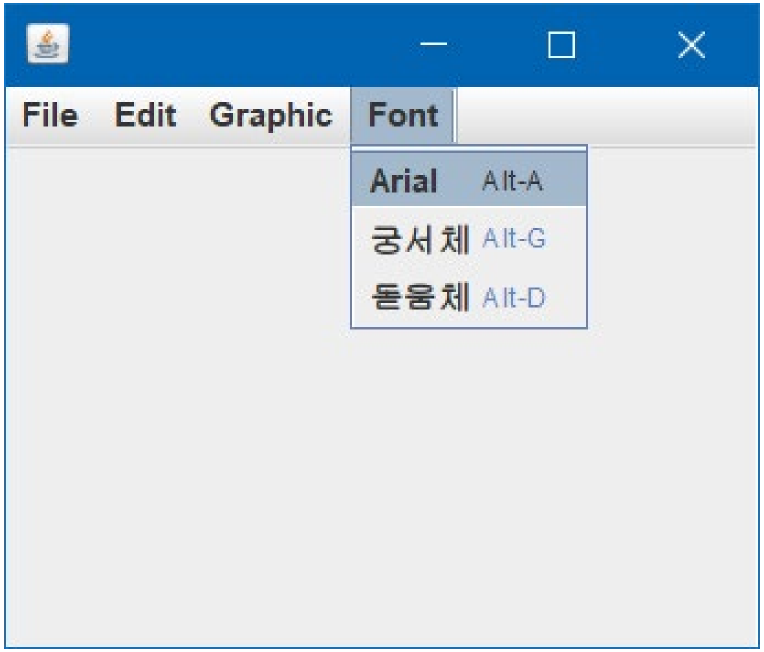
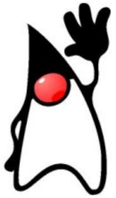
Menu



학습 목표

- 이 강의를 마치면 학생들은
  - ❖ Menu에 대하여 설명할 수 있다.
  - ❖ Menu Event Handler에 대하여 설명할 수 있다.
  - ❖ Graphic Function에 대하여 설명할 수 있다.
  - ❖ PopUp Menu에 대하여 설명할 수 있다.
  - ❖ Font 에 대하여 설명할 수 있다.



Menu (1)

Menu

- Function
  - 폼에 표시할 메뉴를 나타내는 Function

- 구성

Component	Class
메뉴바	Java.awt.MenuBar
메뉴	Java.awt.Menu
메뉴 아이템	Java.awt.MenuItem

- MenuBar 클래스 객체 생성

```
MenuBar mnuBar = new MenuBar();
```

Menu (2)

❖ Menu 클래스 객체 생성

```
Menu m = new Menu(“메뉴 명”);
```

❖ Menu ← MenuItem 추가

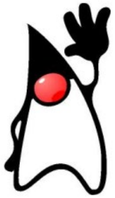
```
MenuItem  item = new MenuItem(“Menu Item 명”  );

m.add(item);
```

❖ ManuBar ← Menu 추가

```
mnuBar.addMenu(m);
```

❖ Frame ← ManuBar 추가

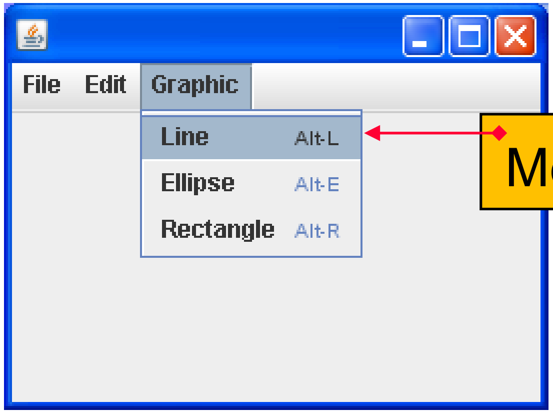
```
Frame fmMain = new Frame();

fmMain.add(mnuBar);
```



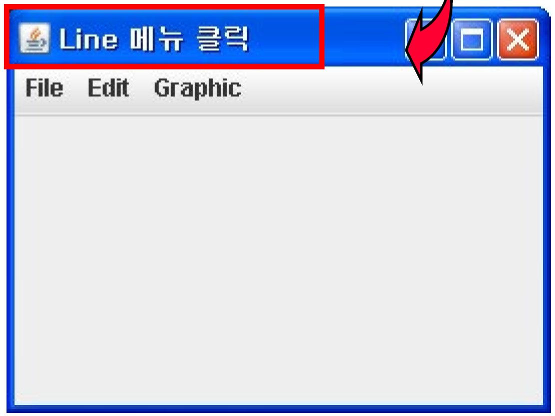
Practice 1 : Menu

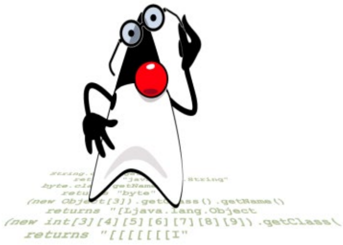
❖ Project Name: Menu_Source(Time: 30 min)

- [Graphic] 메뉴의 하위 메뉴 Click / 단축 키 입력
  - Line : 제목표시줄에 “Line 메뉴 Click” 메시지 출력
  - Ellipse : 제목표시줄에 “Ellipse 메뉴 Click” 메시지 출력
  - Rectangle : 제목표시줄에 “Rectangle 메뉴 Click” 메시지 출력



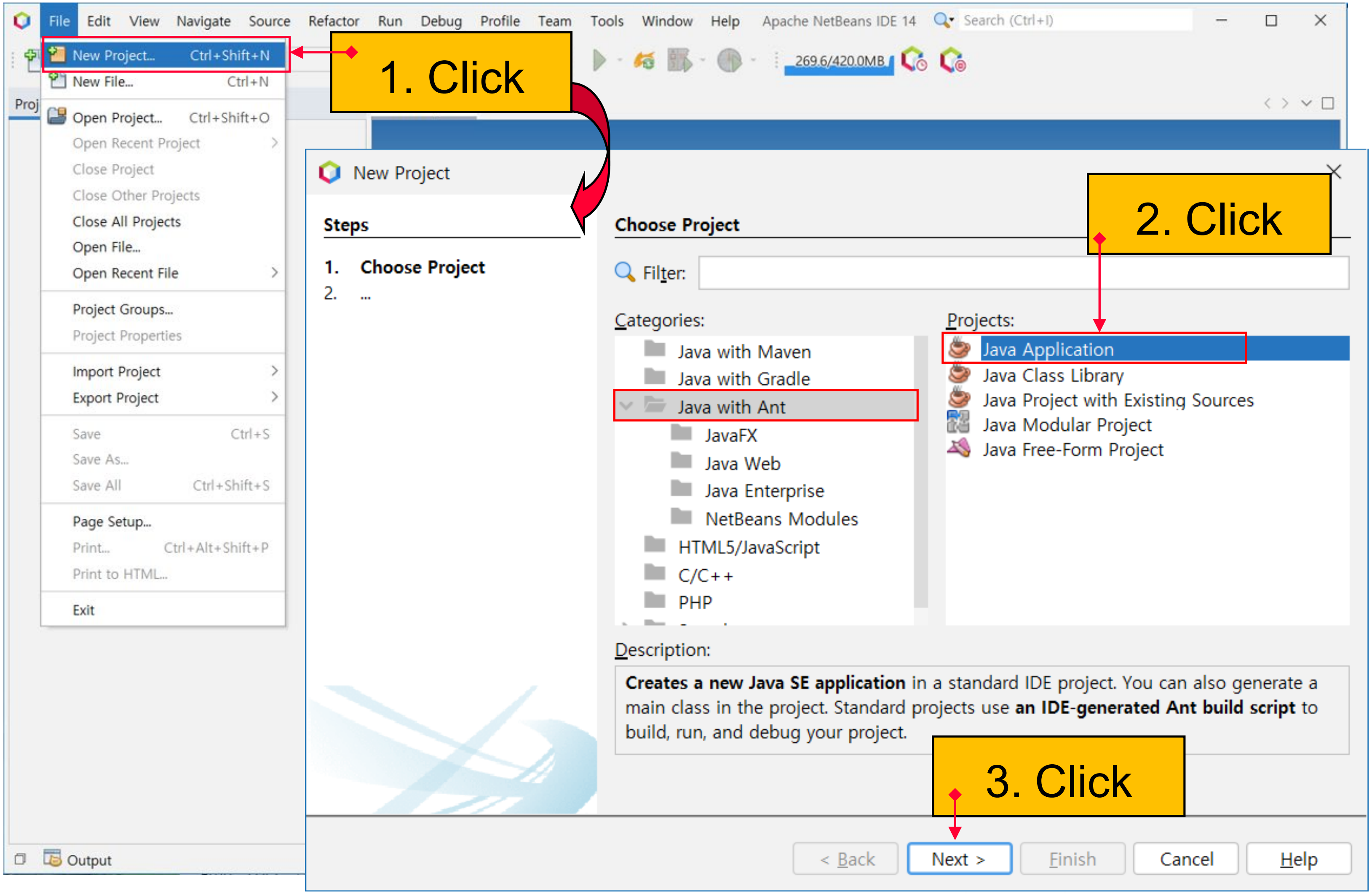
Menu Click or Shortcut input





Practice 1 : Menu (1)

Create Project

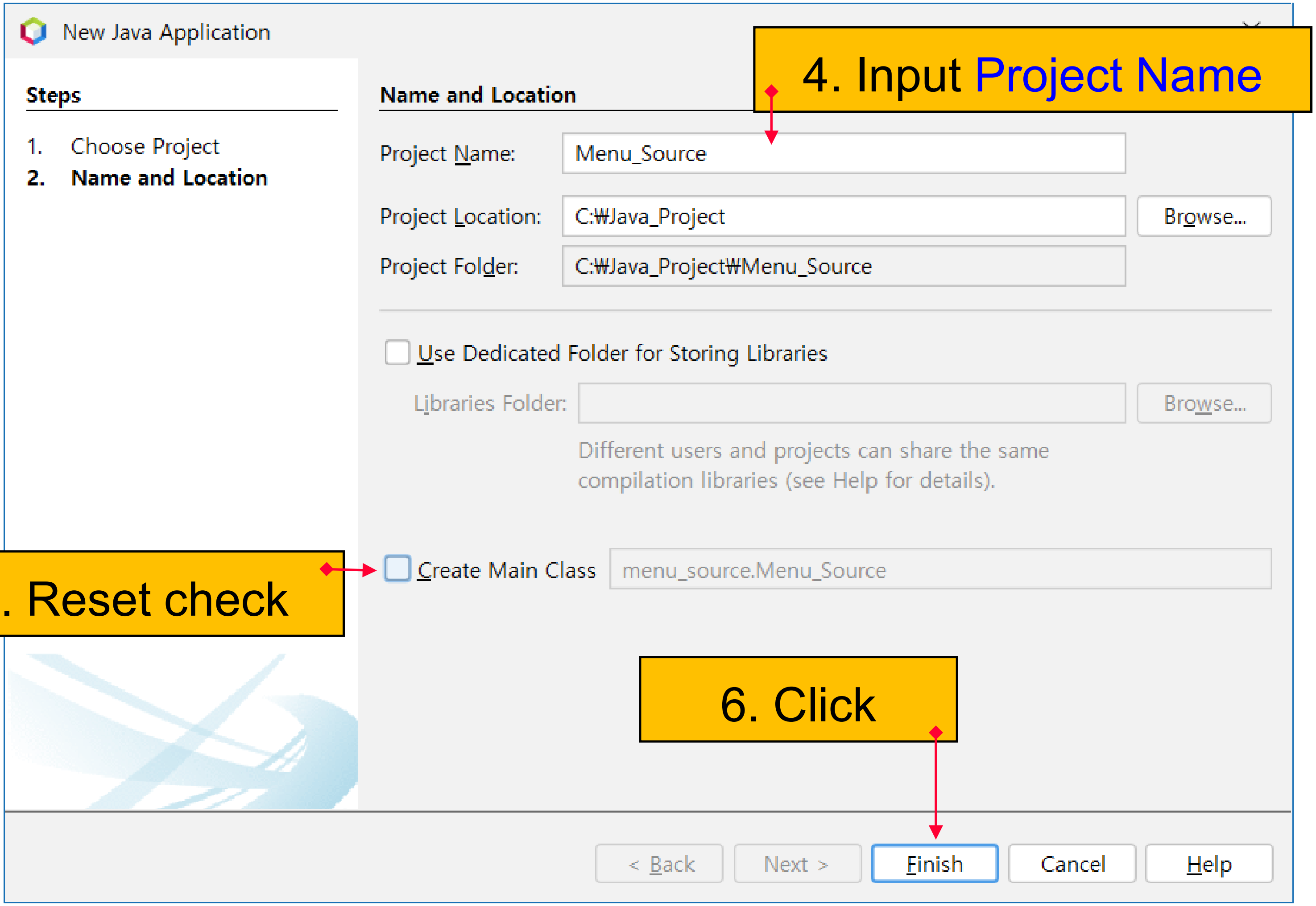


Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Practice 1 : Menu (2)

Project Name and Location

- Project name: Menu_Source



New Java Application

Steps

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

Name and Location

Project Name: Menu_Source

Project Location: C:₩Java_Project

Project Folder: C:₩Java_Project₩Menu_Source

□ 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 menu_source.Menu_Source

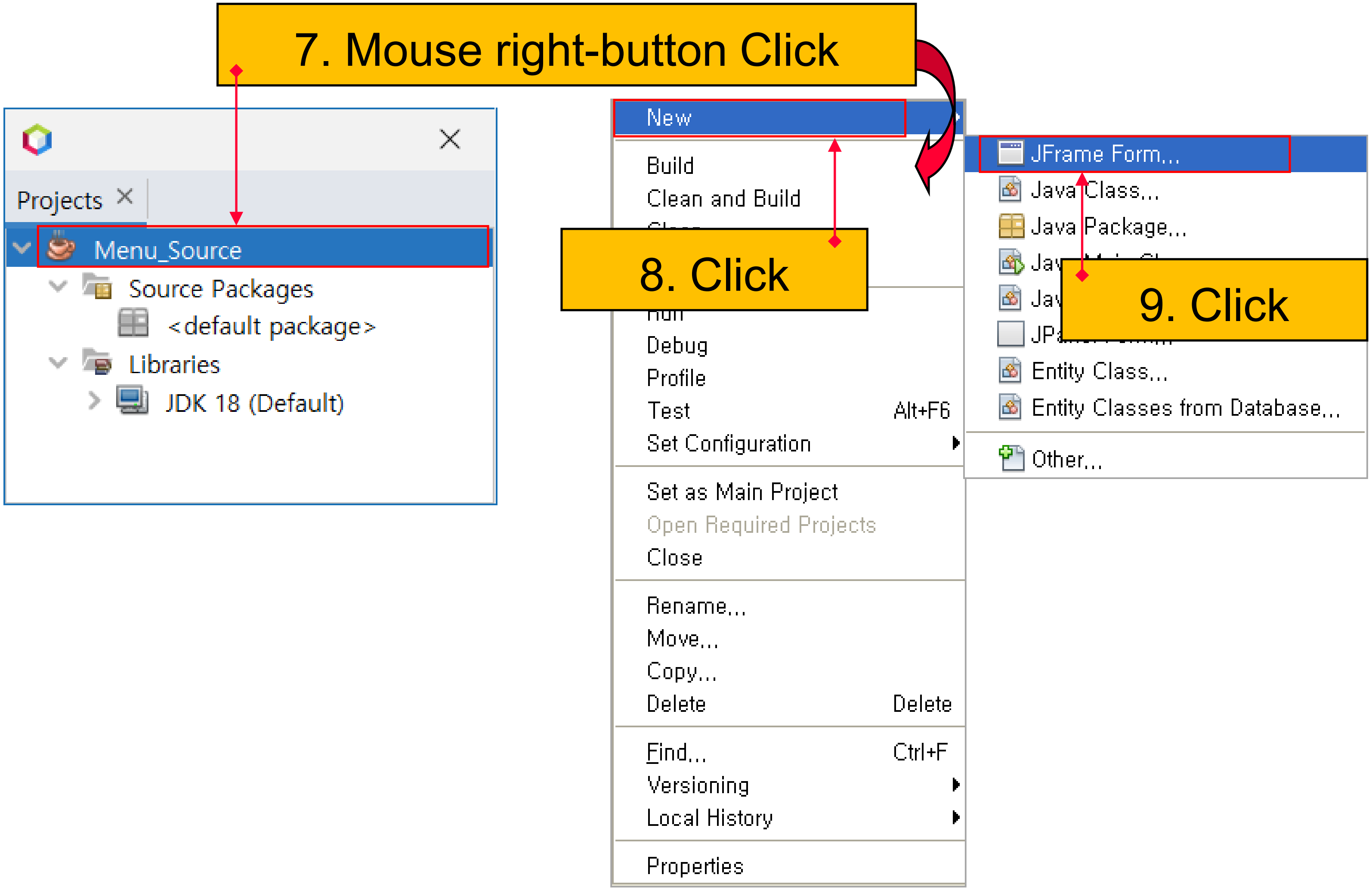
4. Input Project Name

5. Reset check

6. Click

Practice 1 : Menu (3)

Create JFrame Form



7. Mouse right-button Click

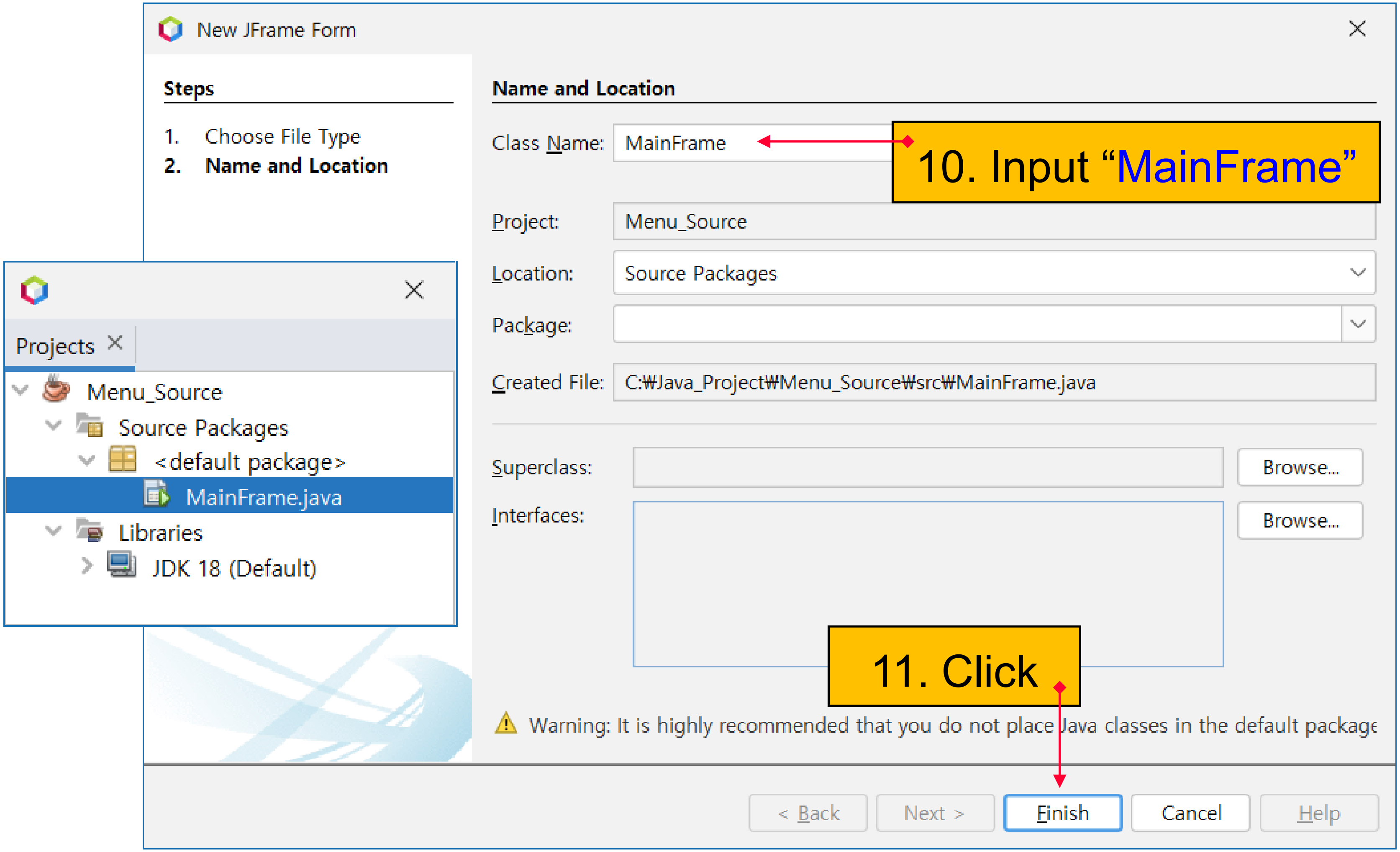
8. Click

9. Click

Practice 1 : Menu (4)

Setting JFrame Form Name

Create MainFrame.java



New JFrame Form

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: MainFrame

Project: Menu_Source

Location: Source Packages

Package:

Created File: C:₩Java_Project₩Menu_Source₩src₩MainFrame.java

Superclass:

Interfaces:

