
JAVA

GUI 컨트롤 IV

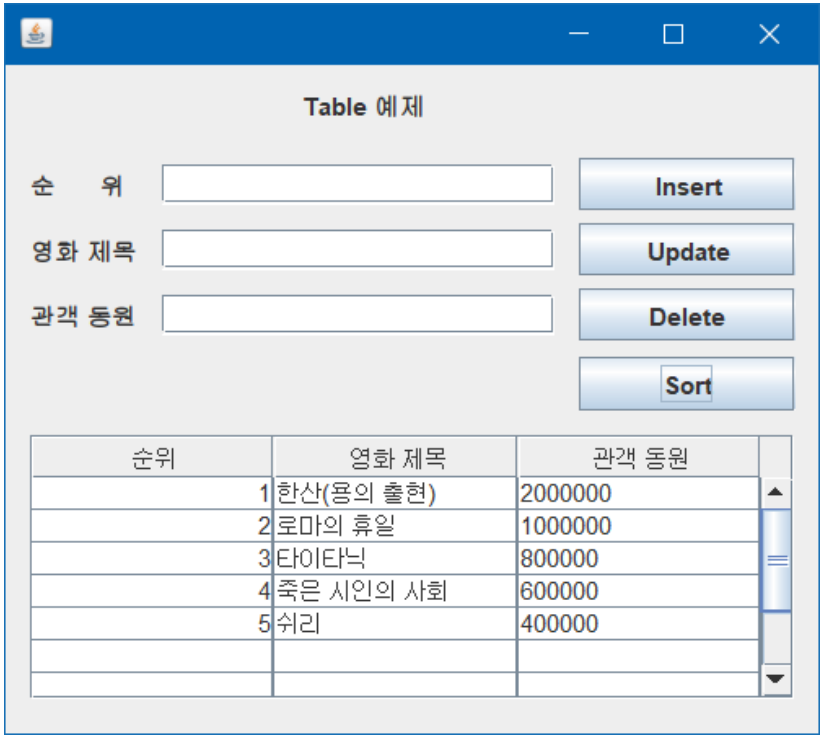
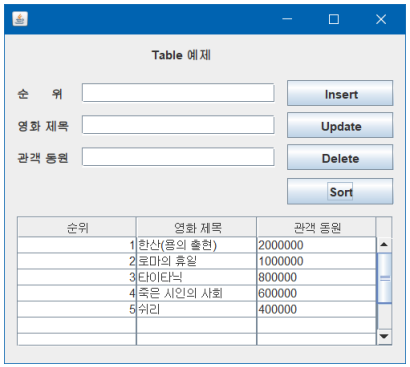


Table 예제
순 위
영화 제목
관객 동원
Insert
Update
Delete
Sort
순위
영화 제목
관객 동원
1 한산(용의 출현) 2000000
2 로마의 휴일 1000000
3 타이타닉 800000
4 죽은 시인의 사회 600000
5 쉬리 400000

학습 목표

이 강의를 마치면 학생들은

- Table Control에 대하여 설명할 수 있다.
- Tree Control에 대하여 설명할 수 있다.



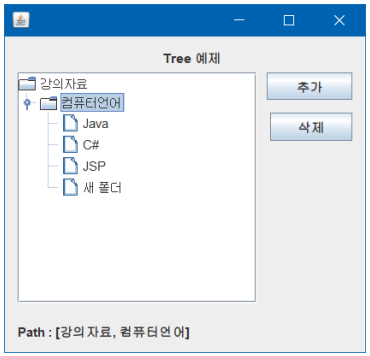


Table Control

Table Control 기능

- Table 형태의 데이터 저장

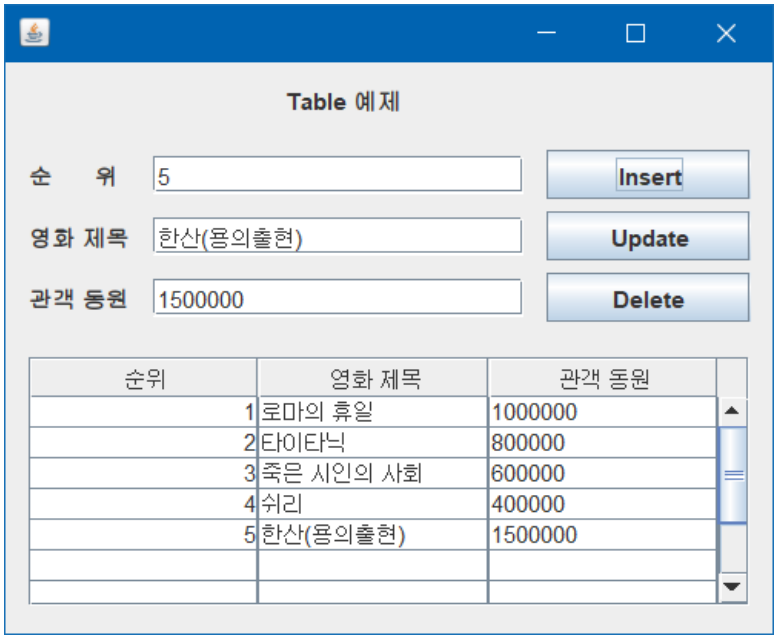
Method

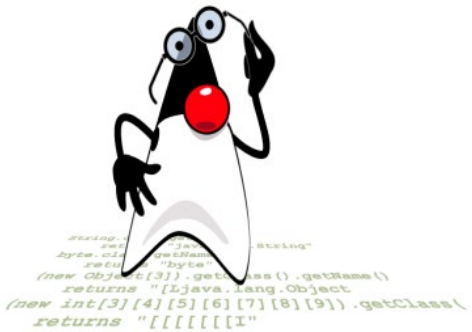
Method	기 능
int getColumnCount()	열의 수 반환
int getRowCount()	행의 수 반환
Object getValueAt(int row, int column)	해당 데이터 반환
String getColumnName(int column)	열 제목 반환
int getSelectedRow()	Table의 선택 행 번호 반환
int getSelectedColumn()	Table의 선택 열 번호 반환

Practice 1 : Table Control

❖ Project Name: Table_Source(Time: 20 min)

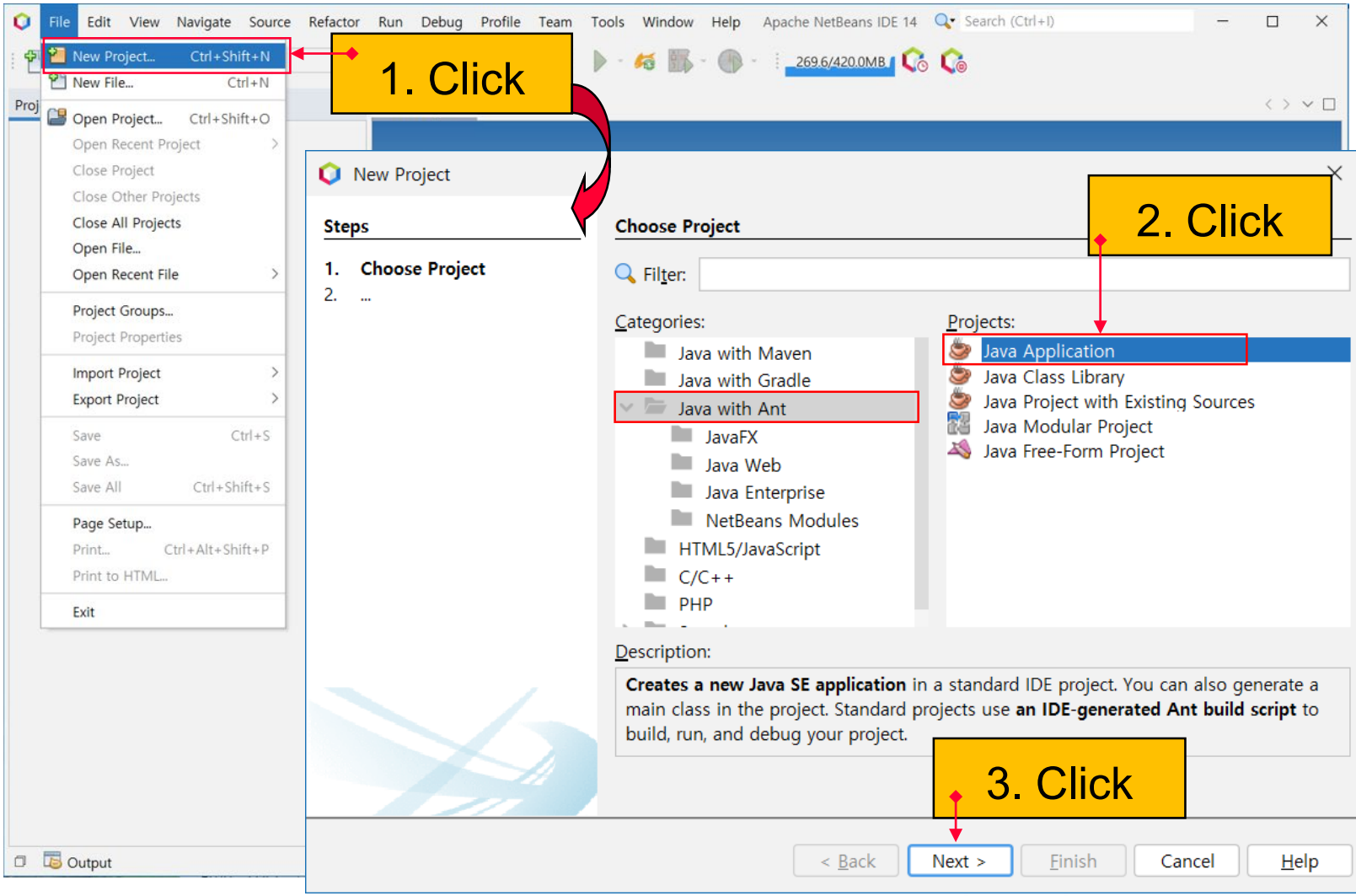
- [Insert] Button
  - 순위, 영화제목, 관객동원을 순위 입력값과 같은 행에 삽입한다.
- [Update] Button
  - Table에서 선택한 행의 순위, 영화제목, 관객동원을 데이터를 수정한다.
- [Delete] Button
  - Table에서 선택한 행의 순위, 영화제목, 관객동원을 데이터를 삭제한다.





Practice 1 : Table Control (1)

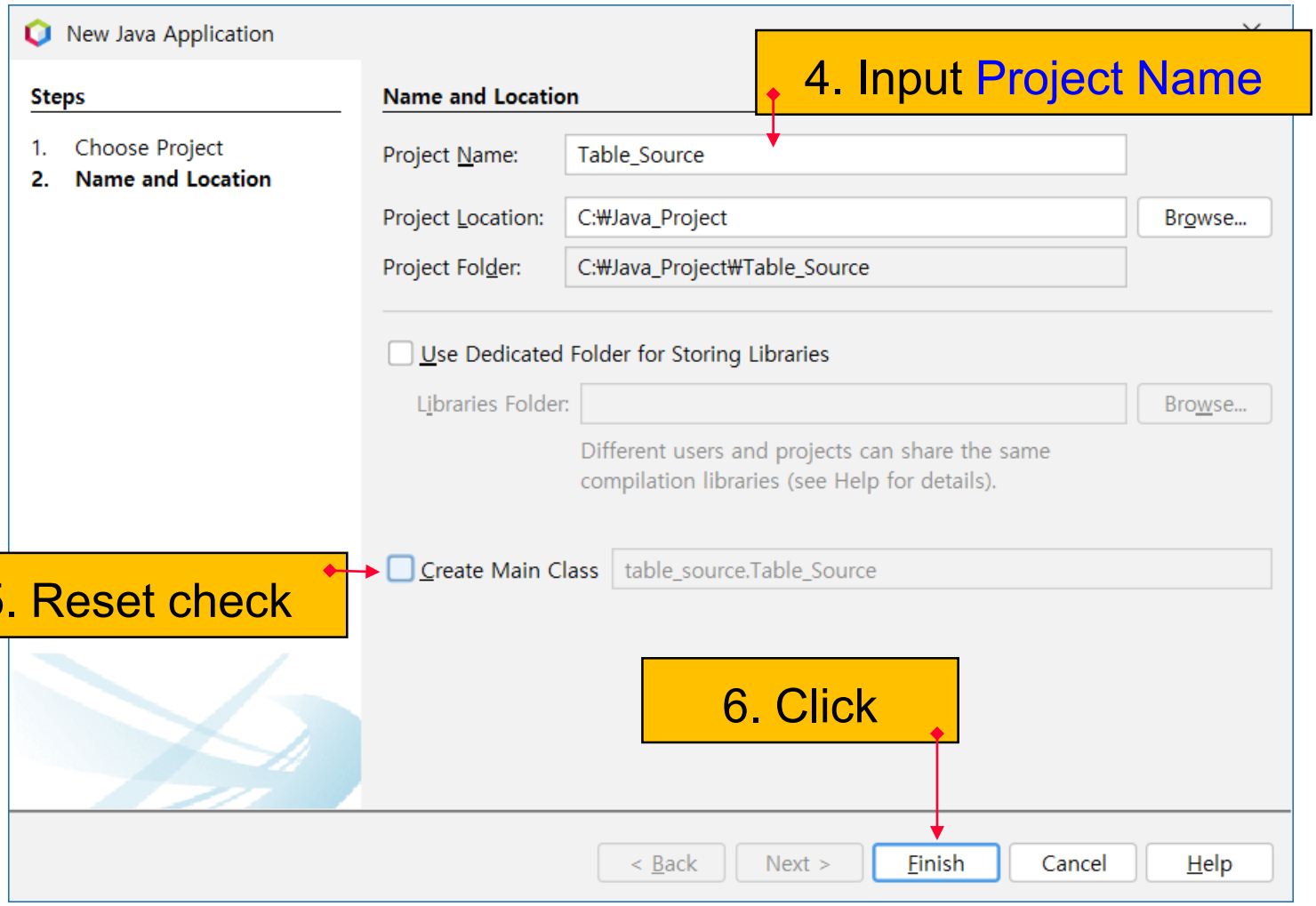
Create Project



Practice 1 : Table Control (2)

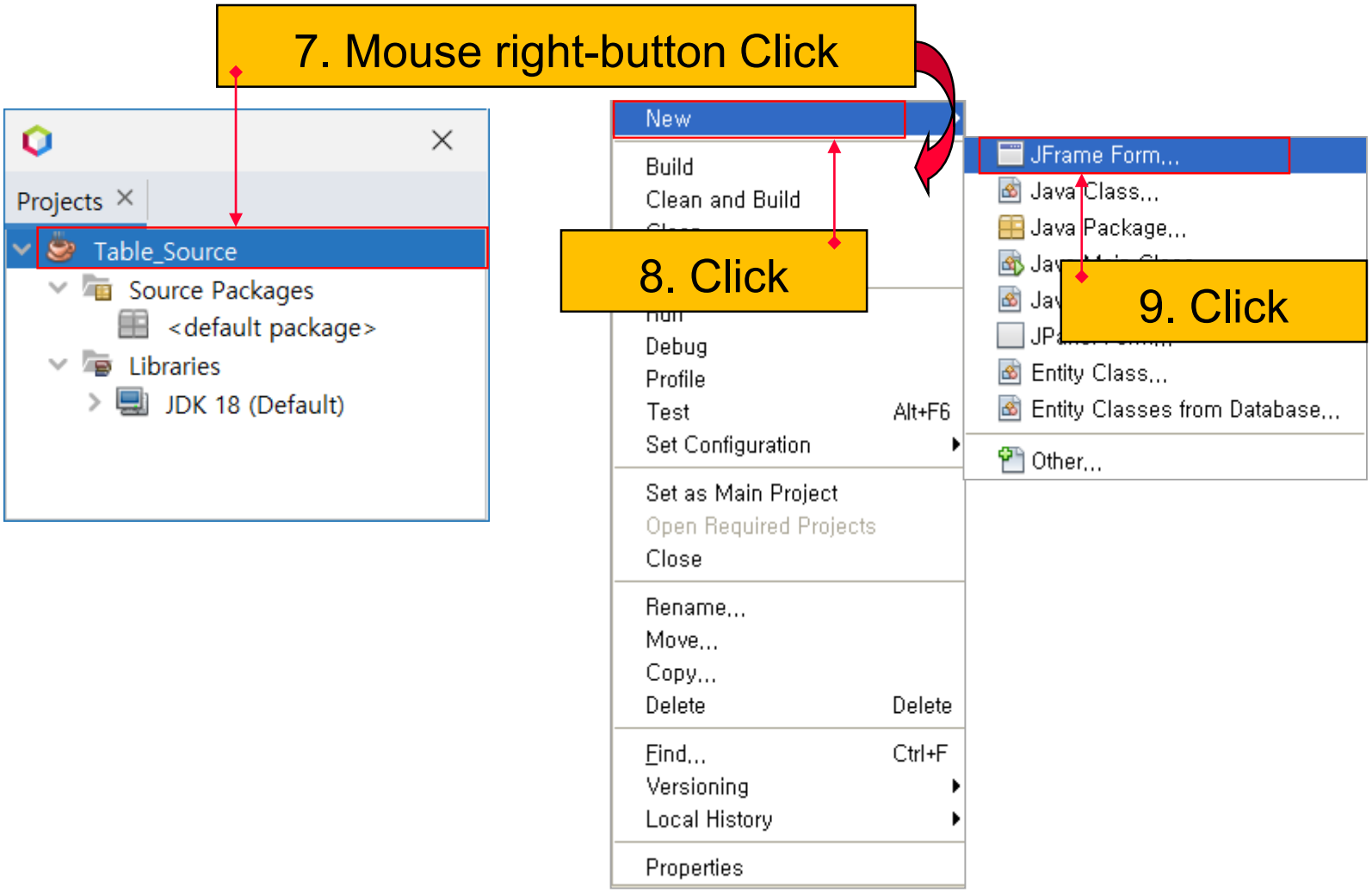
Project Name and Location

- Project name: Table_Source



Practice 1 : Table Control (3)

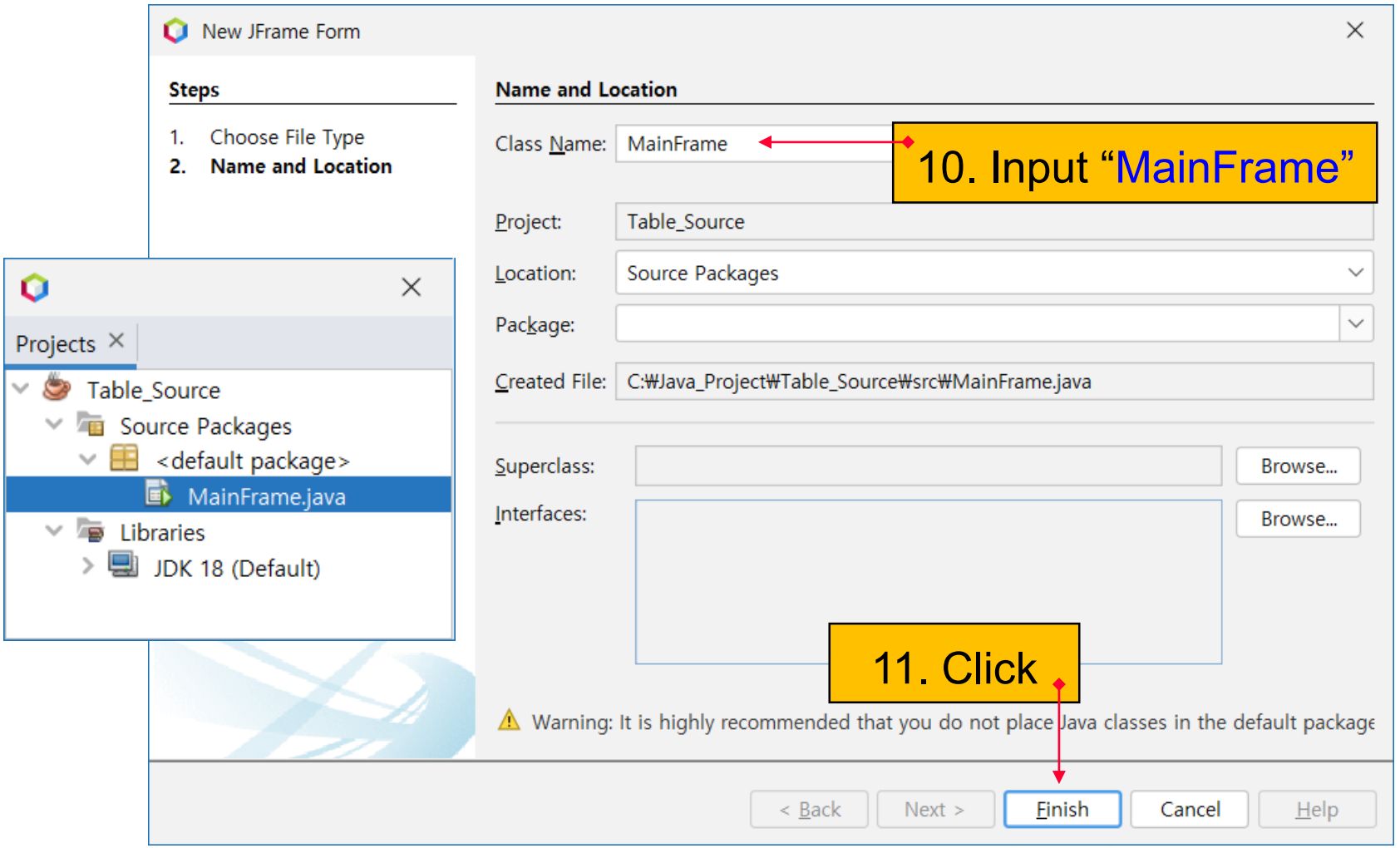
- Create JFrame Form



Practice 1 : Table Control (4)

