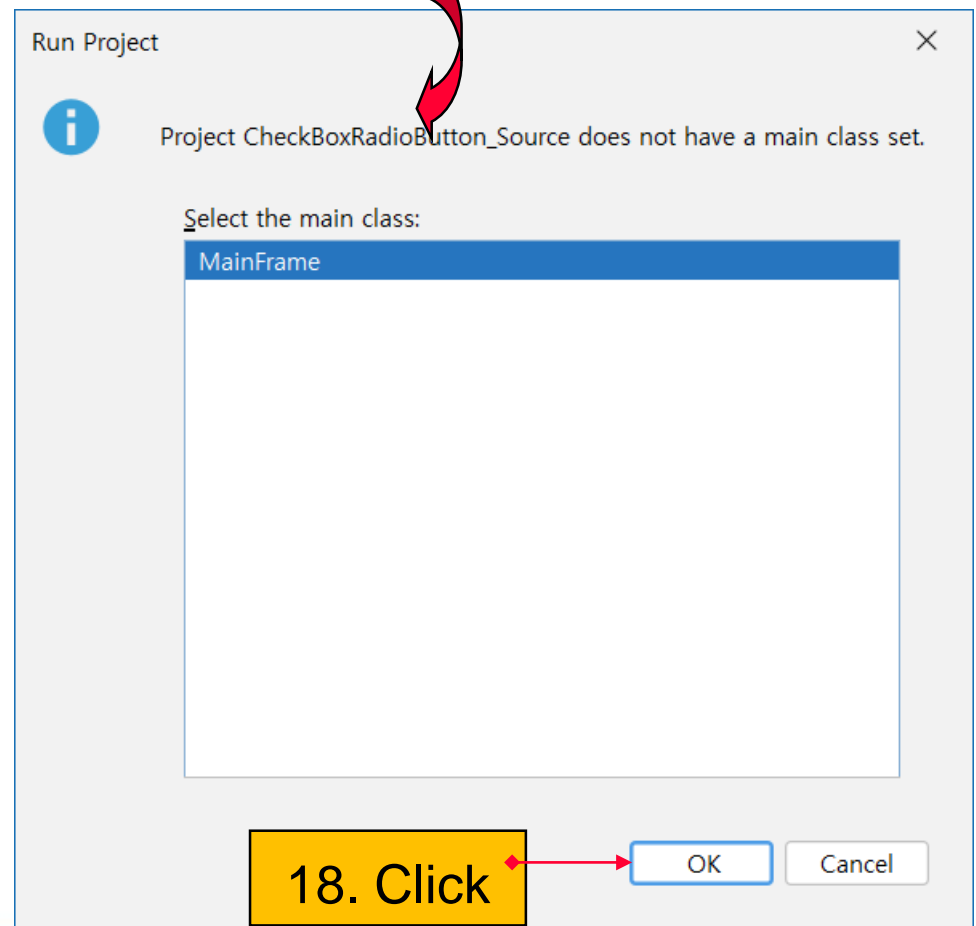
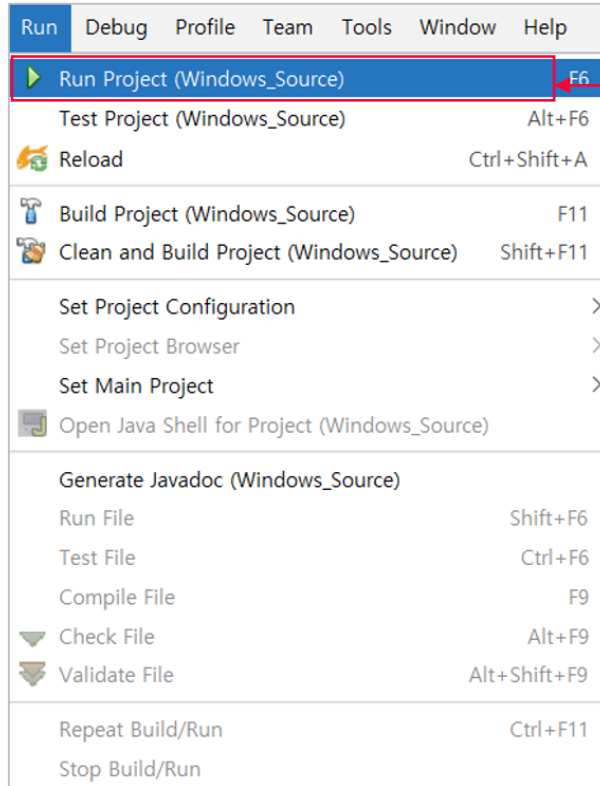
14. Coding





# Practice 1: CheckBox & Radio Button Control (13)

## Run





# Practice 1: CheckBox & Radio Button Control(14)

## ■ Run

- ❖ 결재 방법 선택
- ❖ 회원 구분 선택
- ❖ 확인(OK) Button Click

CheckBox & RadioButton 예제

결재 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

☒ 회원

☐ 비회원

확인(OK)

결재 내역 확인





# Practice 1: CheckBox & Radio Button Control (15)

■ Run

CheckBox & RadioButton 예제

결제 방법

☒ 현금 이체

☐ 카드

☐ 포인트

회원 구분

☒ 회원

☐ 비회원

확인 (OK)

결제 내역 확인

회원 자격으로 현금 이체 결제 방법을 선택하였습니다.

CheckBox & RadioButton 예제

결제 방법

☐ 현금 이체

☒ 카드

☐ 포인트

회원 구분

☐ 회원

☒ 비회원

확인 (OK)

결제 내역 확인

비회원 자격으로 카드 결제 방법을 선택하였습니다.





# Homework # 1

## ❖ 회원 가입 모듈 구현 (Project 명: Member\_Source)

### ■ [회원 가입] 버튼 Click

- 비밀번호가 일치하면 아이디, 비밀번호, 이름, 회원 구분, 관심 분야를 member.txt 파일에 저장한다.
- 회원의 관심분야는 아래와 같이 bit연산을 이용하여 계산한 값을 member.txt 파일에 저장한다.

