Student name: Lê Ngọc Anh Vũ

Student ID: 20236014

REPORT LAB 04

1 Swing components

Note: For the exercises in this lab (excluding the AIMS exercises), you will create a new Java project named `GUIProject`, and put all your source code in a package called "**`hust.soict.dsai.swing`**" (for DS & AI).

Note: From this section onwards, it is assumed that you are a DS-AI student, so your folder structure will contain the "**`dsai`**" package. If you are an HEDSPI or ICT student, you should replace the "**`dsai`**" string with "**`hedspi`**" or "**`globalict`**".

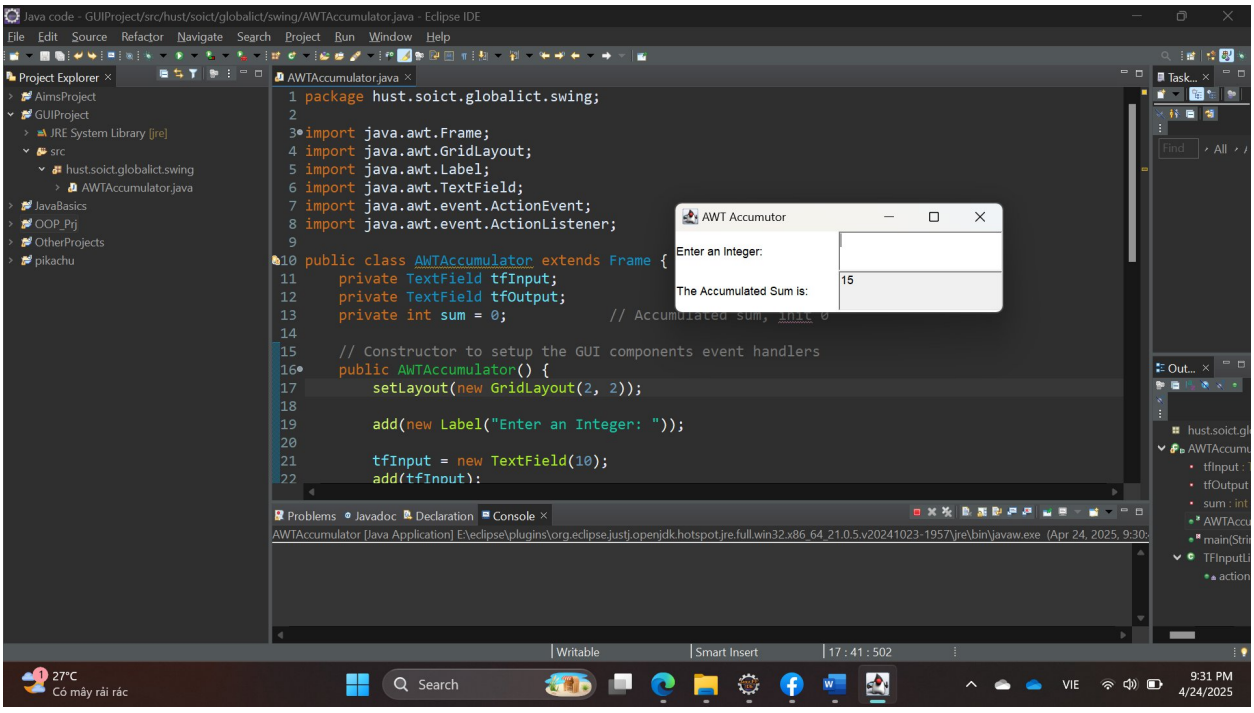
In this exercise, we revisit the elements of the Swing API and compare them with those of AWT by implementing the same mini-application using the two libraries. The application is an accumulator which accumulates the values entered by the user and displays the sum.