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

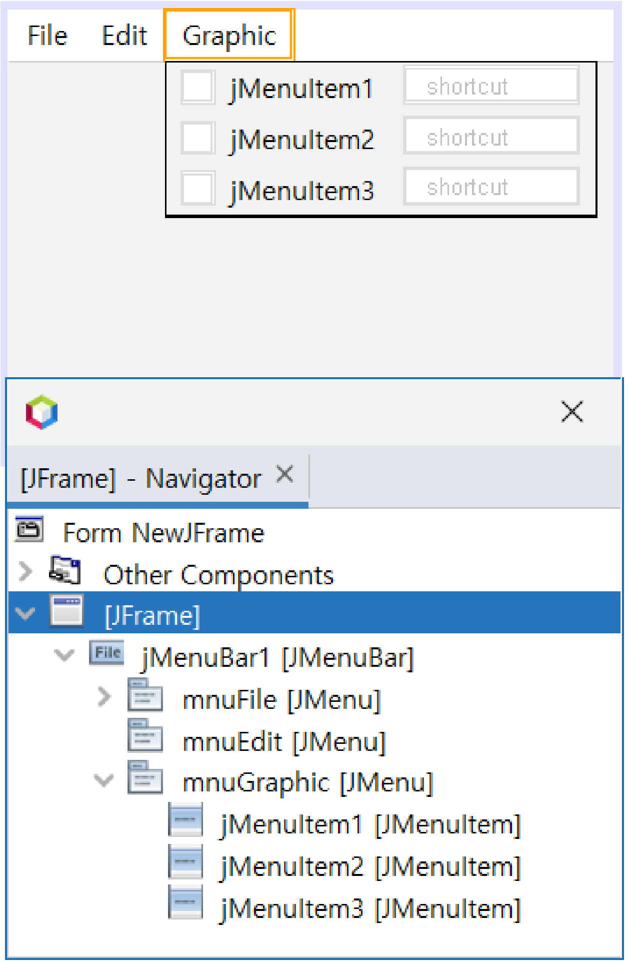
10. Input "MainFrame"

11. Click

Practice 1 : Menu (5)

Control Layout & Property Setting

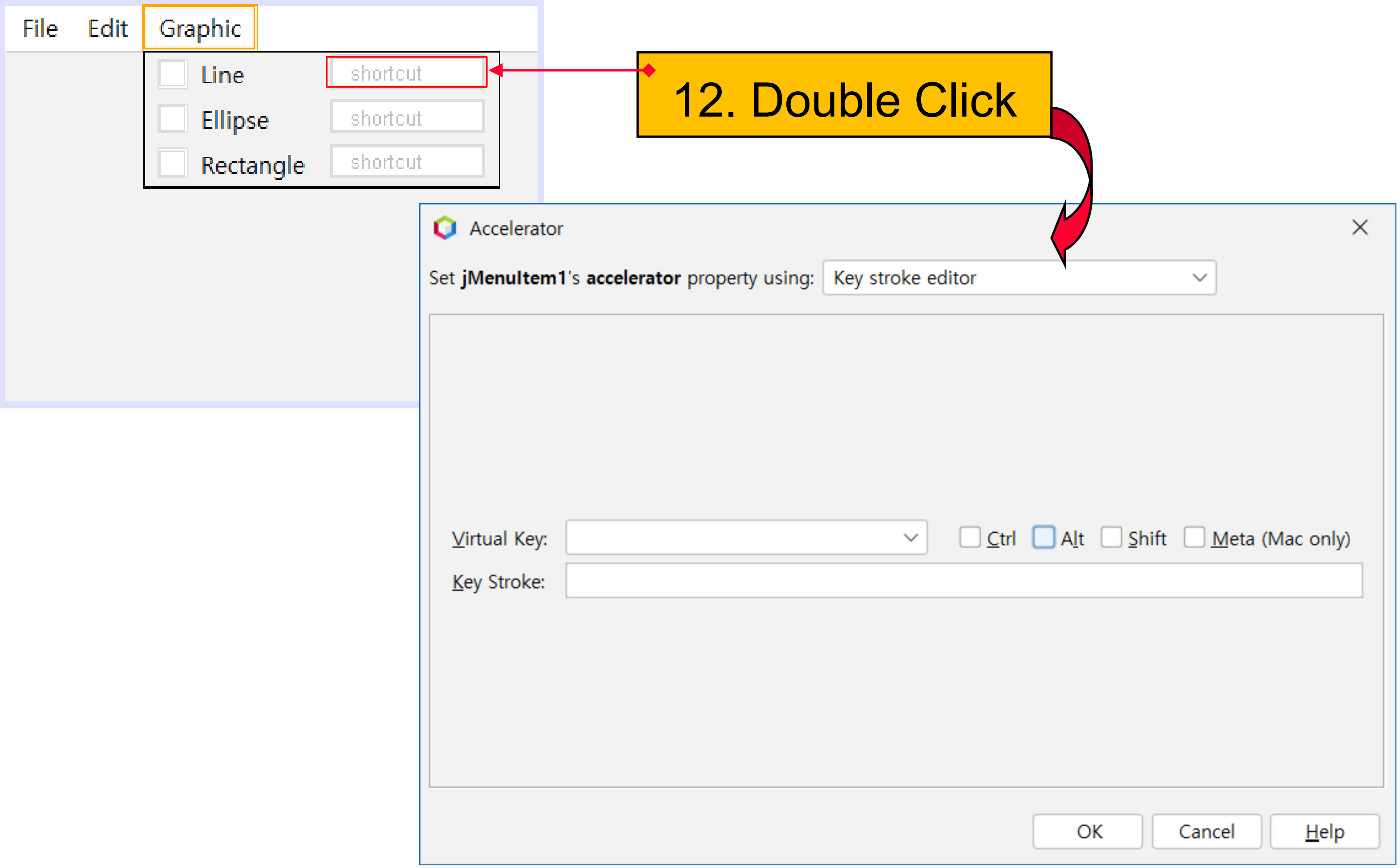
MainFrame



Control	Properties Setting
jMenuBar1	• Variable Name : jMenuBar1
jMenu1	• Variable Name : mnuFile • Text : File
jMenu2	• Variable Name : mnuEdit • Text : Edit
jMenu3	• Variable Name : mnuGraphic • Text : Graphic
jMenuItem1	• Variable Name : mnuGraphic_Line • Text : Line
jMenuItem2	• Variable Name : mnuGraphic_Ellipse • Text : Ellipse
jMenuItem3	• Variable Name : mnuGraphic_Rectangle • Text : Rectangle

Practice 1 : Menu (6)

- 단축키(Shortcut) 지정
  - Line : Alt + L



12. Double Click

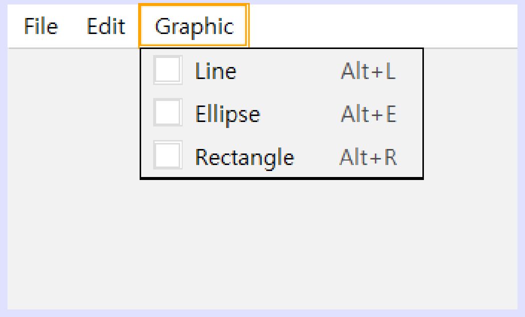
Set **jMenuItem1**'s **accelerator** property using: Key stroke editor

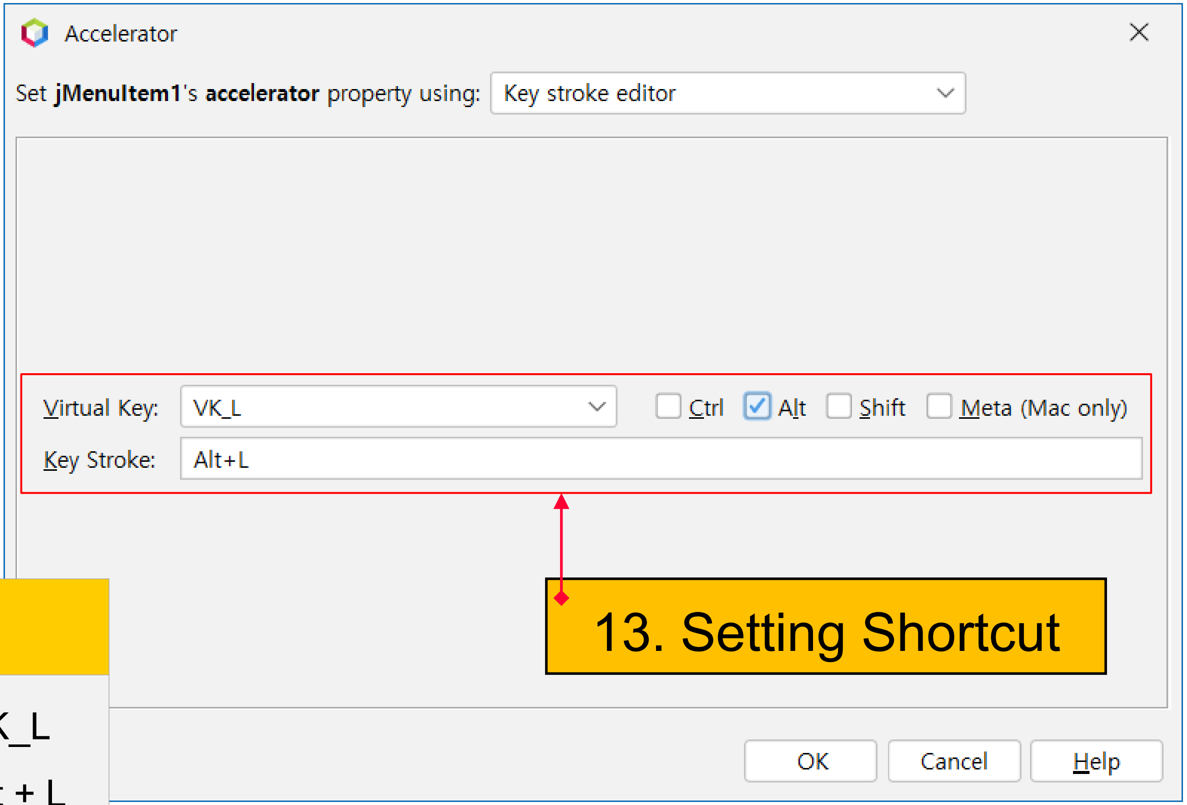
Virtual Key:

□ Ctrl □ Alt □ Shift □ Meta (Mac only)

Key Stroke:

Practice 1 : Menu (7)




Accelerator

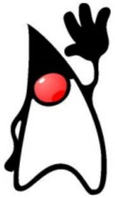
Set **jMenuItem1**'s **accelerator** property using: Key stroke editor

Virtual Key: VK_L □ Ctrl ☑ Alt □ Shift □ Meta (Mac only)

Key Stroke: Alt+L

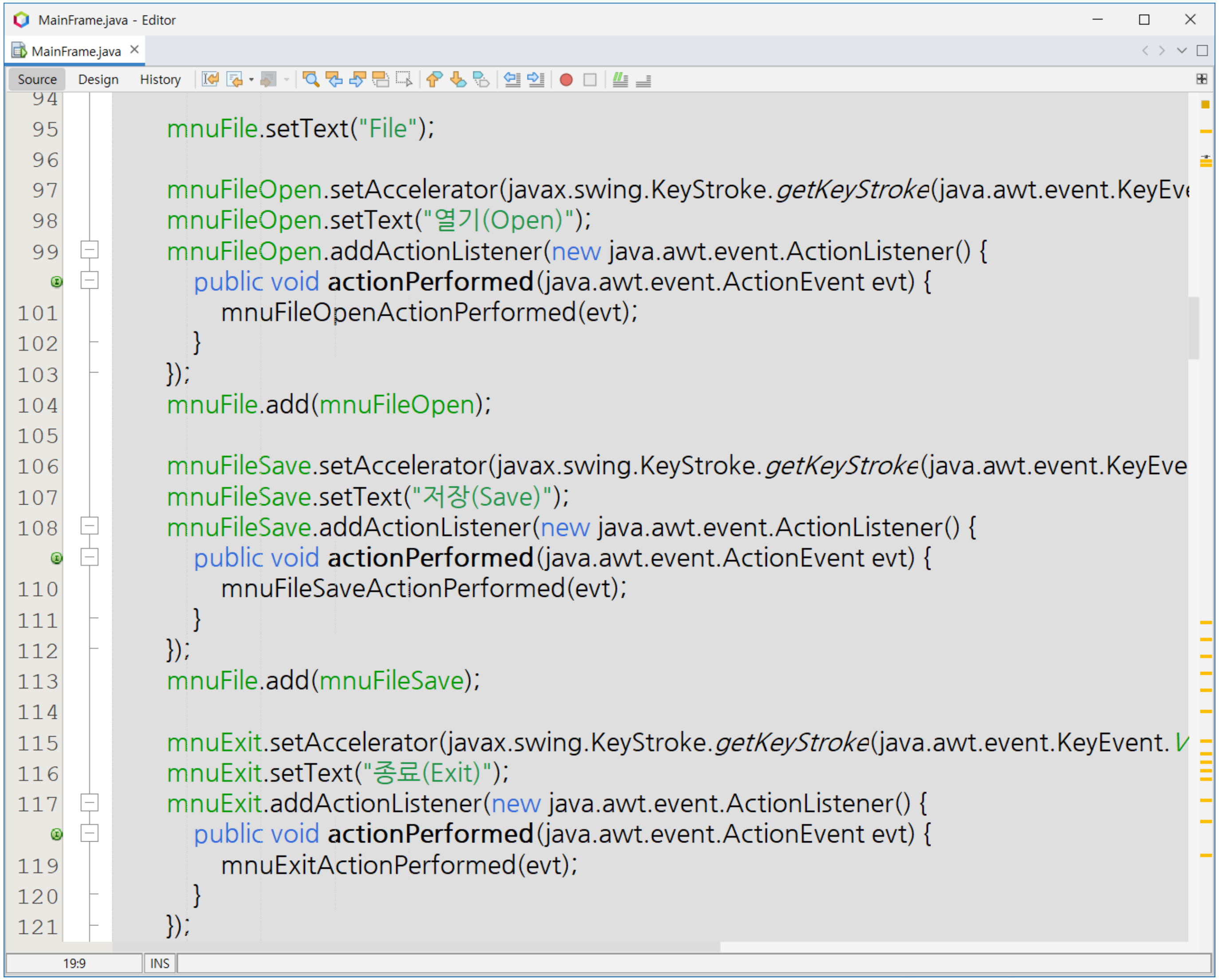
13. Setting Shortcut

Menu	Shortcut
Line	• Virtual Key : VK_L • Key Stroke : Alt + L
Ellipse	• Virtual Key : VK_E • Key Stroke : Alt + E
Rectangle	• Virtual Key : VK_R • Key Stroke : Alt + R



Practice 1 : Menu (8)

❖ Setting Menu Control Property Code



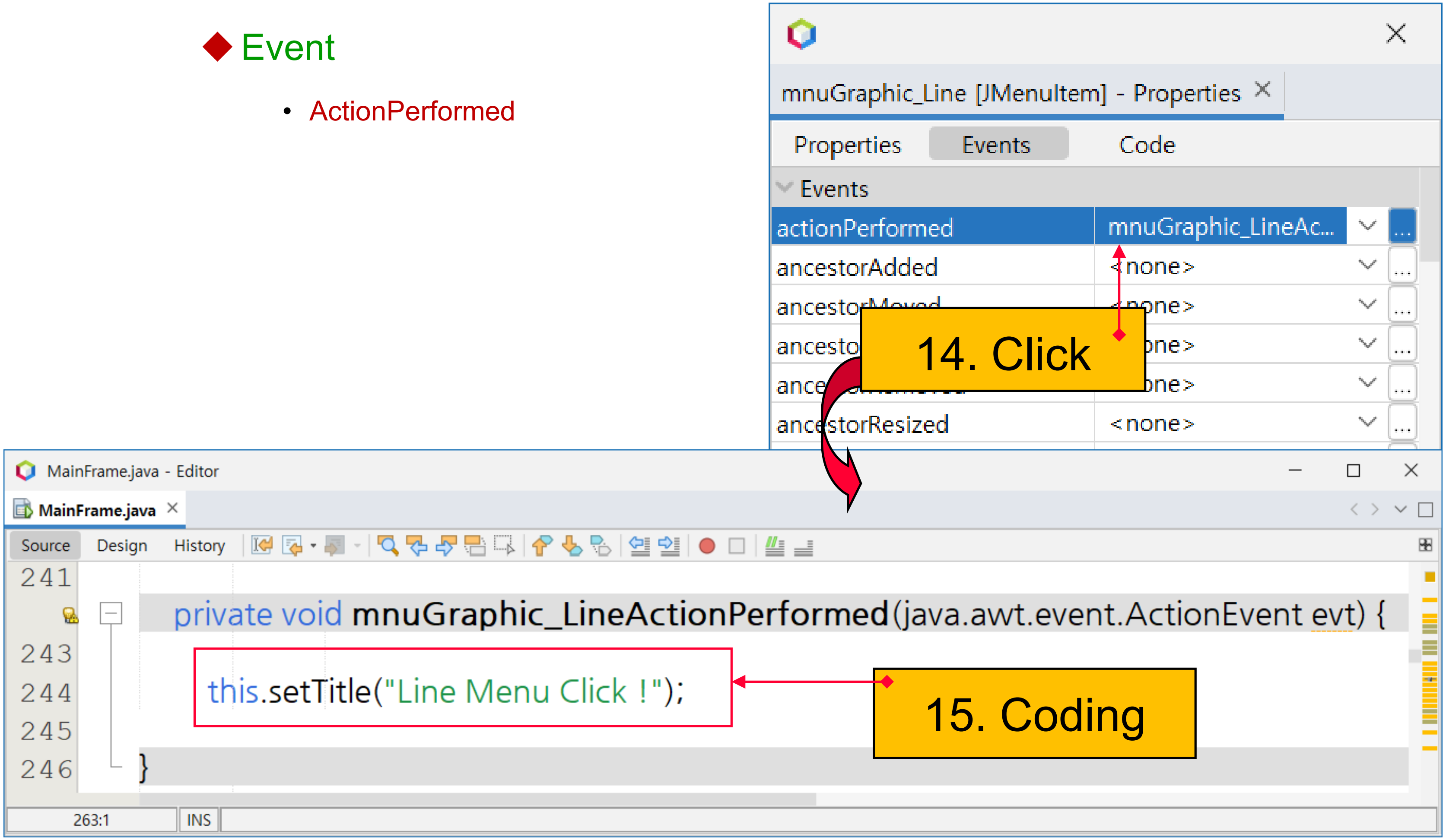
```
mnuFile.setText("File");

mnuFileOpen.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEv
mnuFileOpen.setText("열기(Open)");
mnuFileOpen.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mnuFileOpenActionPerformed(evt);
    }
});
mnuFile.add(mnuFileOpen);

mnuFileSave.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEve
mnuFileSave.setText("저장(Save)");
mnuFileSave.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mnuFileSaveActionPerformed(evt);
    }
});
mnuFile.add(mnuFileSave);

mnuExit.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.V
mnuExit.setText("종료(Exit)");
mnuExit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mnuExitActionPerformed(evt);
    }
});
```

Practice 1 : Menu (9)

❖ [Line] menu Event Handler

◆ Event

- ActionPerformed



mnuGraphic_Line [JMenuItem] - Properties

Events

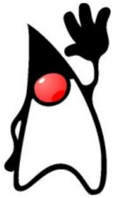
actionPerformed	mnuGraphic_LineAc...
ancestorAdded	<none>
ancestorMoved	<none>
ancesto	ne>
ance	ne>
ancestorResized	<none>

14. Click

MainFrame.java - Editor

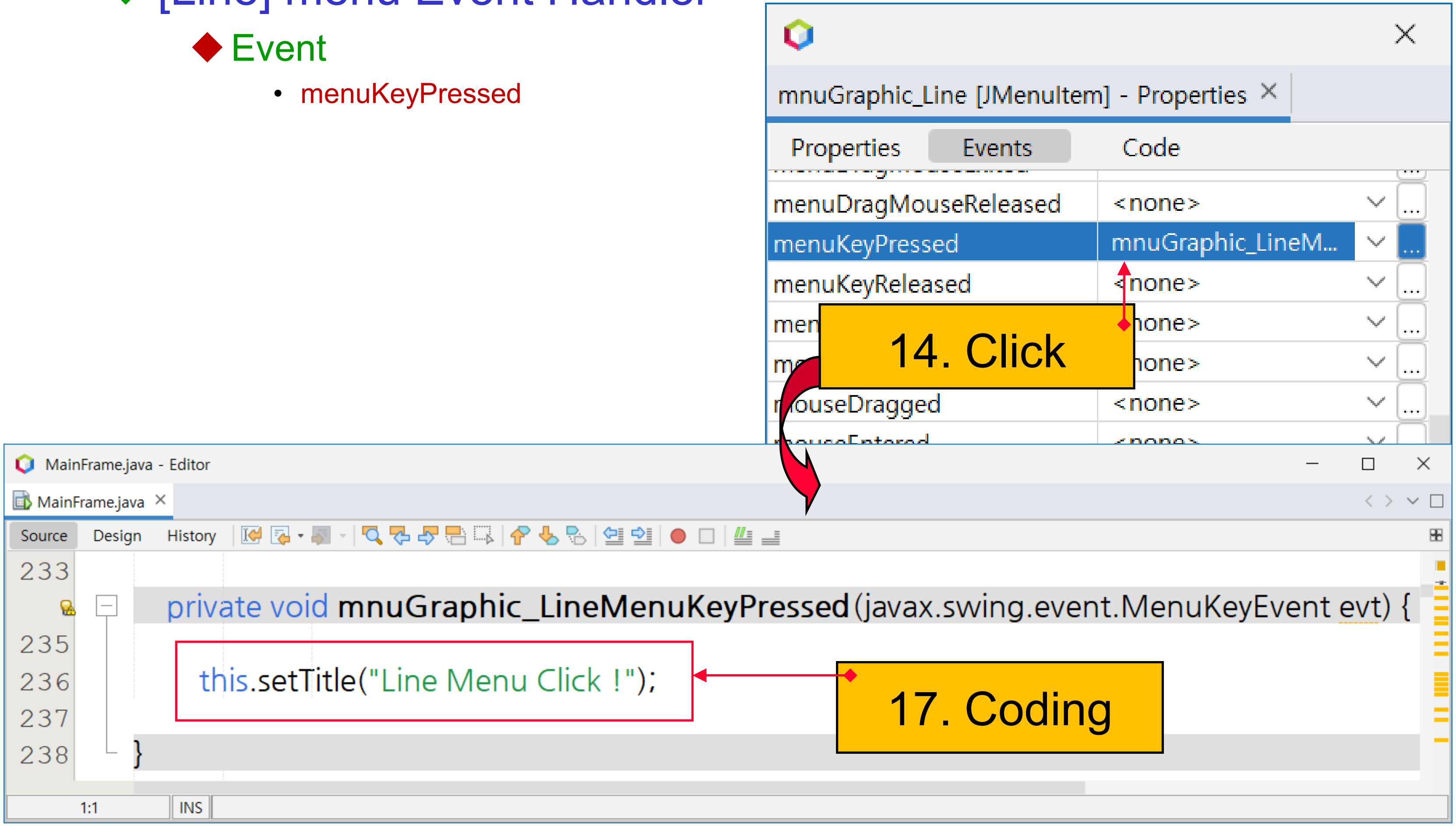
```java
private void mnuGraphic_LineActionPerformed(java.awt.event.ActionEvent evt) {

    this.setTitle("Line Menu Click !");

}
```