수영	게임	골프	스키	독서	테니스
32	16	8	4	2	1

회원 가입

아이디

kdhong

중복 확인

비밀번호

\*\*\*\*

비밀번호 확인

\*\*\*\*

성명

홍길동

회원 구분

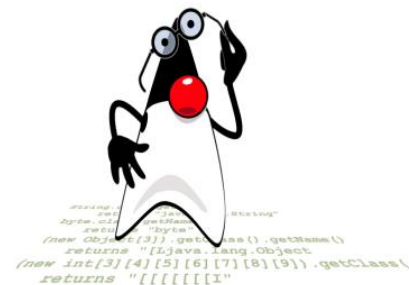
☒ 정회원 ☐ 준회원 ☐ 학생회원

관심 분야

☒ 수영 ☐ 게임 ☐ 골프

☐ 스키 ☐ 독서 ☒ 테니스

회원 가입



member.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

kdhong 1234 홍길동 정회원 33





# Combo Box 컨트롤

## ■ Combo Box 컨트롤

### ❖ Function

◆ Drop-down 리스트 구현

◆ addItem() 메서드를 사용하여 drop-down 항목 추가

### ❖ Method

Method	Function
<code>int getItemCount()</code>	Item 수 반환
<code>String getItemAt(int index)</code>	Index에 해당하는 Item 반환
<code>void addItem(Object obj)</code>	Obj를 Combo Box Item으로 추가
<code>Object getSelectedItem()</code>	선택된 Item 반환
<code>int getSelectedIndex()</code>	선택된 Item의 Index를 반환

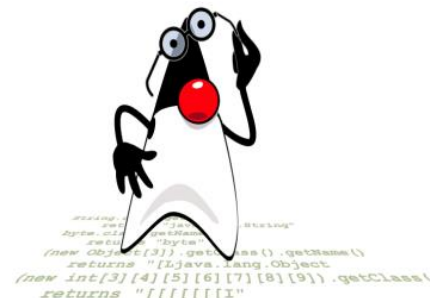
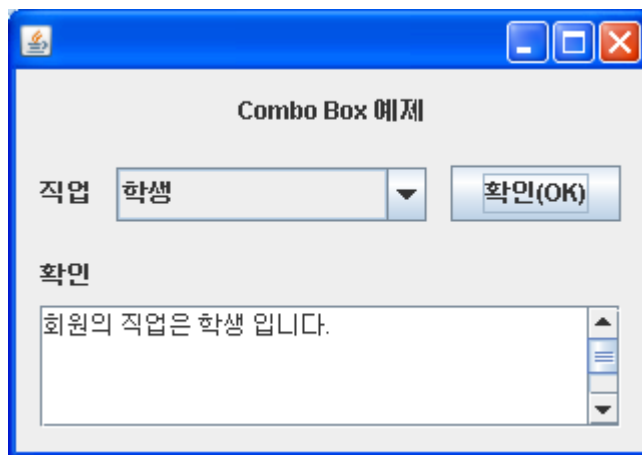




# Practice 2 : Combo Box Control

## ❖ Project Name : ComboBox\_Source (Time: 20 min)

- Combo Box에서 직업 선택
- 확인(OK) 버튼 Click
  - Combo Box에서 선택한 직업을 Text Area에 표현한다.







# Practice 2 : Combo Box Control (1)

## Create Project

The screenshot illustrates the process of creating a new project in the Apache NetBeans IDE. The interface includes a menu bar at the top with options like File, Edit, View, and Run. A 'File' menu is open, showing 'New Project...' as the first option. A yellow callout box labeled '1. Click' points to this option. Below the menu, the 'New Project' dialog is displayed. It has a 'Steps' section on the left listing '1. Choose Project' and '2. ...'. The main area is titled 'Choose Project' and contains a 'Filter' field, a 'Categories' list on the left, and a 'Projects' list on the right. In the 'Categories' list, 'Java with Ant' is expanded, and 'Java Application' is selected in the 'Projects' list. A yellow callout box labeled '2. Click' points to 'Java Application'. At the bottom of the dialog, there is a 'Description' section and a row of buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. A yellow callout box labeled '3. Click' points to the 'Next >' button.

1. Click

2. Click

3. Click





# Practice 2 : Combo Box Control (2)

## ■ Project Name and Location

❖ Project name: ComboBox\_Source

New Java Application

**Steps**

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

**Name and Location**

Project Name: ComboBox\_Source

Project Location: C:\Java\_Project Browse...

Project Folder: C:\Java\_Project\ComboBox\_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 combobox\_source.ComboBox\_Source

< Back Next > Finish Cancel Help

5. Reset check

4. Input Project Name

6. Click

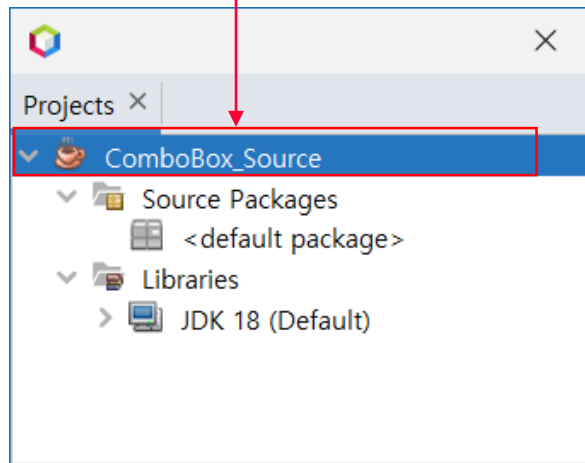




# Practice 2 : Combo Box Control (3)

## ■ Create JFrame Form

7. Mouse right-button Click



New

Build  
Clean and Build

8. Click

Run  
Debug  
Profile  
Test  
Set Configuration

Alt+F6

Set as Main Project  
Open Required Projects  
Close

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

Delete

Find...  
Versioning  
Local History

Ctrl+F

Properties

JFrame Form...

Java Class...

Java Package...

Java Module...

Java Class...

Java Class...

Entity Class...

Entity Classes from Database...