1.1 AWTAccumulator

1.1.1 Create class `AWTAccumulator` with the source code as below

```java
package hust.soict.globalict.swing;

import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class AWTAccumulator extends Frame {
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;            // Accumulated sum, init 0

    // Constructor to setup the GUI components event handlers
    public AWTAccumulator() {
        setLayout(new GridLayout(2, 2));

        add(new Label("Enter an Integer: "));

        tfInput = new TextField(10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label("The Accumulated Sum is: "));

        tfOutput = new TextField(10);
        tfOutput.setEditable(false);
        add(tfOutput);
```

```
30
31          setTitle("AWT Accumutor");
32          setSize(350, 120);
33          setVisible(true);
34      }
35      public static void main(String[] args) {
36          new AWTAccumulator();
37      }
38
39      public class TFInputListener implements ActionListener{
40
41          @Override
42          public void actionPerformed(ActionEvent e) {
43              int numberIn = Integer.parseInt(tfInput.getText());
44              sum += numberIn;
45              tfInput.setText("");
46              tfOutput.setText(sum + "");
47          }
48      }
49  }
```

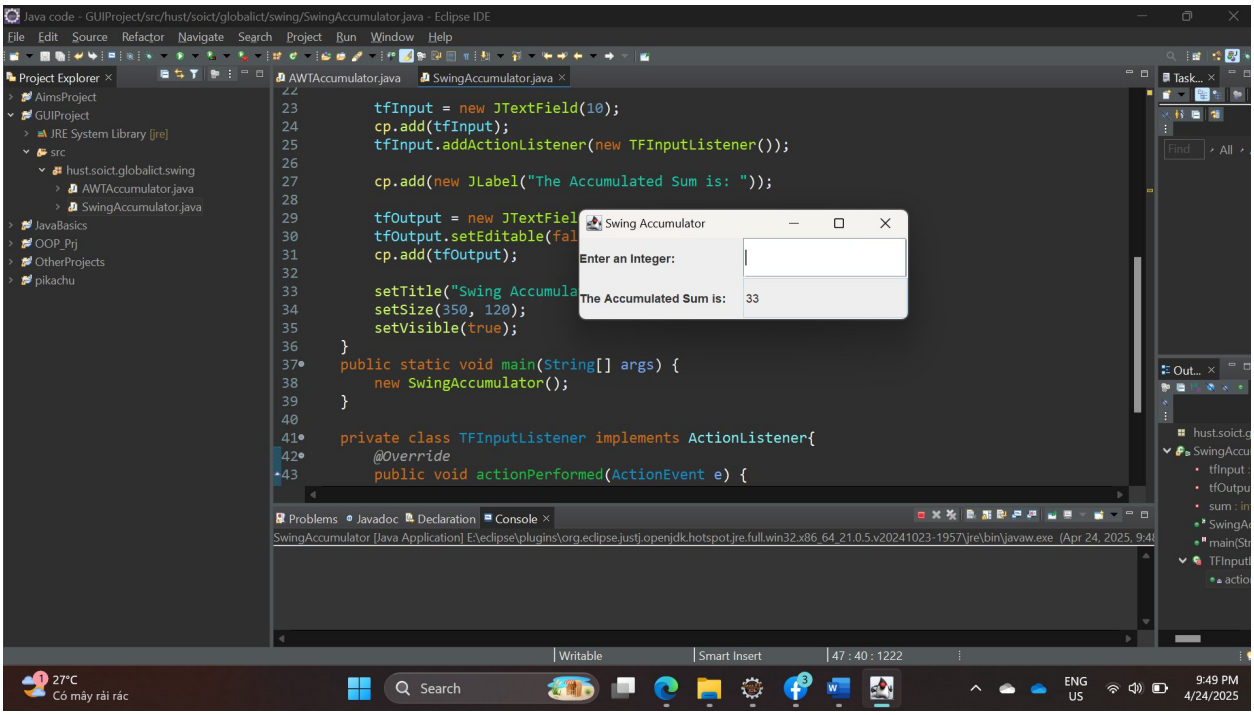


1.2 *SwingAccumulator*

1.2.1 Create class `SwingAccumulator` with the source code as below:

```java
package hust.soict.globalict.swing;

import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class SwingAccumulator extends JFrame {
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum = 0;

    public SwingAccumulator() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(2, 2));

        cp.add(new JLabel("Enter an Integer: "));

        tfInput = new JTextField(10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        cp.add(new JLabel("The Accumulated Sum is: "));