Setting JFrame Form Name

❖ Create MainFrame.java



Practice 1 : Table Control (5)

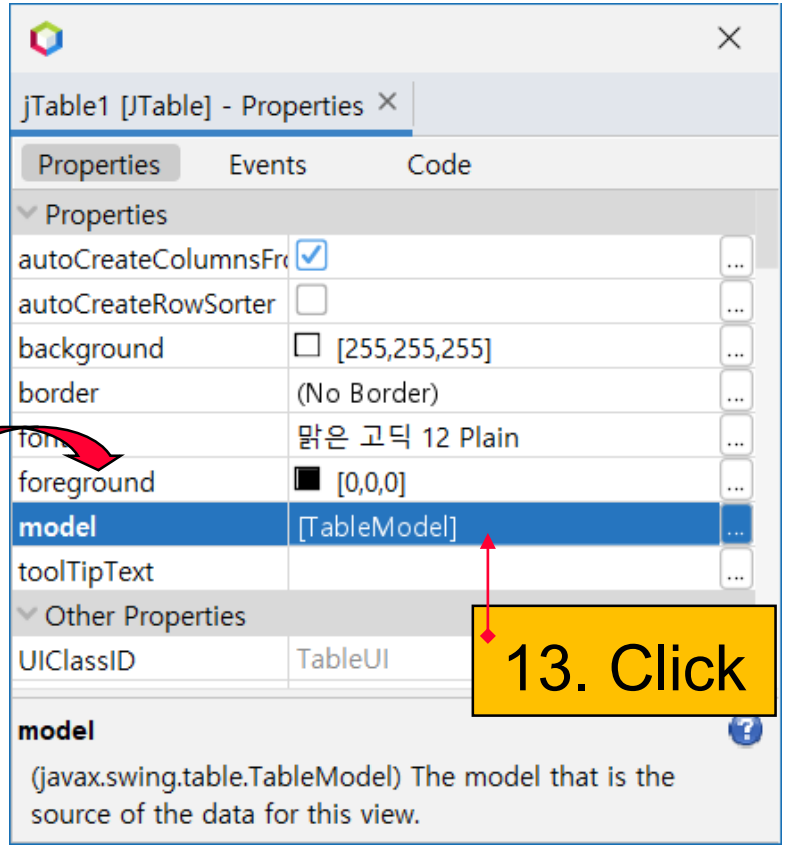
Control Layout & Property Setting

Control	Properties Setting
jLabel1	• Variable Name : lblTitle • Text : Table 예제
jLabel2	• Variable Name : lblOrder • Text : 순위
jLabel3	• Variable Name : lblMovie • Text : 영화 제목
jLabel4	• Variable Name : lblGallery • Text : 관객 동원

Control	Properties Setting
jTextField1	• Variable Name : txtOrder • Text :
jTextField2	• Variable Name : txtMovie • Text :
jTextField3	• Variable Name : txtGallery • Text :
jButton1	• Variable Name : btnInsert • Text : Insert
jButton2	• Variable Name : btnUpdate • Text : Update
jButton3	• Variable Name : btnDelete • Text : Delete
jTable	• Variable Name : jTable1

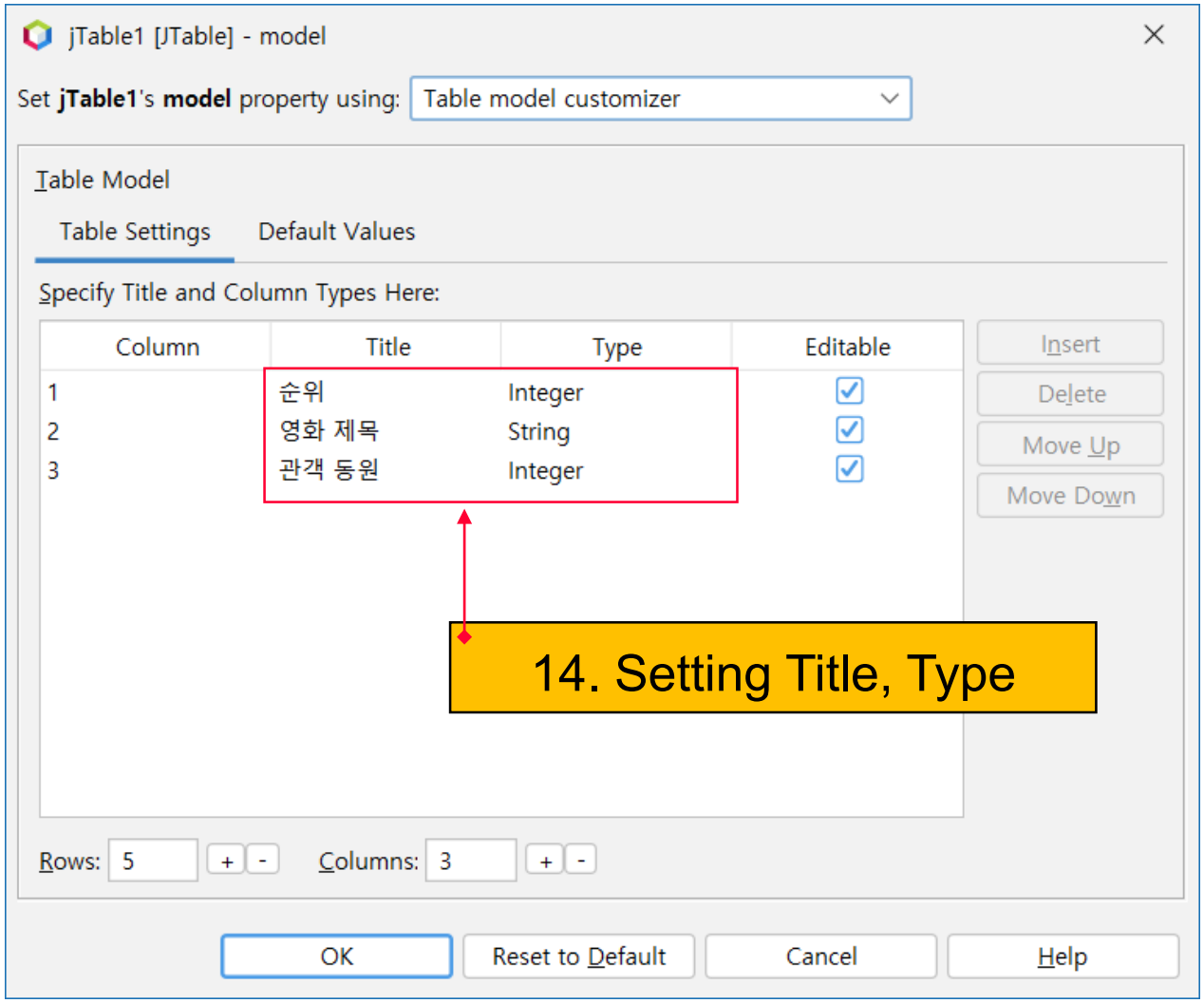
Practice 1 : Table Control (6)

Setting table model property



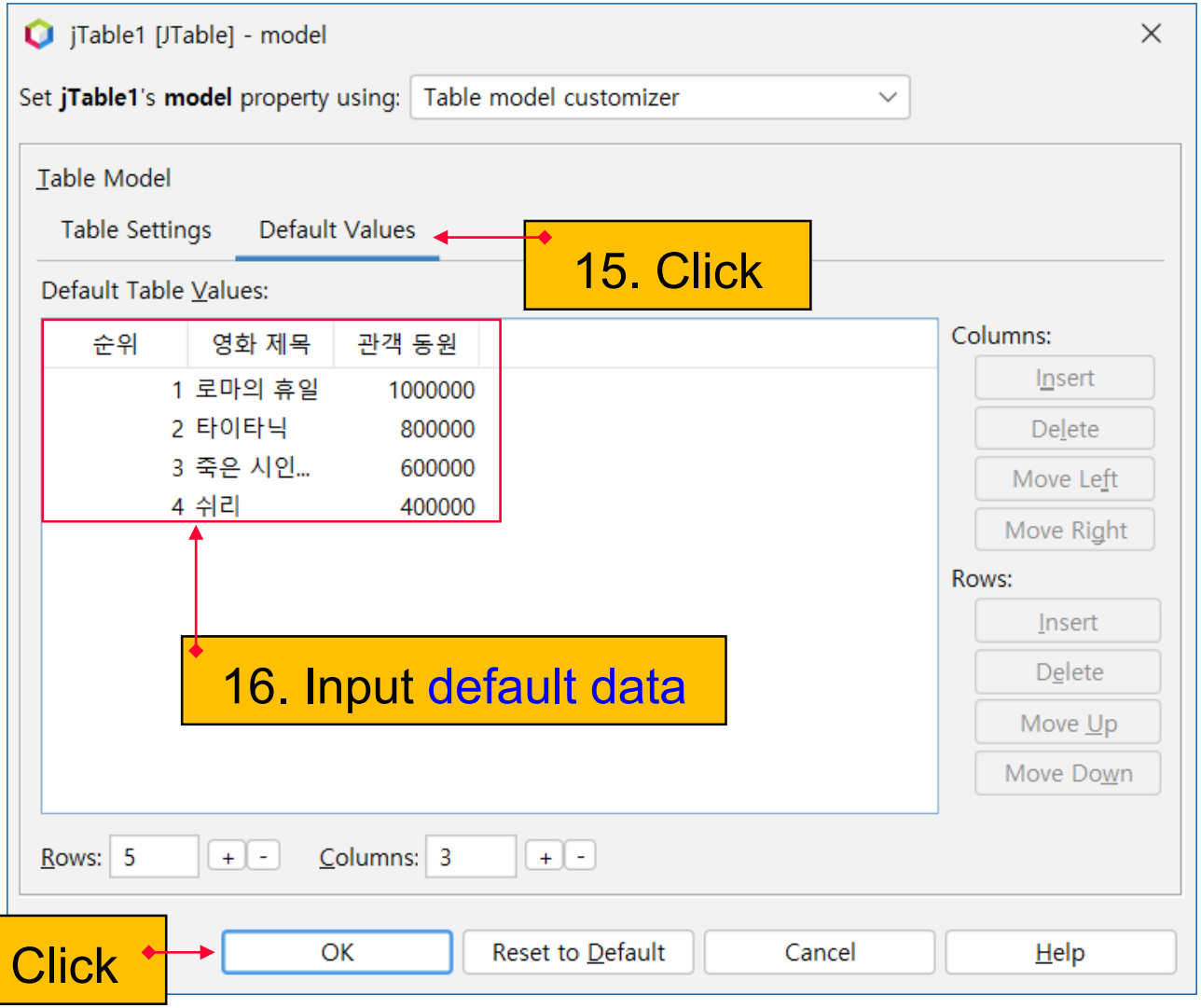
Practice 1 : Table Control (7)

❖ Table Settings



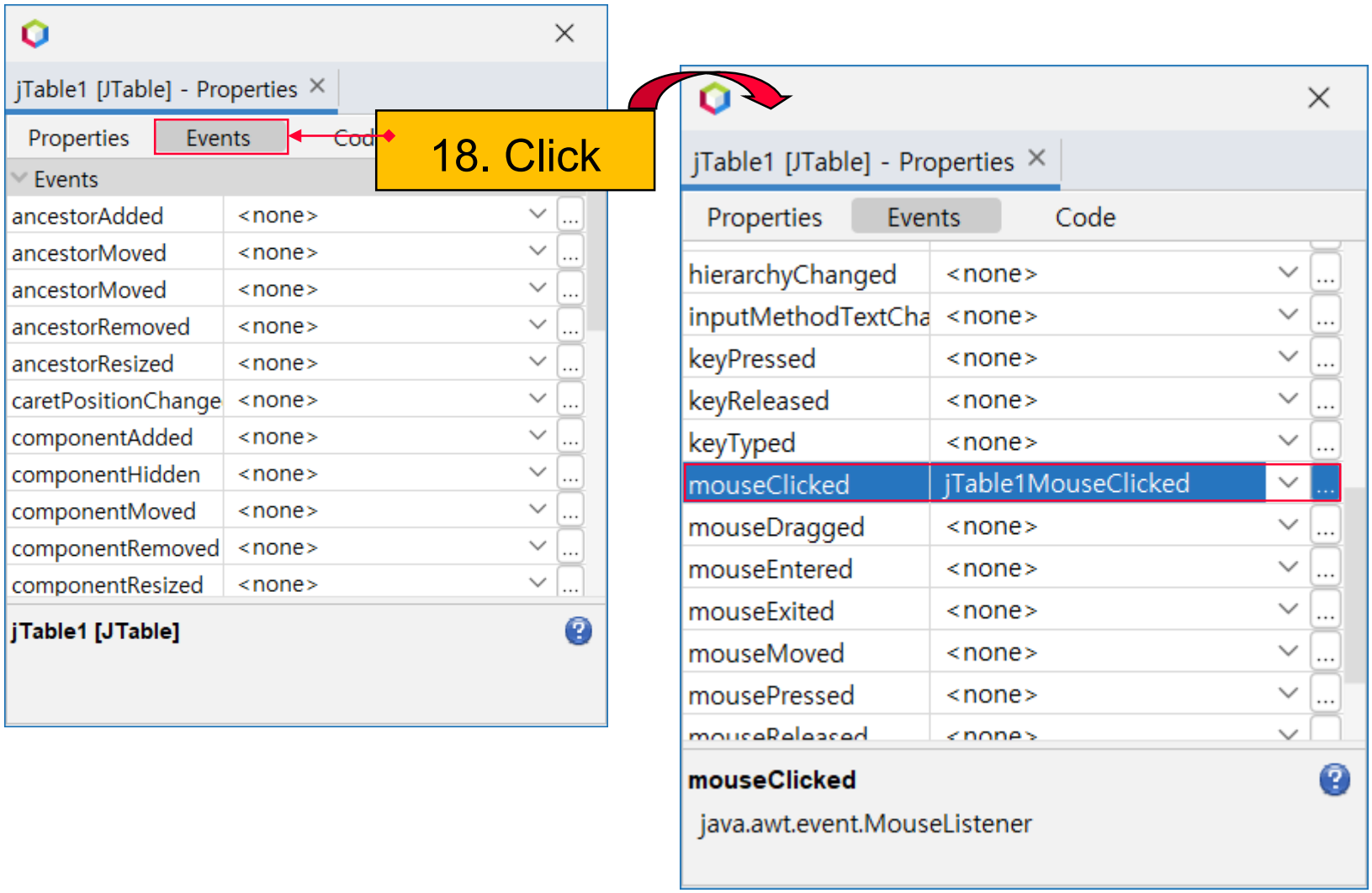
Practice 1 : Table Control (8)

❖ Setting Default Values property



Practice 1 : Table Control (9)

Setting mouseclicked() Event Handler



Practice 1 : Table Control (10)

❖ Setting jTable1 Control property

MainFrame.java - Editor

MainFrame.java

Source | Design | History

```
44
45      jTable1.setModel(new javax.swing.table.DefaultTableModel(
46          new Object [][] {
                {new Integer(1), "로마의 휴일",  new Integer(1000000)},
                {new Integer(2), "타이타닉",  new Integer(800000)},
                {new Integer(3), "죽은 시인의 사회",  new Integer(600000)},
                {new Integer(4), "쉬리",  new Integer(400000)},
51              {null, null, null}
52          },
53          new String [] {
54              "순위", "영화 제목", "관객 동원"
55          }
56      ) {
57          Class[] types = new Class [] {
58              java.lang.Integer.class, java.lang.String.class, java.lang.Integer.class
59          };
60
62          public Class getColumnClass(int columnIndex) {
63              return types [columnIndex];
64          }
65      });
67      jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
68          public void mouseClicked(java.awt.event.MouseEvent evt) {
69              jTable1MouseClicked(evt);
          }
      });
```

77:12 INS

Practice 1 : Table Control (11)

❖ jTable1MouseClicked() Event Handler

MainFrame.java - Editor

MainFrame.java

Source Design History

```
    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        int iCntRow = 0;
260     iCntRow = jTable1.getSelectedRow();   //Return the selected row
261
262     txtOrder.setText(jTable1.getValueAt(iCntRow, 0).toString());   //Return the rank
263     txtMovie.setText(jTable1.getValueAt(iCntRow, 1).toString());  //Return the movie title
264     txtGallery.setText(jTable1.getValueAt(iCntRow, 2).toString()); //Return the number of gallery
265 }
266
    private void btnSortActionPerformed(java.awt.event.ActionEvent evt) {...19 lines }
286
287 /**...3 lines */
290 public static void main(String args[]) {
291     java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
293         new MainFrame().setVisible(true);
294     }
295 });
296 }
```

19. Coding

171:1 INS

Practice 1 : Table Control (12)

❖ [Insert] Button Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

20. Coding

```
171
    private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
        int iCntRow = 0;
        int iOrder = 0;
        int iGallery = 0;
176
177     iCntRow = jTable1.getRowCount();          //Return the number of rows
178     for(int idx = 0; idx < jTable1.getRowCount(); idx++){
179         //Check the values in column 0 of each row
180         if (jTable1.getValueAt(idx, 0) == null){
181             iCntRow = idx;
182             break;
183         }
184     }
185
186     iOrder = Integer.parseInt(txtOrder.getText());
187     jTable1.setValueAt( iOrder, iCntRow, 0);                  //Setting the rank
188
189     jTable1.setValueAt(txtMovie.getText(), iCntRow, 1);  //Setting the movie title
190
191     iGallery = Integer.parseInt(txtGallery.getText());
192     jTable1.setValueAt( iGallery, iCntRow, 2);                //Setting the number of gallery
193 }
```

295:12 INS

Practice 1 : Table Control (13)

❖ [Update] Button Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

```
199
200     iCntRow = jTable1.getSelectedRow();       //Returns the selected row number
201
202     iOrder = Integer.parseInt(txtOrder.getText());
203      jTable1.setValueAt( iOrder, iCntRow, 0);                  //Setting the rank
204
205      jTable1.setValueAt(txtMovie.getText(), iCntRow, 1);  //Setting the movie title
206
207      iGallery = Integer.parseInt(txtGallery.getText());
208      jTable1.setValueAt( iGallery, iCntRow, 2);                //Setting the number of gallery
209  }
210
252
253  public class MakeRowData{...4 lines }
```