Other...

9. Click





# Practice 2 : Combo Box Control (4)

## JFrame Form Name 지정

### MainFrame.java 생성

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

Class Name: MainFrame

Project: ComboBox\_Source

Location: Source Packages

Package:

Created File: C:\Java\_Project\ComboBox\_Source\src\MainFrame.java

Superclass: Browse...

Interfaces: Browse...

Warning: It is highly recommended that you do not place Java classes in the default package

< Back Next > Finish Cancel Help



# Practice 2 : Combo Box Control (5)

## Control Layout & Property Setting

Combo Box 예제

직업

확인

12. UI Design

Control	Setting Properties
jLabel1	• Text: Combo Box 예제
jLabel2	• Text: 직업
jLabel3	• Text: 확인
jCombo Box	
jButton1	• Variable Name: btnOK • Text: 확인(OK)
jText Area	Text:





# Practice 2 : Combo Box Control (6)

## ❖ Combo Box 컨트롤 Item 추가

Combo Box 예제

직업  ▼

확인

13. Click

jComboBox1 [JComboBox] - Properties

Properties	Events	Code
▼ Properties		
background	<input type="checkbox"/>	[255,255,255] ...
editable	<input type="checkbox"/>	...
font		맑은 고딕 12 Plain ...
foreground	<input checked="" type="checkbox"/>	[0,0,0] ...
maximumRowCount		15 ...
model		학생, 회사원(연구 개발),... ...
selectedIndex		0 ...
selectedItem		학생 ...
toolTipText		...

**model**

(javax.swing.ComboBoxModel) Model that the combo box uses to get data to display.

14. Click





# Practice 2 : Combo Box Control (7)

jComboBox1 [JComboBox] - model

Set **jComboBox1**'s **model** property using: Combo Box Model Editor

Enter the textual representation of combo box model content. Each row corresponds to one combo box item.

학생
회사원(연구 개발)
회사원(사무직)
회사원(마케팅)
자영업자
공무원
기타

15. Input Items

16. Click

OK Reset to Default Cancel Help





# Practice 2 : Combo Box Control (8)

## ❖ 확인(OK) Button Event Handler

```
MainFrame.java - Editor
MainFrame.java x
Source Design History
121 private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {
122     String strData = null;
123     strData = "회원의 직업은 " + jComboBox1.getSelectedItem().toString() + "입니다. \n";
124     JTextArea1.setText(strData);
125 }
126
127
128 /**
129  * @param args the command line arguments
130  */
131 public static void main(String args[]) {
132     java.awt.EventQueue.invokeLater(new Runnable() {
133         public void run() {
134             new MainFrame().setVisible(true);
135         }
136     });
137 }
```