        tfOutput = new JTextField(10);
```

```java
        tfOutput.setEditable(false);
        cp.add(tfOutput);

        setTitle("Swing Accumulator");
        setSize(350, 120);
        setVisible(true);
    }
    public static void main(String[] args) {
        new SwingAccumulator();
    }

    private class TFInputListener implements ActionListener{
        @Override
        public void actionPerformed(ActionEvent e) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum + "");
        }
    }
}
```



2 Organizing Swing components with Layout Managers

2.2 Using JPanel as a secondary-level container to organize components:

2.2.1 Create class `NumberGrid`:

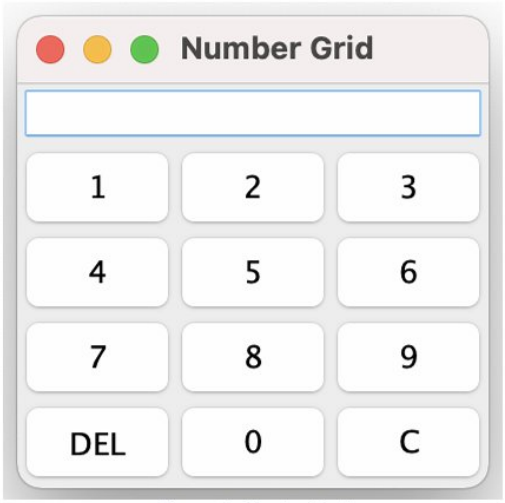


Figure 8. NumberGrid

```java
package hust.soict.globalict.swing;

import java.awt.BorderLayout;

public class NumberGrid extends JFrame {
    private JButton[] btnNumbers = new JButton[10];
    private JButton  btnDelete, btnReset;
    private JTextField tfDisplay;

    public NumberGrid() {

        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(4, 3));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Number Grid");
        setSize(200, 200);
        setVisible(true);
    }
```

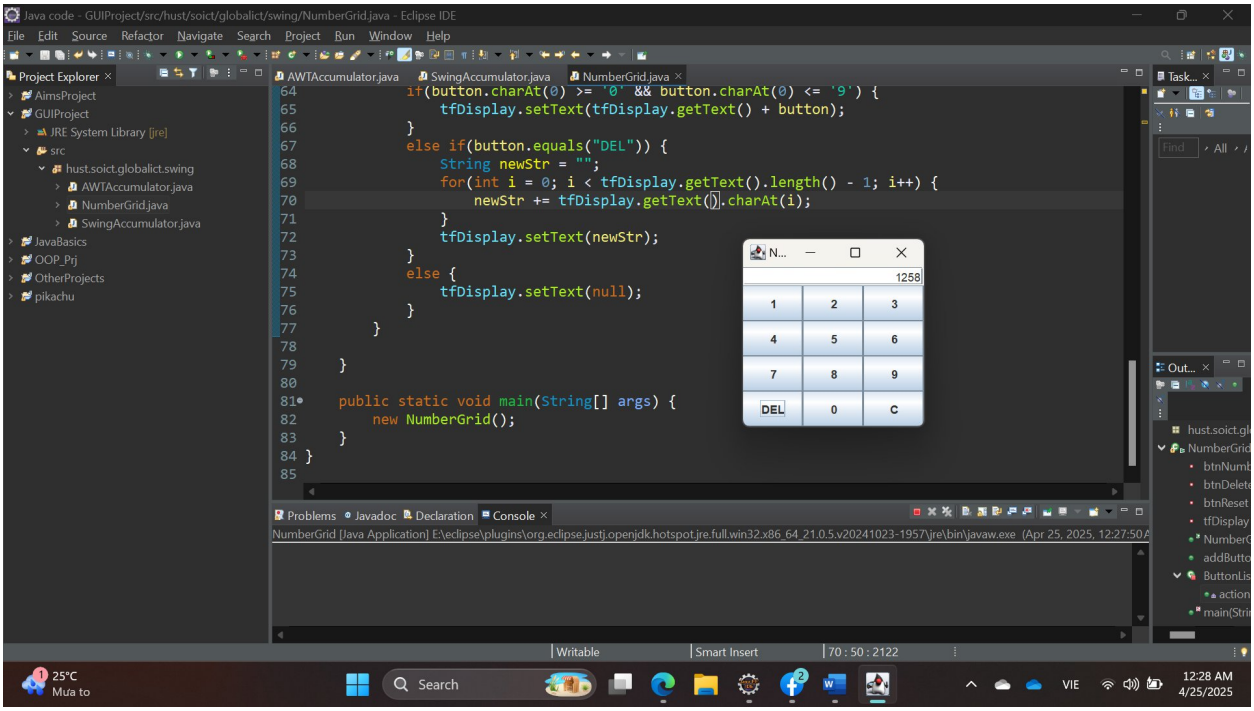
2.2.2 Adding buttons

```java
    public void addButtons(JPanel panelButtons) {
        ButtonListener btnListener = new ButtonListener();
        for(int i = 1; i <= 9; i++) {
            btnNumbers[i] = new JButton(""+i);
            panelButtons.add(btnNumbers[i]);
            btnNumbers[i].addActionListener(btnListener);
        }

        btnDelete = new JButton("DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

        btnNumbers[0] = new JButton("0");
        panelButtons.add(btnNumbers[0]);
        btnNumbers[0].addActionListener(btnListener);

        btnReset = new JButton("C");
        panelButtons.add(btnReset);
        btnReset.addActionListener(btnListener);
    }
```