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    int iCntRow = 0;
    int iOrder = 0;
    int iGallery = 0;
```

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {...41 lines }
```

21. Coding

261:9 | INS | Explict type can be replaced with 'var'

Practice 1 : Table Control (14)

❖ [Delete] Button Event Handler

MainFrame.java - Editor

MainFrame.java

Source Design History

```
212     private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
            MakeRowData objRowData;
            Vector myVC = new Vector();

214
            int iCntRow = 0;
216         iCntRow = jTable1.getSelectedRow();          //Returns the selected row number
217
218         DefaultTableModel jTableModel = (DefaultTableModel)jTable1.getModel();
219
220         //Step 1: jTable1 -> myVC
221         for(int iRow = 0; iRow < jTable1.getRowCount(); iRow++){
222             if (jTable1.getValueAt(iRow, 0) != null) {
223                 objRowData = new MakeRowData();
224                 objRowData.strMovie = jTable1.getValueAt(iRow, 1).toString();
225                 objRowData.iGallery = Integer.parseInt(jTable1.getValueAt(iRow, 2).toString());
226                 myVC.add(objRowData);
227             } else{
228                 break;
229             }
230         }
231
232         //Step 2: Remove seleted row
233         myVC.removeElementAt(iCntRow);
234         jTableModel.removeRow(iCntRow);
235
```

1:1 INS

22. Coding

Practice 1 : Table Control (15)

MainFrame.java - Editor

MainFrame.java

Source | Design | History

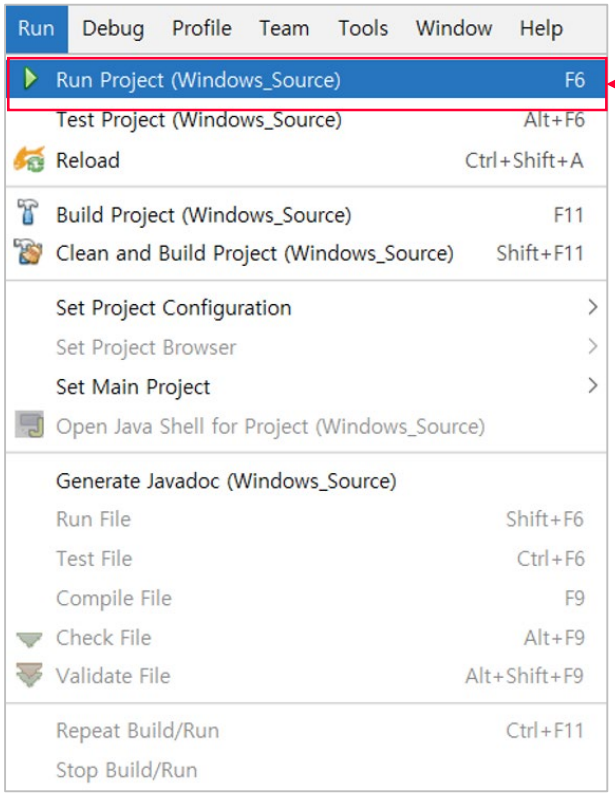
```
235
236     //Step 3: myVC -> jTable1
237     for(int idx = 0; idx < myVC.size(); idx++){
238         objRowData = (MakeRowData) myVC.get(idx);
239         jTable1.setValueAt(idx+1, idx, 0);
240         jTable1.setValueAt(objRowData.strMovie, idx, 1);
241         jTable1.setValueAt(objRowData.iGallery, idx, 2);
242     }
243
244     //Step 4: Add one dummy row to jTable1
245     String[] strRecord = new String[ jTable1.getColumnCount()];
246      jTableModel.addRow(strRecord);
247
248     txtOrder.setText(null);                    // Initialize txtOrder
249     txtMovie.setText(null);                    // Initialize txtMovie
250     txtGallery.setText(null);                  // Initialize txtGallery
251 }
252
253 public class MakeRowData{
254     public String strMovie;
255     public int iGallery;
256 }
```

1:1 INS

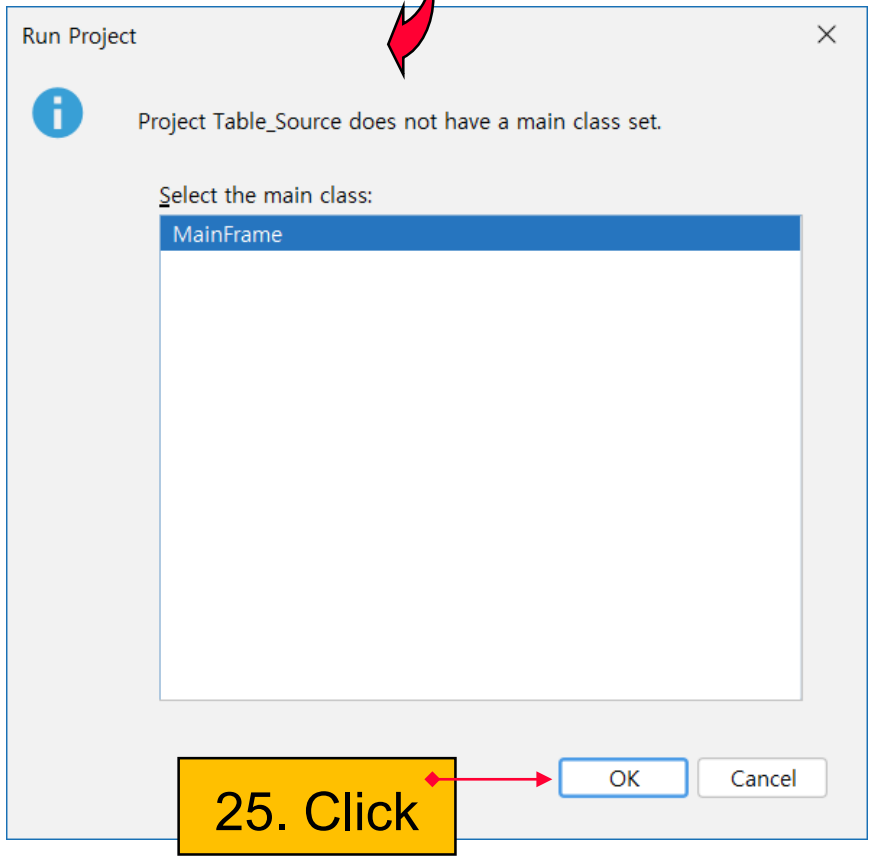
23. Coding

Practice 1 : Table Control (16)

Run



24. Click

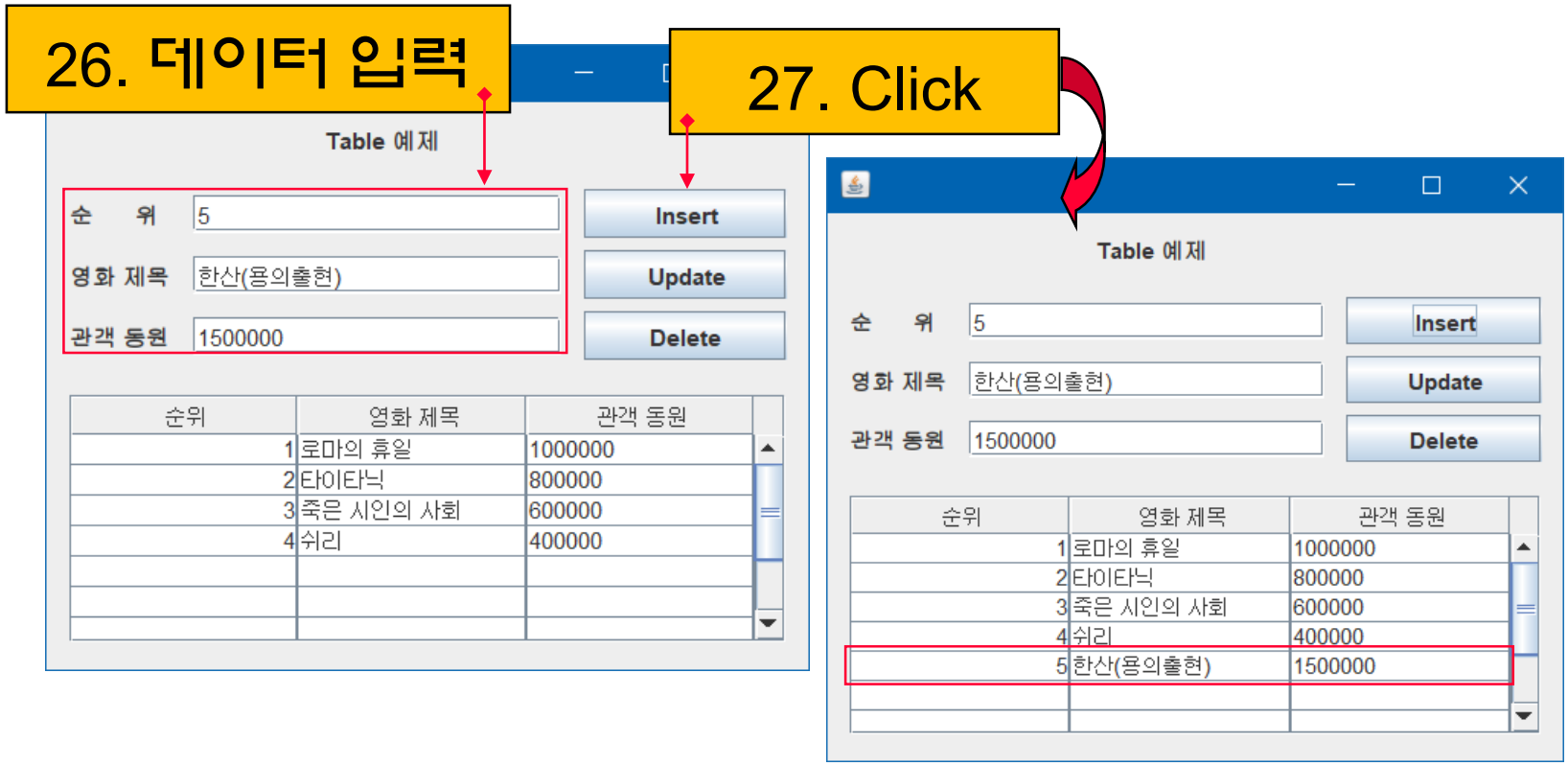


25. Click

Practice 1 : Table Control (17)

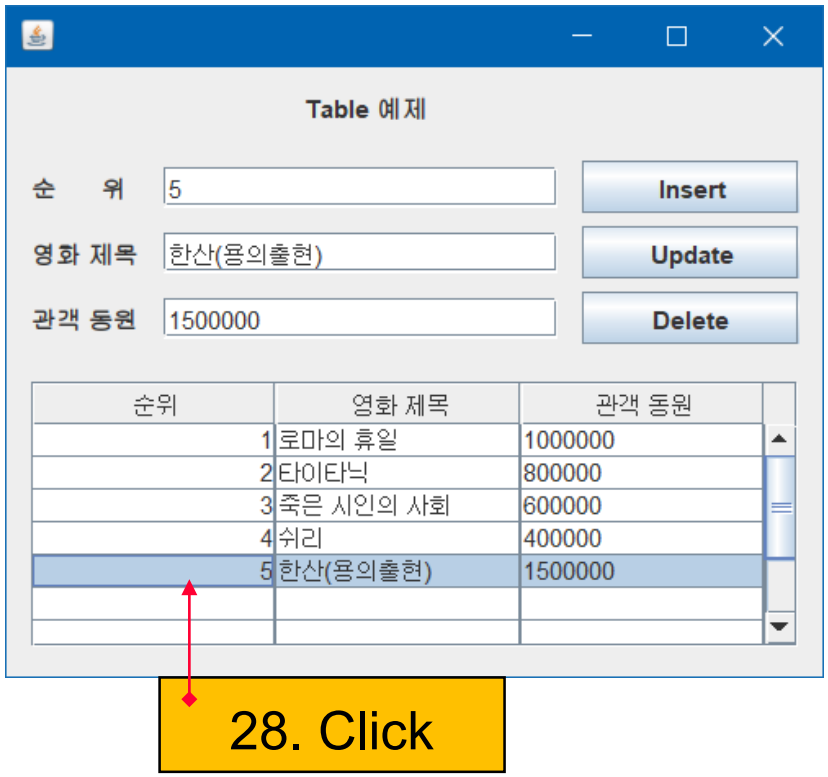
Run

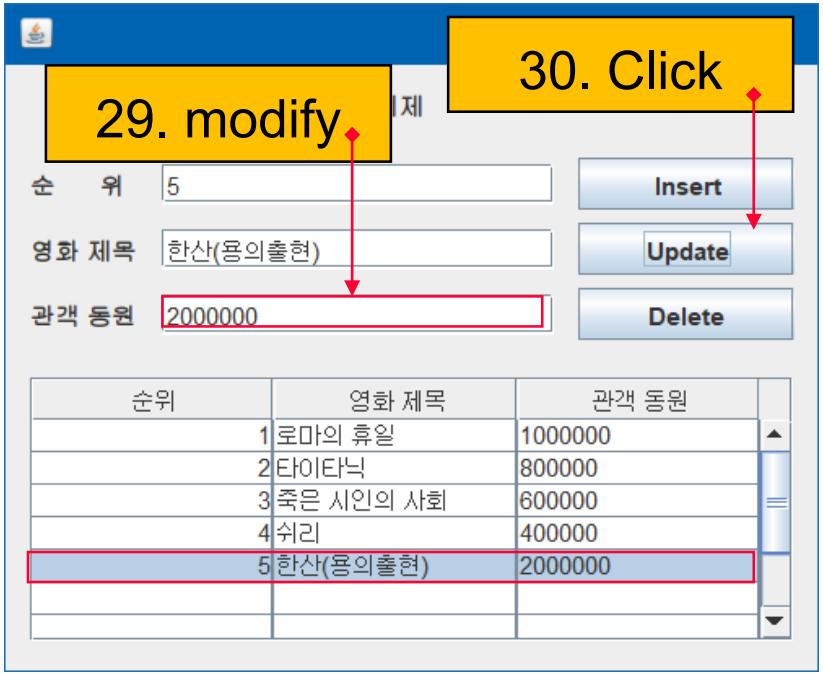
❖ [Insert] Button Click



Practice 1 : Table Control (18)

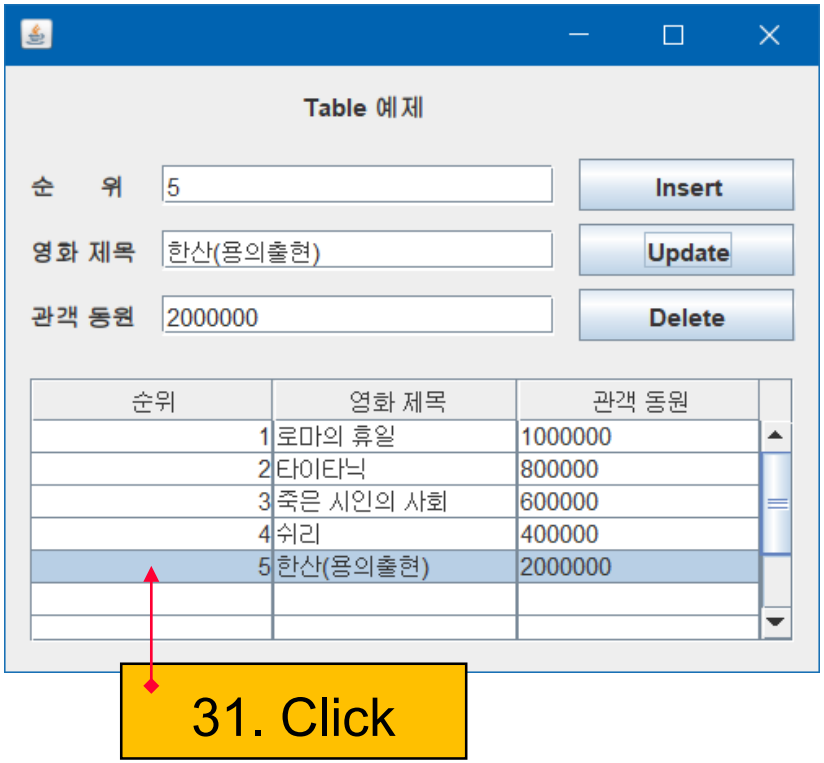
❖ [Update] Button Click

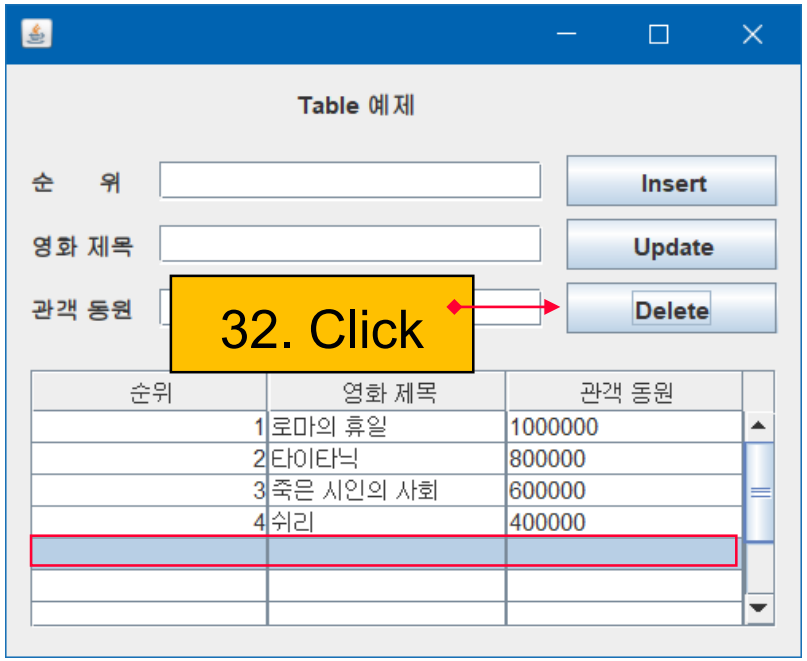




Practice 1 : Table Control (19)

❖ [Delete] Button Click

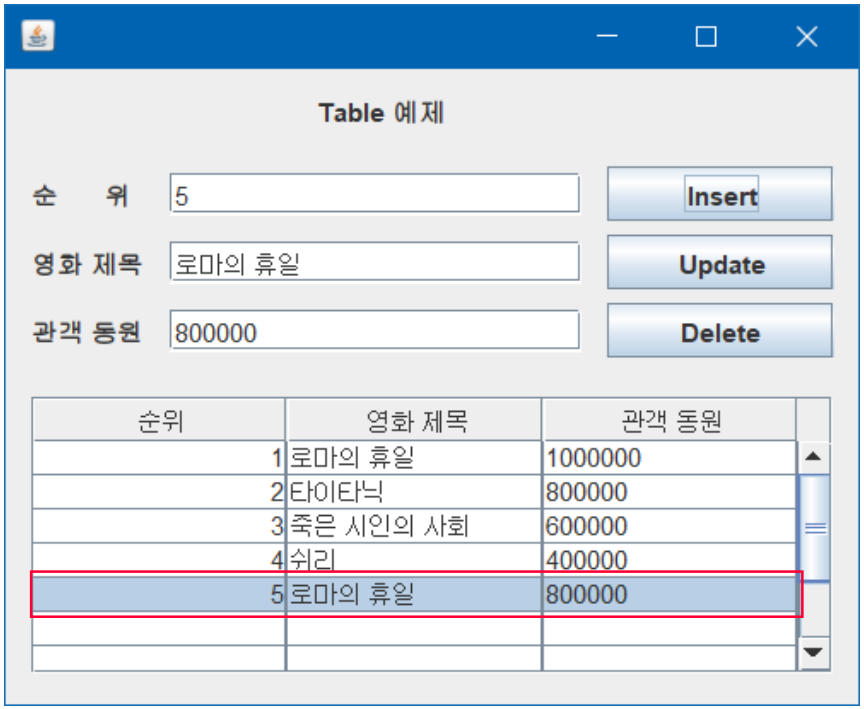


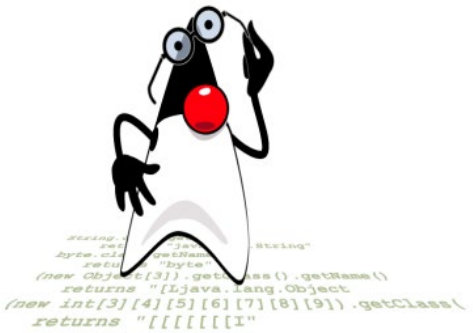


Practice 2: Table Control

❖ Project Name: Table_Source(Time: 30 min)

- [Insert] Button
  - 아래 그림의 실행 결과에서 “로마의 휴일” 과 같이 동일한 데이터가 삽입되지 않도록 [Insert] button event handler를 수정하시오.