17. Coding

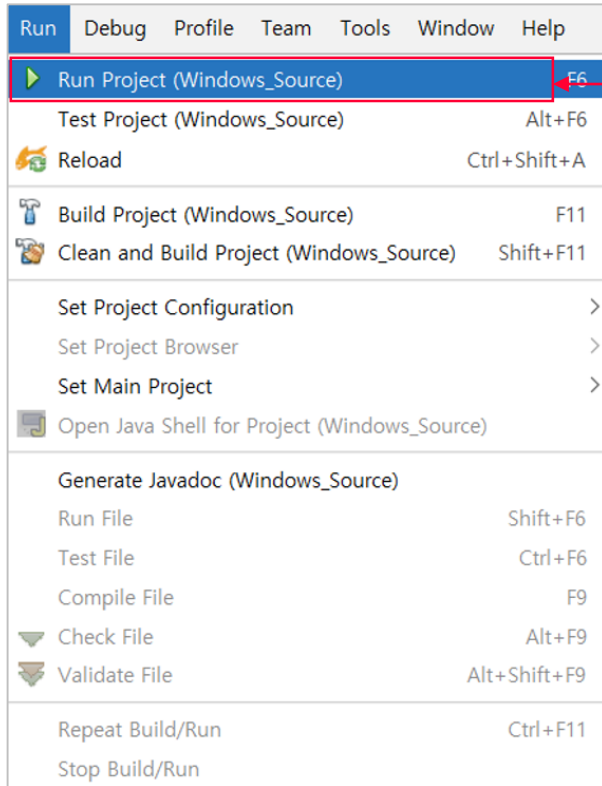




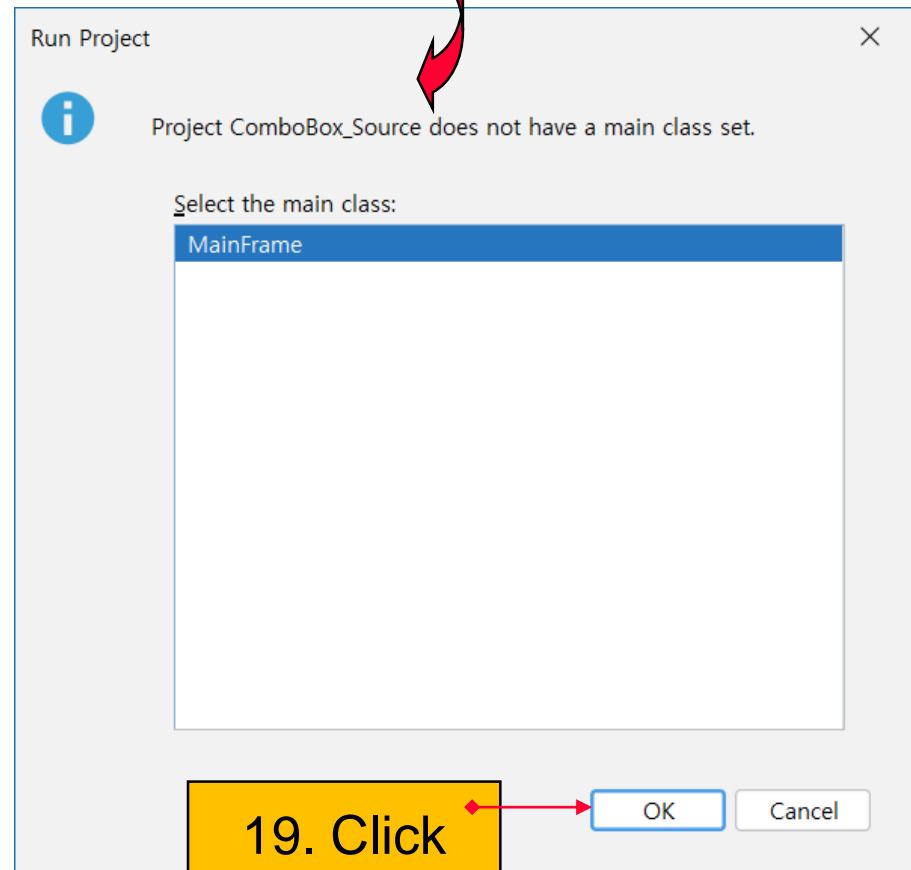


# Practice 2 : Combo Box Control (9)

## Run



18. Click



19. Click





# Practice 2 : Combo Box Control (10)

## ■ Run

### ❖ 확인(OK) Button Click

◆ Combo Box에서 선택한 직업을 Text Area에 표현한다.

Combo Box 예제

직업 학생

확인(OK)

확인

20. Click

Combo Box 예제

직업 학생

확인(OK)

확인

회원의 직업은 학생입니다.



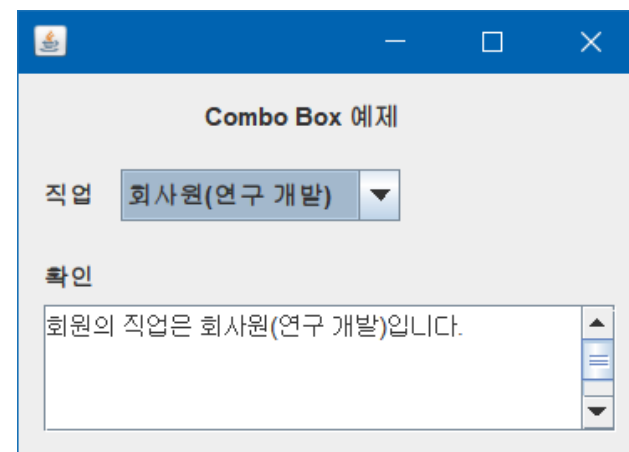
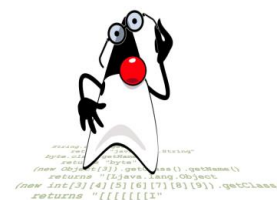
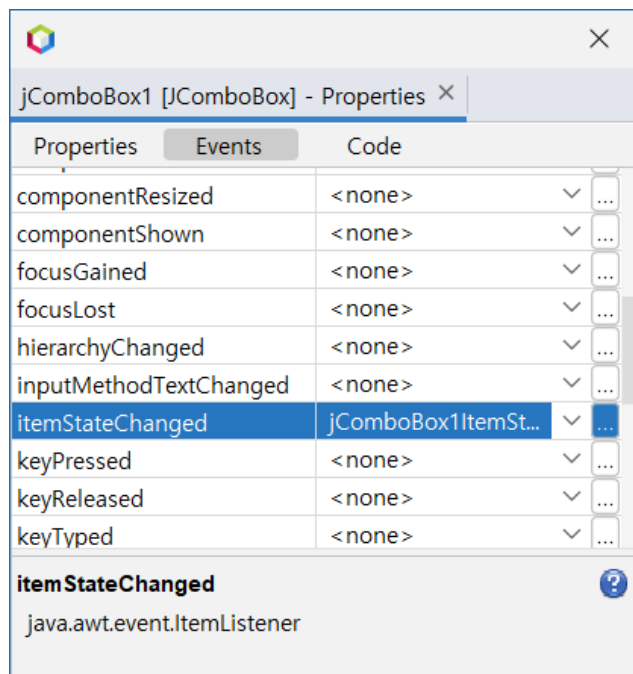


# Practice 3 : Combo Box Control

## ❖ Project Name : Combo Box2\_Source (Time: 20 min)

- Combo Box에서 해당 직업을 선택하면 Text Area에 표현한다.
- Event Handler

Event Handler	Function
ItemStateChanged(ItemEvent evt)	• Item 선택시 발생하는 event 처리





# Choice 컨트롤

## ■ Choice 컨트롤

### ❖ Function

◆ Drop-down 리스트를 구현할 때 사용

◆ addItem() 메서드를 사용하여 drop-down 항목 추가

### ❖ Method

Method	Function
int countItems()	Choice 컨트롤의 Item 수 반환
String getItem(int index)	Index에 해당하는 Item 반환
void addItem(String item)	Item을 choice Item으로 추가
String getSelectedItem()	선택된 Item 반환
int getSelectedIndex()	선택된 항목의 Index 반환
void select(int index)	index에 해당하는 Item 선택
void select(String str)	str에 해당하는 Item 선택





# List 컨트롤 (1)

## List 컨트롤

### ❖ Function

◆ 다중 Item이 리스트에 나타나도록 구현

### ❖ Constructor

Constructor	Comment
<code>JList();</code>	전달 인자가 없는 Constructor
<code>JList(Object[] listData);</code>	배열 형태의 데이터를 갖는 list 생성
<code>JList(Vector&lt;?&gt; listData);</code>	벡터를 데이터로 갖는 list 생성
<code>JList(ListModel dataModel);</code>	데이터 모델을 갖는 list 생성





# List 컨트롤 (2)

## ❖ Method

Method	Function
getSize()	List의 Item 수 반환
void setListData(Object[] listData)	Object 클래스로 List에 추가
void setListData(Vector<?> listData)	Vector 클래스로 List에 추가
int getSelectedIndex()	선택된 Item의 index 반환
int[] getSelectedIndices()	다중 선택 모드일때, 선택된 Item의 index 를 배열로 반환
int setSelectedIndex()	선택된 Item의 index 지정
int[] setSelectedIndices()	다중 선택 모드일때, 선택된 Item의 index 배열 지정