2.2.3 Complete inner class `ButtonListener`

```java
    private class ButtonListener implements ActionListener{
        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            if(button.charAt(0) >= '0' && button.charAt(0) <= '9') {
                tfDisplay.setText(tfDisplay.getText() + button);
            }
            else if(button.equals("DEL")) {
                String newStr = "";
                for(int i = 0; i < tfDisplay.getText().length() - 1; i++) {
                    newStr += tfDisplay.getText().charAt(i);
                }
                tfDisplay.setText(newStr);
            }
            else {
                tfDisplay.setText(null);
            }
        }

    }
```

Testing:



3 Create a graphical user interface for AIMS with Swing

Additional Requirement: From this lab, we will split the AIMS system into two applications according to the role of the user: Customer and Store Manager

- Store Manager can view and update the store (i.e: Add Book, Add CD, Add DVD).
- Customers can view the store as the store manager, but they can choose to add media to the cart. Customers can also perform some functions related to the cart.

3.1 View Store Screen

For the view store screen, we will use the `BorderLayout`:
In the NORTH component, there will be the menu bar and the header
In the CENTER component, there will be a panel that uses the `GridLayout`, each cell is an item in the store.

3.1.1 Create the `StoreManagerScreen` class:

```
public class StoreManagerScreen{
    private Store store;
```

Figure 17. Declaration of StoreManagerScreen class

This will be our view store screen while logging in with the role Store Manager.
Declare one attribute in the `StoreManagerScreen` class: `Store store`. This is because we need information on the items in the store to display them.

```java
package hust.soict.globalict.aims.screen.manager;

import hust.soict.globalict.aims.store.Store;

public class StoreManagerScreen {
    private Store store;
}
```