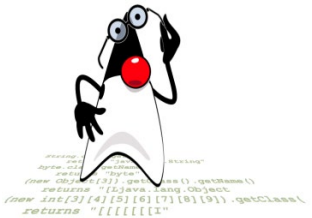


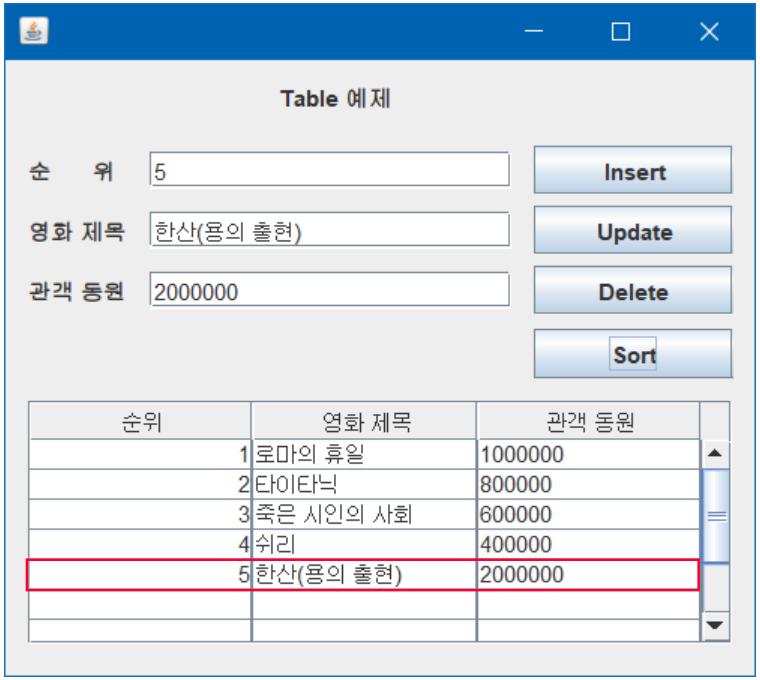


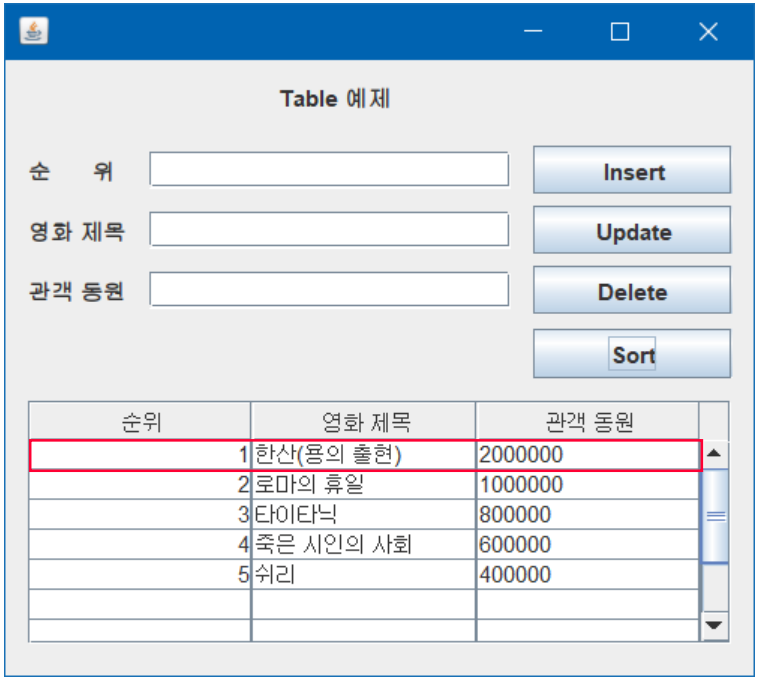
Homework

❖ Project Name: TableSorting_Source

- [Sort] Button
  - 관객 동원" 수에 따라 내림차순으로 정렬하도록 구현하시오.

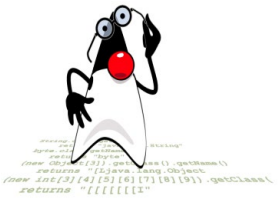






Homework (1)

❖ Solution 1



MainFrame.java - Editor

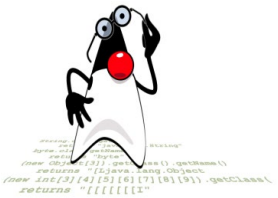
MainFrame.java

Source | Design | History

```
218     private void btnSortActionPerformed(java.awt.event.ActionEvent evt) {
            MakeRowData objRowData;
            Vector myVC = new Vector();
220
221         //1. jTable1 -> myVC
222         for(int iRow = 0; iRow < jTable1.getRowCount(); iRow++){
223             if (jTable1.getValueAt(iRow, 0) != null) {
224                 objRowData = new MakeRowData();
225                 objRowData.strMovie = jTable1.getValueAt(iRow, 1).toString();
226                 objRowData.iGallery = Integer.parseInt(jTable1.getValueAt(iRow, 2).toString());
227                 myVC.add(objRowData);
228             } else{
229                 break;
230             }
231         }
232
233         //2. Sorting
234                                 ❶
235
236         //3. myVC ->  jTable1
237         for(int idx = 0; idx < myVC.size(); idx++){
238             objRowData = (MakeRowData) myVC.get(idx);
239             jTable1.setValueAt(idx+1, idx, 0);
240             jTable1.setValueAt(objRowData.strMovie, idx, 1);
241             jTable1.setValueAt(objRowData.iGallery, idx, 2);
242         }
243
244         //4. Initialize TextField
245         txtOrder.setText(null);                    // Initialize txtOrder
246         txtMovie.setText(null);                    // Initialize txtMovie
247         txtGallery.setText(null);                  // Initialize txtGallery
248     }
```

1:1 INS

Homework (2)



MainFrame.java - Editor

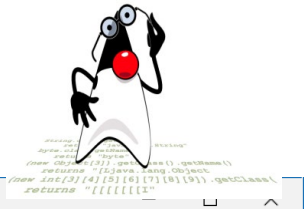
MainFrame.java

Source | Design | History

```
250  public class MakeRowData{
251      public String strMovie;
252      public int iGallery;
253  }
254  public class GalleryDescCompare implements Comparator {
         public int compare(Object arg0, Object arg1){
256          //Ascending(<) : Descending(>)
257          if(((MakeRowData)arg0).iGallery > ((MakeRowData)arg1).iGallery )
258                  return -1;
259          else  if(((MakeRowData)arg0).iGallery == ((MakeRowData)arg1).iGallery)
260                  return Integer.compare(((MakeRowData)arg0).iGallery, ((MakeRowData)arg1).iGallery);
261          else
262                 return 1;
263      }
264  }
265
266  /**...3 lines */
269  public static void main(String args[]) {
         java.awt.EventQueue.invokeLater(new Runnable() {
             public void run() {
272              new MainFrame().setVisible(true);
273          }
274      });
275  }
```

219:1 INS

Homework (3)



❖ Solution 2

MainFrame.java - Editor

MainFrame.java

Source | Design | History

```
266
    private void btnSortActionPerformed(java.awt.event.ActionEvent evt) {
268
269     TableRowSorter<TableModel> objSorter = new TableRowSorter<>(jTable1.getModel());
270     jTable1.setRowSorter(objSorter);
271     ArrayList<RowSorter.SortKey> SortKeys = new ArrayList<>();
272
273
274                                  ❶
275
276
277
278     for(int idx = 0; idx < jTable1.getRowCount(); idx++){
279         if (jTable1.getValueAt(idx, 0) != null)
280             jTable1.setValueAt(idx+1, idx, 0);
281         else
282             break;
283     }
284
285 }
286
```

252:5 INS

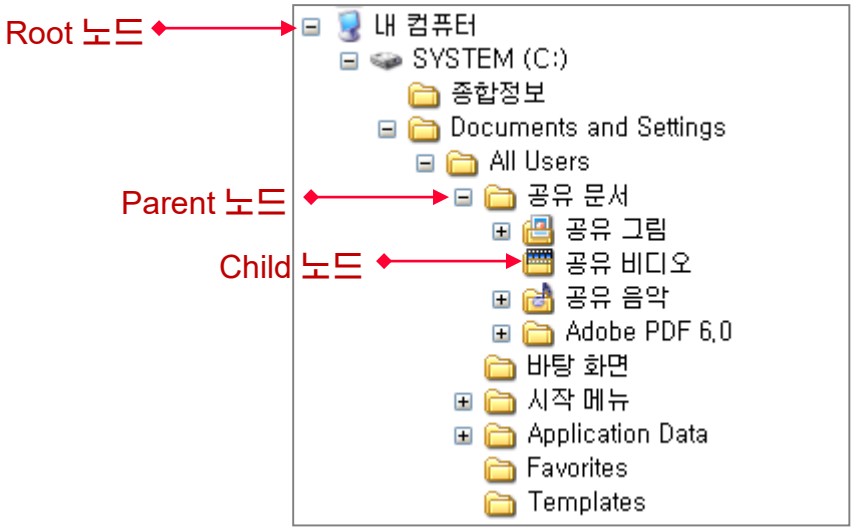
Tree Control (1)

Tree Control 기능

- 계층적인 자료를 보여주는 기능
  - 폴더 구조와 같이 트리 형태의 데이터 구조

Tree 구조

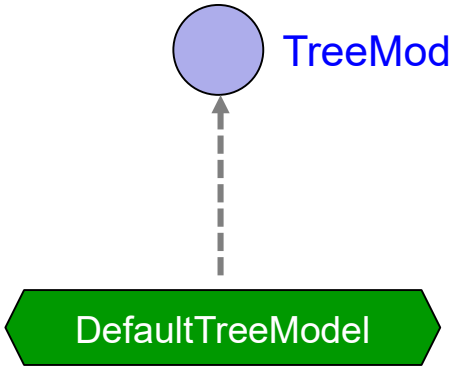
- Root 노드
- Child 노드
- Parent 노드



Tree Control (2)

Model

- TreeModel
  - Tree 노드 간의 관계를 알려 주는 메서드가 정의되어 있는 인터페이스
  - 데이터 보관 관리
- DefaultTreeModel
  - TreeModel의 모든 메서드 미리 구현
- Method



Method	Function
Object getChild(Object parent, int index)	Parent node의 자식들중에 index에 해당하는 자식 노드 반환
int getChildCount(Object parent)	자식 노드 수 반환
boolean isLeaf(Object node)	말단 노드 여부 반환

Tree Control (3)

TreeSelectionModel

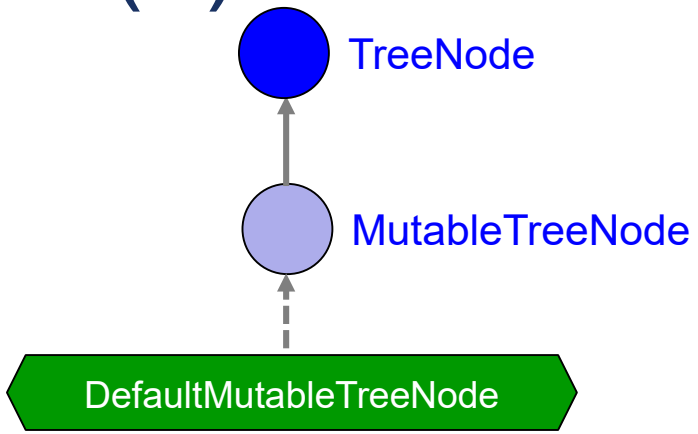
- 노드 선택 관리
- Mode
  - SINGLE_TREE_SELECTION
    - 한번에 한 노드만 선택
  - CONTIGUOUS_TREE_SELECTION
    - 연속된 여러 노드 선택
  - DISCONTIGUOUS_TREE_SELECTION
    - 연속하지 않은 여러 노드 선택
- Method

Method	Function
void setSelectionPath(TreePath path)	한번에 하나의 노드 선택
TreePath getSelectionPath()	선택한 노드의 path 반환
void setSelectionPaths(TreePath[] paths)	여러 개의 노드를 다 선택
TreePath[] getSelectionPaths()	선택한 여러 노드의 path 반환

Tree Control (4)

TreeNode

- 노드의 상하관계 표현
- DefaultMutableTreeNode
- Method

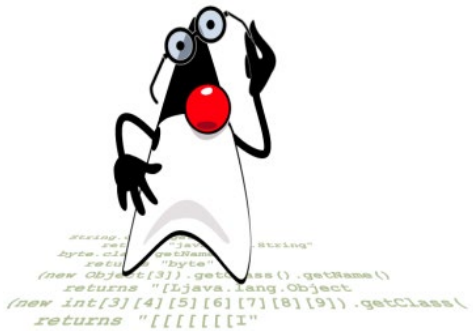


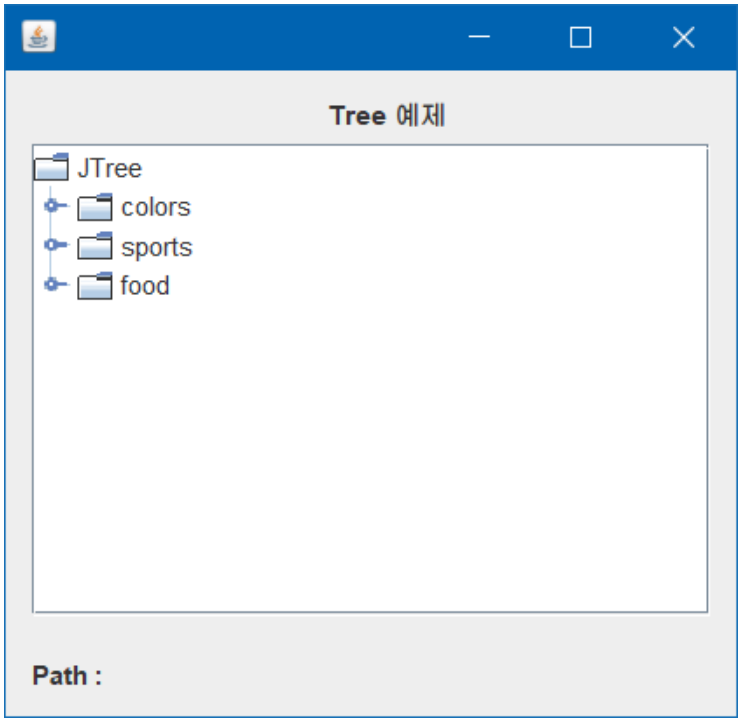
Method	Function
int getParent()	부모 노드 index 반환
int getChildAt()	자식 노드 index 반환
void add(MutableTreeNode newChild)	기존 tree의 말단 노드로 newChild 추가
void add(MutableTreeNode newChild, int index)	기존 tree의 index 위치에 newChild 추가
void remove(int index)	index 위치에 노드 삭제
void remove(MutableTreeNode aChild)	aChild 노드 삭제
void removeAllChildren()	모든 Child 노드 삭제
void insertNodeInto(MutableTreeNode newChild, MutableTreeNode parent, int index)	Parent 노드의 index 위치에 자식 노드로 newChild 추가
void reload()	변경된 node 를 포함한 젠처 tree 구조 반영
void nodeStructureChanged(TreeNode node)	Node의 하위 노드만을 update

Practice 2 : Tree Control

❖ Project Name: Tree_Source(Time: 30 min)

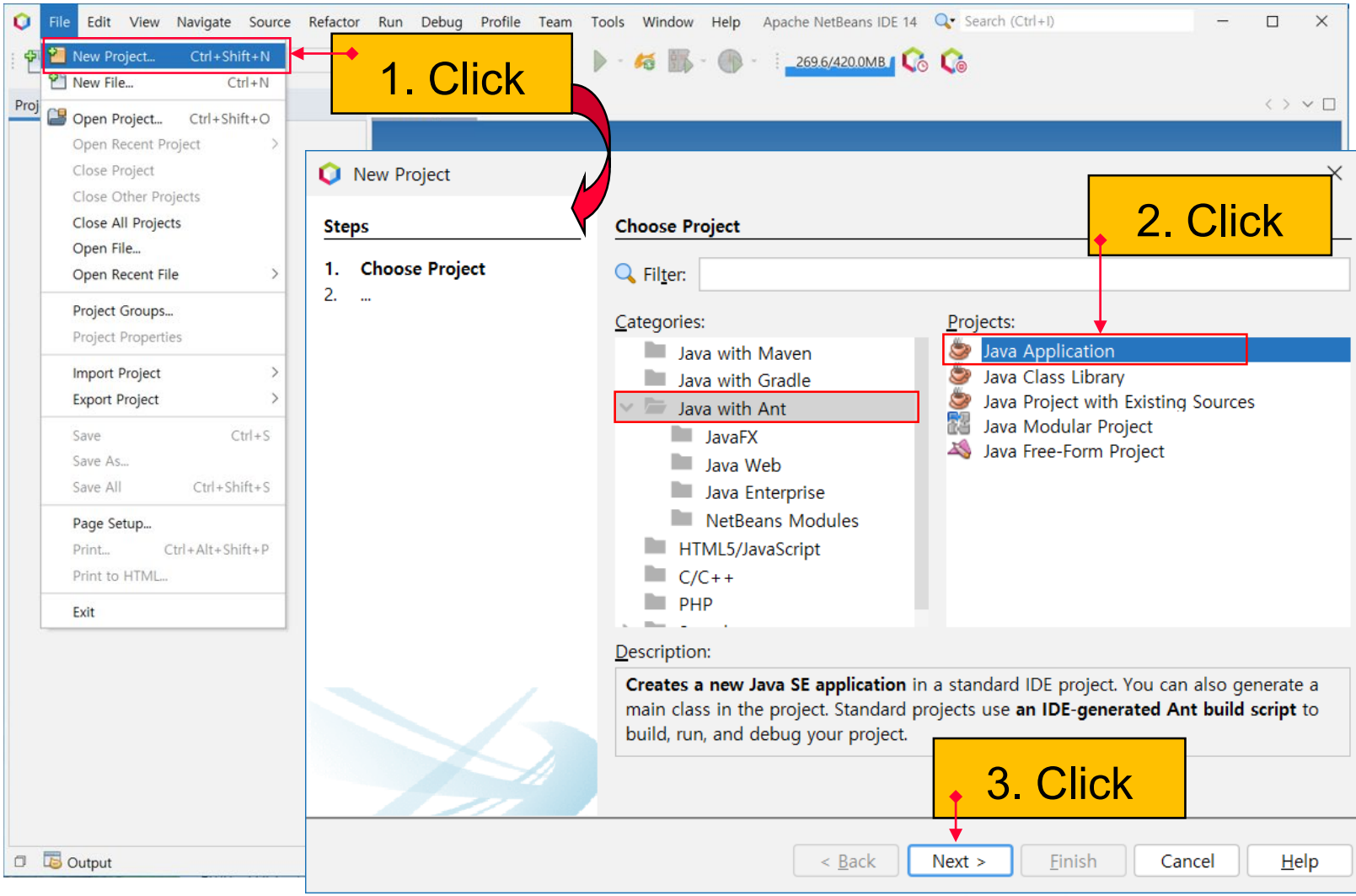
- Tree 폴더 Click
  - Path : Tree의 해당 폴더 선택 경로 출력
- Tree Expanded
  - Tree의 폴더 확장 경로 출력
- Tree Collapsed
  - Tree의 폴더 축소