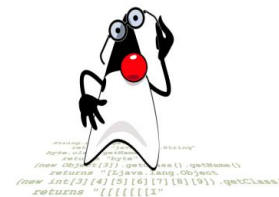




# Practice 4 : List Control

## ❖ Project Name : List\_Source (Time: 20 min)

- List의 Item을 추가, 삭제하는 Function을 구현한다.
- Button Event Handler



List 예제

New Item

Item List

추가(Add)

삭제(Del)

NHN  
Daum  
Yahoo  
Google

List 예제

New Item

Item List

추가(Add)

삭제(Del)

NHN  
Daum  
Yahoo  
Google  
NAVER





# Practice 4 : List Control (1)

## Create Project

The screenshot shows the Apache NetBeans IDE interface. The 'File' menu is open, and the 'New Project...' option is highlighted. A yellow callout box labeled '1. Click' points to this option. The 'New Project' dialog box is open, showing the 'Choose Project' tab. The 'Java with Ant' category is selected in the 'Categories' list. A yellow callout box labeled '2. Click' points to the 'Java Application' project in the 'Projects' list. The 'Description' section at the bottom explains that this creates a new Java SE application. A yellow callout box labeled '3. Click' points to the 'Next >' button at the bottom right of the dialog.

1. Click

2. Click

3. Click







# Practice 4 : List Control (2)

## ■ Project Name and Location

❖ Project name: List\_Source

New Java Application

**Steps**

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

**Name and Location**

Project Name: List\_Source

Project Location: C:\Java\_Project Browse...

Project Folder: C:\Java\_Project\List\_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 list\_source.List\_Source

< Back Next > Finish Cancel Help

5. Reset check

4. Input Project Name

6. Click

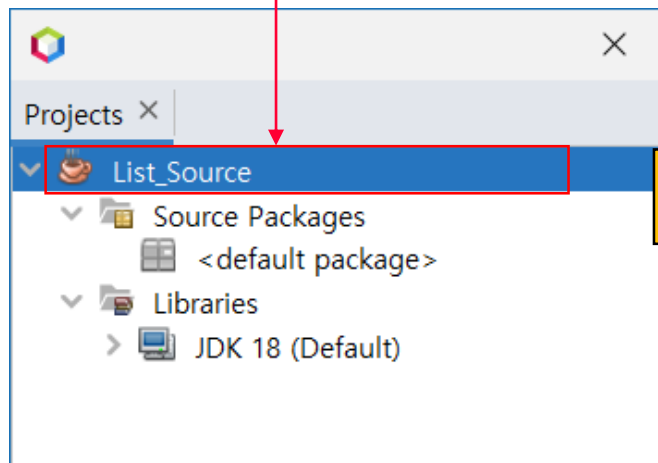




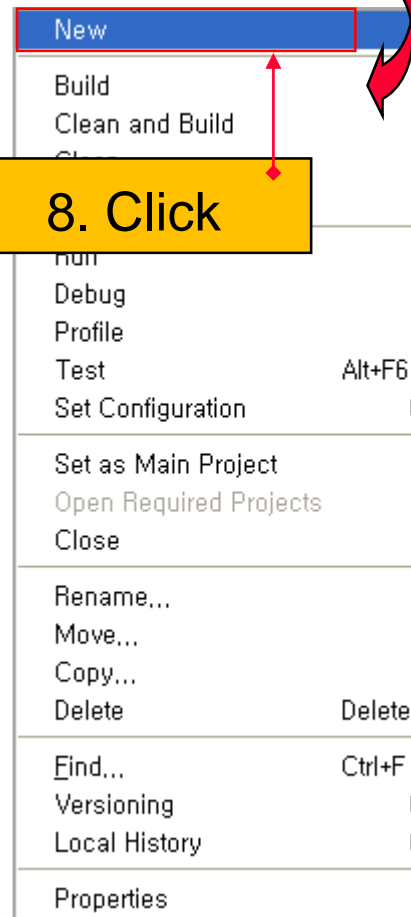
# Practice 4 : List Control (3)

## Create JFrame Form

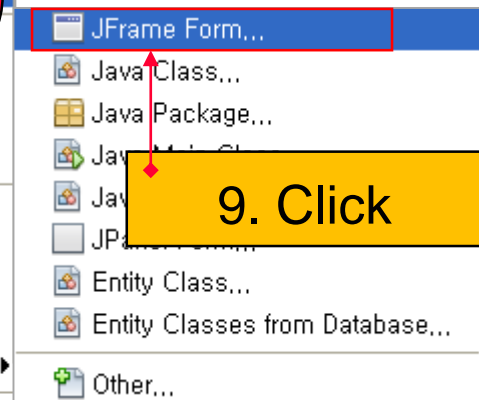
7. Mouse right-button Click



8. Click



9. Click





# Practice : List Control (4)

## JFrame Form Name 지정

### MainFrame.java 생성

**New JFrame Form**

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

Class Name:  10. Input "MainFrame"

Project:

Location:

Package:

Created File:

Superclass:  Browse...

Interfaces:  Browse...

11. Click

Warning: It is highly recommended that you do not place java classes in the default package

< Back Next > Finish Cancel Help



# Practice 4 : List Control (5)

## Control Layout & Property Setting

List 예제

New Item  추가(Add)

Item List 삭제(Del)