3.1.2 The NORTH component:

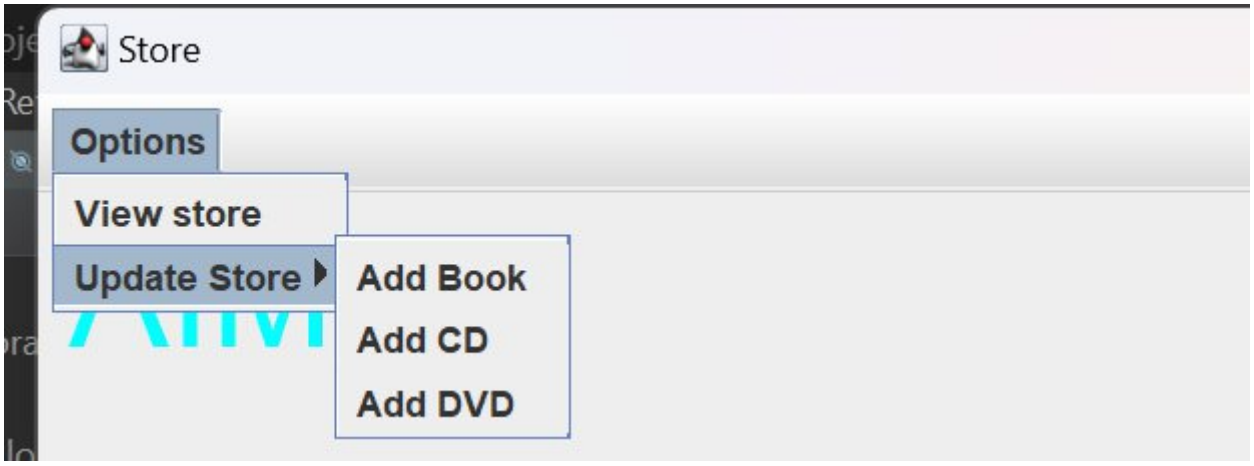
Create the method createNorth(), which will create our NORTH component:

```java
    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }
```

Create the method `createMenuBar()`:

```java
    JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");

        menu.add(new JMenuItem("View store"));

        JMenu smUpdateStore = new JMenu("Update Store");
        smUpdateStore.add(new JMenuItem("Add Book"));
        smUpdateStore.add(new JMenuItem("Add CD"));
        smUpdateStore.add(new JMenuItem("Add DVD"));
        menu.add(smUpdateStore);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }
```

The resulting menu bar will look something like this:



Create the method `createHeader()`:

```
JPanel createHeader() {
    JPanel header = new JPanel();
    header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

    JLabel title = new JLabel("AIMS");
    title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
    title.setForeground(Color.CYAN);

    header.add(Box.createRigidArea(new Dimension(10, 10)));
    header.add(title);
    header.add(Box.createHorizontalGlue());
    header.add(Box.createRigidArea(new Dimension(10, 10)));

    return header;
}
```

3.1.3 The CENTER component:

```
83•     JPanel createCenter() {
84          JPanel center = new JPanel();
85          center.setLayout(new GridLayout(3, 3, 2, 2));
86
87          ArrayList<Media> mediaInStore = store.getItemsInStore();
88          for(int i = 0; i < 9; i++) {
89              MediaStore cell = new MediaStore(mediaInStore.get(i));
90              center.add(cell);
91          }
92
93          return center;
94      }
```

3.1.4 The MediaStore class:

Here, since the `MediaStore` is a GUI element, it extends the `JPanel` class. It has one attribute: `Media media`.

```
1  package hust.soict.globalict.aims.screen.manager;
2
3• import java.awt.Color;
16
17 public class MediaStore extends JPanel {
18     private Media media;
19•    public MediaStore(Media media) {
20         this.media = media;
21         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
22
23         JLabel title = new JLabel(media.getTitle());
24         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 15));
25         title.setAlignmentX(CENTER_ALIGNMENT);
26
27         JLabel cost = new JLabel("" + media.getCost() + " $");
28         cost.setAlignmentX(CENTER_ALIGNMENT);
29
30         JPanel container = new JPanel();
31         container.setLayout(new FlowLayout(FlowLayout.CENTER));
32
33         if(media instanceof Playable) {
34             JButton playButton = new JButton("Play");
35             container.add(playButton);
36         }
```

```
38          this.add(Box.createVerticalGlue());
39          this.add(title);
40          this.add(cost);
41          this.add(Box.createHorizontalGlue());
42          this.add(container);
43
44          this.setBorder(BorderFactory.createLineBorder(Color.black));
45      }
46  }
```