Practice 2 : Tree Control (1)

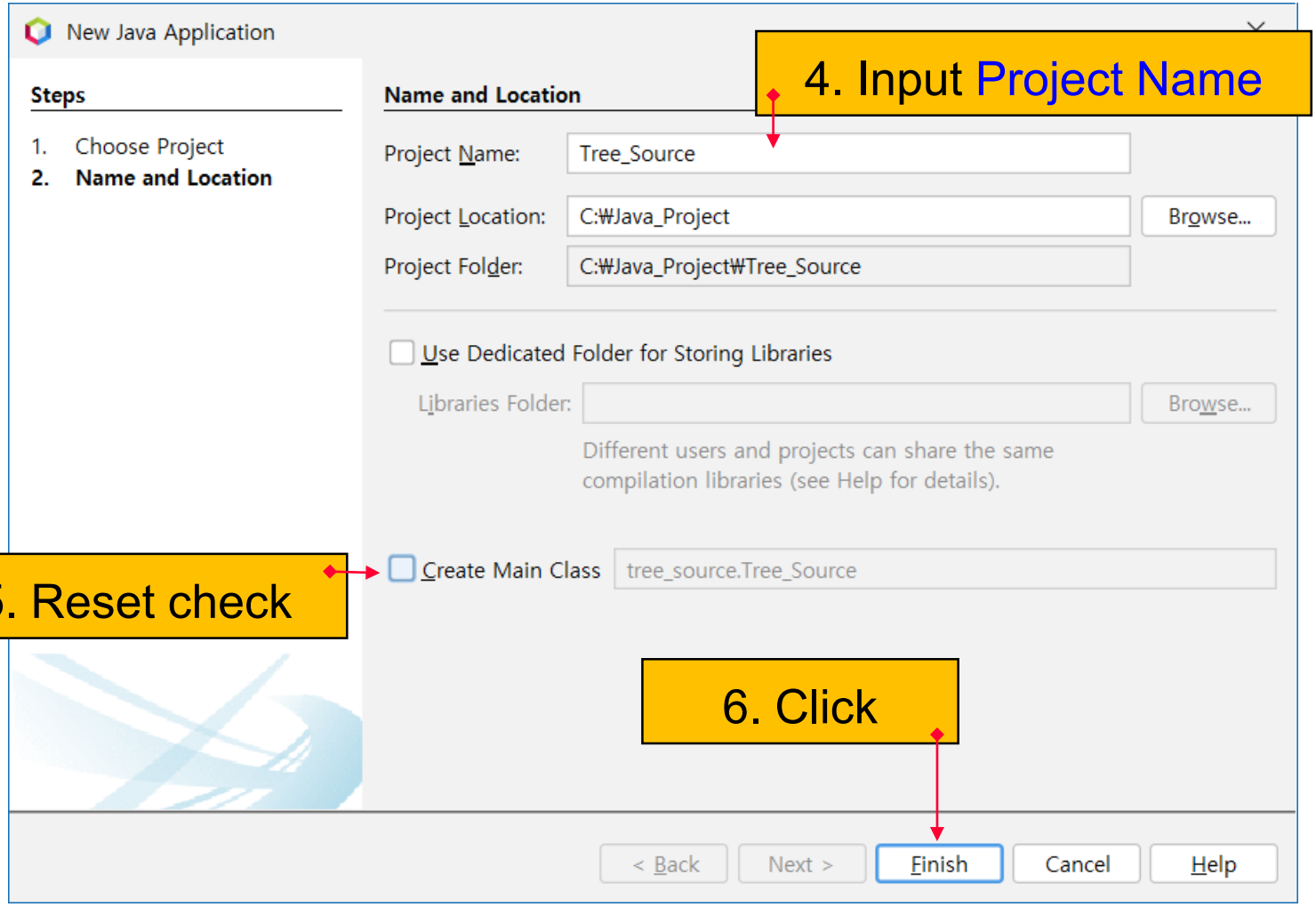
Create Project



Practice 2 : Tree Control (2)

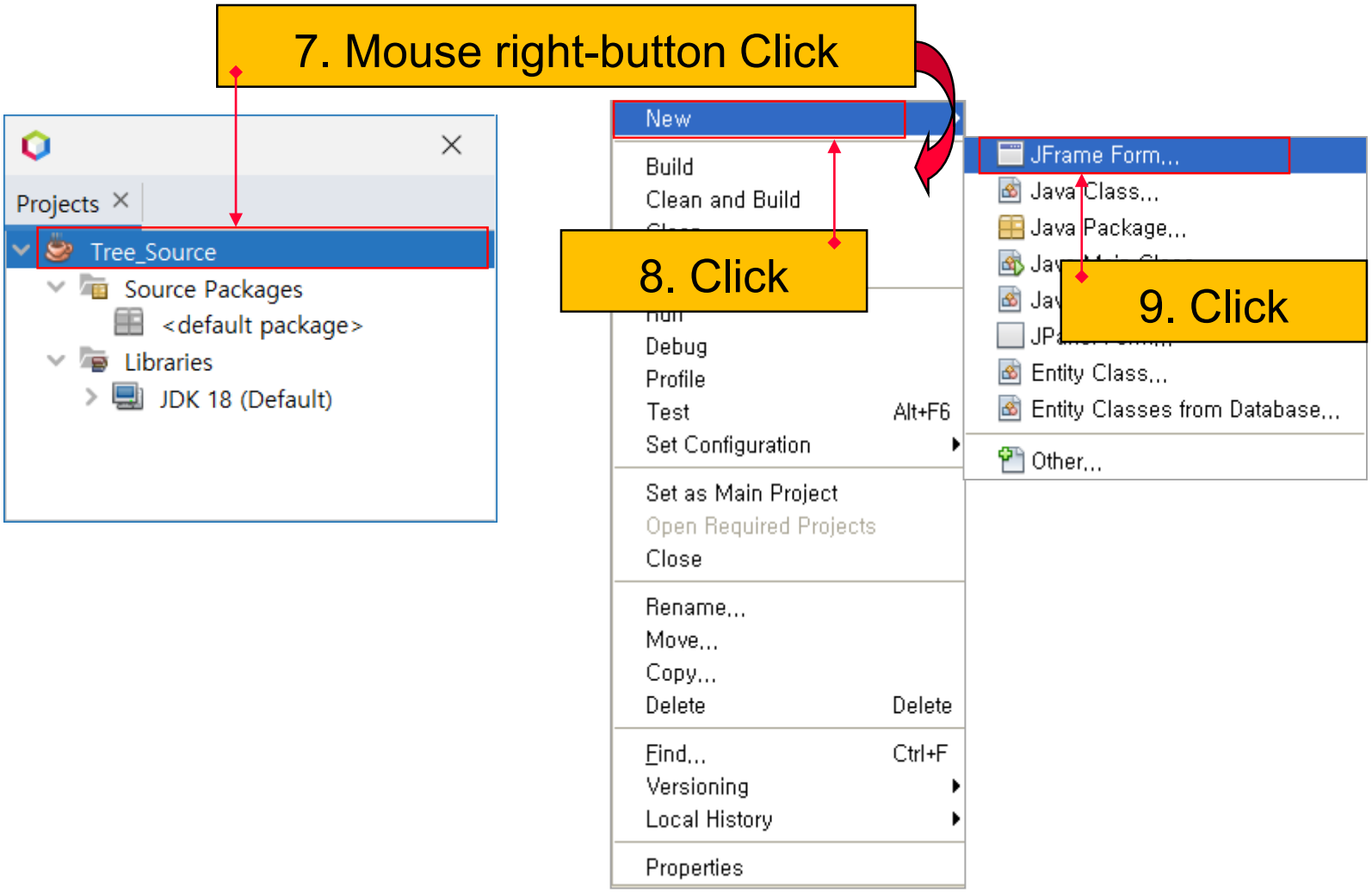
Project Name and Location

- Project name: Tree_Source



Practice 2 : Tree Control (3)

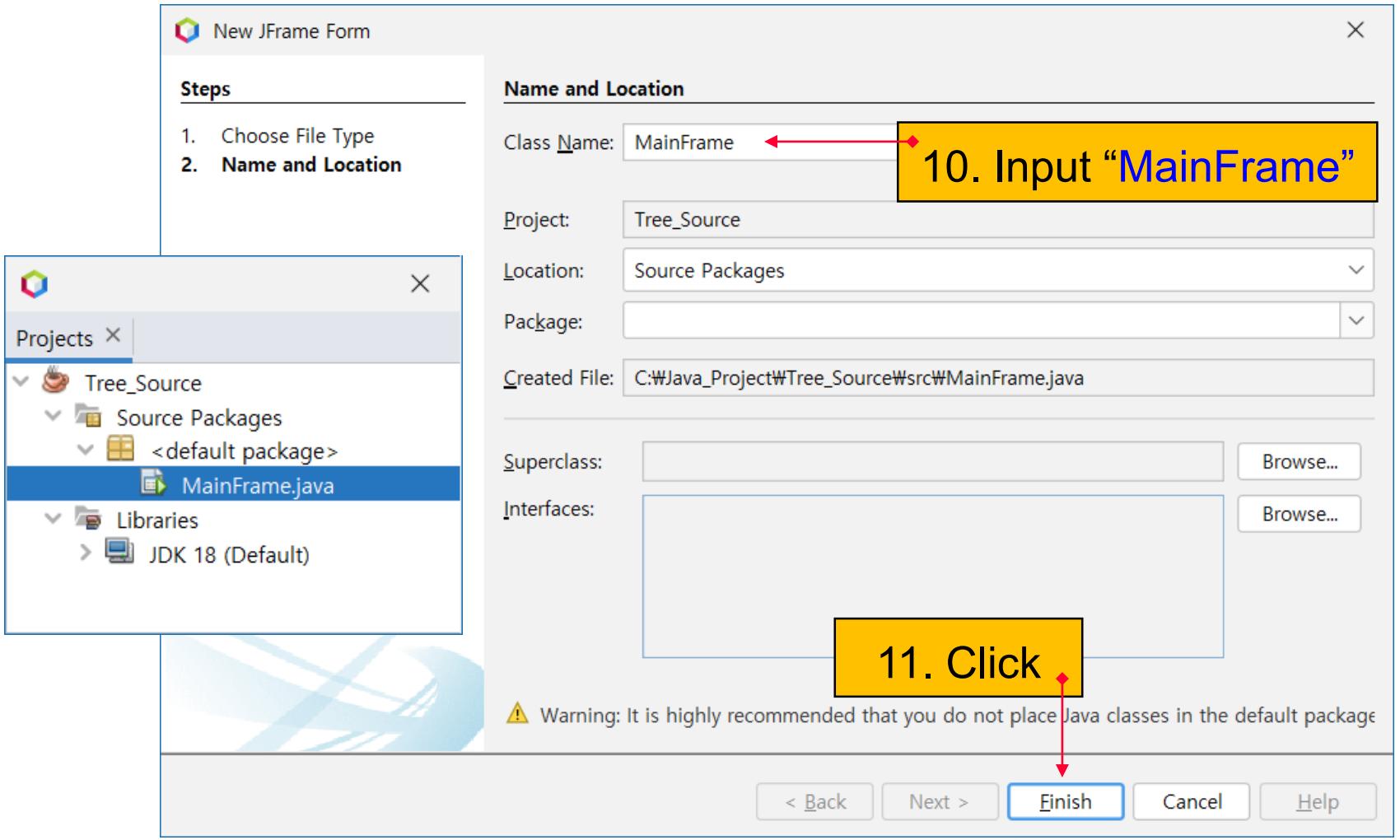
Create JFrame Form



Practice 2 : Tree Control (4)

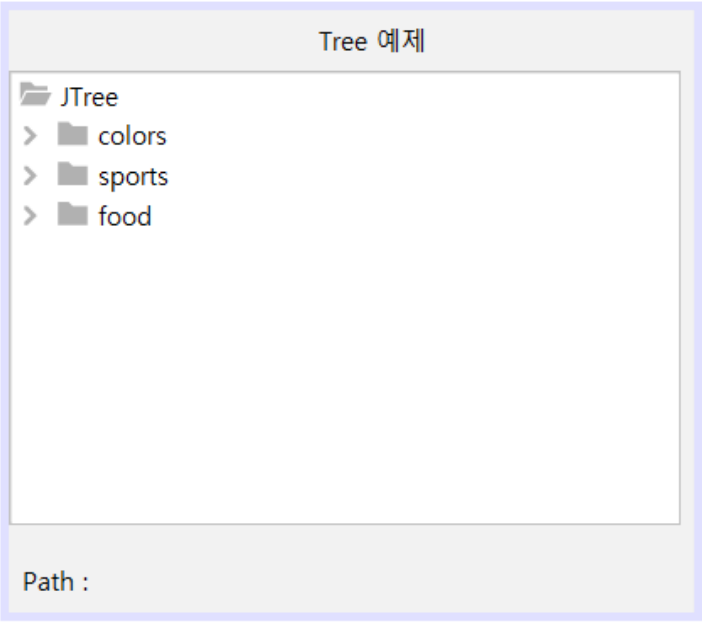
Setting JFrame Form Name

Create MainFrame.java



Practice 2 : Tree Control (5)

Control Layout & Property Setting



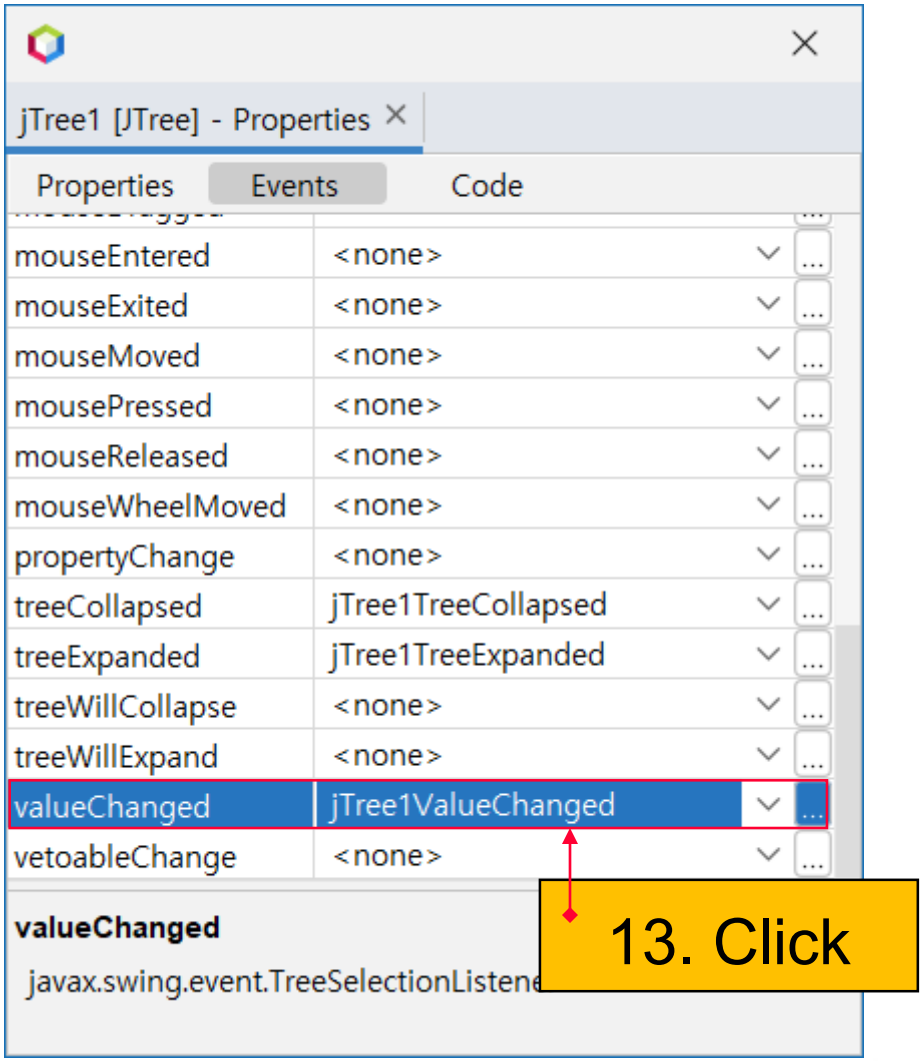
12. UI Design

Control	Properties Setting
jLabel1	• Variable Name : lblTitle • Text : Tree 예제
jLabel2	• Variable Name : lblPath • Text : Path :
jTree1	• Variable Name : jTree1

Event	Properties Setting
treeExpanded	• tree 확장시 발생:
treeCollapsed	• tree 축소시 발생
valueChanged	• Tree 값 변할때 발생

Practice 2 : Tree Control (6)

- Setting jTree1 Control valueChanged() Event Handler



Practice 2 : Tree Control (7)

- jTree1ValueChanged() Event Handler

MainFrame.java - Editor

MainFrame.java

Source Design History

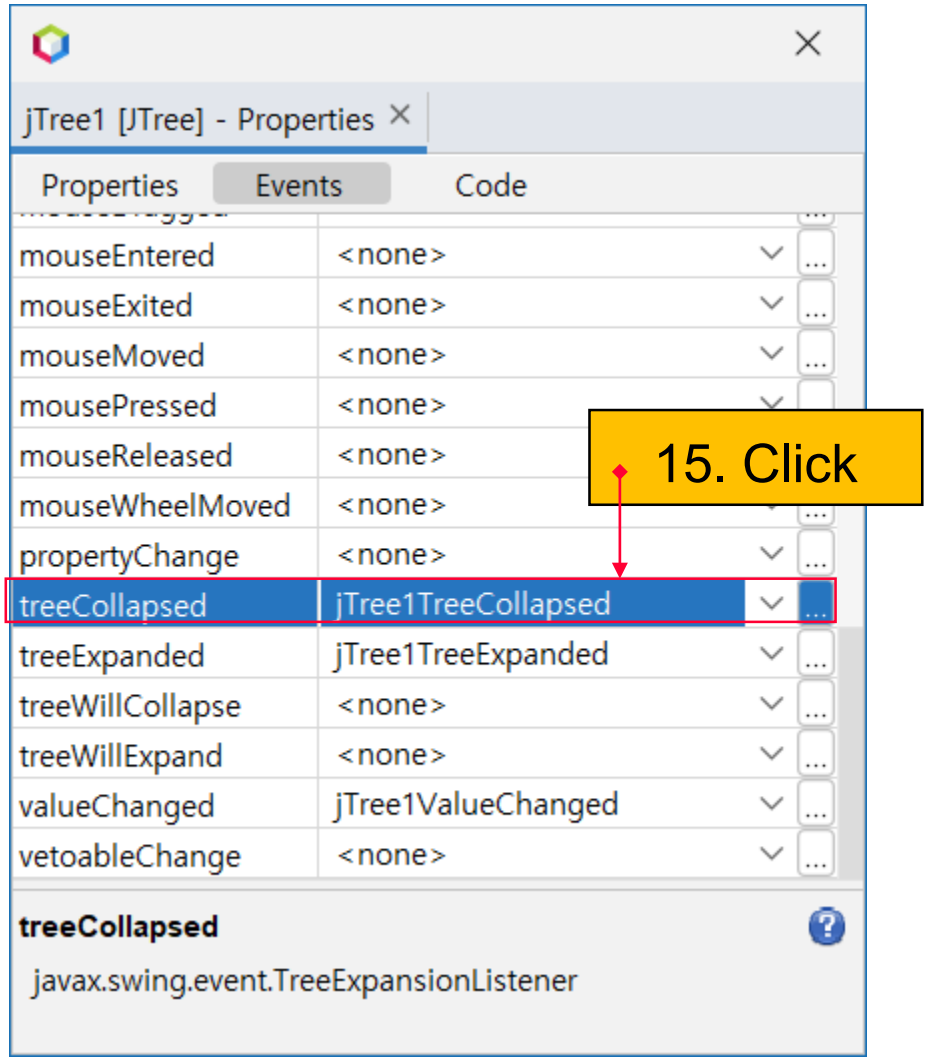
```
1
2   import javax.swing.tree.*;
3
4   public class MainFrame extends javax.swing.JFrame {
6       /** Creates new form MainFrame */
7       public MainFrame() {
8           initComponents();
9       }
10
11      /** This method is called from within the constructor to ...5 lines */
16      Generated Code
77
        private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {
79          DefaultMutableTreeNode node;
80          node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();
81          if (node == null) return;
82
83          lblPath.setText("Path : " + node.getUserObject() + " 선택 !!! ₩n");
84      }
```