15. Coding



Practice 1 : Menu (10)

- [Line] menu Event Handler
  - Event
    - menuKeyPressed

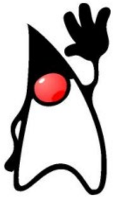


menuDragMouseReleased	<none>
menuKeyPressed	mnuGraphic_LineM...
menuKeyReleased	<none>
men	<none>
m	<none>
mouseDragged	<none>

14. Click

```
private void mnuGraphic_LineMenuKeyPressed(javax.swing.event.MenuKeyEvent evt) {

    this.setTitle("Line Menu Click !");

}
```

17. Coding

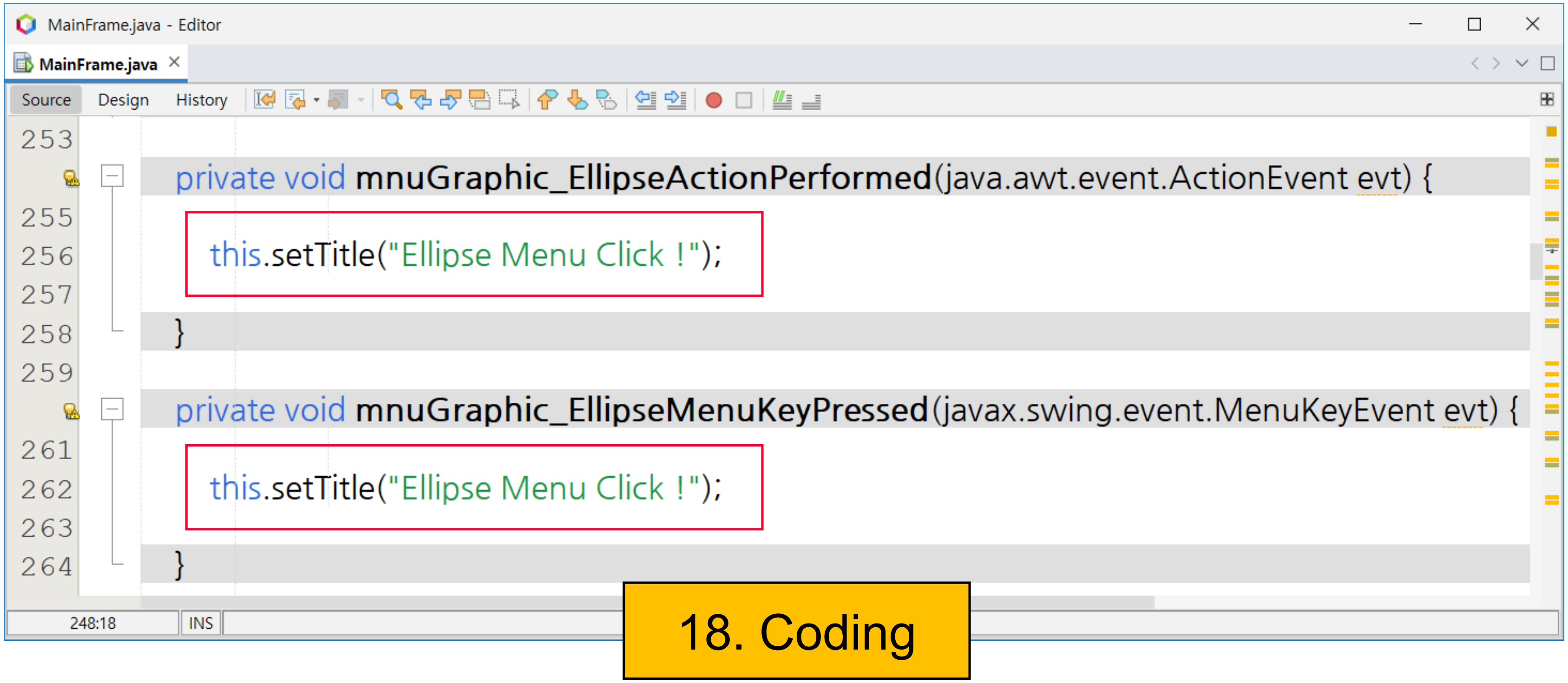


Practice 1 : Menu (11)

❖ [Ellipse] menu Event Handler

◆ Event

- ActionPerformed
- menuKeyPressed



```
private void mnuGraphic_EllipseActionPerformed(java.awt.event.ActionEvent evt) {
    this.setTitle("Ellipse Menu Click !");
}

private void mnuGraphic_EllipseMenuKeyPressed(javax.swing.event.MenuKeyEvent evt) {
    this.setTitle("Ellipse Menu Click !");
}
```

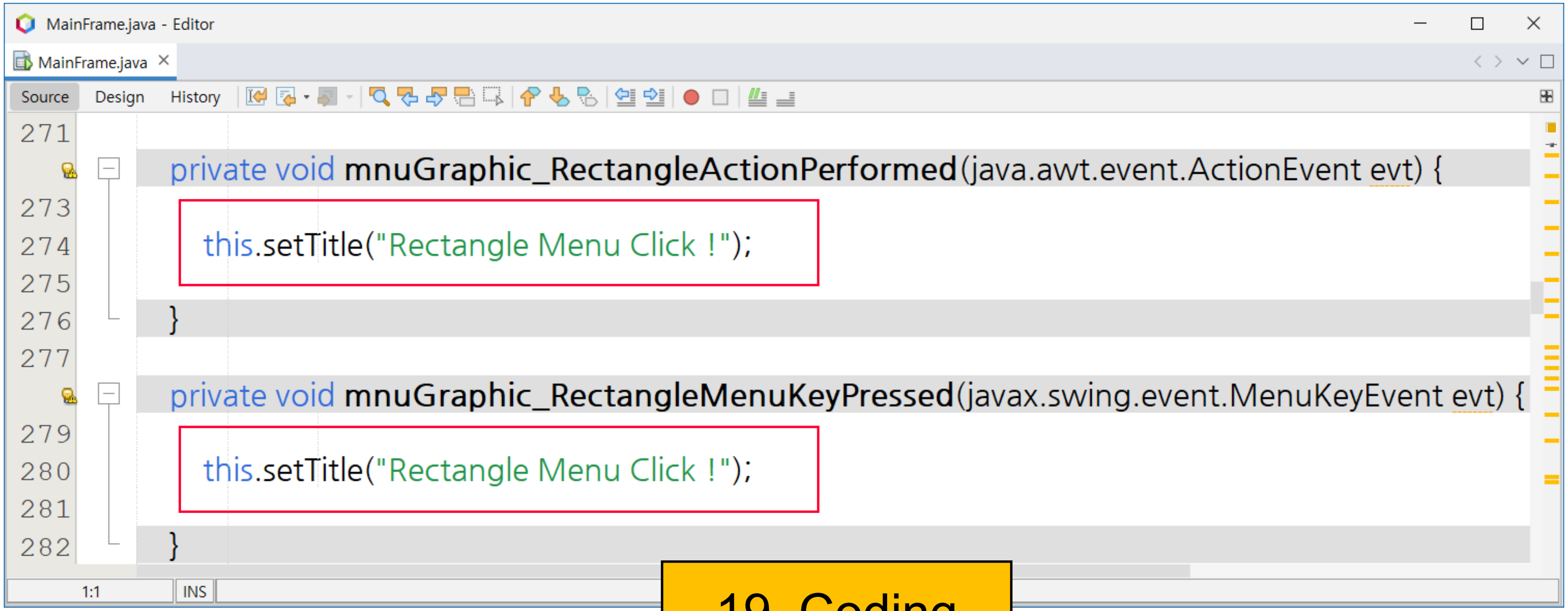
18. Coding

Practice 1 : Menu (12)

❖ [Rectangle] 메뉴 Event Handler

◆ Event

- ActionPerformed
- menuKeyPressed

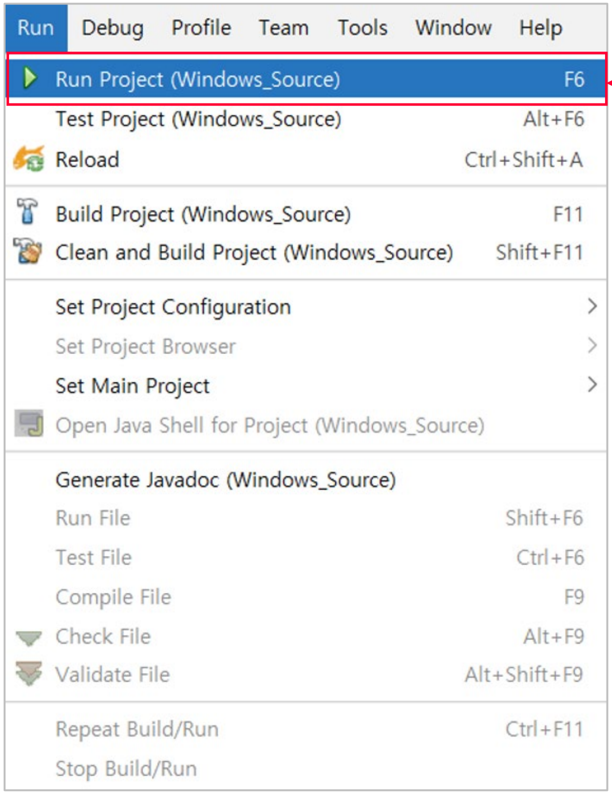


```java
private void mnuGraphic_RectangleActionPerformed(java.awt.event.ActionEvent evt) {
    this.setTitle("Rectangle Menu Click !");
}

private void mnuGraphic_RectangleMenuKeyPressed(javax.swing.event.MenuKeyEvent evt) {
    this.setTitle("Rectangle Menu Click !");
}
```

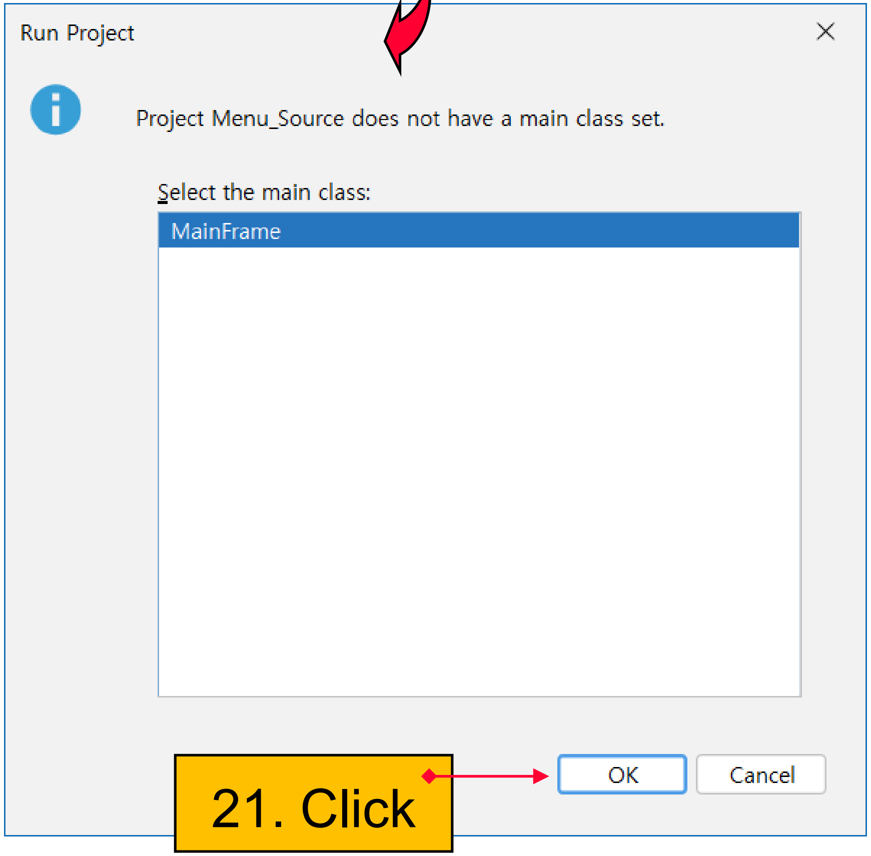
19. Coding

Practice 1 : Menu (13)

Run


Run
Debug
Profile
Team
Tools
Window
Help
Run Project (Windows_Source) F6
Test Project (Windows_Source) Alt+F6
Reload Ctrl+Shift+A
Build Project (Windows_Source) F11
Clean and Build Project (Windows_Source) Shift+F11
Set Project Configuration
Set Project Browser
Set Main Project
Open Java Shell for Project (Windows_Source)
Generate Javadoc (Windows_Source)
Run File Shift+F6
Test File Ctrl+F6
Compile File F9
Check File Alt+F9
Validate File Alt+Shift+F9
Repeat Build/Run Ctrl+F11
Stop Build/Run

20. Click

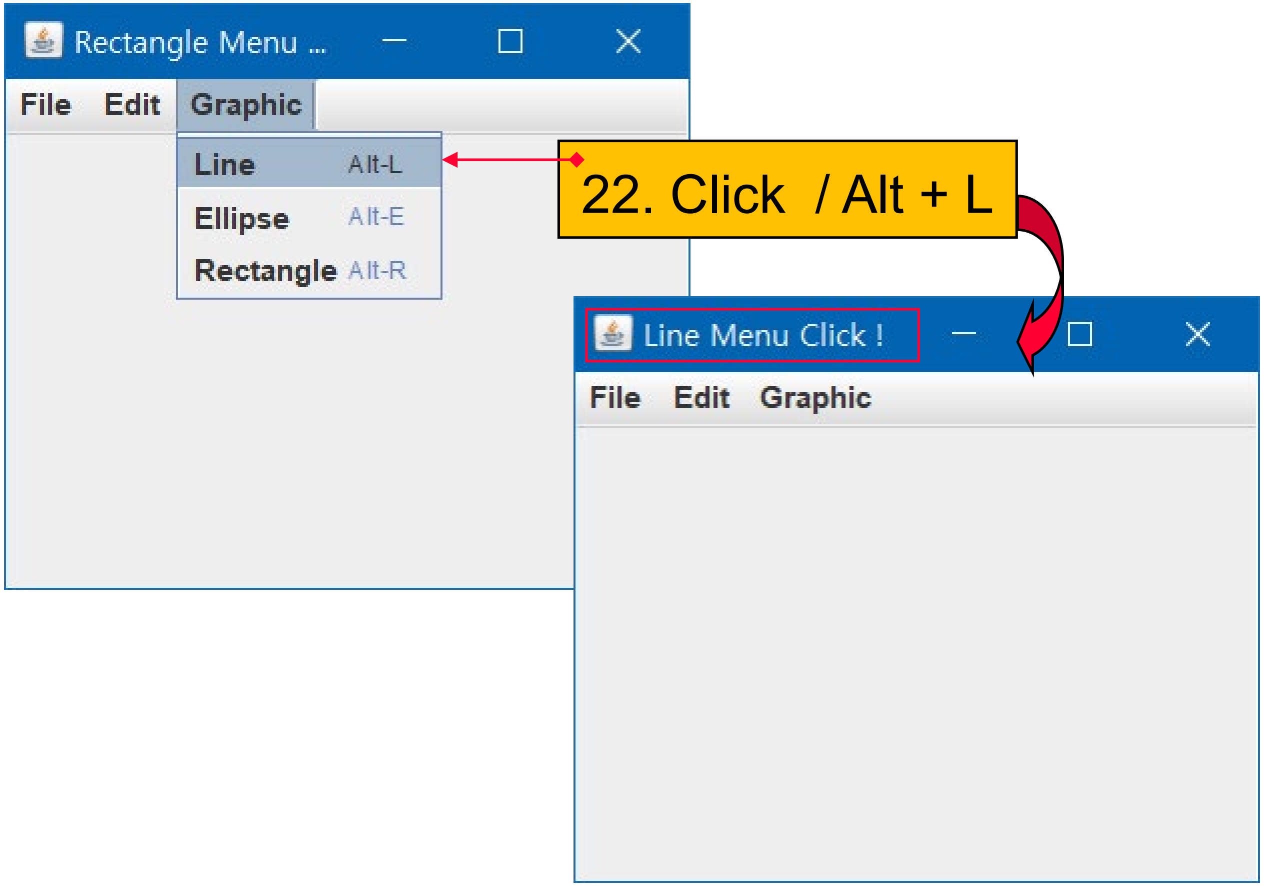


Project Menu_Source does not have a main class set.

21. Click

Practice 1 : Menu (14)

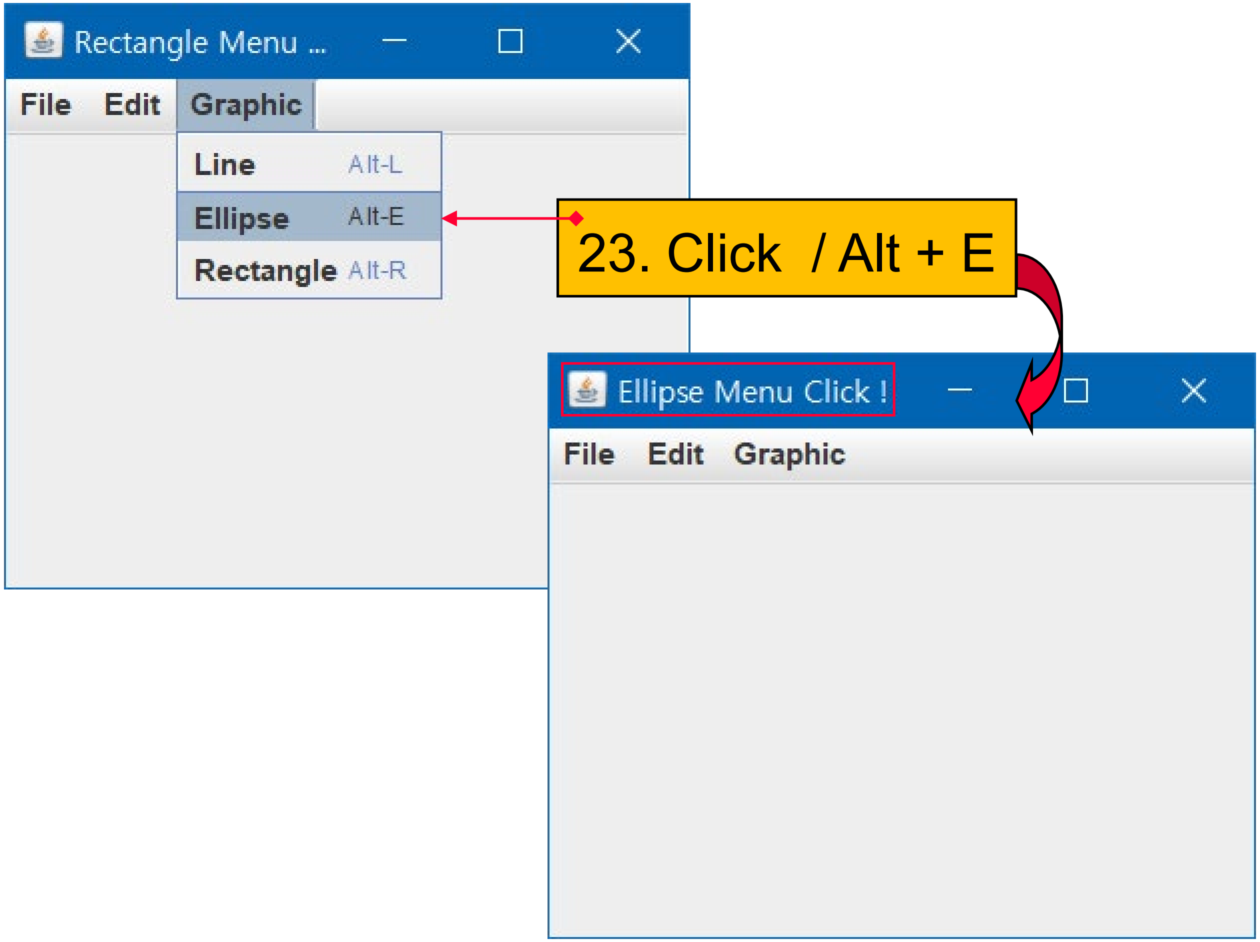
- [Line] Menu Click



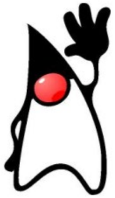
22. Click / Alt + L

Practice 1 : Menu (15)