3.1.5 Putting it all together:

Finally, we have all the component methods to use in the constructor of `StoreScreen`:

```
public StoreManagerScreen(Store store) {
    this.store = store;

    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());
    cp.add(createNorth(), BorderLayout.NORTH);
    cp.add(createCenter(), BorderLayout.CENTER);

    setTitle("Store");
    setSize(1024, 768);
    setLocationRelativeTo(null);
    setVisible(true);
}
```

3.2 Update Store Screen

We have successfully set up all the components for our store, but they are just static – buttons and menu items don't respond when clicked. Now, it's your task to implement the handling of the event when the user interacts with:

- The menu bar:
  - When a user clicks on one of the items of the "Update Store" menu on the menu bar (such as "Add Book", "Add CD", or "Add DVD"), the application should switch to an appropriate new screen for the user to add the new item. This screen should have the same menu bar as the View Store Screen, so the user can go back to view the store.
  - When the user clicks on "View Store", the application should switch to the View Store Screen.
- The buttons on MediaHome:
  - When the user clicks on the "Play" button, the Media should be played in a dialog window. You can use JDialog here.

Note:

- For simplicity:
  - You only need to do the "Add item to store" screen, the "Remove item from store" is omitted. As shown above, there is no option of "Remove item from store" in the "Update Store" menu.
  - In the "Add item to store" Screen, you don't need to do data type validation yet.

I create three more classes: AddDigitalVideoDiscToStoreScreen, AddCompactDiscToStoreScreen and AddBookToStoreScreen

All of them extend from class AddItemToStoreScreen

For AddItemToStoreScreen class:

```java
package hust.soict.globalict.aims.screen.manager;

import java.awt.BorderLayout;

public abstract class AddItemToStoreScreen extends JFrame {
    private JLabel titleLabel, categoryLabel, costLabel;
    private JTextField titleInput, categoryInput, costInput;
    private JPanel main;
    private JButton buttonFinish = new JButton();

    public AddItemToStoreScreen() {
        main = new JPanel();
        main.setLayout(new GridLayout(0, 1, 5, 5));

        titleLabel = new JLabel("Title*: ");
        titleInput = new JTextField(30);
        titleInput.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(titleInput.getText().equals("") || costInput.getText().equals("")) {
                    buttonFinish.setEnabled(false);
                }
                else {
                    buttonFinish.setEnabled(true);
                }
            }
        });
        JPanel title = new JPanel(new GridLayout(1, 2, 10, 0));
        title.add(titleLabel);
        title.add(titleInput);
        main.add(title);

        categoryLabel = new JLabel("Category: ");
        categoryInput = new JTextField(20);
        JPanel category = new JPanel(new GridLayout(1, 2, 10, 0));
        category.add(categoryLabel);
        category.add(categoryInput);
        main.add(category);

        costLabel = new JLabel("Cost*: ");
        costInput = new JTextField(20);
        costInput.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(titleInput.getText().equals("") || costInput.getText().equals("")) {
                    buttonFinish.setEnabled(false);
                }
                else {
                    buttonFinish.setEnabled(true);
                }
            }
        });
        JPanel cost = new JPanel(new GridLayout(1, 2, 10, 0));
        cost.add(costLabel);
        cost.add(costInput);
        main.add(cost);
    }
```

```
74•     public void add(JPanel panel) {
75          main.add(panel);
76      }
77
78•     public JPanel getMainPanel() {
79          return main;
80      }
81
82•     public JPanel request() {
83          JPanel panel = new JPanel();
84          panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));
85
86          JLabel rqt = new JLabel("You have to fill all (*) fields");
87          rqt.setFont(new Font(rqt.getFont().getName(), Font.PLAIN, 15));
88          rqt.setForeground(Color.red);
89
90          panel.add(rqt);
91          return panel;
92      }
93
94•     public JPanel createNorth(String str) {
95          JPanel panelNorth = new JPanel(new BorderLayout());
96          panelNorth.add(createMenuBar(), BorderLayout.NORTH);
97          panelNorth.add(createHeader(str), BorderLayout.CENTER);
98          return panelNorth;
99      }
100
101•     public JPanel createHeader(String str) {
102         JPanel header = new JPanel();
```

```
104         JLabel title = new JLabel(str);
105         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
106         title.setForeground(Color.CYAN);
107
108         header.add(Box.createRigidArea(new Dimension(10, 10)));
109         header.add(title, BorderLayout.CENTER);
110         header.add(Box.createHorizontalGlue());
111         header.add(Box.createRigidArea(new Dimension(10, 10)));
112
113         return header;
114     }
115
116•     public JPanel createSouth() {
117         JPanel panel = new JPanel(new BorderLayout());
118         panel.add(request(), BorderLayout.WEST);
119         buttonFinish = createButton();
120         buttonFinish.setEnabled(false);
121         panel.add(buttonFinish, BorderLayout.EAST);
122         panel.setBorder(new EmptyBorder(0, 5, 5, 5));
123         return panel;
124     }
125
126•     public String getTitle() {
127         return titleInput.getText();
128     }
129
130•     public String getCategory() {
131         return categoryInput.getText();
132     }
```

```
134•     public String getCost() {
135          return costInput.getText();
136      }
137
138•     public void setTitle(String str) {
139          titleInput.setText(str);
140      }
141
142•     public void setCategory(String str) {
143          categoryInput.setText(str);
144      }
145
146•     public void setCost(String str) {
147          costInput.setText(str);
148      }
149
150      public abstract JButton createButton();
151
152      public abstract JMenuBar createMenuBar();
153  }
```