91:59 INS Unused Import

14. Coding

Practice 2 : Tree Control (8)

- Setting jTree1 Control Tree Collapsed() Event Handler



Practice 2 : Tree Control (9)

❖ jTree1TreeCollapsed() Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

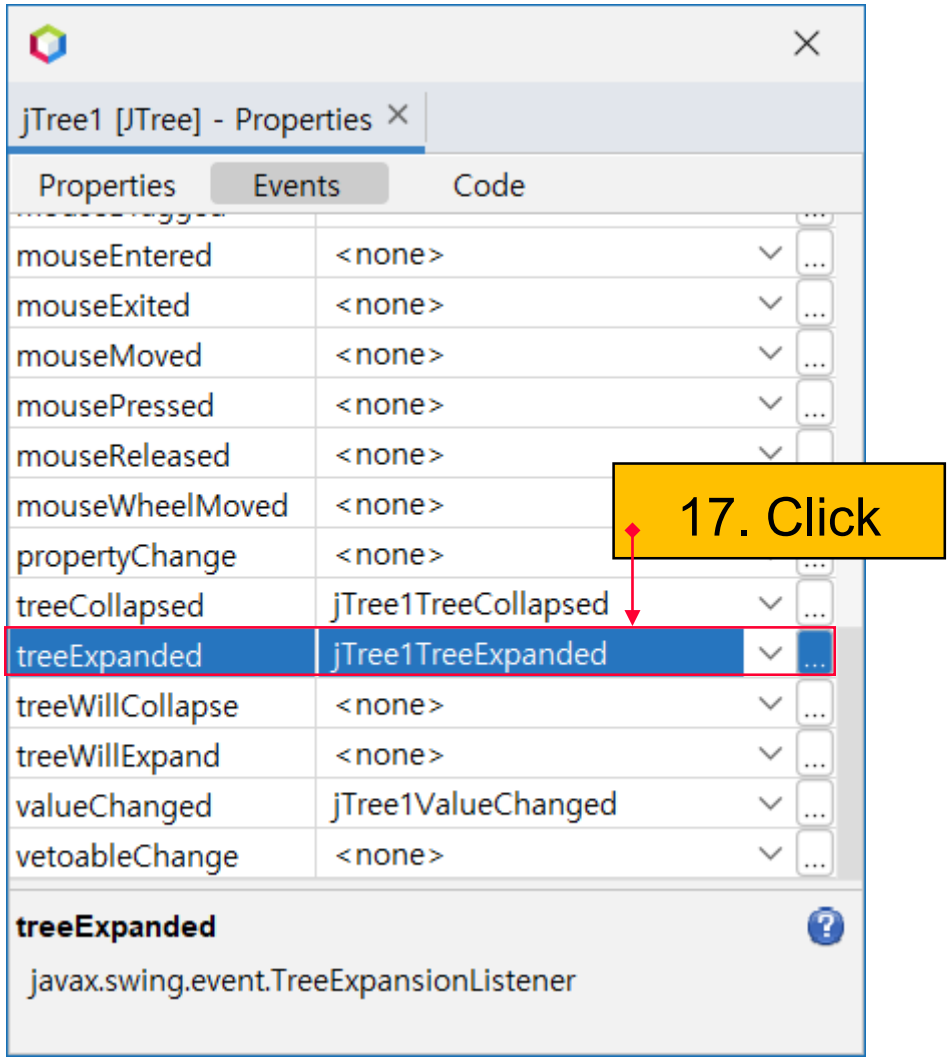
```
79  private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {
80      DefaultMutableTreeNode node;
81      node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();
82       if (node == null) return;
83
84       lblPath.setText("Path : " + node.getUserObject() + " 선택 !!! ₩n");
85   }
86
87   private void jTree1TreeCollapsed(javax.swing.event.TreeExpansionEvent evt) {
88       lblPath.setText("Path : " + evt.getPath() + "₩n");
89   }
90
91   private void jTree1TreeExpanded(javax.swing.event.TreeExpansionEvent evt) {
92       lblPath.setText("Path : " + evt.getPath() + "₩n");
93   }
94
97   public static void main(String args[]) {...7 lines }
```

16. Coding

8:34 | INS | Unused Import

Practice 2 : Tree Control (10)

- Setting jTree1 Control TreeExpanded() Event Handler



Practice 2 : Tree Control (11)

❖ jTree1TreeExpanded() Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

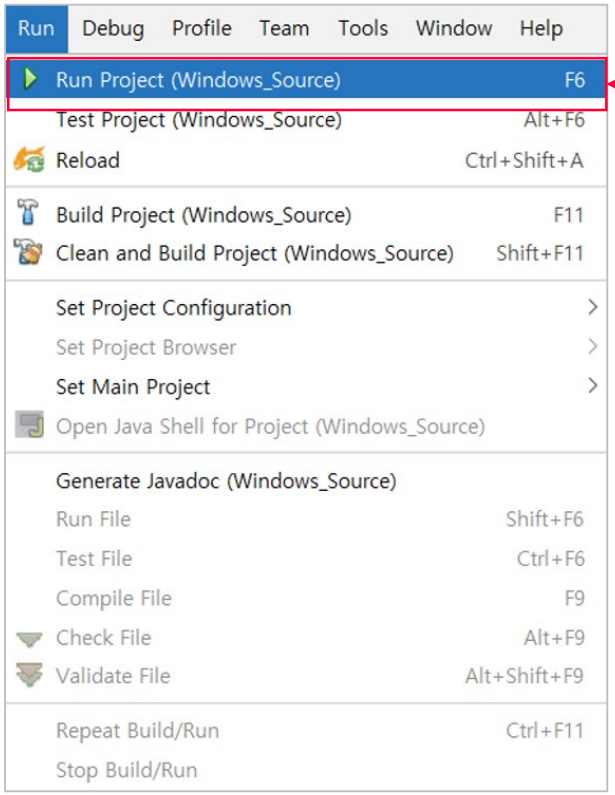
```
79  private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {
80      DefaultMutableTreeNode node;
81      node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();
82       if (node == null) return;
83
84       lblPath.setText("Path : " + node.getUserObject() + " 선택 !!! ₩n");
85  }
86
87  private void jTree1TreeCollapsed(javax.swing.event.TreeExpansionEvent evt) {
88      lblPath.setText("Path : " + evt.getPath() + "₩n");
89  }
90
91  private void jTree1TreeExpanded(javax.swing.event.TreeExpansionEvent evt) {
92      lblPath.setText("Path : " + evt.getPath() + "₩n");
93  }
94
97  /**...3 lines */
    public static void main(String args[]) {...7 lines }
```

18. Coding

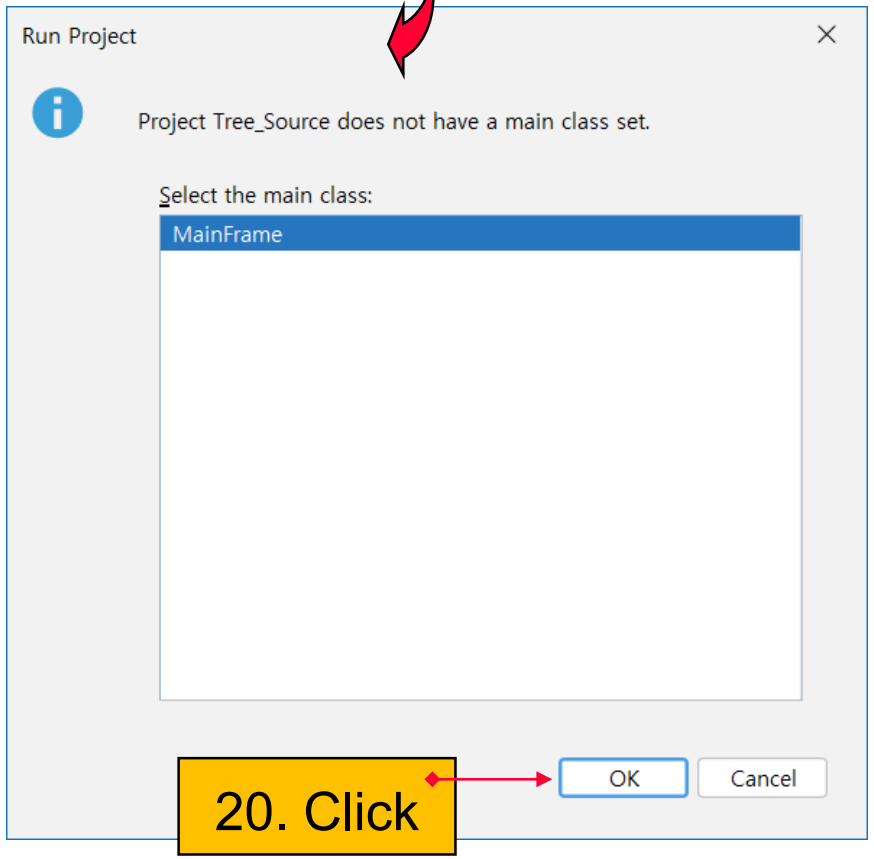
8:34 | INS | Unused Import

Practice 2 : Tree Control (12)

Run



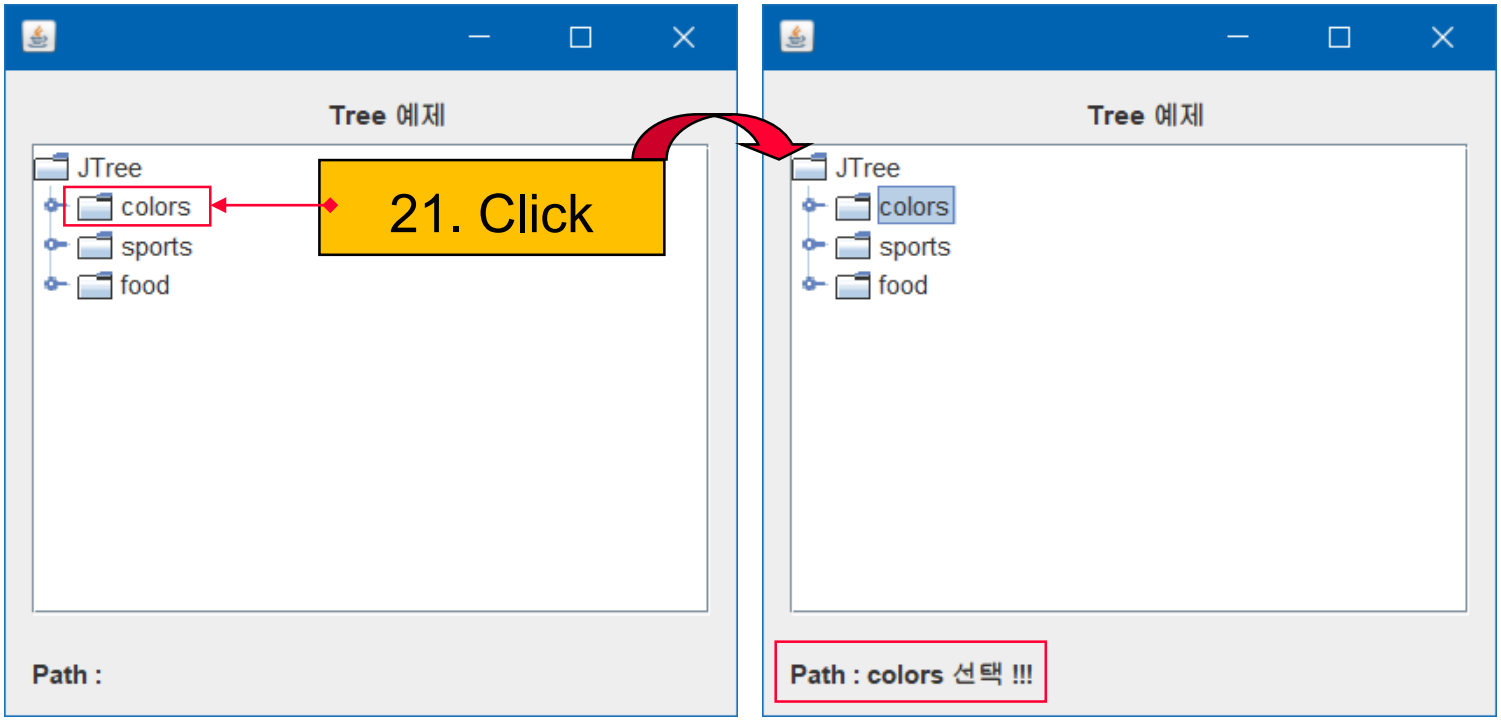
19. Click



20. Click

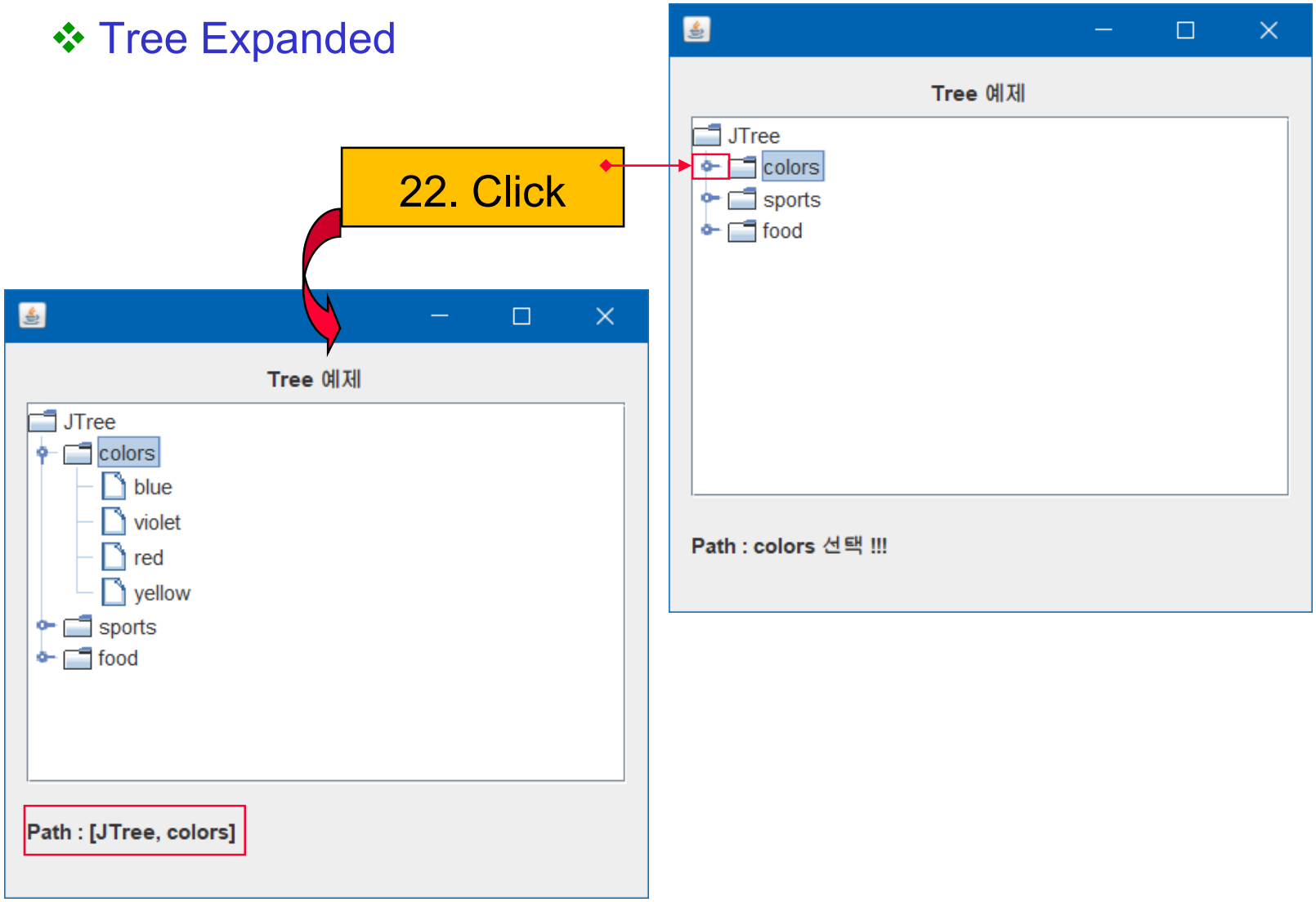
Practice 2 : Tree Control (13)

❖ colors 폴더 선택



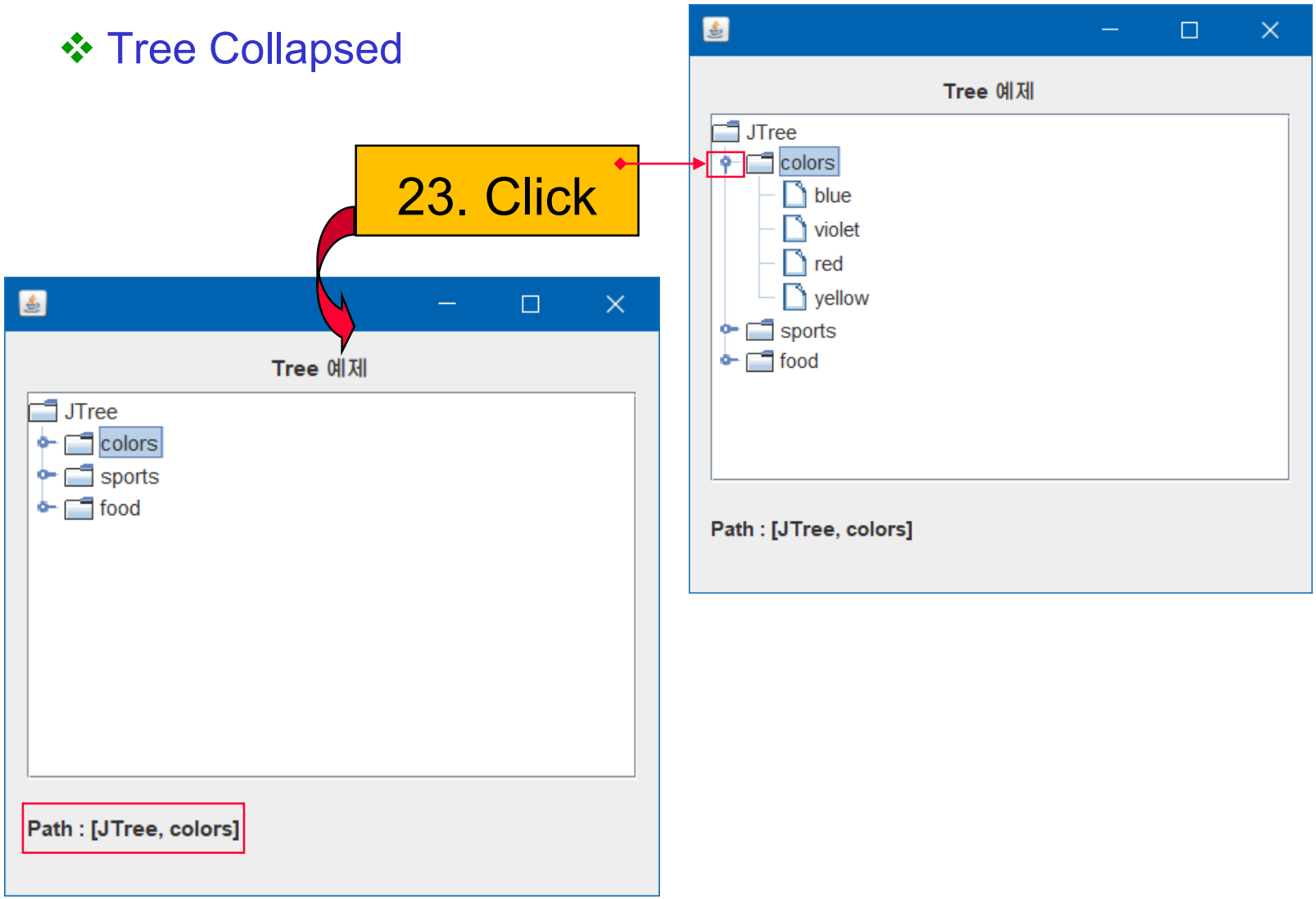
Practice 2 : Tree Control (14)