❖ [Ellipse] Menu Click

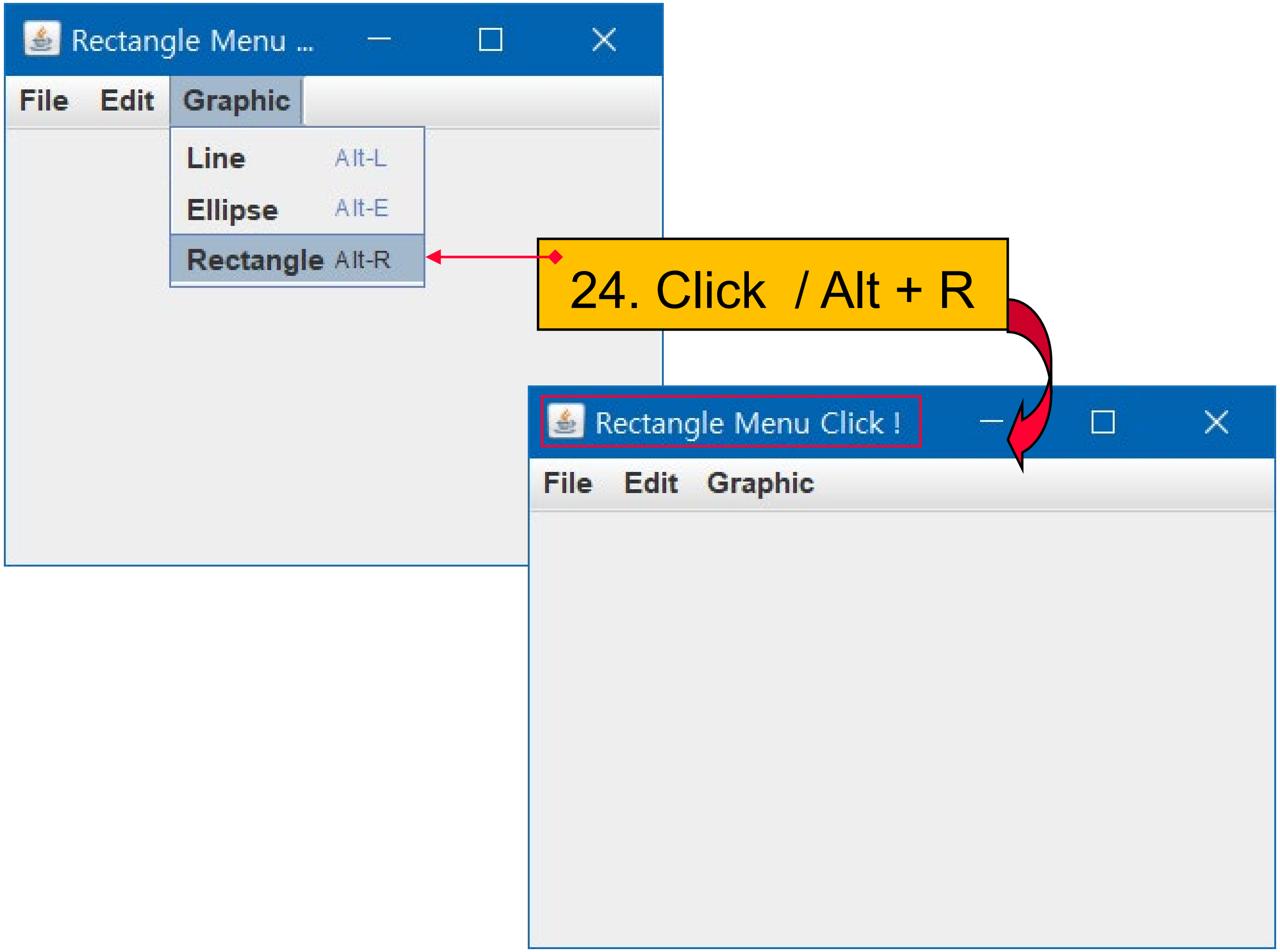


23. Click / Alt + E



Practice 1 : Menu (16)

❖ [Rectangle] Menu Click

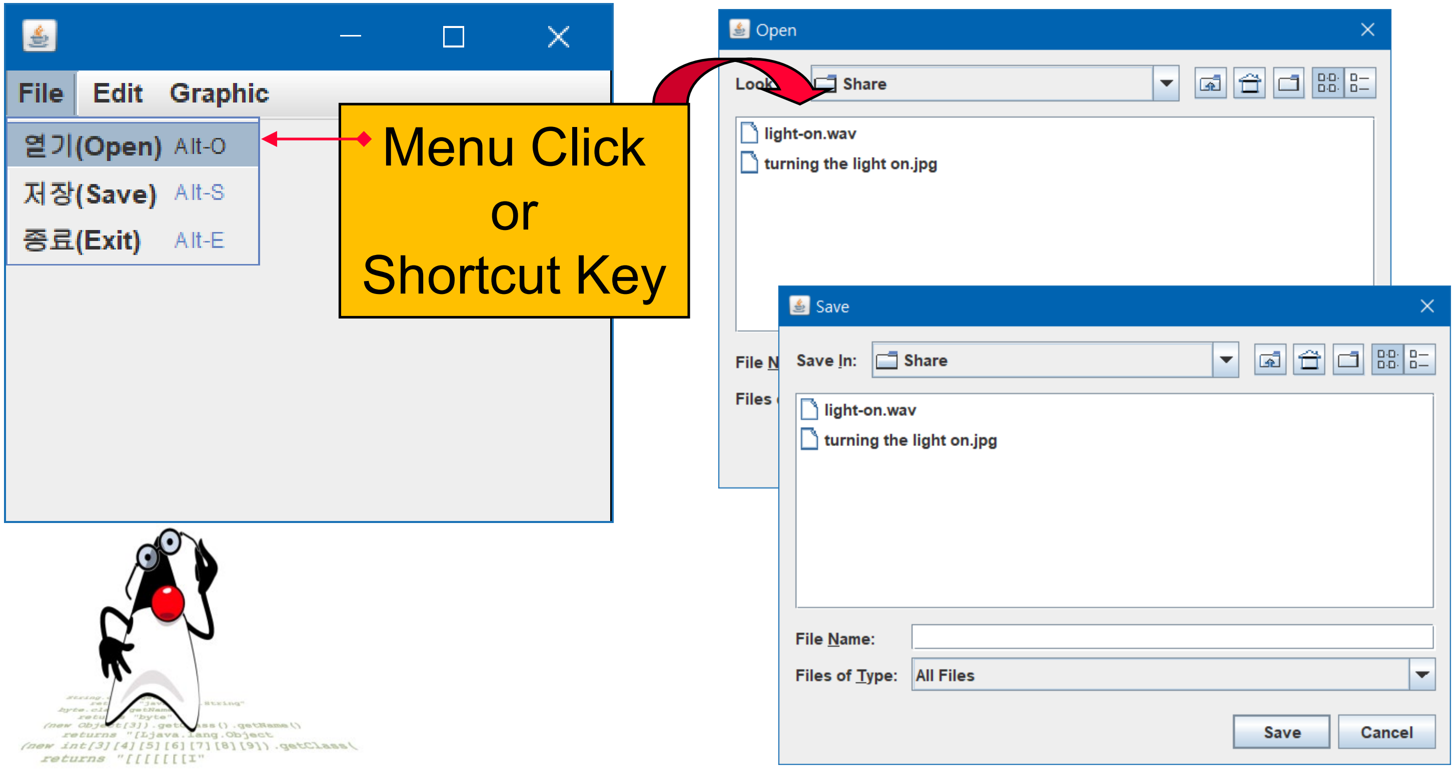


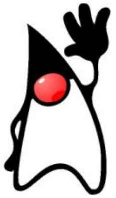
24. Click / Alt + R

Practice 2 : Menu

❖ Project Name: Menu_Source(Time: 30 min)

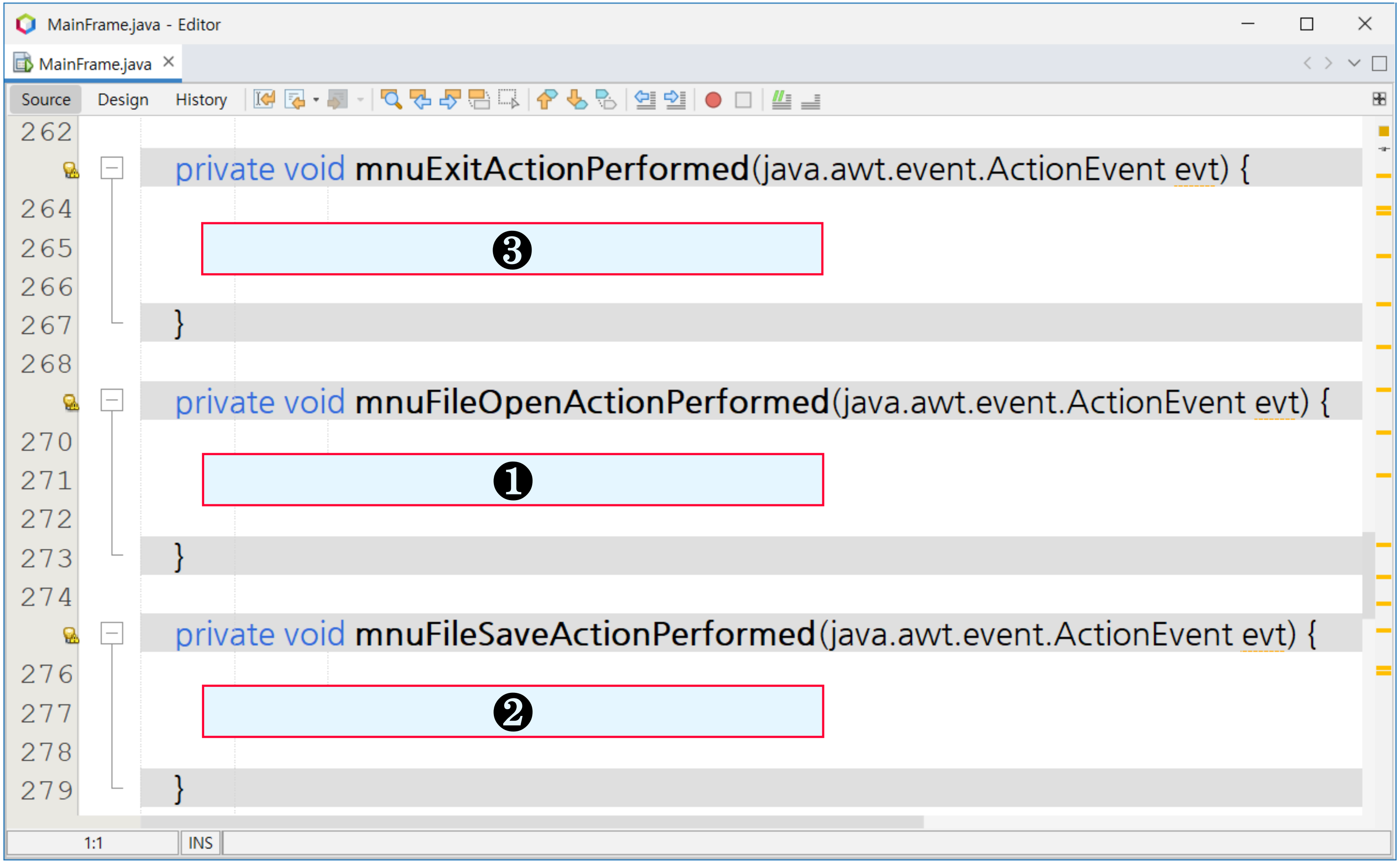
- Click Sub memu & Press Shortcut Key of the [File] menu
  - 열기(Open) : FileDialog Open Run // mnuFile_Open
  - 저장(Save) : FileDialog Save Run // mnuFile_Save
  - 종료(Exit) : 프로그램 종료 // mnuFile_Exit


Menu Click or Shortcut Key



Practice 3 : Menu (1)

❖ [File] Menu Event Handler



```
private void mnuExitActionPerformed(java.awt.event.ActionEvent evt) {

    ❸

}

private void mnuFileOpenActionPerformed(java.awt.event.ActionEvent evt) {

    ❶

}

private void mnuFileSaveActionPerformed(java.awt.event.ActionEvent evt) {

    ❷

}
```

Color Class (1)

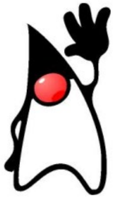
Color 클래스

구성

```
Color(int r, int g, int b)
```

Method

Method	Comment
Color getColor()	현재 설정된 Color 반환
void setColor(Color c)	새로운 Color 설정
Color getBackground()	컴포넌트의 배경색 반환
Color getForeground()	컴포넌트의 전경색 반환
void setBackground(Color c)	컴포넌트의 배경색 지정
void setForeground(Color c)	컴포넌트의 전경색 지정



Color Class (2)

❖ 색상값

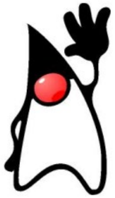
Constant	Color	Constant	Color
Color.black, Color.BLACK	검정색	Color.magenta, Color.MAGENTA	진홍색
Color.blue, Color.BLUE	파란색	Color.orange, Color.ORANGE	주황색
Color.cyan, Color.CYAN	하늘색	Color.pink, Color.PINK	분홍색
Color.darkGray, Color.DARK_GRAY	짙은 회색	Color.red, Color.RED	빨간색
Color.gray, Color.GRAY	회색	Color.white, Color.WHITE	하얀색
Color.green, Color.GREEN	녹색	Color.yellow, Color.YELLOW	노란색

Graphic Class

Graphics 클래스

- Function
  - 도형(선, 타원, 사각형, 다각형) 그리기 구현 Function
- Method

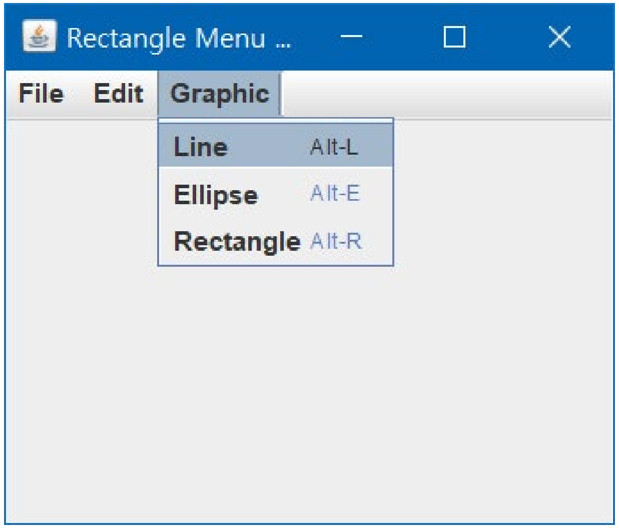
Method	Comment
draw3DRect(), fill3DRect()	입체 사각형 그리기
drawLine()	선 그리기
drawRect(), fillRect()	사각형 그리기, 채운 사각형 그리기
drawOval(), fillOval()	타원 그리기, 채운 타원 그리기
drawArc (), fillArc()	부채꼴 그리기, 채운 부채꼴 그리기
drawPolygon(), fillPolygon()	다각형 그리기, 채운 다각형 그리기
drawPolyline()	연결선 그리기
drawString()	문자쓰기
drawRoundRect(), fillRoundRect()	둥근 사각형 그리기, 둥근 사각형 채우기
drawImage()	이미지 그리기

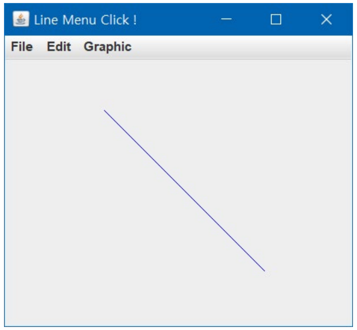


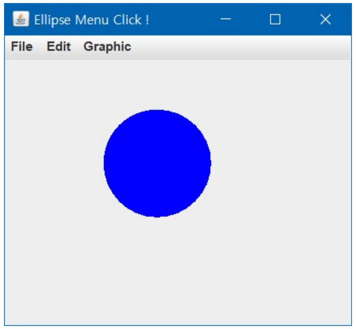
Practice 3 : Menu

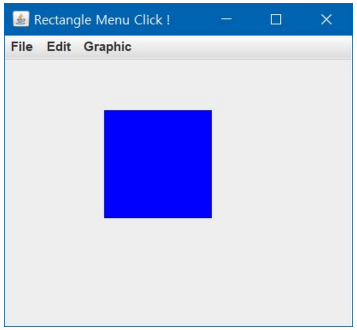
❖ Project Name: Menu_Source(Time: 30 min)

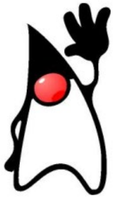
- [Graphic] 메뉴의 하위 메뉴 Click / 단축 키 입력
  - Line : Line 그리기
  - Ellipse : Ellipse 그리기
  - Rectangle : Rectangle 그리기





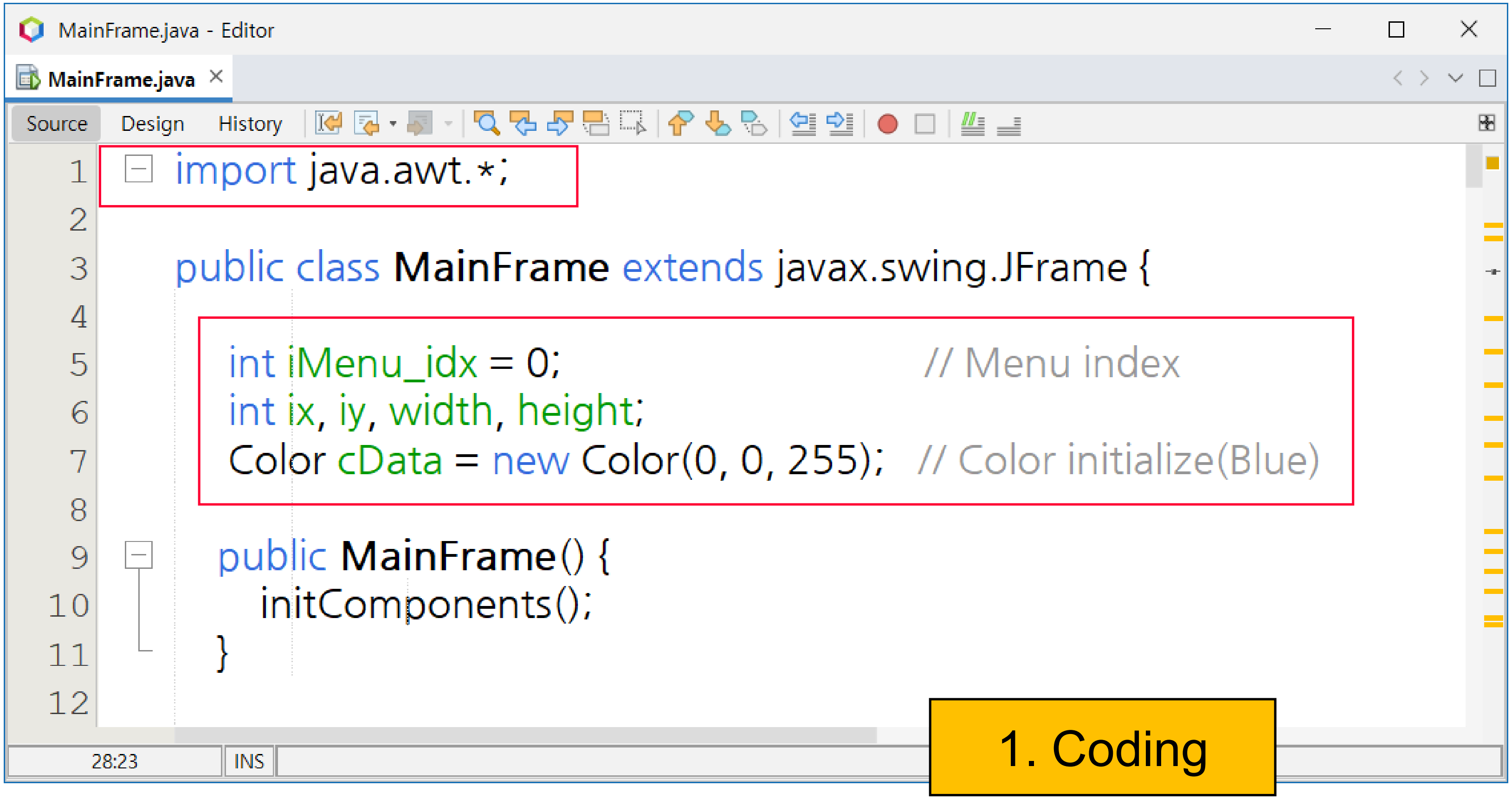






Practice 3 : Menu (1)