For AddDigitalVideoDiscToStoreScreen class:

```java
package hust.soict.globalict.aims.screen.manager;

import java.awt.BorderLayout;

public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
    private JLabel lengthLabel, directorLabel;
    private JTextField lengthInput, directorInput;
    private JFrame frame = new JFrame();

    public AddDigitalVideoDiscToStoreScreen() {
        super();
        lengthLabel = new JLabel("Length: ");
        lengthInput = new JTextField(10);
        JPanel length = new JPanel(new GridLayout(1, 2, 10, 0));
        length.add(lengthLabel);
        length.add(lengthInput);
        add(length);

        directorLabel = new JLabel("Director: ");
        directorInput = new JTextField(20);
        JPanel director = new JPanel(new GridLayout(1, 2, 10, 0));
        director.add(directorLabel);
        director.add(directorInput);
        add(director);

        JPanel panel = new JPanel(new BorderLayout());

        panel.add(createNorth(), BorderLayout.NORTH);
        getMainPanel().setBorder(new EmptyBorder(10, 250, 250, 250));
```

```java
        panel.add(getMainPanel(), BorderLayout.CENTER);
        panel.add(createSouth(), BorderLayout.SOUTH);

        frame.add(panel);
        frame.setTitle("Adding DVD");
        frame.setSize(800, 600);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    public JPanel createNorth() {
        JPanel panelNorth = new JPanel(new BorderLayout());
        panelNorth.add(createMenuBar(), BorderLayout.NORTH);
        panelNorth.add(createHeader("Digital Video Disc"), BorderLayout.CENTER);
        return panelNorth;
    }

    @Override
    public JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");

        JMenuItem viewStore = new JMenuItem("View store");
        viewStore.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new StoreManagerScreen(Main.store);
            }
        });
```

```
        menu.add(viewStore);

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBook = new JMenuItem("Add Book");
        addBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddBookToStoreScreen();
            }
        });
        smUpdateStore.add(addBook);

        JMenuItem addCD = new JMenuItem("Add CD");
        addCD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddCompactDiscToStoreScreen();
            }
        });
        smUpdateStore.add(addCD);

        JMenuItem addDVD = new JMenuItem("Add DVD");
        addDVD.setEnabled(false);
        smUpdateStore.add(addDVD);
        menu.add(smUpdateStore);

        JMenuBar menuBar = new JMenuBar();
```

```
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }

    @Override
    public JButton createButton() {
        JButton btnFinish = new JButton("Finish");
        btnFinish.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String length = lengthInput.getText();
                if(length.equals("")) {
                    length = "0";
                }
                    Main.store.addMedia(new DigitalVideoDisc(0, getTitle(),
                            getCategory(), directorInput.getText(),
                            Integer.parseInt(length),
                            Float.parseFloat(getCost())));
                    btnFinish.setEnabled(false);
                    setTitle("");
                    setCategory("");
                    directorInput.setText("");
                    lengthInput.setText("");
                    setCost("");
            }
        });
        return btnFinish;
```