❖ Tree Expanded



Practice 2 : Tree Control (15)

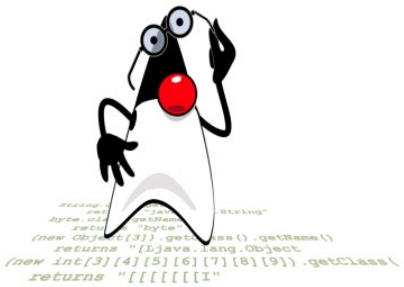
❖ Tree Collapsed

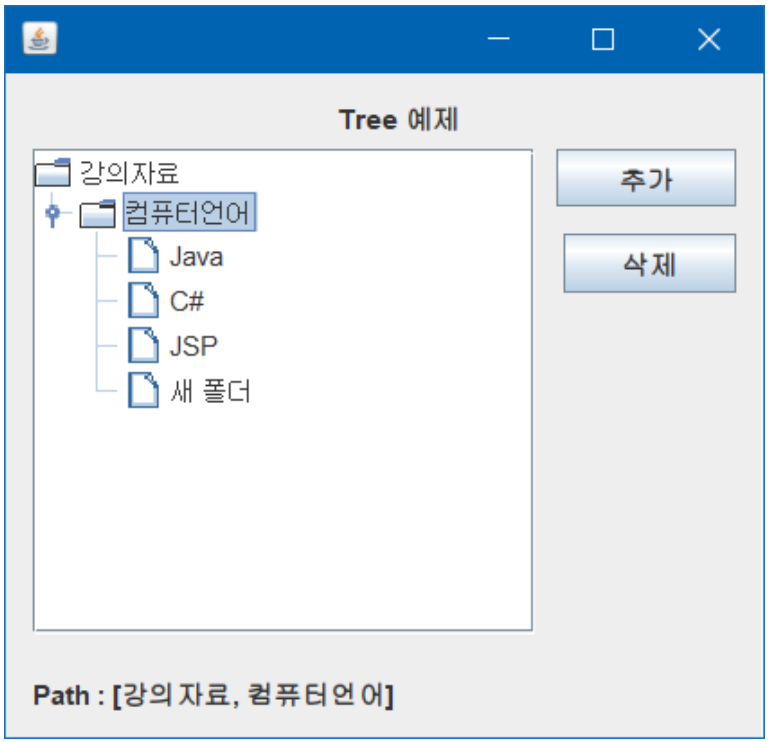


Practice 3 : Tree Control

❖ Project Name: TreeNode_Source(Time: 60 min)

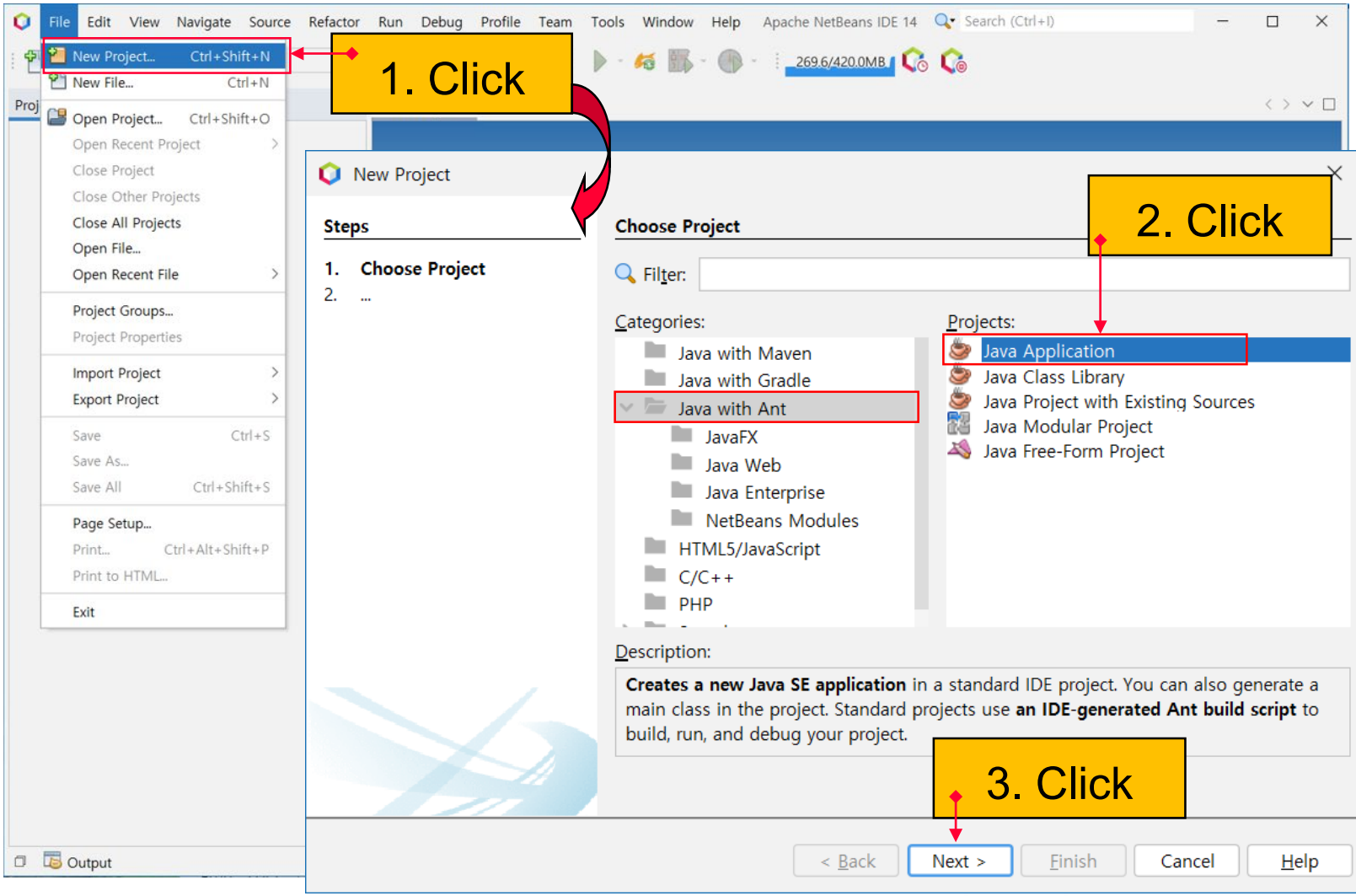
- [추가] Button Click
  - 선택한 폴더의 Child Node로 "새 폴더" 추가
- [삭제] Button Click
  - 선택한 폴더 삭제





Practice 3 : Tree Control (1)

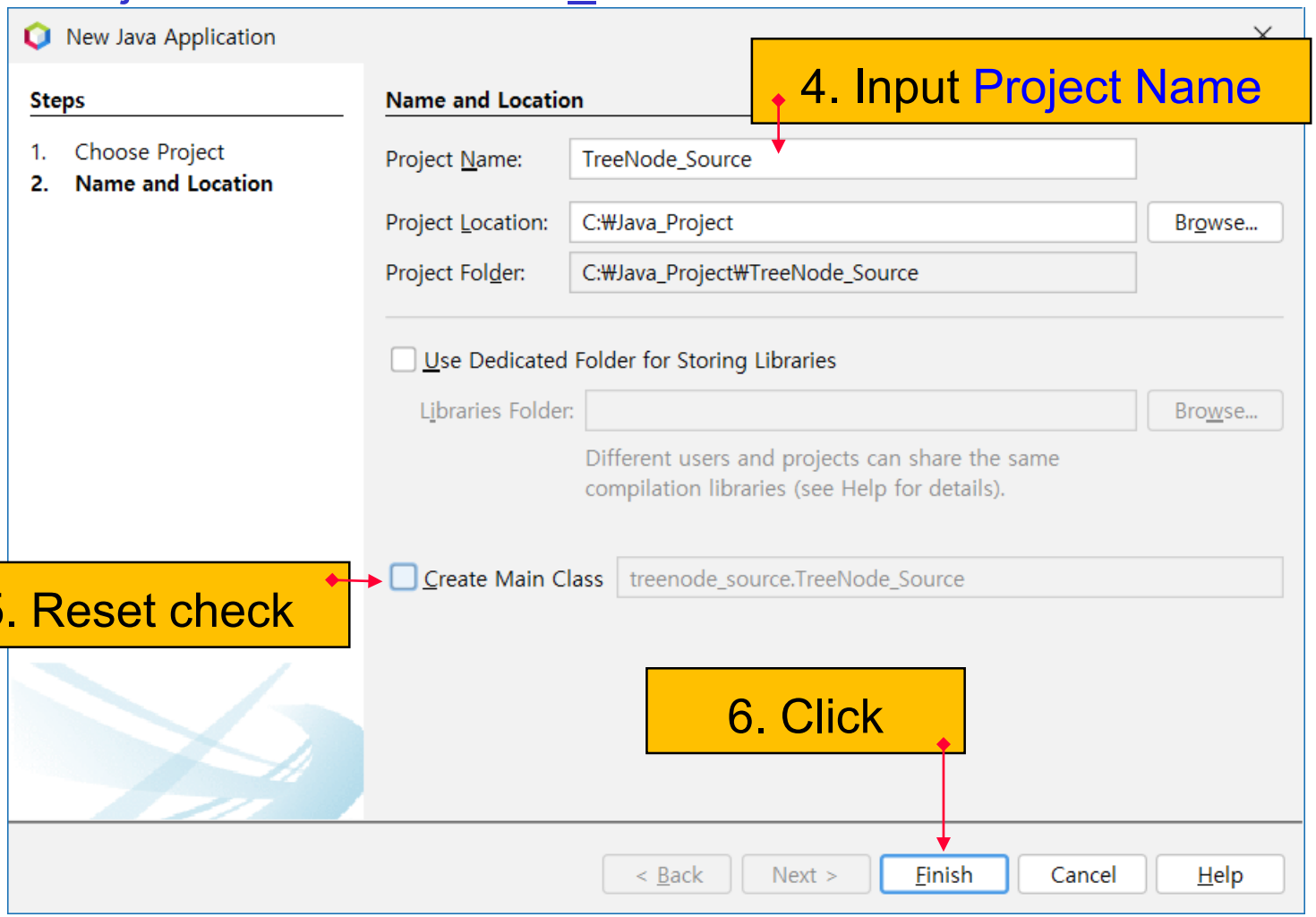
Create Project



Practice 3 : Tree Control (2)

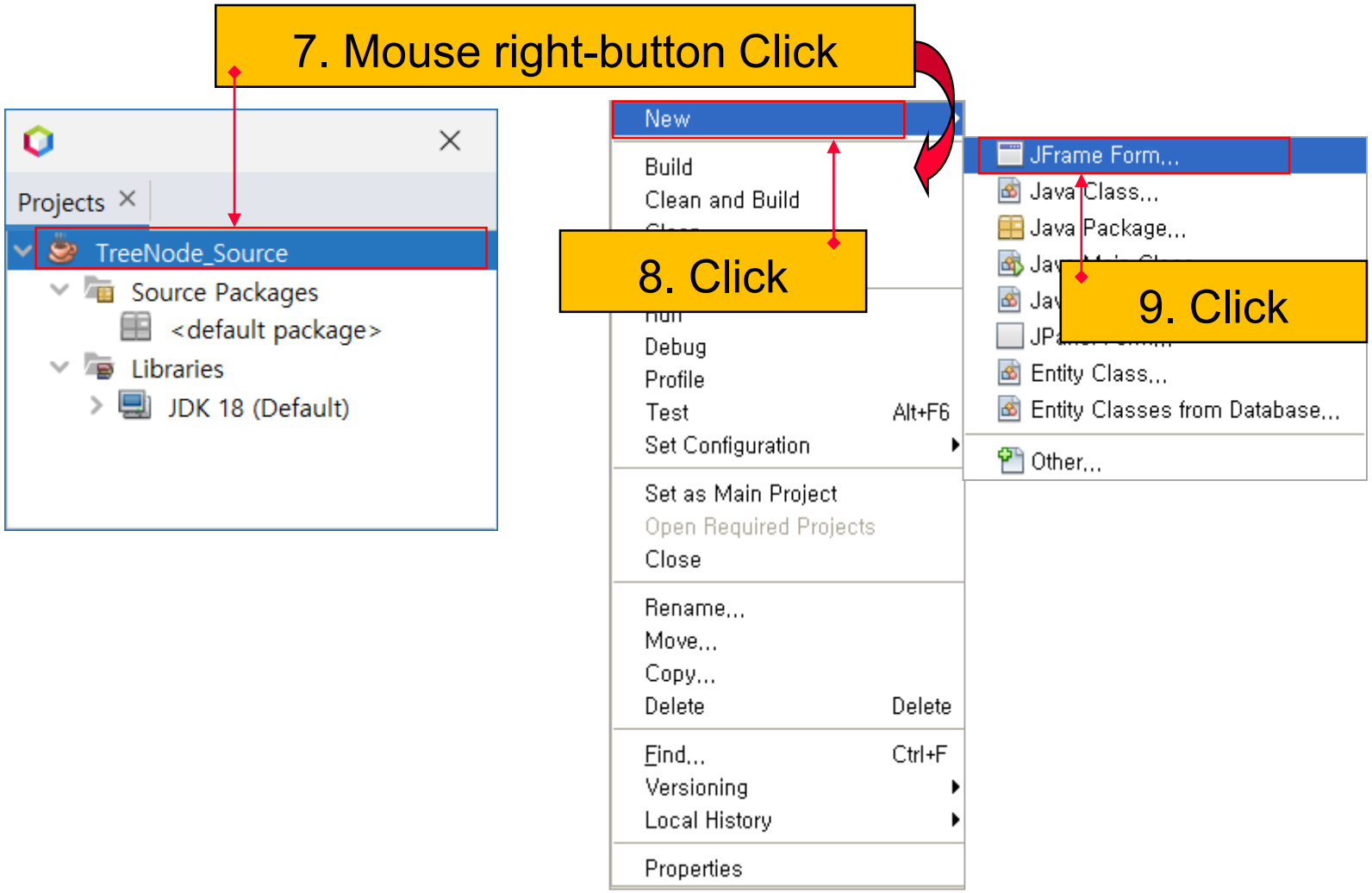
Project Name and Location

- Project name: TreeNode_Source



Practice 3 : Tree Control (3)

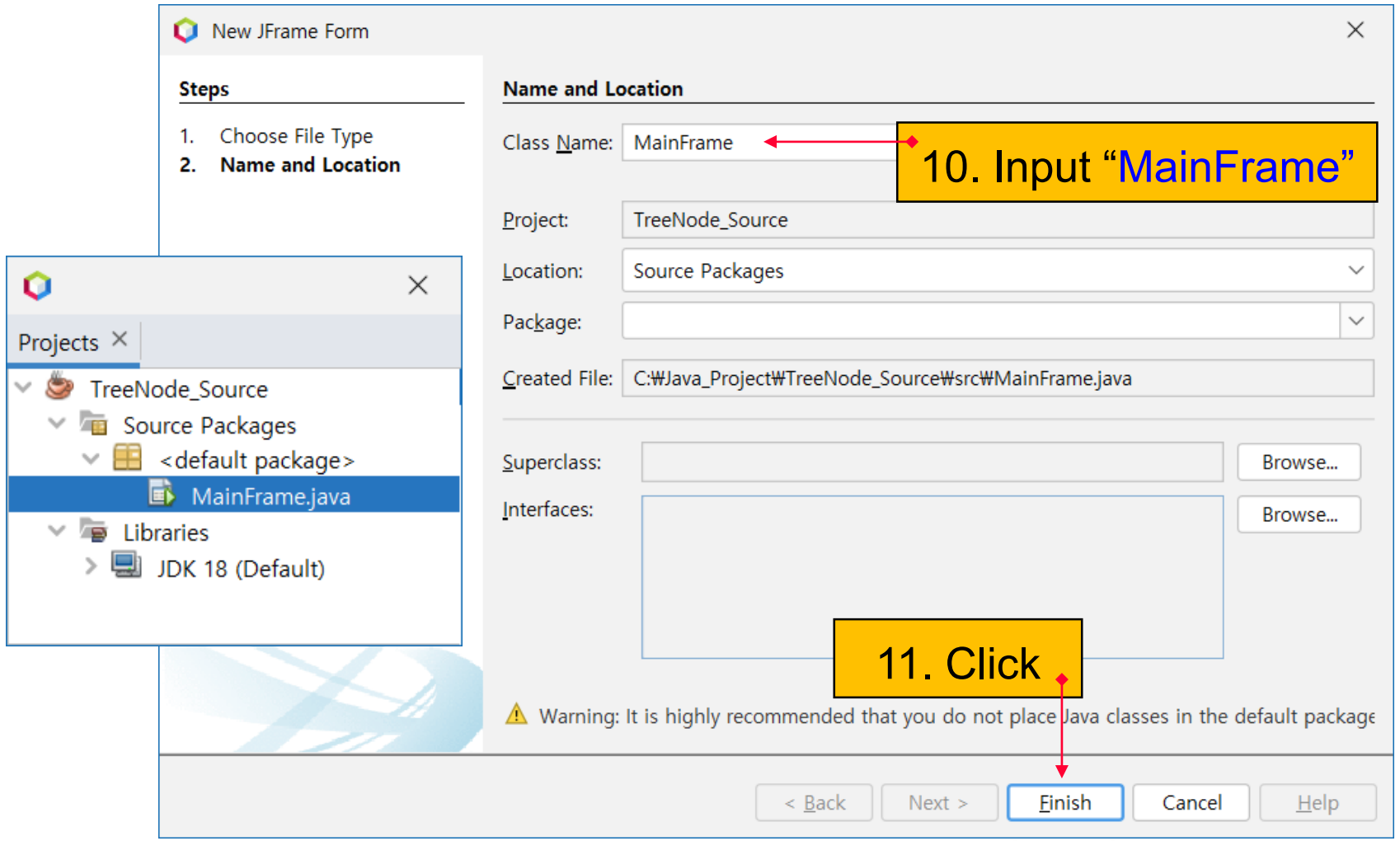
Create JFrame Form



Practice 3 : Tree Control (4)

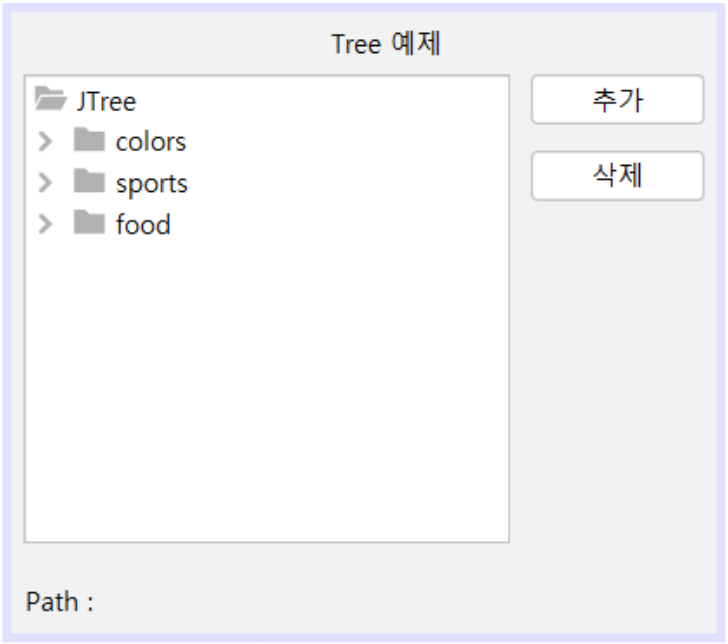
Setting JFrame Form Name

❖ Create MainFrame.java



Practice 3 : Tree Control (5)