❖ Setting Member Variable

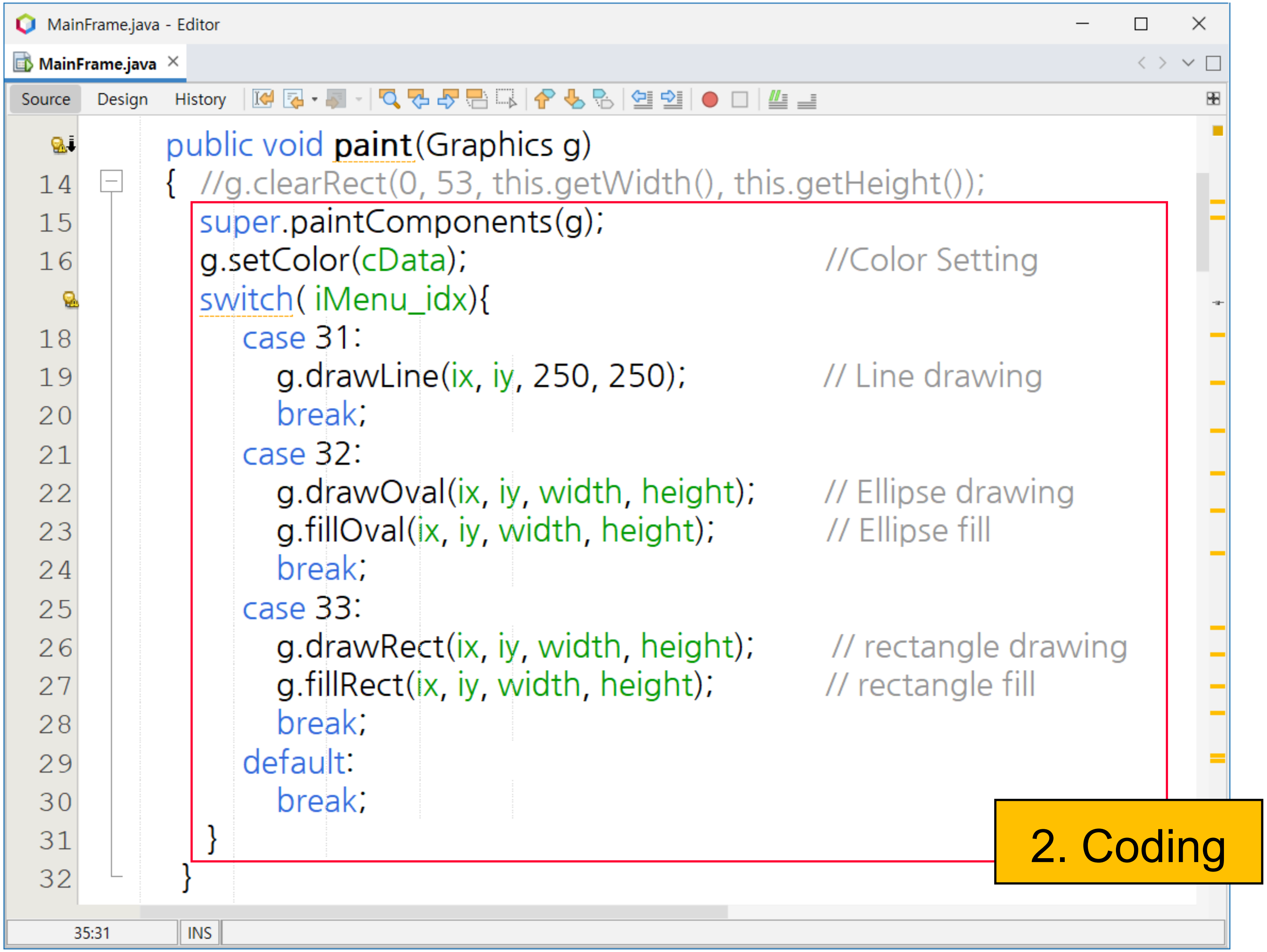


```java
import java.awt.*;

public class MainFrame extends javax.swing.JFrame {

    int iMenu_idx = 0;                    // Menu index
    int ix, iy, width, height;
    Color cData = new Color(0, 0, 255);   // Color initialize(Blue)

    public MainFrame() {
        initComponents();
    }

```

1. Coding

Practice 3 : Menu (2)

❖ Paint() method



```
public void paint(Graphics g)
{   //g.clearRect(0, 53, this.getWidth(), this.getHeight());
    super.paintComponents(g);
    g.setColor(cData);                                  //Color Setting
    switch( iMenu_idx){
        case 31:
            g.drawLine(ix, iy, 250, 250);               // Line drawing
            break;
        case 32:
            g.drawOval(ix, iy, width, height);          // Ellipse drawing
            g.fillOval(ix, iy, width, height);          // Ellipse fill
            break;
        case 33:
            g.drawRect(ix, iy, width, height);          // rectangle drawing
            g.fillRect(ix, iy, width, height);          // rectangle fill
            break;
        default:
            break;
    }
}
```

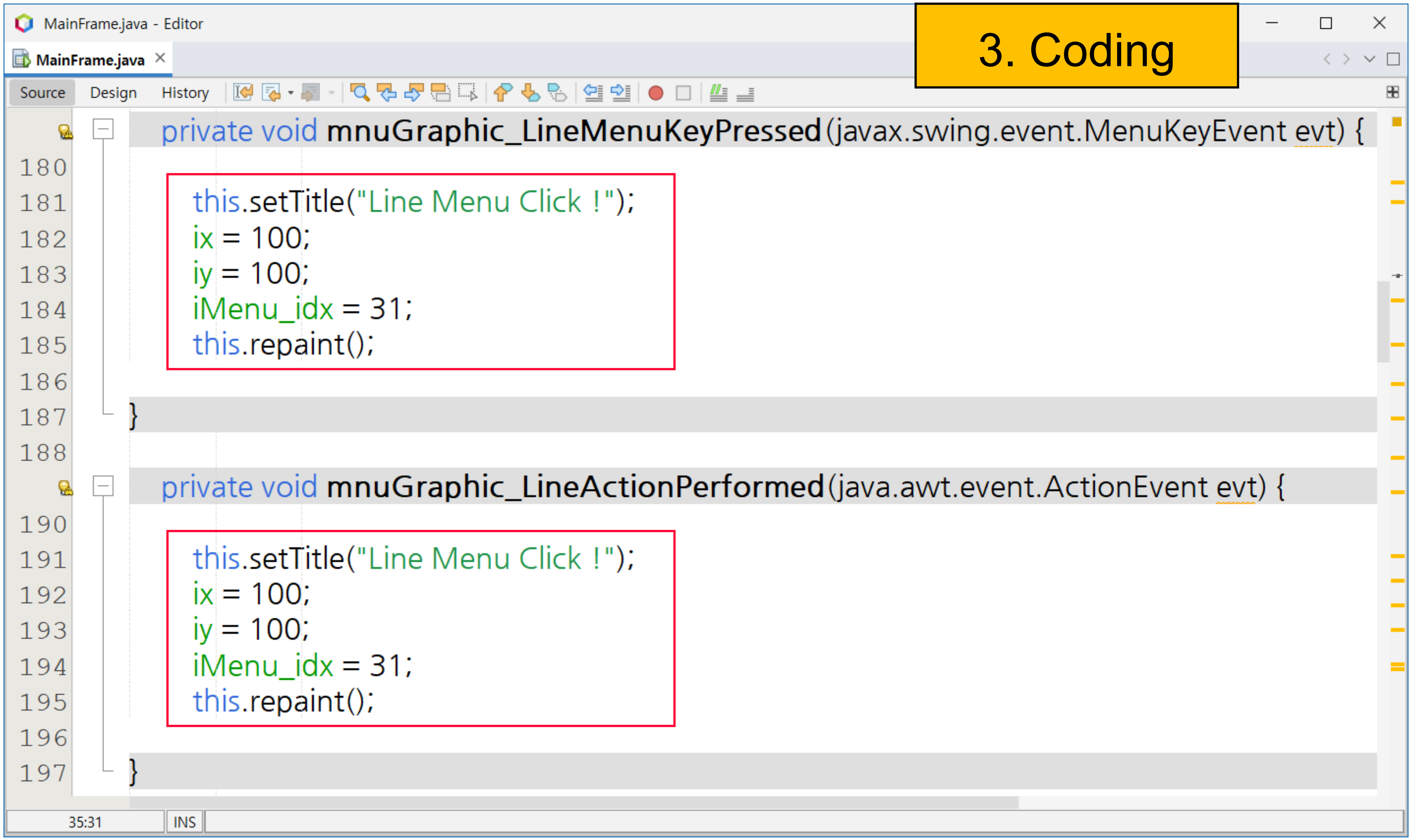
2. Coding

Practice 3 : Menu (3)

[Line] Menu Event Handler

Event

- ActionPerformed
- menuKeyPressed



3. Coding

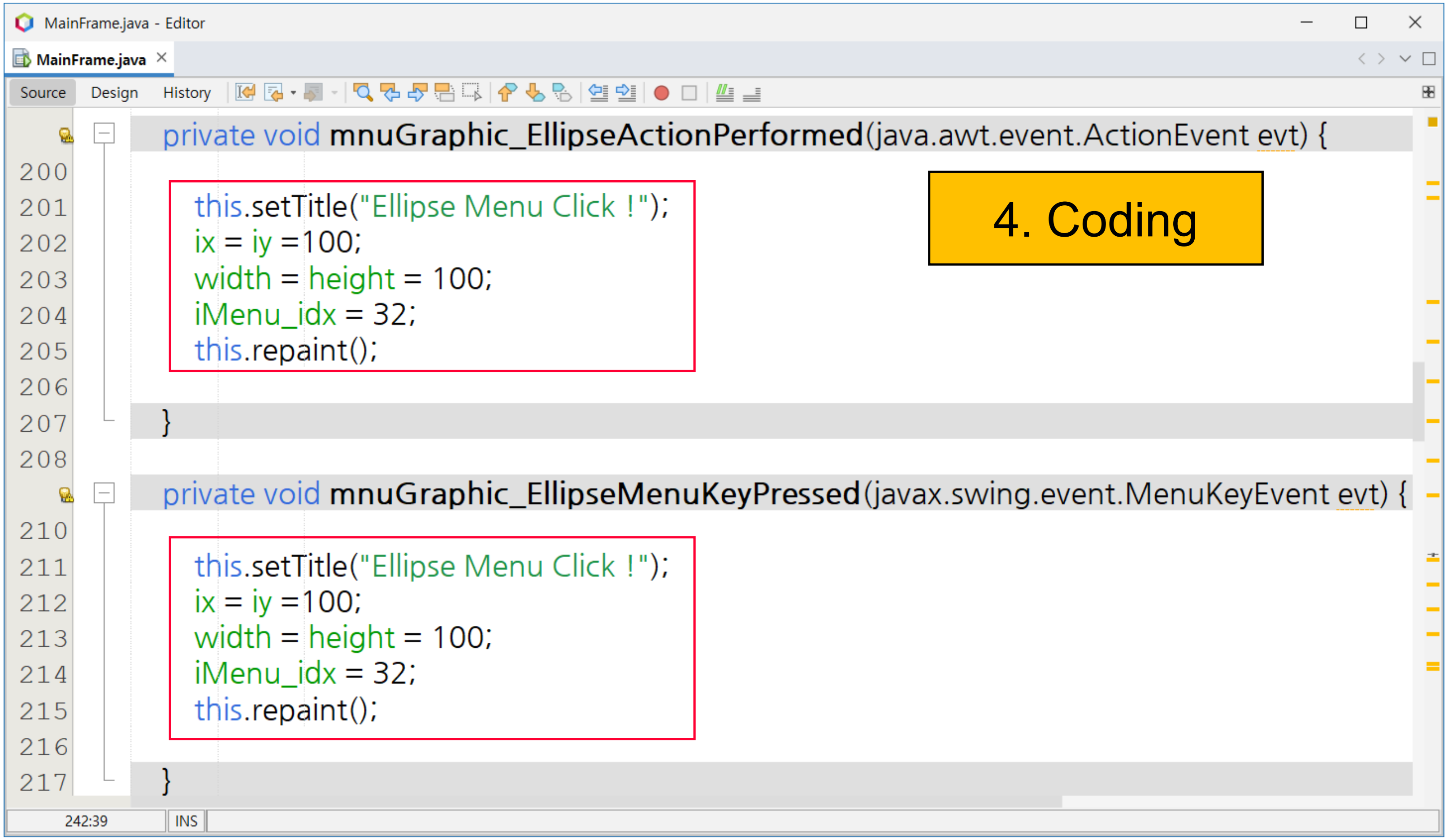
```
private void mnuGraphic_LineMenuKeyPressed(javax.swing.event.MenuKeyEvent evt) {

    this.setTitle("Line Menu Click !");
    ix = 100;
    iy = 100;
    iMenu_idx = 31;
    this.repaint();

}

private void mnuGraphic_LineActionPerformed(java.awt.event.ActionEvent evt) {

    this.setTitle("Line Menu Click !");
    ix = 100;
    iy = 100;
    iMenu_idx = 31;
    this.repaint();

}
```

Practice 3 : Menu (4)

❖ [Ellipse] Menu Event Handler

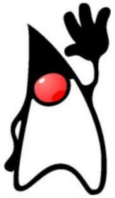
◆ Event

- ActionPerformed
- menuKeyPressed



```
private void mnuGraphic_EllipseActionPerformed(java.awt.event.ActionEvent evt) {

    this.setTitle("Ellipse Menu Click !");
    ix = iy =100;
    width = height = 100;
    iMenu_idx = 32;
    this.repaint();

}

private void mnuGraphic_EllipseMenuKeyPressed(javax.swing.event.MenuKeyEvent evt) {

    this.setTitle("Ellipse Menu Click !");
    ix = iy =100;
    width = height = 100;
    iMenu_idx = 32;
    this.repaint();

}
```

4. Coding

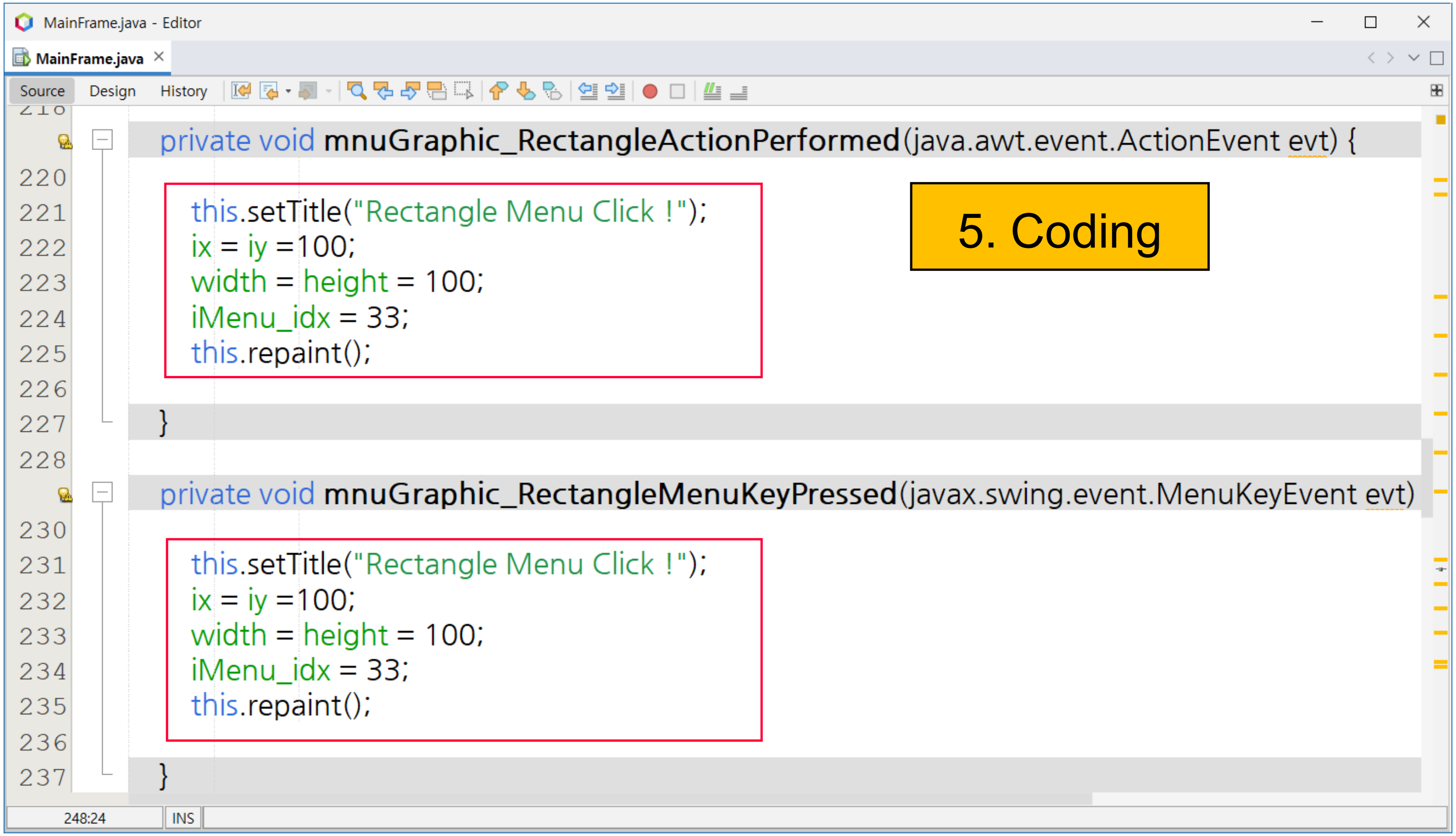


Practice 3 : Menu (5)

❖ [Rectangle] Menu Event Handler

◆ Event

- ActionPerformed
- menuKeyPressed



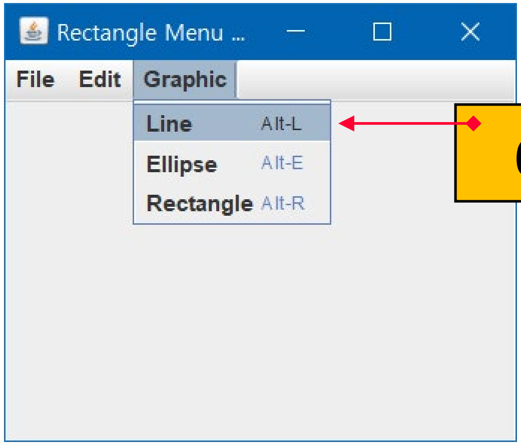
5. Coding

```
private void mnuGraphic_RectangleActionPerformed(java.awt.event.ActionEvent evt) {

    this.setTitle("Rectangle Menu Click !");
    ix = iy =100;
    width = height = 100;
    iMenu_idx = 33;
    this.repaint();

}

private void mnuGraphic_RectangleMenuKeyPressed(javax.swing.event.MenuKeyEvent evt)

    this.setTitle("Rectangle Menu Click !");
    ix = iy =100;
    width = height = 100;
    iMenu_idx = 33;
    this.repaint();

}
```

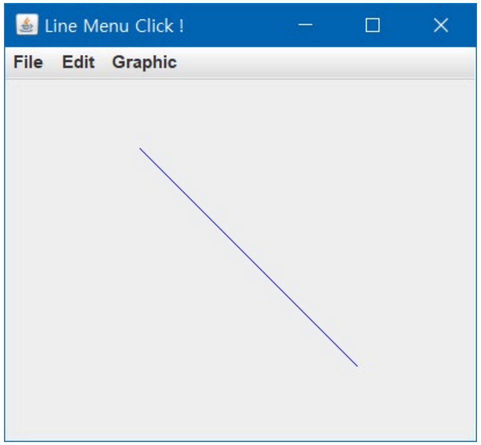
Practice 3 : Menu (6)

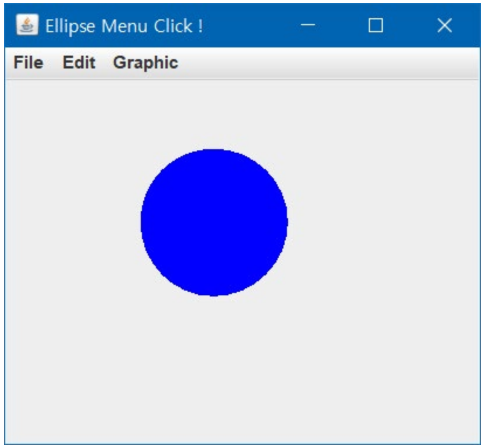
Run

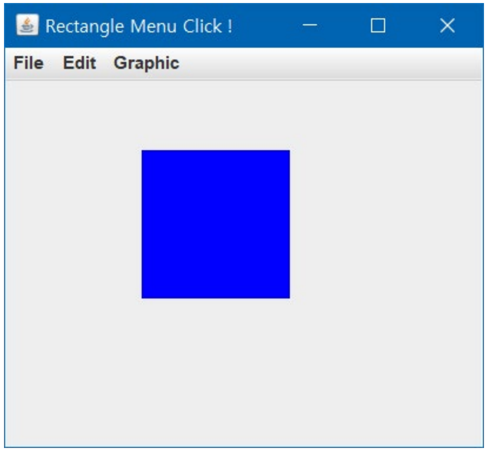
❖ [Line/Ellipse/Rectangle] Menu Click


Rectangle Menu ...

6. Click / Alt+L Key Input


Line Menu Click !


Ellipse Menu Click !


Rectangle Menu Click !