- NHN
- Daum
- Yahoo
- Google

12. UI Design

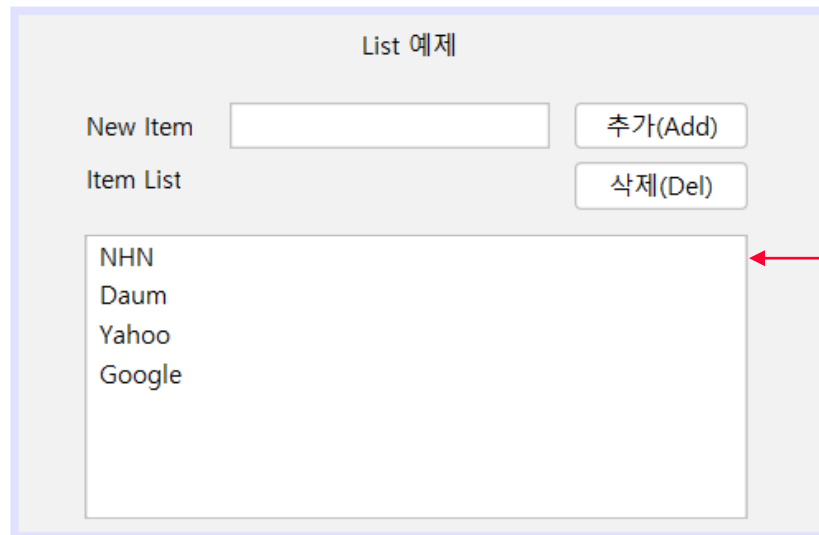
Control	Properties Setting
jLabel1	<ul style="list-style-type: none"><li>Variable Name: lblTitle</li><li>Text: List 예제</li></ul>
jTextField1	<ul style="list-style-type: none"><li>Variable Name: txtInput</li><li>Text:</li></ul>
jLabel2	<ul style="list-style-type: none"><li>Variable Name: lblItem</li><li>Text: New Item</li></ul>
jLabel3	<ul style="list-style-type: none"><li>Variable Name: lblItemlist</li><li>Text: Item List</li></ul>
List	<ul style="list-style-type: none"><li>Variable Name: lstCompany</li><li>model: 4개 Item 추가</li></ul>
jButton1	<ul style="list-style-type: none"><li>Variable Name: btnAdd</li><li>Text: 추가(Add)</li></ul>
jButton2	<ul style="list-style-type: none"><li>Variable Name: btnDel</li><li>Text: 삭제(Del)</li></ul>



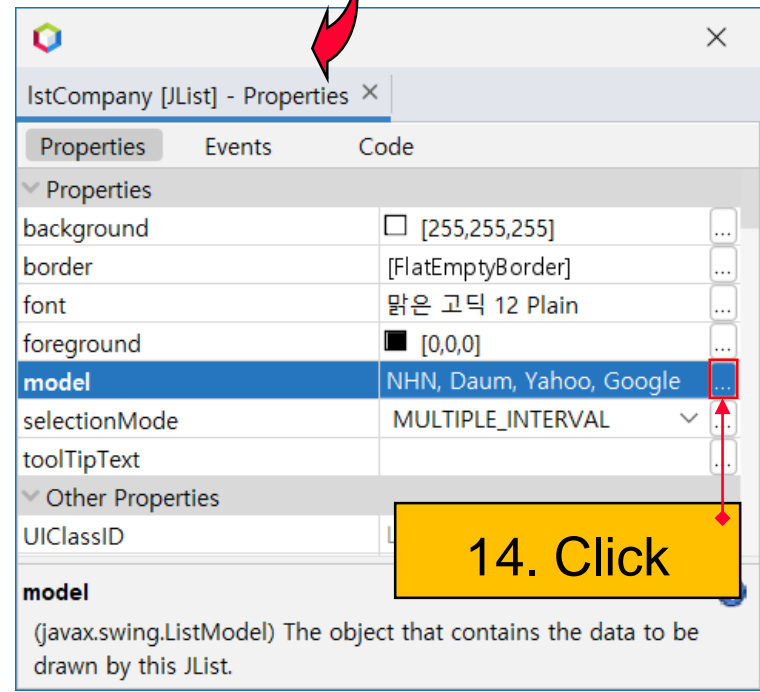


# Practice 4 : List Control (6)

## ❖ Insert model of List Control



13. Click



14. Click





# Practice 4 : List Control (7)

IstCompany [JList] - model

Set **IstCompany's model** property using: List Model Editor

Enter the textual representation of list model content. Each row corresponds to one list item.

NHN	15. Input Items
Daum	
Yahoo	
Google	

16. Click

OK Reset to Default Cancel Help





# Practice 4 : List Control (8)

## ❖ 추가(Add) Button Event Handler

```
19  lstCompany.setModel(new javax.swing.AbstractListModel() {  
20      String[] strings = { "NHN", "Daum", "Yahoo", "Google" };  
21      public int getSize() { return strings.length; }  
22      public Object getElementAt(int i) { return strings[i]; }  
23  });  
24  lstCompany.setValuesAdjusting(true);  
25  jScrollPane1.setViewportViewView(lstCompany);  
  
112 private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {  
113     Vector myVC = new Vector();           //New Item 저장을 위한 Vector 클래스  
114     int iListSize = 0;                     //List Size 저장  
  
116     iListSize = lstCompany.getModel().getSize();           //List Size 반환  
117     for(int idx = 0; idx < iListSize; idx++)               //List의 Item -> Vector 전송  
118         myVC.addElement( lstCompany.getModel().getElementAt(idx));  
119     myVC.addElement(txtInput.getText());                   //txtInput 문자열 -> Vector 추가  
120     lstCompany.setListData(myVC);                         //List의 Data 지정  
121     txtInput.setText(null);  
122 }  
123
```

127:54    INS

17. Coding



# Practice 4 : List Control (9)

## ❖ 삭제(Del) Button Event Handler

```
MainFrame.java - Editor
MainFrame.java
18
23
24
25
112
124
129
130
131
132
133
134
135
136
137
138

lstCompany.setModel(new javax.swing.AbstractListModel() {
    String[] strings = { "NHN", "Daum", "Yahoo", "Google" };
    public int getSize() { return strings.length; }
    public Object getElementAt(int i) { return strings[i]; }
});
lstCompany.setValuesAdjusting(true);
jScrollPane1.setViewportViewView(lstCompany);

private void btnDelActionPerformed(java.awt.event.ActionEvent evt) {
    Vector myVC = new Vector();
    int iList_index = 0;
    int iListSize = 0;

    iListSize = lstCompany.getModel().getSize();
    for(int idx = 0; idx < iListSize; idx++)
        myVC.addElement( lstCompany.getModel().getElementAt(idx));

    iList_index = lstCompany.getSelectedIndex();
    myVC.removeElementAt(iList_index);
    lstCompany.setListData(myVC);
    txtInput.setText(null);
}

16:25  INS
```