Control Layout & Property Setting



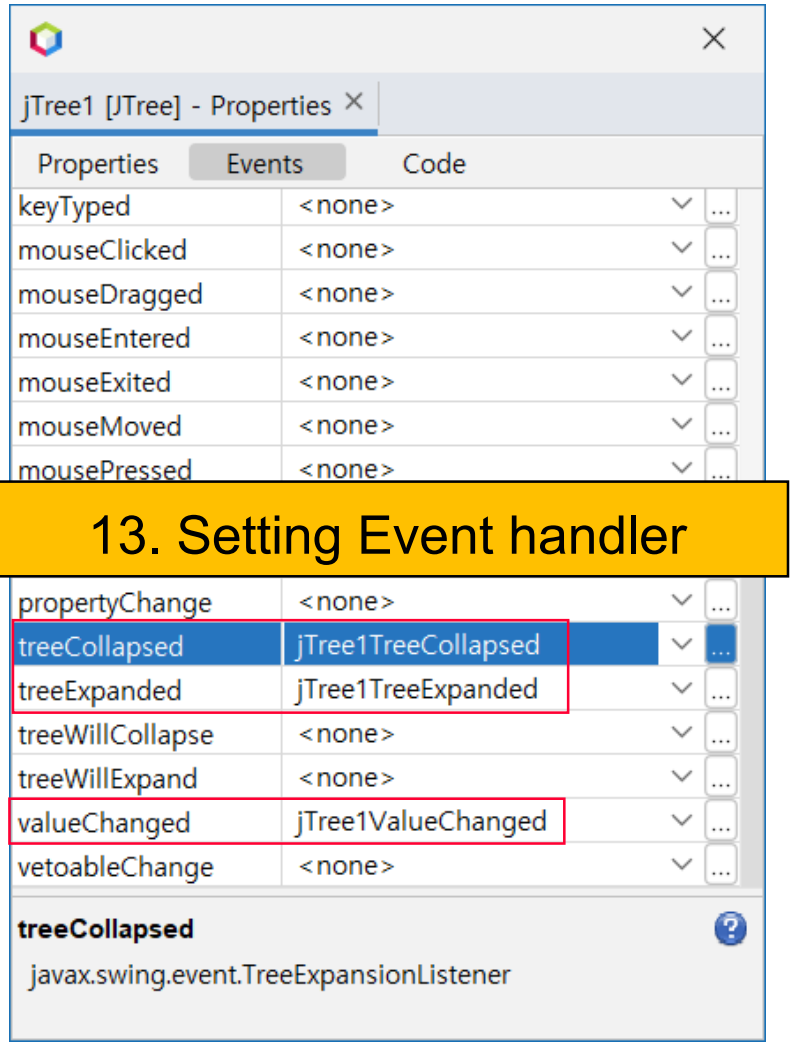
12. UI Design

Control	Properties Setting
jLabel1	• Variable Name : lblTitle • Text : Tree 예제
jLabel2	• Variable Name : lblPath • Text : Path :
jTree1	• Variable Name : jTree1
jButton1	• Variable Name : btnInsert • Text : Path : 추가
jButton2	• Variable Name : btnDelete • Text : Path : 삭제

Event	Properties Setting
treeExpanded	• tree 확장시 발생
treeCollapsed	• tree 축소시 발생
valueChanged	• Tree 값 변할때 발생

Practice 3 : Tree Control (6)

- Setting jTree1 Control Event Handler
  - jTree1TreeCollapsed()
  - jTree1TreeExpanded()
  - jTree1ValueChanged()



Practice 3 : Tree Control (7)

❖ Declaration Member Variable

MainFrame.java - Editor

MainFrame.java

Source Design History

```
1
2   import javax.swing.tree.*;
3   import java.util.*;
4
5   public class MainFrame extends javax.swing.JFrame {
6
7       //Setting Member Variable
8       DefaultTreeModel dftTreeModel = null;
9       DefaultTreeSelectionModel dftTreeSelectionModel = new DefaultTreeSelectionModel();
10      DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("강의자료");
11      DefaultMutableTreeNode  comNode= new DefaultMutableTreeNode("컴퓨터언어");
12
13      /** Creates new form MainFrame */
14      public MainFrame() {
15          initComponents();
16
```

19:4 INS A breakpoint cannot be set at this location.

14. Coding

Practice 3 : Tree Control (8)

- Constructor

MainFrame.java - Editor

MainFrame.java

Source | Design | History

```
13  /** Creates new form MainFrame */
14  public MainFrame() {
15      initComponents();
16
17      //Selection Mode = SINGLE_SELECTION_MODE Setting
18      dftTreeSelectionModel.setSelectionMode(1);
19      //Create 컴퓨터언어 node의 childNode
        Vector<DefaultMutableTreeNode> myVC = new Vector<>();
21      String [] objFolder = {"Java", "C#", "JSP"};
22      for(int idx=0; idx < objFolder.length; idx++)
23      {
24          //Create node object to add Vector element
25          myVC.add(new  DefaultMutableTreeNode(objFolder[idx]));
26          comNode.add(myVC.get(idx));
27      }
28
29      //Add comNode as child node of rootNode
30      rootNode.add(comNode);
31      //Setting rootNode as DefaultTreeModel
32      dftTreeModel = new DefaultTreeModel(rootNode);
33      //Setting dftTreeModel as TreeModel of jTree1
34      jTree1.setModel(dftTreeModel);
35      jTree1.setEditable(true);
36      jScrollPane1.getViewport().add(jTree1, null);
37  }
```

9:47 INS

15. Coding

Practice 3 : Tree Control (9)

❖ jTree1의 Event Handler

16. Coding

MainFrame.java - Editor

MainFrame.java

```
37      }
130
        private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {
132         DefaultMutableTreeNode node;
133         node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();
134          if (node == null) return;
135
136          lblPath.setText("Path : " + node.getUserObject() + " 선택 !!! ₩n");
137     }
138
139     private void jTree1TreeCollapsed(javax.swing.event.TreeExpansionEvent evt) {
140       lblPath.setText("Path : " + evt.getPath() + "₩n");
141     }
142
143     private void jTree1TreeExpanded(javax.swing.event.TreeExpansionEvent evt) {
144       lblPath.setText("Path : " + evt.getPath() + "₩n");
145     }
146
```

14:24 INS

Practice 3 : Tree Control (10)

- Setting [추가] Button Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

```
private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {...7 lines }
138
private void jTree1TreeCollapsed(javax.swing.event.TreeExpansionEvent evt) {...3 lines }
142
private void jTree1TreeExpanded(javax.swing.event.TreeExpansionEvent evt) {...3 lines }
146
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    int index =0;        //Save the index of the "new folder" to be added
149
    DefaultMutableTreeNode select_Node;
150 select_Node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();
151
152 DefaultMutableTreeNode newFolder = new DefaultMutableTreeNode("새 폴더");
153 index = select_Node.getChildCount();
154 dftTreeModel.insertNodeInto(newFolder, select_Node, index);
155
156
157
158 jTree1.setModel(dftTreeModel);
159 jTree1.setSelectionModel(dftTreeSelectionModel);
160 jScrollPane1.getViewport().add(jTree1, null);
161 }
```

1:1 INS

17. Coding

Practice 3 : Tree Control (11)

- Setting [삭제] Button Event Handler

MainFrame.java - Editor

MainFrame.java

Source | Design | History

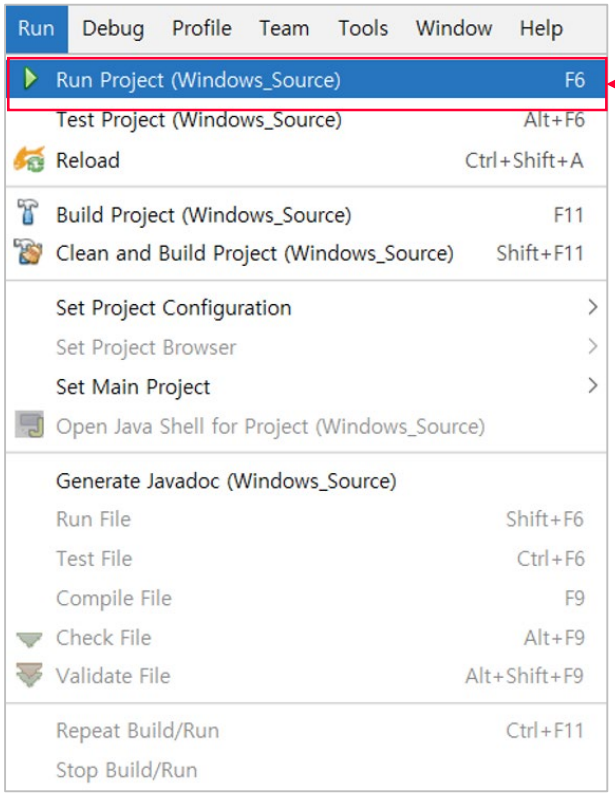
```
138
139  private void jTree1ValueChanged(javax.swing.event.TreeSelectionEvent evt) {...7 lines }
142
143  private void jTree1TreeCollapsed(javax.swing.event.TreeExpansionEvent evt) {...3 lines }
146
162  private void jTree1TreeExpanded(javax.swing.event.TreeExpansionEvent evt) {...3 lines }
164  private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {...15 lines }
172  private void btbDeleteActionPerformed(java.awt.event.ActionEvent evt) {
         DefaultMutableTreeNode select_Node;
         select_Node = (DefaultMutableTreeNode) jTree1.getLastSelectedPathComponent();

         dftTreeModel.removeNodeFromParent(select_Node); //Delete selected Node

         jTree1.setModel(dftTreeModel);
         jTree1.setSelectionModel(dftTreeSelectionModel);
         jScrollPane1.getViewport().add(jTree1, null);
     }
```

1:1 INS

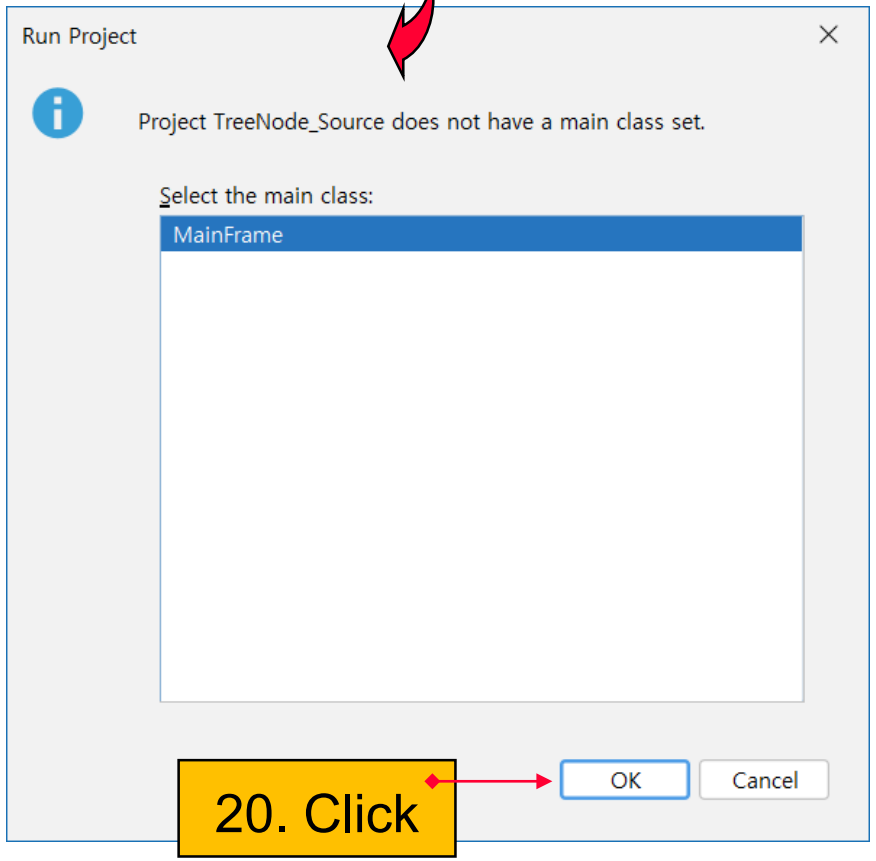
18. Coding

Practice 3 : Tree Control (12)

Run



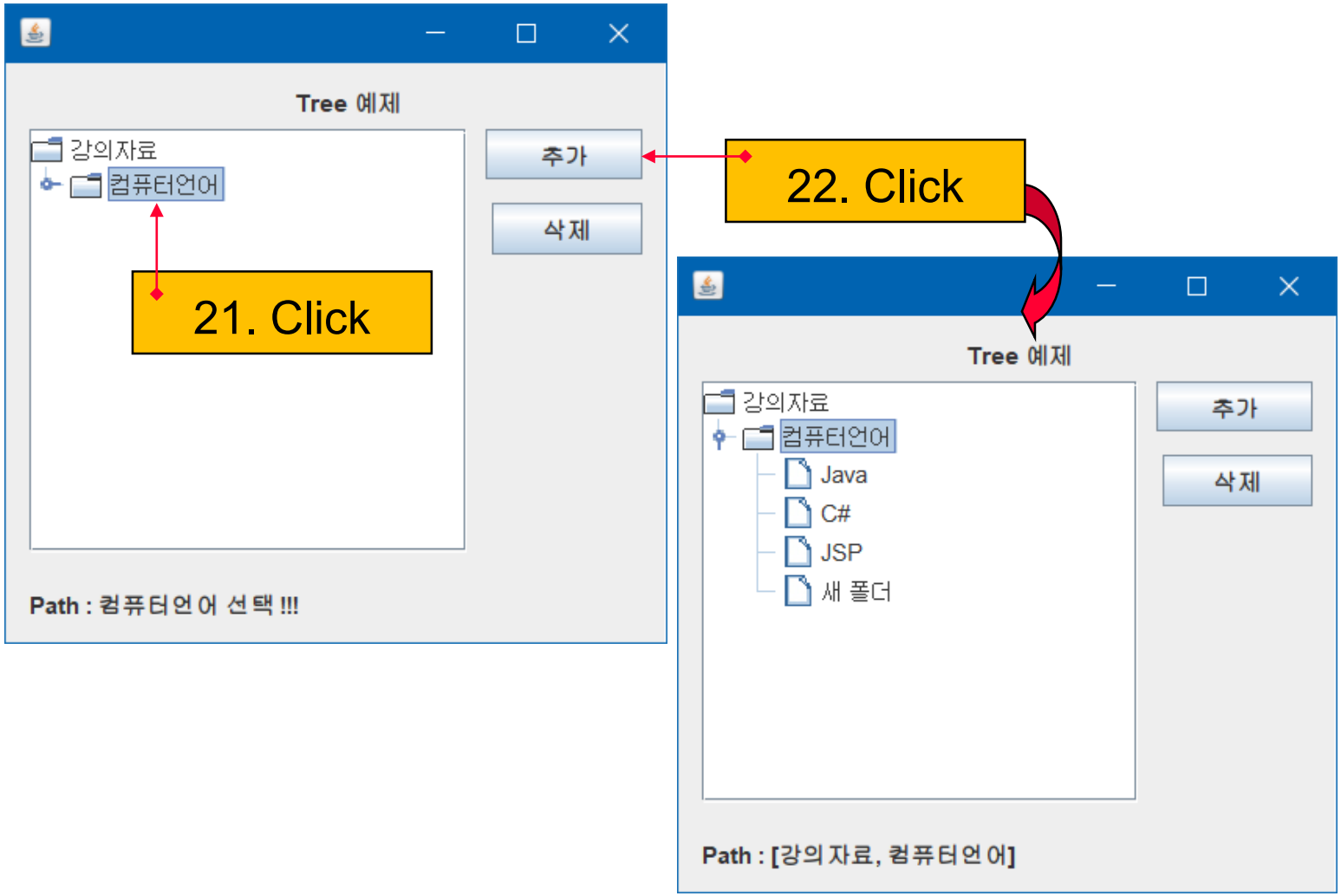
19. Click



20. Click

Practice 3 : Tree Control (13)

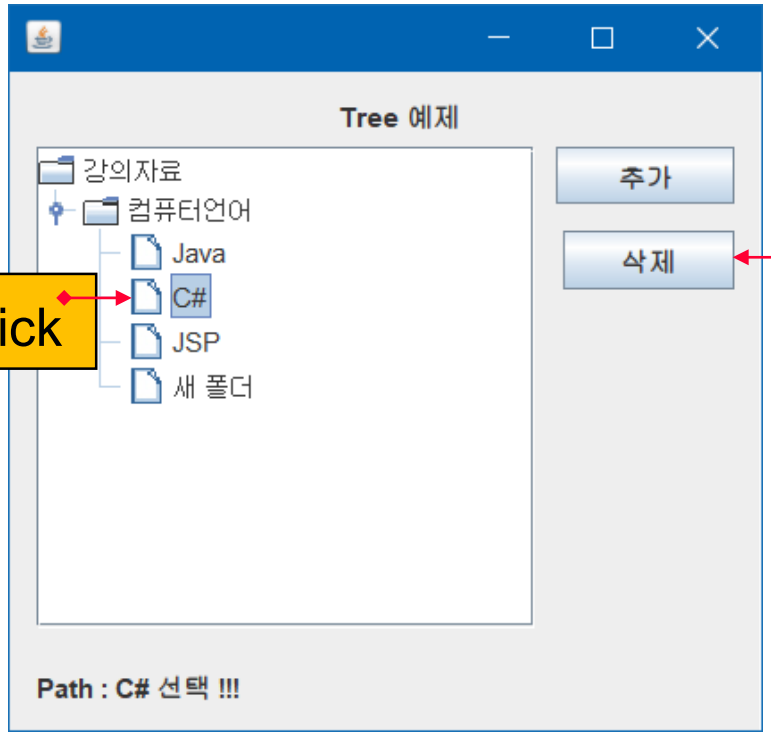
- [추가] Button
  - 새 폴더 추가



Practice 3 : Tree Control (14)

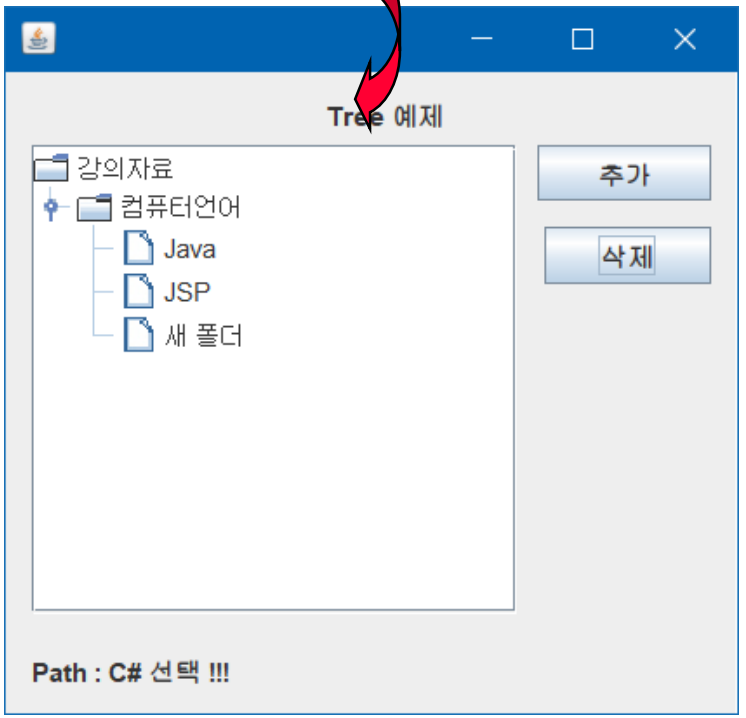
❖ [삭제] Button

◆ 선택 폴더 삭제



23. Click

24. Click



학습 요약

- Table Control
- Tree Control