Pop Up Menu

Pop Up Menu

❖ 구성

Component	Class
메뉴바	Java.awt.MenuBar
메뉴	Java.awt.Menu
메뉴 아이템	Java.awt.MenuItem

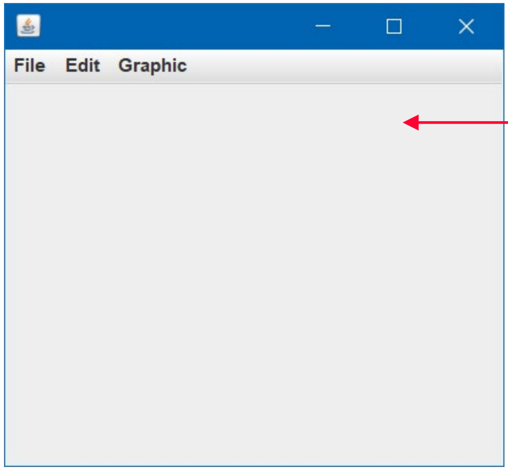
❖ MenuBar 클래스 객체 생성

```
MenuBar mnuBar = new MenuBar();
```

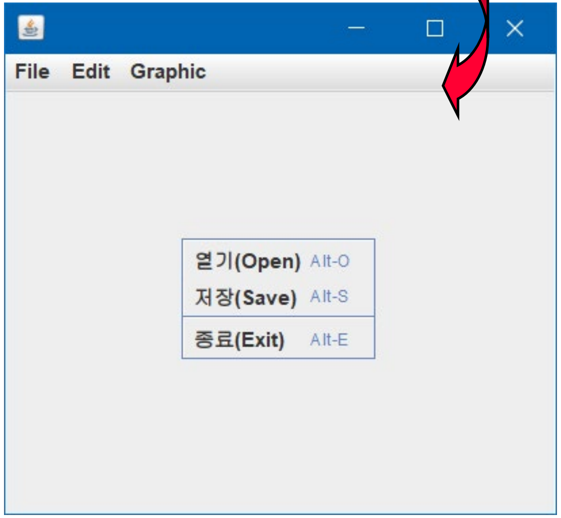
Practice 4 : Pop Up Menu

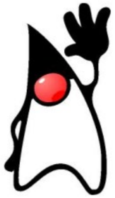
❖ Project Name: Menu_Source(Time: 30 min)

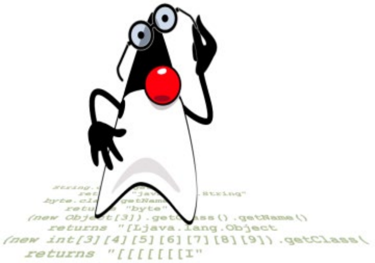
- [Pop UP] 메뉴 구현
  - Mouse right-button Click
  - [File] menu의 sub menu로 구성된 Pop Up menu output



3. Mouse right-button Click



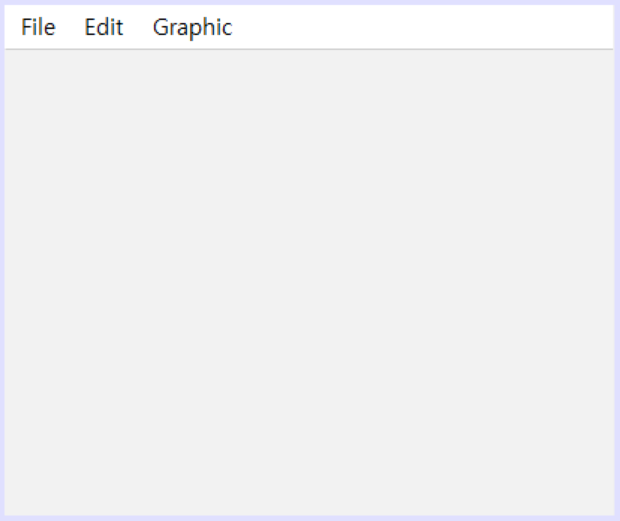




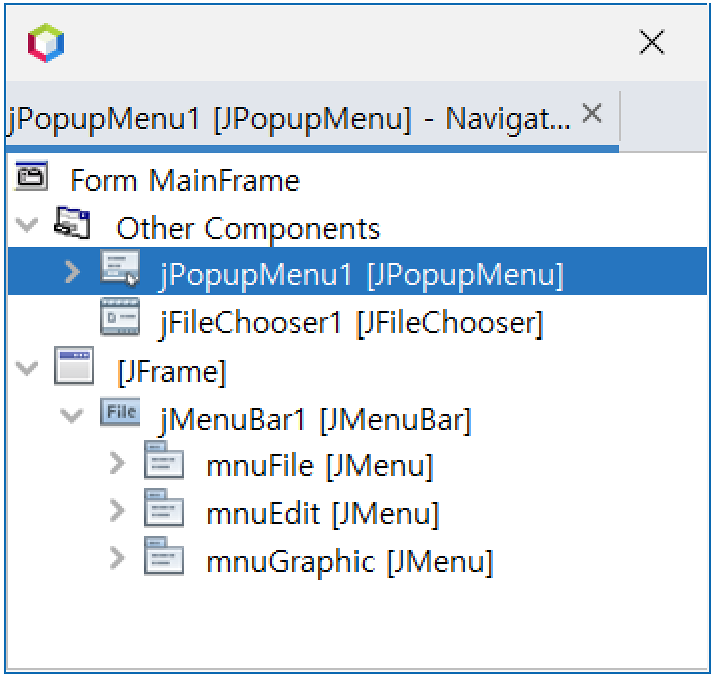
Practice 4 : Pop Up Menu (1)

Control Layout & Property Setting

Popup Menu



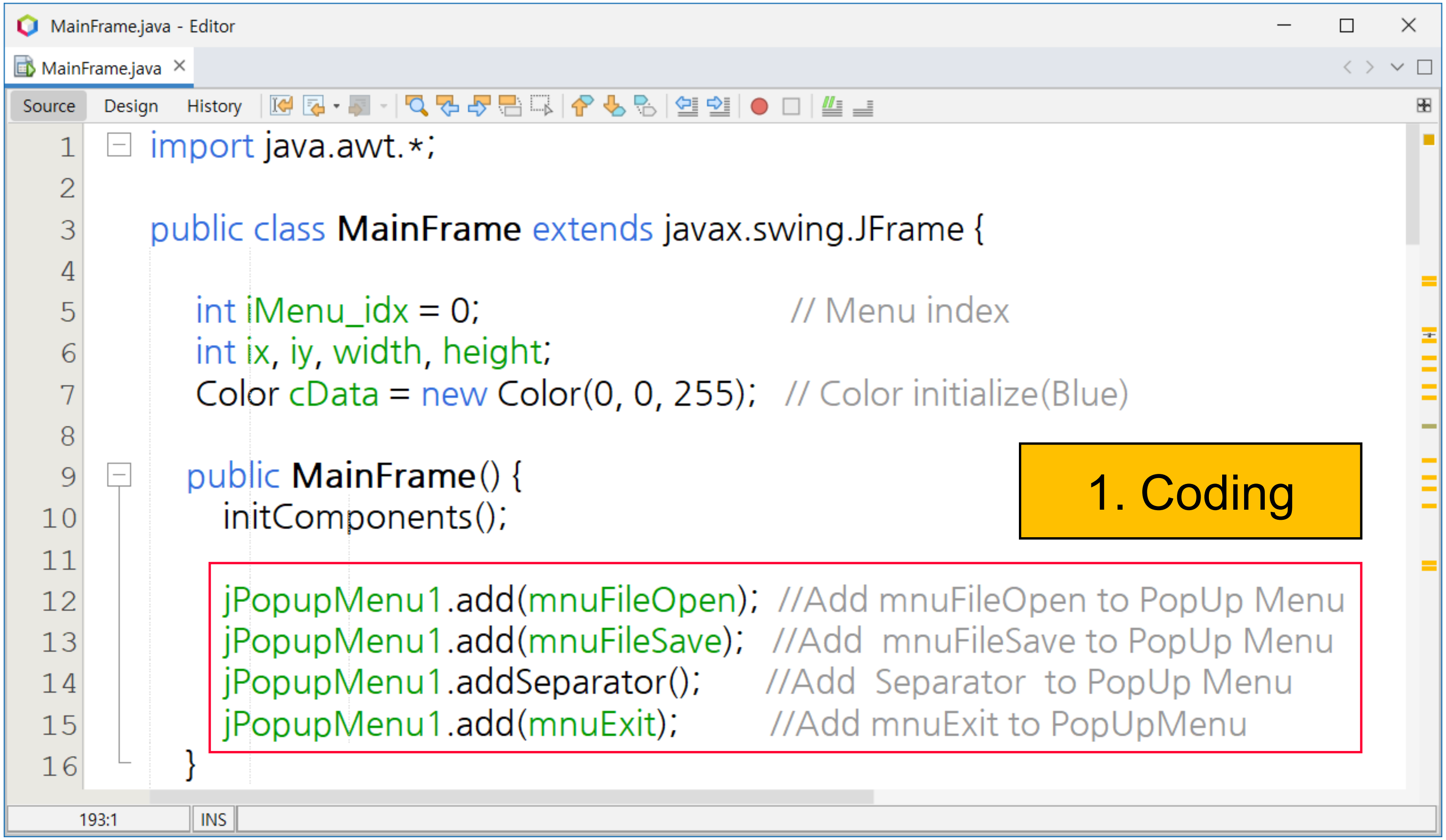
Control	Properties Setting
jPopupMenu1	• Variable Name : jPopupMenu1


jPopupMenu1 [JPopupMenu] - Navigat...

- Form MainFrame
  - Other Components
    - jPopupMenu1 [JPopupMenu]
    - jFileChooser1 [JFileChooser]
  - [JFrame]
    - jMenuBar1 [JMenuBar]
      - mnuFile [JMenu]
      - mnuEdit [JMenu]
      - mnuGraphic [JMenu]

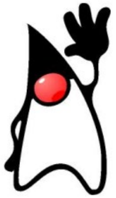
Practice 4 : Pop Up Menu (2)

- Add menuitem to jPopupMenu1
  - mnuFileOpen
  - mnuFileSave
  - mnuExit



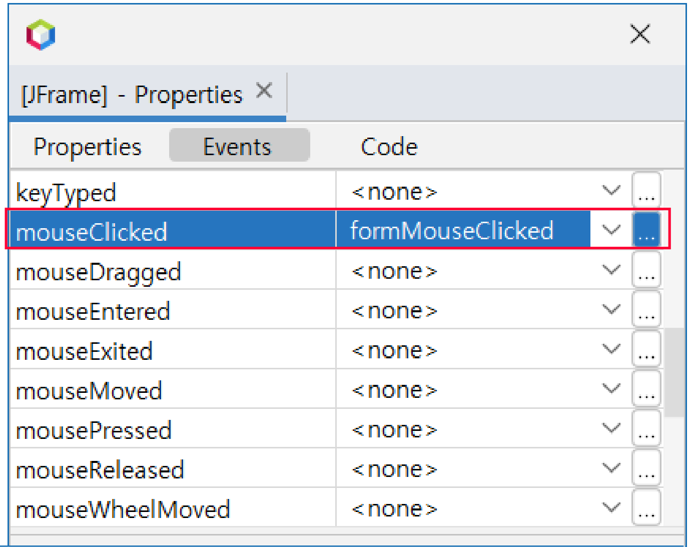
```java
import java.awt.*;

public class MainFrame extends javax.swing.JFrame {

    int iMenu_idx = 0;                       // Menu index
    int ix, iy, width, height;
    Color cData = new Color(0, 0, 255);   // Color initialize(Blue)

    public MainFrame() {
        initComponents();

        jPopupMenu1.add(mnuFileOpen);  //Add mnuFileOpen to PopUp Menu
        jPopupMenu1.add(mnuFileSave);   //Add  mnuFileSave to PopUp Menu
        jPopupMenu1.addSeparator();       //Add  Separator  to PopUp Menu
        jPopupMenu1.add(mnuExit);          //Add mnuExit to PopUpMenu
    }
```

1. Coding

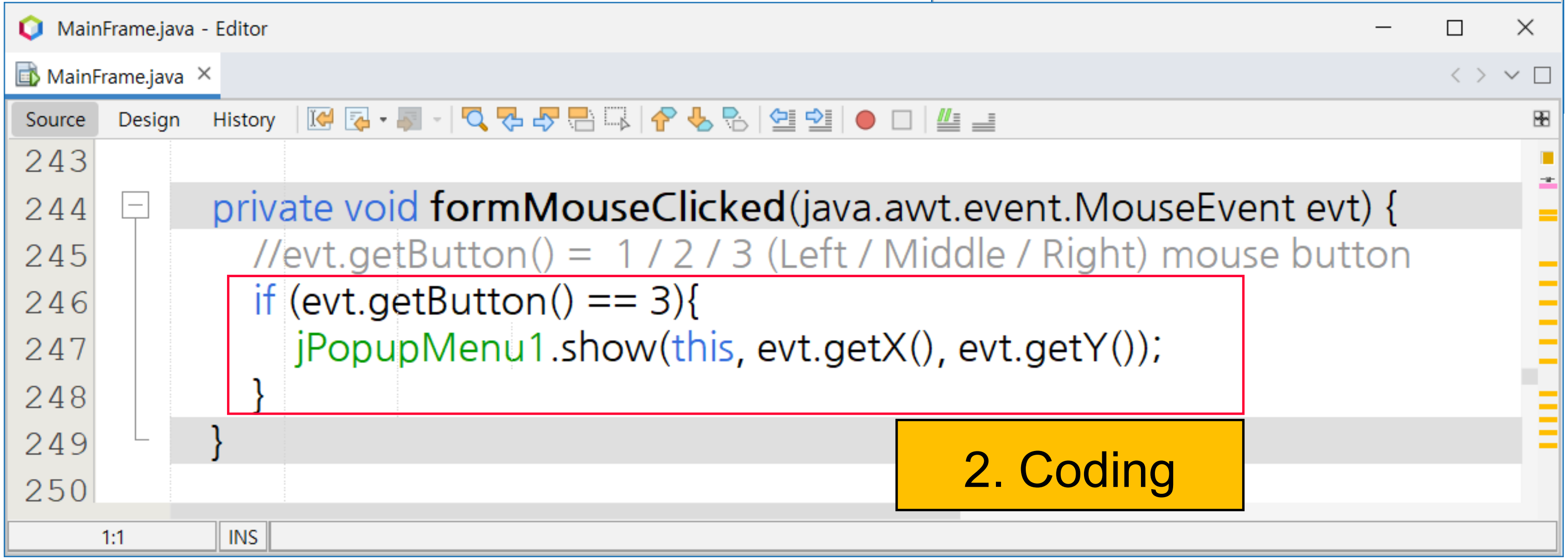


Practice 4 : Pop Up Menu (3)

- formMouseClicked Event Handler
  - Event
    - MouseClicked



keyTyped	<none>
mouseClicked	formMouseClicked
mouseDragged	<none>
mouseEntered	<none>
mouseExited	<none>
mouseMoved	<none>
mousePressed	<none>
mouseReleased	<none>
mouseWheelMoved	<none>



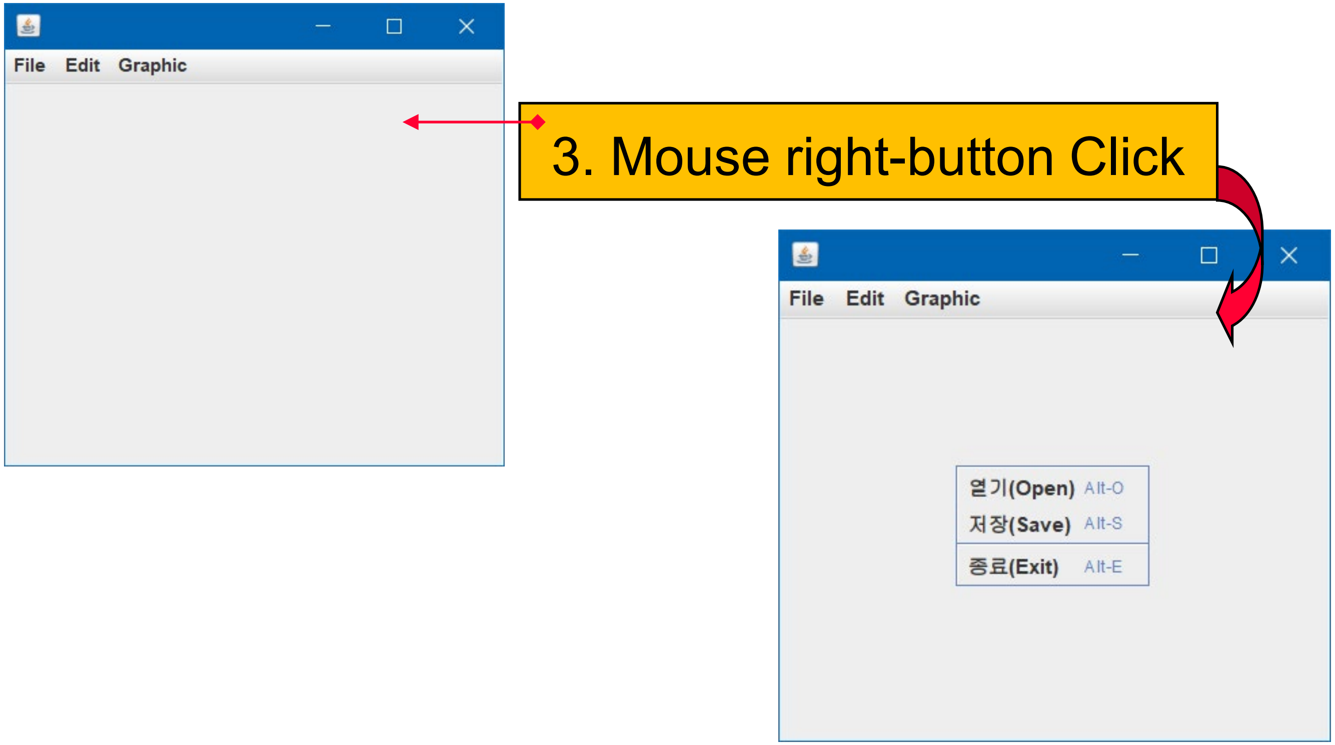
```
private void formMouseClicked(java.awt.event.MouseEvent evt) {
    //evt.getButton() =  1 / 2 / 3 (Left / Middle / Right) mouse button
    if (evt.getButton() == 3){
        jPopupMenu1.show(this, evt.getX(), evt.getY());
    }
}
```

2. Coding

Practice 4 : Pop Up Menu (4)

Run

- Mouse right-button Click
  - evt.getButton() = 1 / 2 / 3 (Left / Middle / Right) mouse button
- Popup Menu



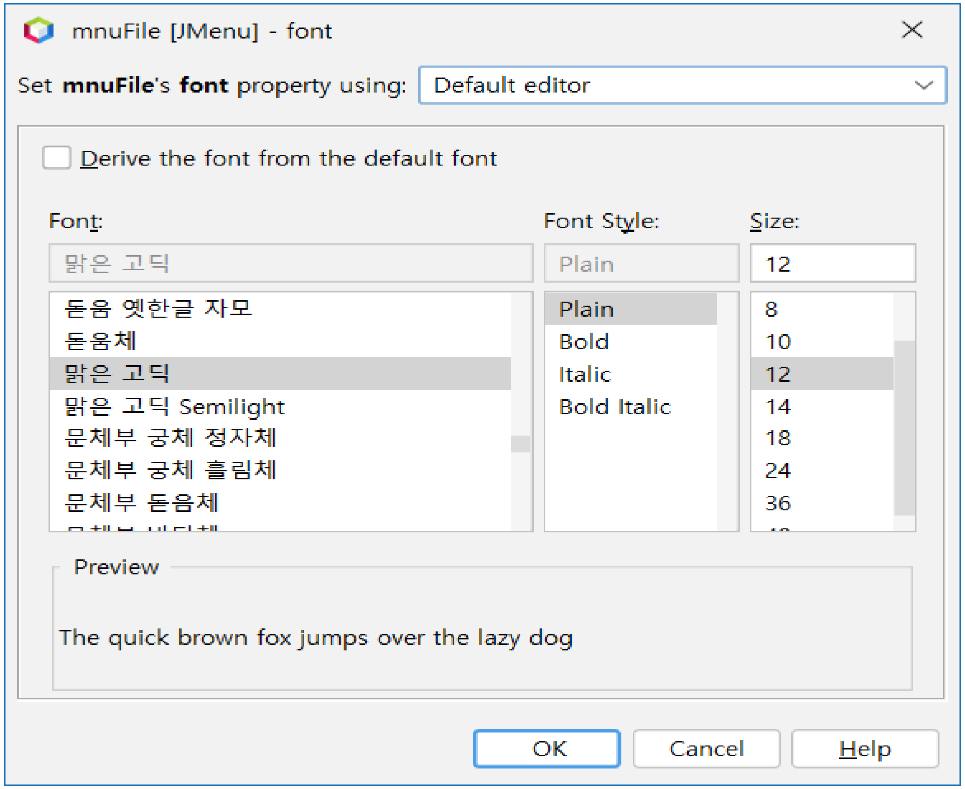
3. Mouse right-button Click

Font Class

Font 클래스

- Function
  - 글꼴의 속성이나 스타일 설정
- 구성

```
Font(String strFontName, int style, int size)
```



mnuFile [JMenu] - font

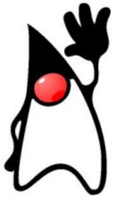
Set **mnuFile**'s **font** property using: Default editor

☐ Derive the font from the default font

Font:	Font Style:	Size:
맑은 고딕	Plain	12
돋움 옛한글 자모	Plain	8
돋움체	Bold	10
맑은 고딕	Italic	12
맑은 고딕 Semilight	Bold Italic	14
문체부 궁체 정자체		18
문체부 궁체 흘림체		24
문체부 돋음체		36

Preview

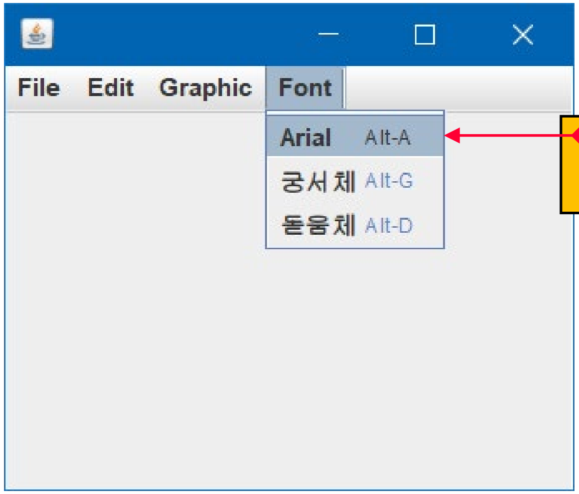
The quick brown fox jumps over the lazy dog