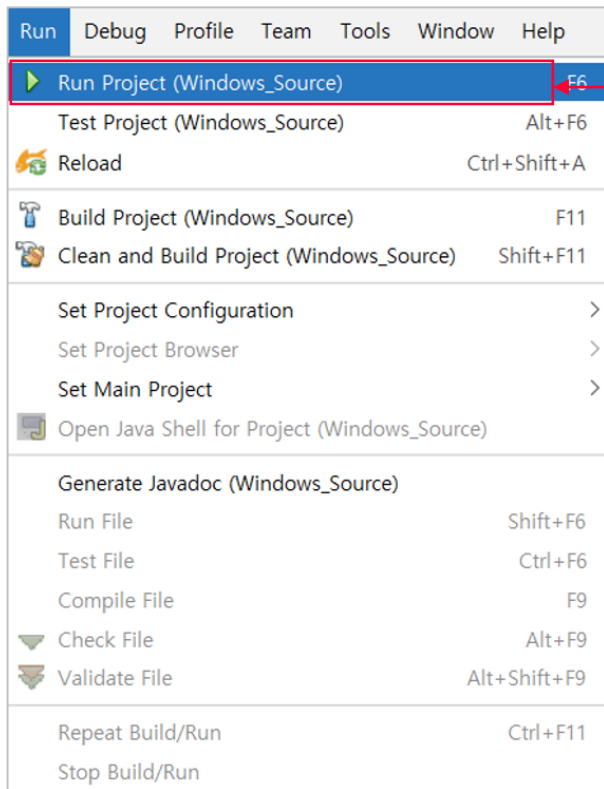




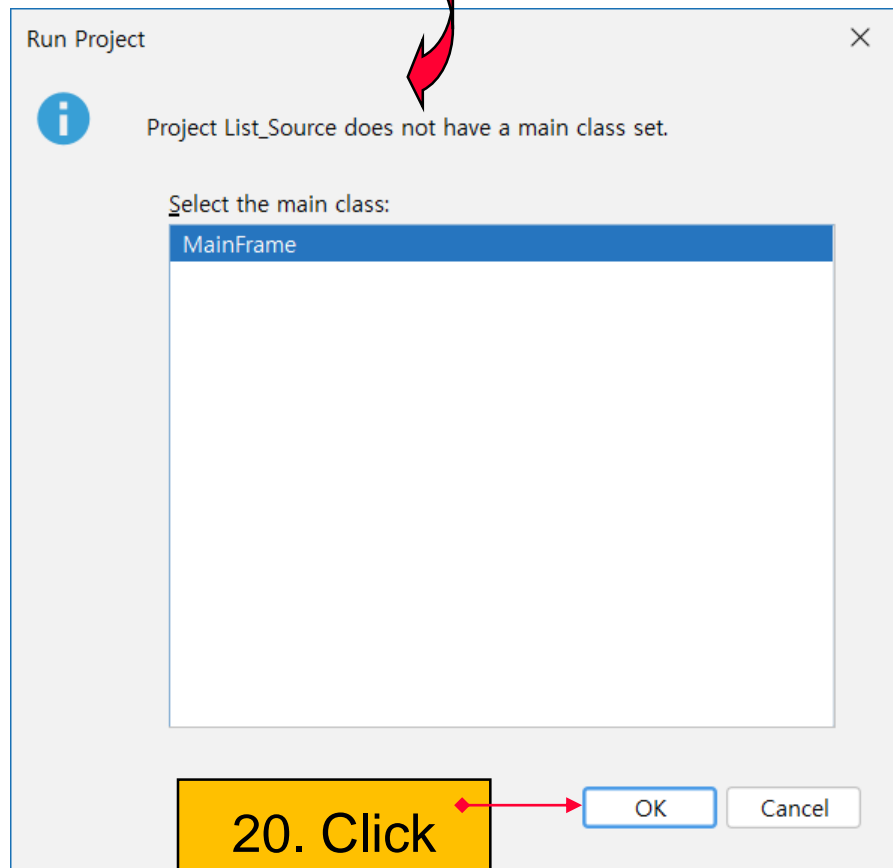


# Practice 4 : List Control (10)

## Run



19. Click



20. Click

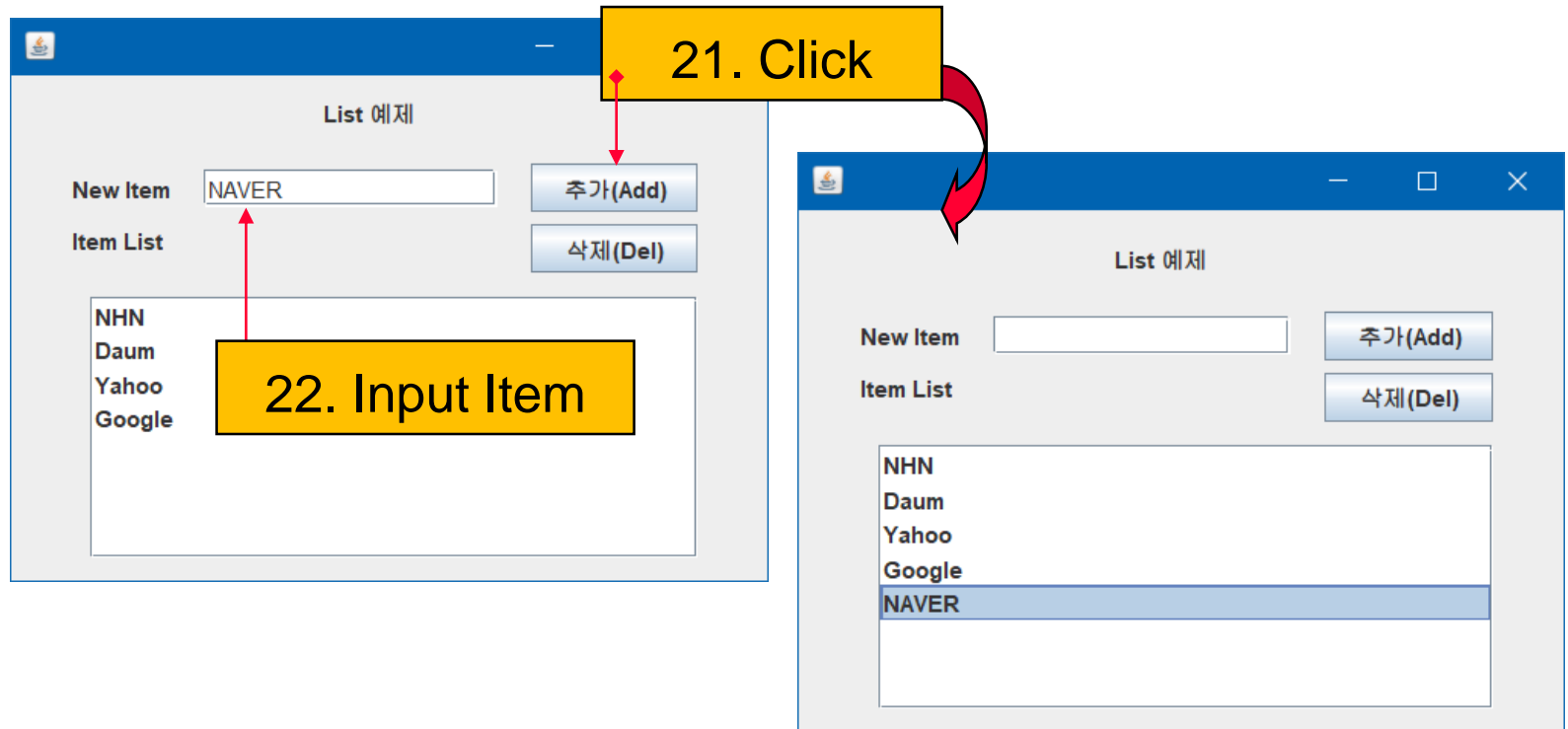




# Practice 4 : List Control (11)

## ❖ 추가(Add) Button

◆ Text Field에 입력한 New Item 을 Item List에 추가한다.

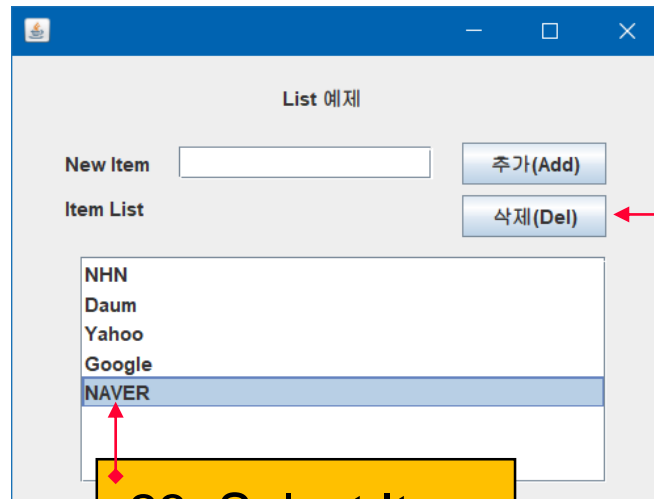




# Practice 4 : List Control (12)

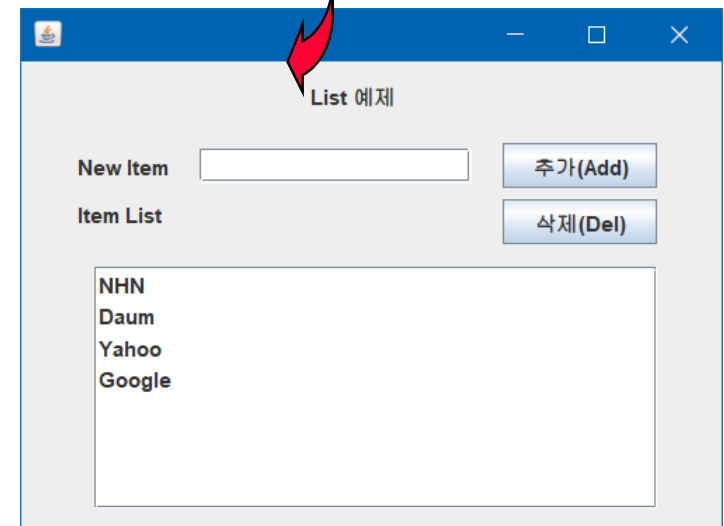
## ❖ 삭제(Add) Button

◆ Item List에서 선택한 Item을 삭제한다.



23. Select Item

24. Click





# Homework # 2

## ❖ 회원 가입 모듈 구현 (Project 명: Member\_ID)

### ■ [회원 가입] 버튼 Click

- 비밀번호가 일치하면 아이디, 비밀번호, 이름, 회원 구분, 관심 분야, 직업, 결제방법을 member.txt 파일에 저장한다.

회원 가입

아이디: kdgong 중복 확인

비밀번호: \*\*\*\*

비밀번호 확인: \*\*\*\*

성명: 홍길동

회원 구분: ☒ 정회원 ☐ 준회원 ☐ 학생회원

관심 분야: ☒ 수영 ☐ 게임 ☐ 골프  
☐ 스키 ☐ 독서 ☒ 테니스

직업: 학생

결제 방법: 현금  
카드  
포인트

회원 가입



```
String s = "1234567890";  
int i = 0;  
while (i < s.length())  
{  
    System.out.println(s.charAt(i));  
    i++;  
}  
return "JavaLangObject";  
(new Integer(4, 5, 6, 7, 8, 9)).getClass().  
getName()
```

member.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

kdhong 1234 홍길동 정회원 33 학생 현금





# 학습 요약

- ❖ CheckBox 컨트롤
- ❖ RadioButton 컨트롤
- ❖ Choice와 Combo Box 컨트롤
- ❖ List 컨트롤