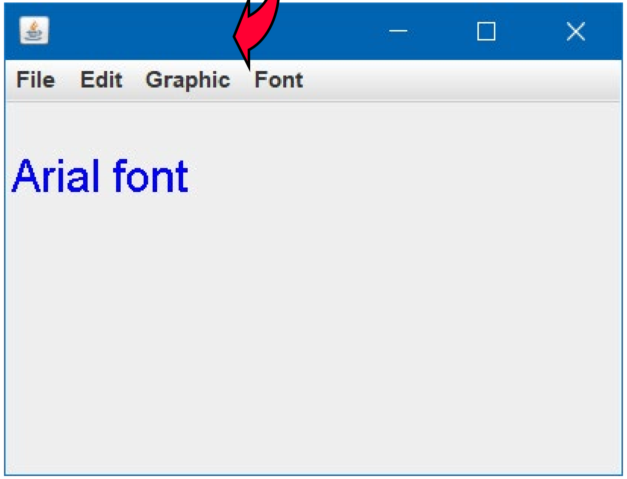
Practice 5 : Font

❖ Project Name: Menu_Source(Time: 30 min)

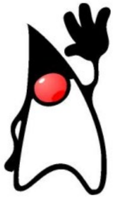
- [Font] menu implementation
  - Arial String Paint
  - 궁서체 String Paint
  - 돋움체 String Paint



Click



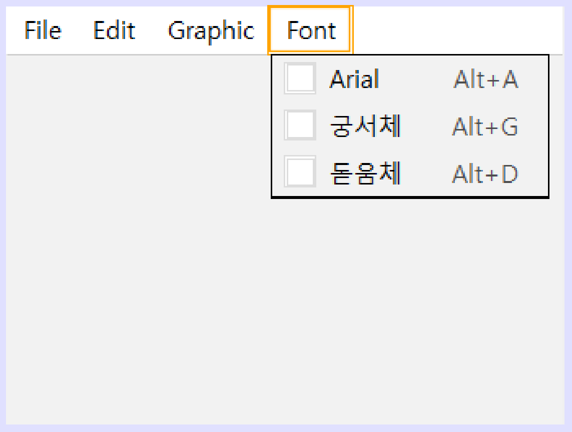
Arial font



Practice 5 : Font (1)

Control Layout & Property Setting

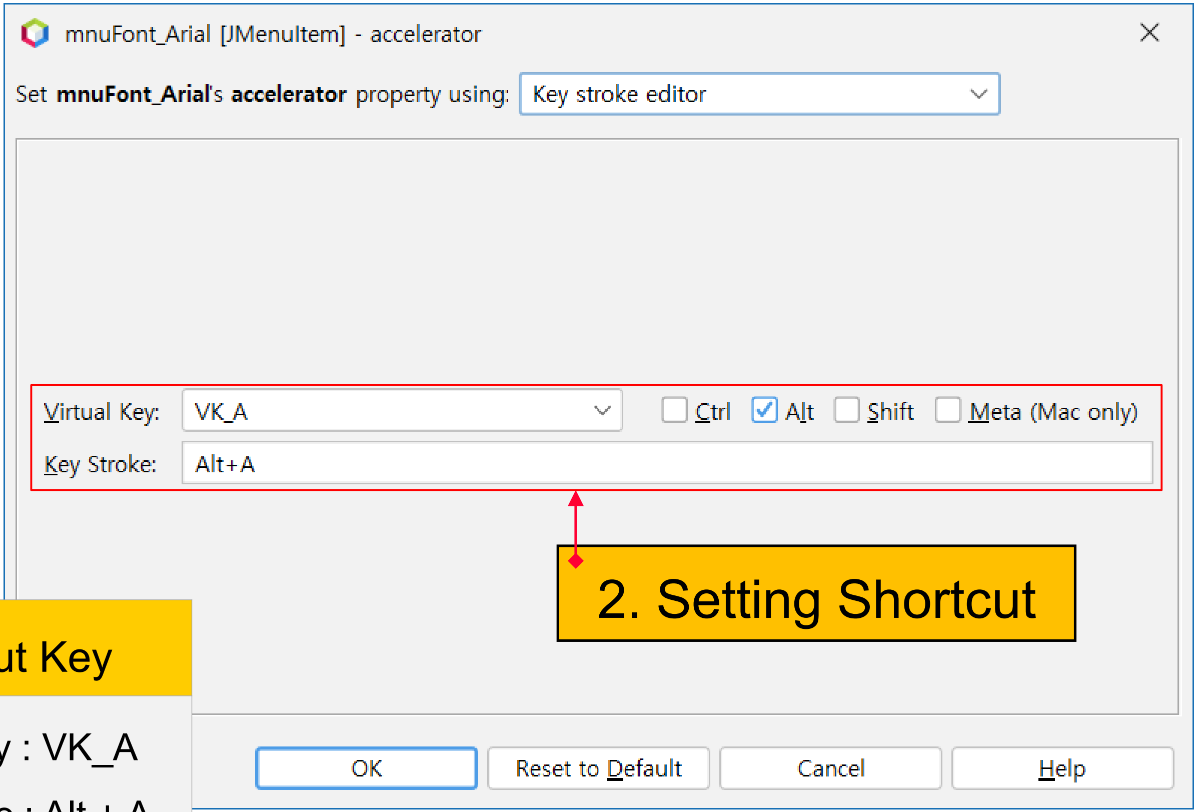
❖ Font Menu



1. UI Design

Control	Properties Setting
jMenu1	• Variable Name : mnuFont • Text : Font
jMenuItem1	• Variable Name : mnuFont_Arial • Text : Arial
jMenuItem2	• Variable Name : mnuFont_Gungseo • Text : 궁서체
jMenuItem3	• Variable Name : mnuFont_Dotum • Text : 돋움체

Practice 5 : Font (2)



mnuFont_Arial [JMenuItem] - accelerator

Set **mnuFont_Arial**'s **accelerator** property using: Key stroke editor

Virtual Key: VK_A ☐ Ctrl ☑ Alt ☐ Shift ☐ Meta (Mac only)

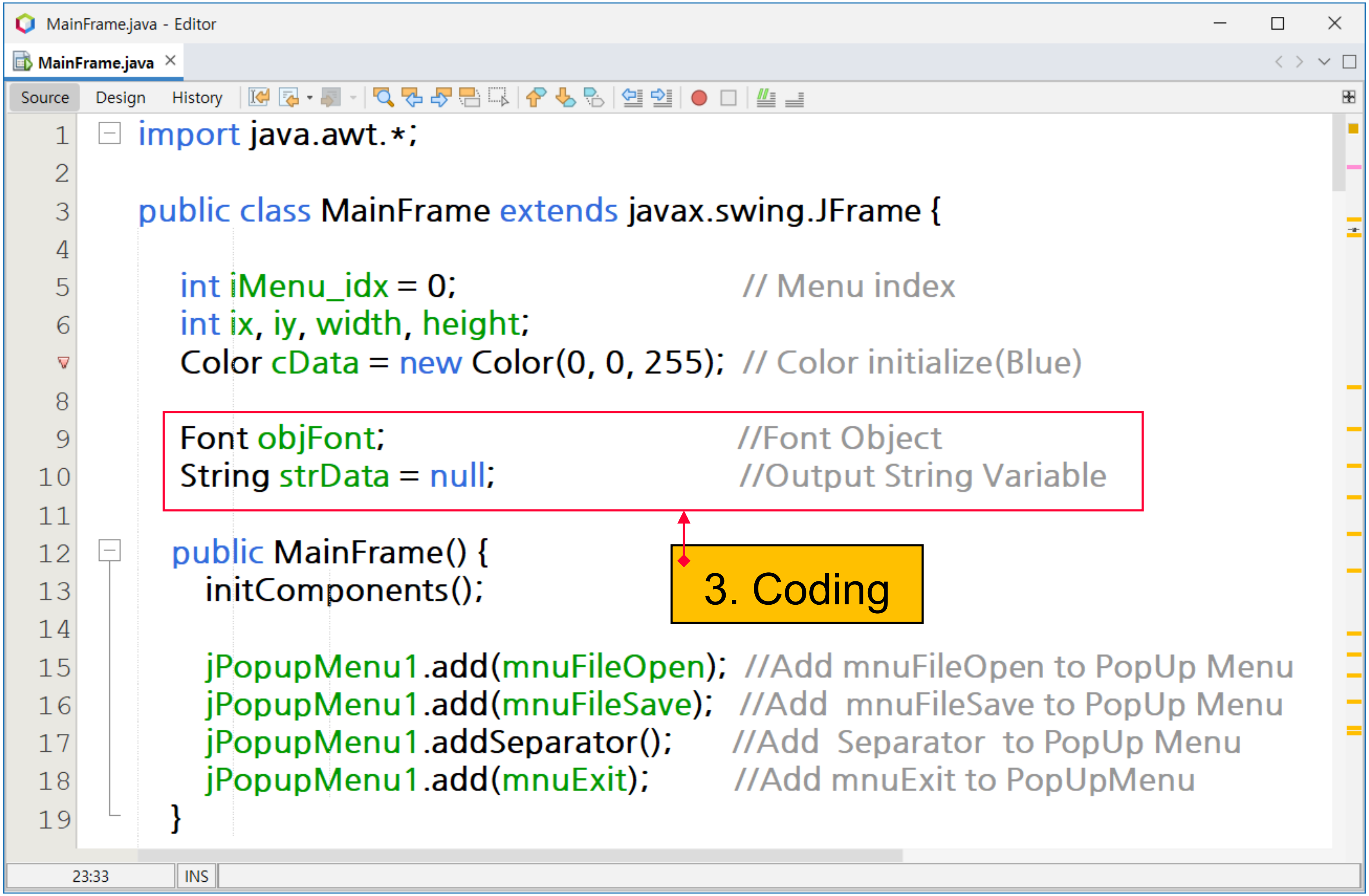
Key Stroke: Alt+A

2. Setting Shortcut

Menu	Shortcut Key
Arial	• Virtual Key : VK_A • Key Stroke : Alt + A
궁서체	• Virtual Key : VK_G • Key Stroke : Alt + G
돋움체	• Virtual Key : VK_D • Key Stroke : Alt + D

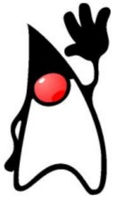
Practice 5 : Font (3)

❖ Font 구현을 위한 멤버 변수 선언



```
import java.awt.*;

public class MainFrame extends javax.swing.JFrame {

    int iMenu_idx = 0;                        // Menu index
    int ix, iy, width, height;
    Color cData = new Color(0, 0, 255);  // Color initialize(Blue)

    Font objFont;                              //Font Object
    String strData = null;                     //Output String Variable

    public MainFrame() {
        initComponents();

        jPopupMenu1.add(mnuFileOpen);  //Add mnuFileOpen to PopUp Menu
        jPopupMenu1.add(mnuFileSave);  //Add  mnuFileSave to PopUp Menu
        jPopupMenu1.addSeparator();      //Add  Separator  to PopUp Menu
        jPopupMenu1.add(mnuExit);        //Add mnuExit to PopUpMenu
    }
```

3. Coding

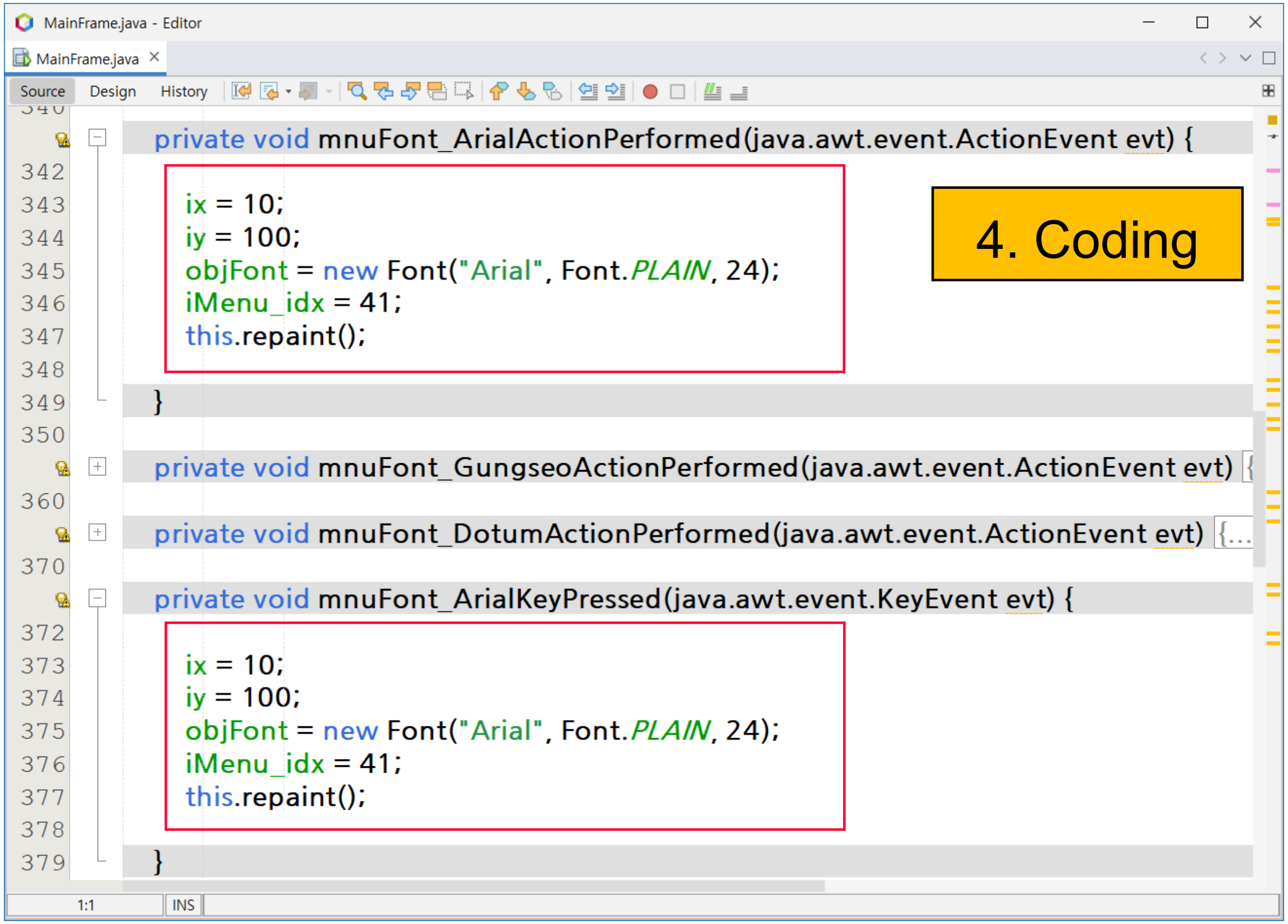


Practice 5 : Font (4)

❖ mnuFont_Arial Event Handler

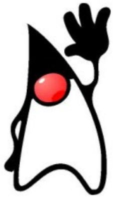
◆ Event

- ActionPerformed, KeyPressed



4. Coding

```
    private void mnuFont_ArialActionPerformed(java.awt.event.ActionEvent evt) {

        ix = 10;
        iy = 100;
        objFont = new Font("Arial", Font.PLAIN, 24);
        iMenu_idx = 41;
        this.repaint();

    }

    private void mnuFont_GungseoActionPerformed(java.awt.event.ActionEvent evt) {...

    private void mnuFont_DotumActionPerformed(java.awt.event.ActionEvent evt) {...

    private void mnuFont_ArialKeyPressed(java.awt.event.KeyEvent evt) {

        ix = 10;
        iy = 100;
        objFont = new Font("Arial", Font.PLAIN, 24);
        iMenu_idx = 41;
        this.repaint();

    }
```

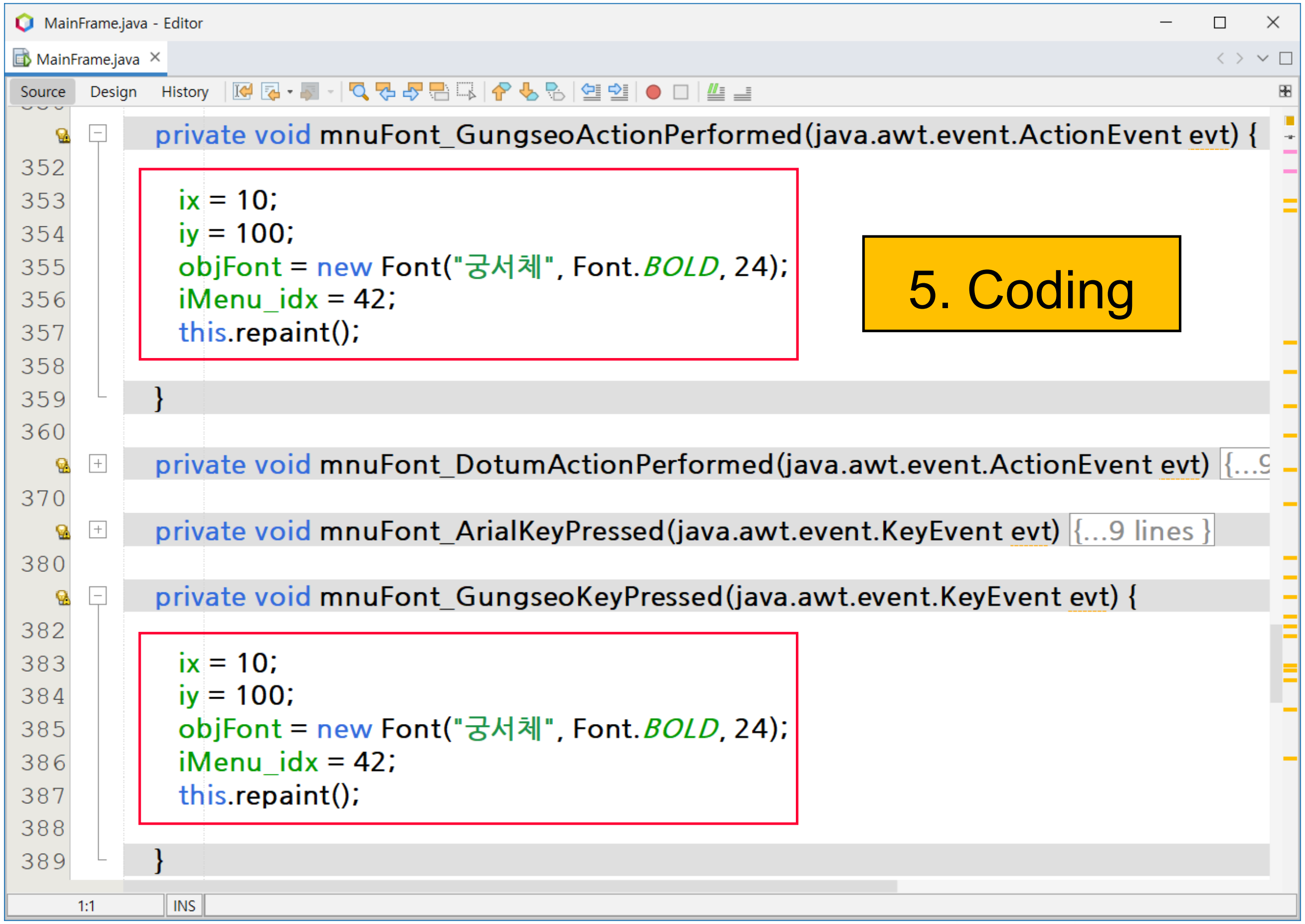


Practice 5 : Font (5)

mnuFont_Gungseo Event Handler

Event

- ActionPerformed, KeyPressed



```
    private void mnuFont_GungseoActionPerformed(java.awt.event.ActionEvent evt) {

        ix = 10;
        iy = 100;
        objFont = new Font("궁서체", Font.BOLD, 24);
        iMenu_idx = 42;
        this.repaint();

    }

    private void mnuFont_DotumActionPerformed(java.awt.event.ActionEvent evt) {...9

    private void mnuFont_ArialKeyPressed(java.awt.event.KeyEvent evt) {...9 lines }

    private void mnuFont_GungseoKeyPressed(java.awt.event.KeyEvent evt) {

        ix = 10;
        iy = 100;
        objFont = new Font("궁서체", Font.BOLD, 24);
        iMenu_idx = 42;
        this.repaint();

    }
```

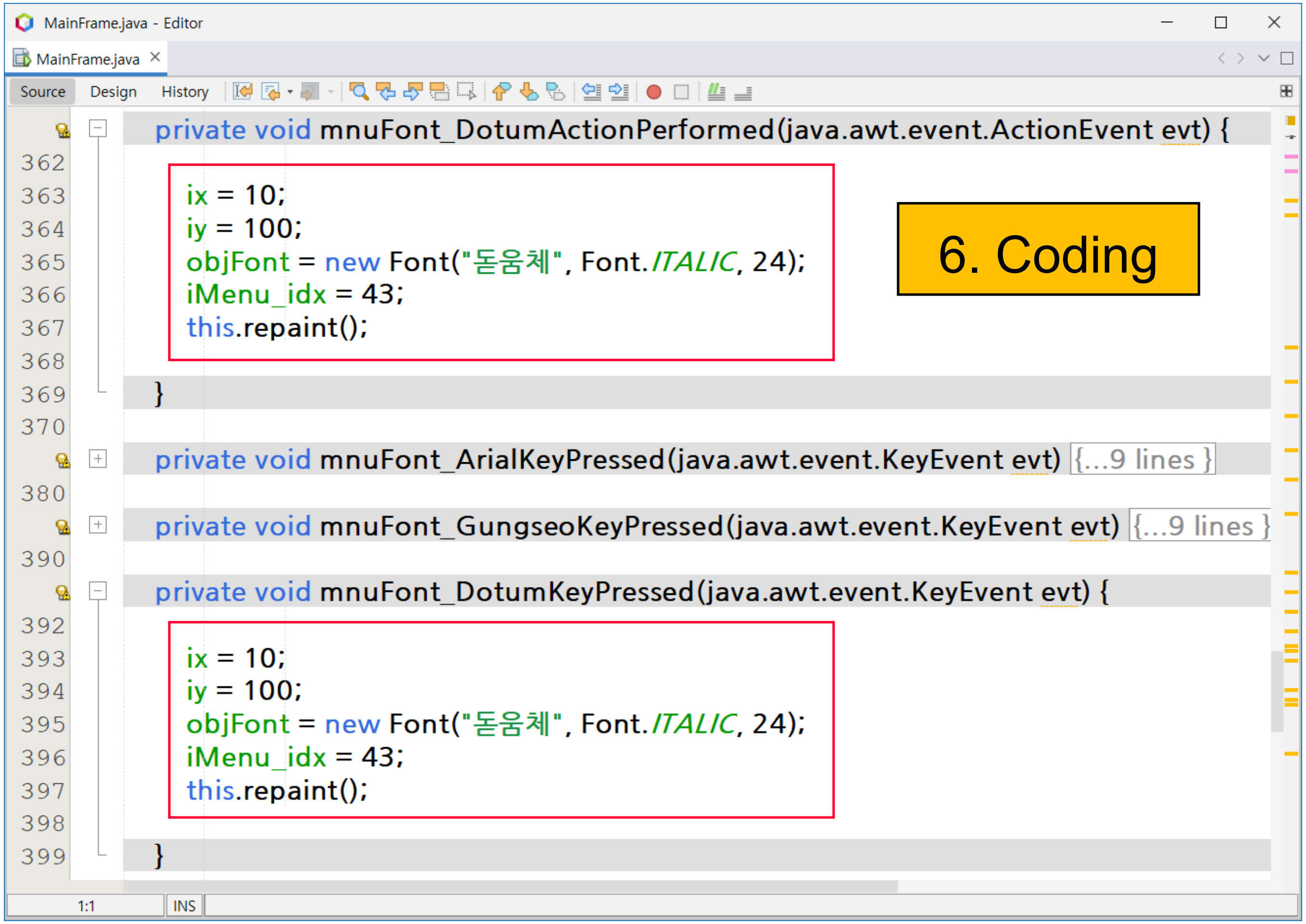
5. Coding

Practice 5 : Font (6)

❖ mnuFont_Dotum Event Handler

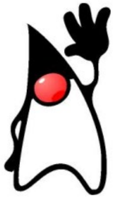
◆ Event

- ActionPerformed, KeyPressed



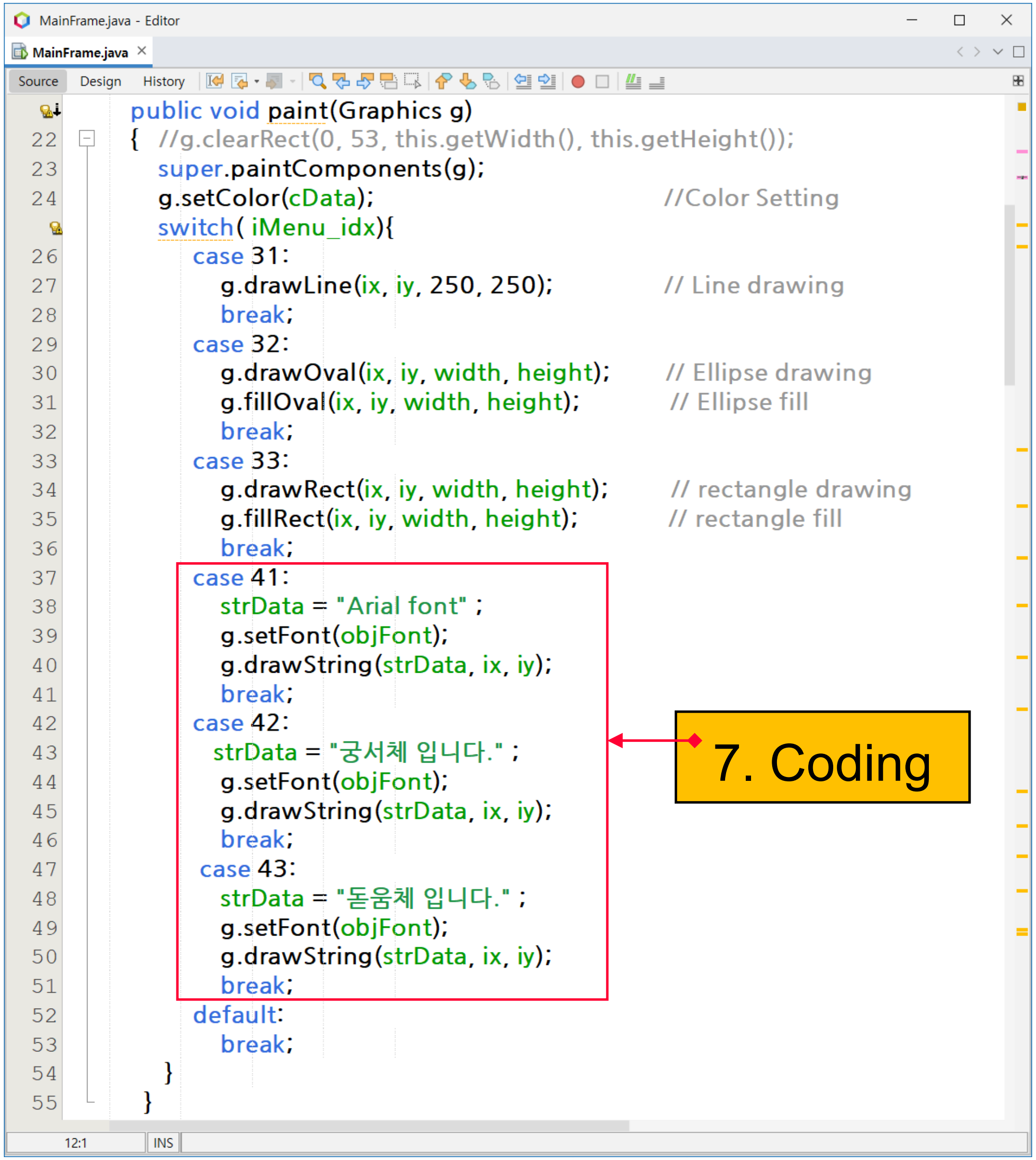
```java
private void mnuFont_DotumActionPerformed(java.awt.event.ActionEvent evt) {

    ix = 10;
    iy = 100;
    objFont = new Font("돋움체", Font.ITALIC, 24);
    iMenu_idx = 43;
    this.repaint();

}

private void mnuFont_ArialKeyPressed(java.awt.event.KeyEvent evt) {...9 lines }

private void mnuFont_GungseoKeyPressed(java.awt.event.KeyEvent evt) {...9 lines }

private void mnuFont_DotumKeyPressed(java.awt.event.KeyEvent evt) {

    ix = 10;
    iy = 100;
    objFont = new Font("돋움체", Font.ITALIC, 24);
    iMenu_idx = 43;
    this.repaint();

}
```

6. Coding



Practice 5 : Font (7)

❖ Paint() method

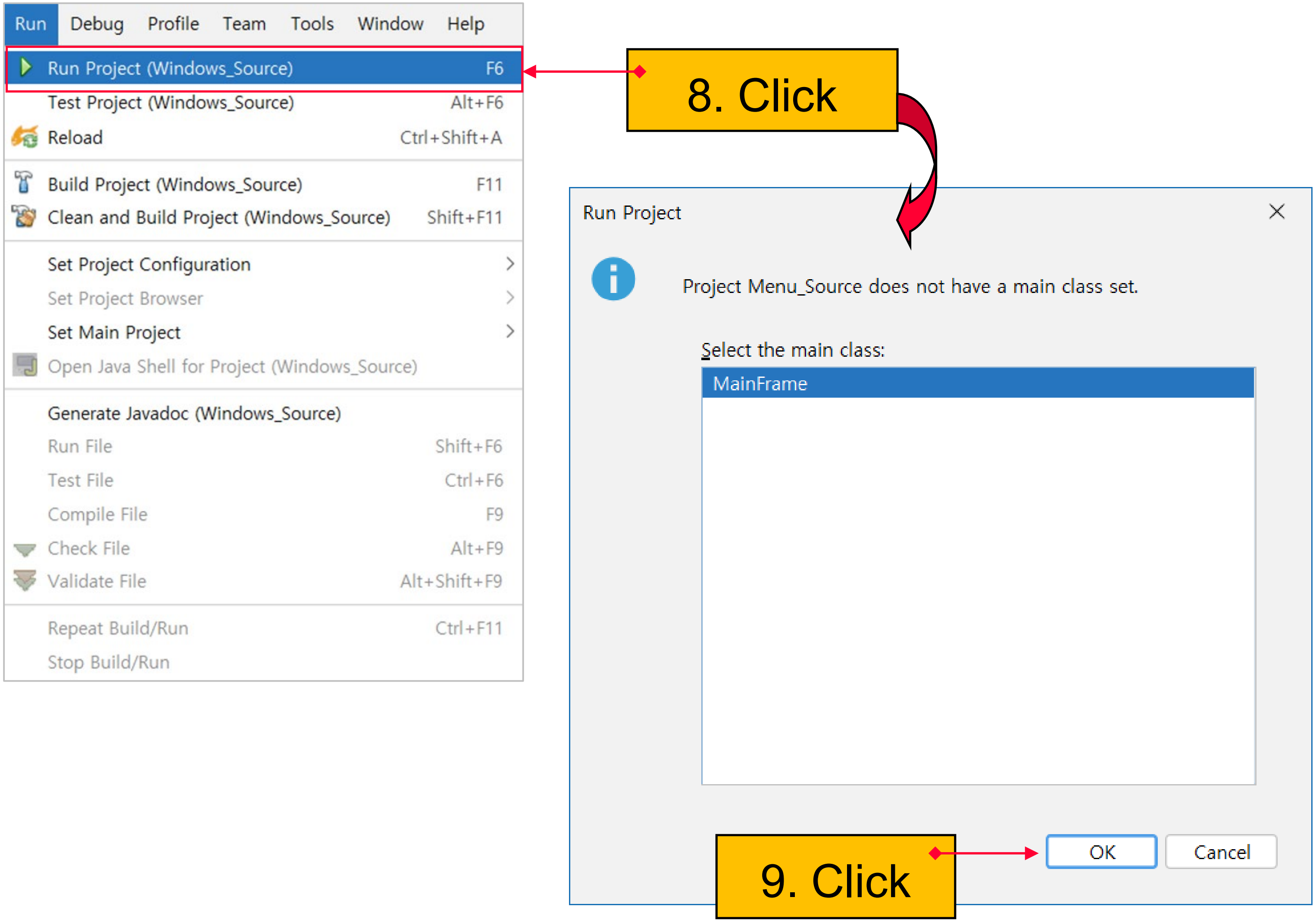


```
public void paint(Graphics g)
{  //g.clearRect(0, 53, this.getWidth(), this.getHeight());
    super.paintComponents(g);
    g.setColor(cData);                                   //Color Setting
    switch( iMenu_idx){
        case 31:
            g.drawLine(ix, iy, 250, 250);                // Line drawing
            break;
        case 32:
            g.drawOval(ix, iy, width, height);           // Ellipse drawing
            g.fillOval(ix, iy, width, height);           // Ellipse fill
            break;
        case 33:
            g.drawRect(ix, iy, width, height);           // rectangle drawing
            g.fillRect(ix, iy, width, height);           // rectangle fill
            break;
        case 41:
            strData = "Arial font" ;
            g.setFont(objFont);
            g.drawString(strData, ix, iy);
            break;
        case 42:
           strData = "궁서체 입니다." ;
            g.setFont(objFont);
            g.drawString(strData, ix, iy);
            break;
         case 43:
            strData = "돋움체 입니다." ;
            g.setFont(objFont);
            g.drawString(strData, ix, iy);
            break;
        default:
            break;
    }
  }
```

7. Coding

Practice 5 : Font (8)

Run


Run
Debug
Profile
Team
Tools
Window
Help
Run Project (Windows_Source) F6
Test Project (Windows_Source) Alt+F6
Reload Ctrl+Shift+A
Build Project (Windows_Source) F11
Clean and Build Project (Windows_Source) Shift+F11
Set Project Configuration
Set Project Browser
Set Main Project
Open Java Shell for Project (Windows_Source)
Generate Javadoc (Windows_Source)
Run File Shift+F6
Test File Ctrl+F6
Compile File F9
Check File Alt+F9
Validate File Alt+Shift+F9
Repeat Build/Run Ctrl+F11
Stop Build/Run
Run Project
OK
Cancel

8. Click

Project Menu_Source does not have a main class set.

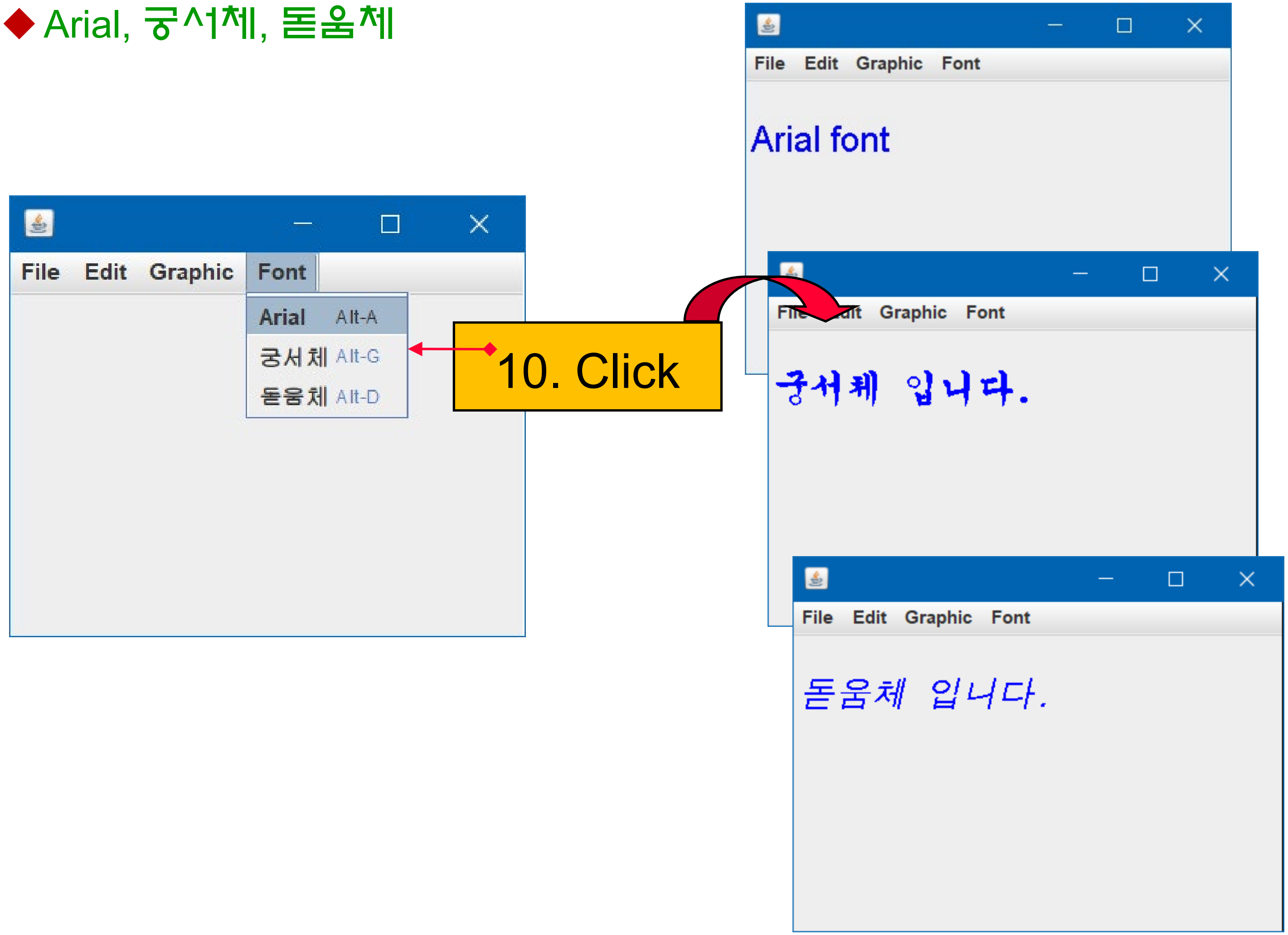
Select the main class:

MainFrame

9. Click

Practice 5 : Font (9)

- [Font] menu Click
  - Arial, 궁서체, 돋움체



10. Click

Arial font

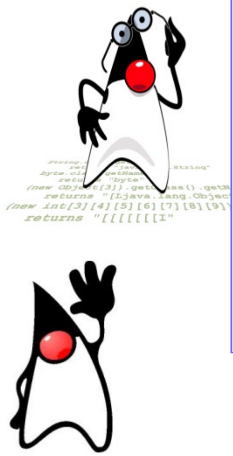
궁서체 입니다.

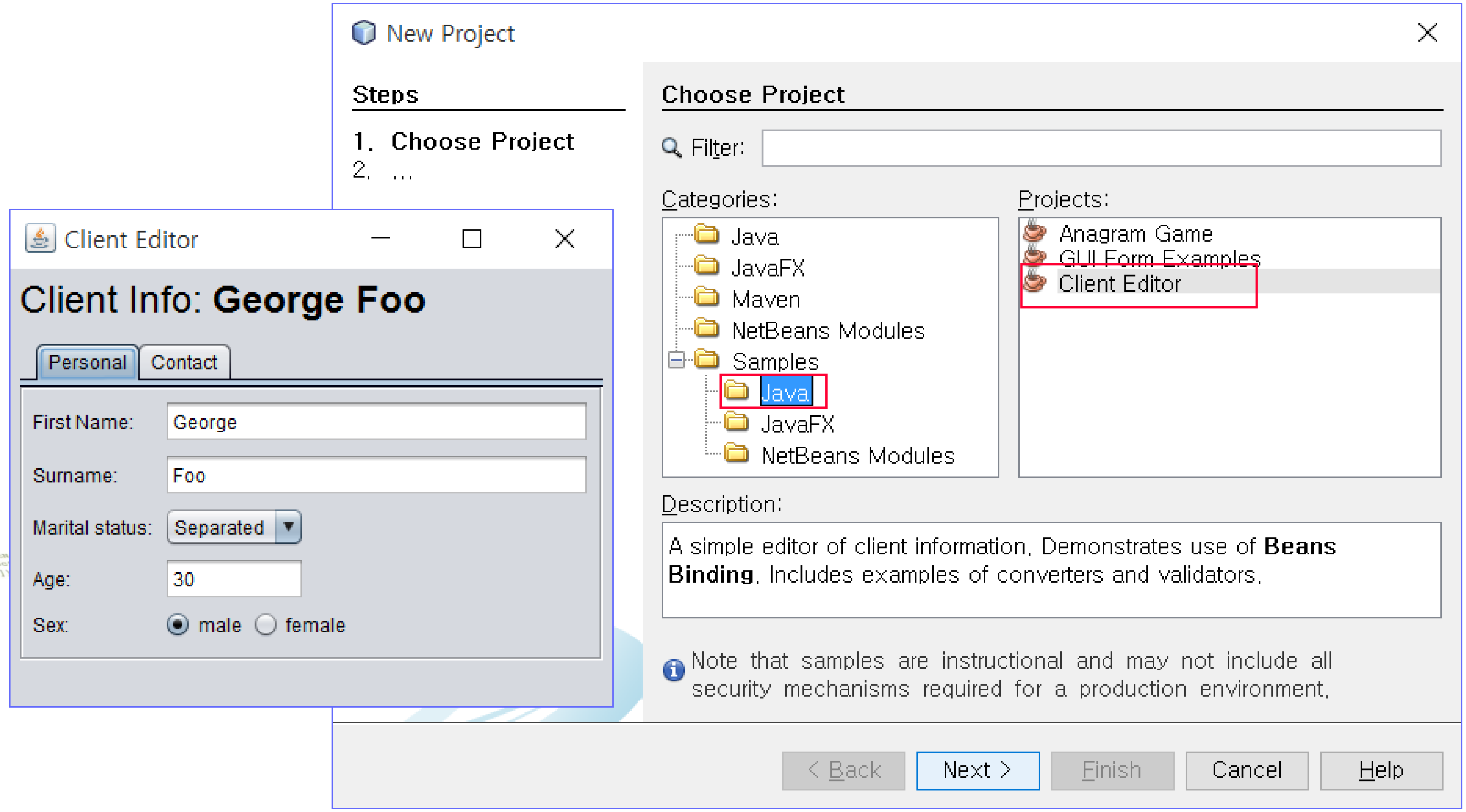
돋움체 입니다.

Homework # 1

❖ Project Name: Client Editor 소스 분석

- Client Editor 프로젝트
  - New Project > Samples > Java > Client Editor





New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- Maven
- NetBeans Modules
- Samples
  - Java
  - JavaFX
  - NetBeans Modules

Projects:

- Anagram Game
- GUI Form Examples
- Client Editor

Description:

A simple editor of client information. Demonstrates use of **Beans Binding**. Includes examples of converters and validators.

Note that samples are instructional and may not include all security mechanisms required for a production environment.

Client Editor

Client Info: **George Foo**

First Name: George

Surname: Foo

Marital status: Separated ▼

Age: 30

Sex: ● male ○ female

학습 요약

- Menu
- Menu Event Handler
- PopUp Menu
- Graphic
- Font