For AddCompactDiscToStoreScreen class:

```
package hust.soict.globalict.aims.screen.manager;

import java.awt.BorderLayout;

public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
    private JLabel artistLabel,lengthLabel, directorLabel;
    private JTextField artistInput, lengthInput, directorInput;
    private JFrame frame = new JFrame();

    public AddCompactDiscToStoreScreen() {
        super();
        artistLabel = new JLabel("Artist: ");
        artistInput = new JTextField(20);
        JPanel artist = new JPanel(new GridLayout(1, 2, 10, 0));
        artist.add(artistLabel);
        artist.add(artistInput);
        add(artist);
```

```
        lengthLabel = new JLabel("Length: ");
        lengthInput = new JTextField(10);
        JPanel length = new JPanel(new GridLayout(1, 2, 10, 0));
        length.add(lengthLabel);
        length.add(lengthInput);
        add(length);

        directorLabel = new JLabel("Director: ");
        directorInput = new JTextField(20);
        JPanel director = new JPanel(new GridLayout(1, 2, 10, 0));
        director.add(directorLabel);
        director.add(directorInput);
        add(director);

        JPanel panel = new JPanel(new BorderLayout());

        panel.add(createNorth(), BorderLayout.NORTH);
        getMainPanel().setBorder(new EmptyBorder(10, 250, 200, 250));
        panel.add(getMainPanel(), BorderLayout.CENTER);
        panel.add(createSouth(), BorderLayout.SOUTH);

        frame.add(panel);
        frame.setTitle("Adding CD");
        frame.setSize(800, 600);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    public JPanel createNorth() {
```

```
        JPanel panelNorth = new JPanel(new BorderLayout());
        panelNorth.add(createMenuBar(), BorderLayout.NORTH);
        panelNorth.add(createHeader("Compact Disc"), BorderLayout.CENTER);
        return panelNorth;
    }

    @Override
    public JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");

        JMenuItem viewStore = new JMenuItem("View store");
        viewStore.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new StoreManagerScreen(Main.store);
            }
        });
        menu.add(viewStore);

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBook = new JMenuItem("Add Book");
        addBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddBookToStoreScreen();
            }
        });
```

```
        smUpdateStore.add(addBook);

        JMenuItem addCD = new JMenuItem("Add CD");
        addCD.setEnabled(false);
        smUpdateStore.add(addCD);

        JMenuItem addDVD = new JMenuItem("Add DVD");
        addDVD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddDigitalVideoDiscToStoreScreen();
            }
        });
```

```
        smUpdateStore.add(addDVD);
        menu.add(smUpdateStore);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }

    @Override
    public JButton createButton() {
        JButton btnFinish = new JButton("Finish");
        btnFinish.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String length = lengthInput.getText();
                if(length.equals("")) {
                    length = "0";
                }
                    Main.store.addMedia(new CompactDisc(0, getTitle(),
                            getCategory(), artistInput.getText(), directorInput.getText(),
                            Integer.parseInt(length),
                            Float.parseFloat(getCost())));
                btnFinish.setEnabled(false);
                setTitle("");
                setCategory("");
                artistInput.setText("");
                directorInput.setText("");
```

```
                lengthInput.setText("");
                setCost("");
            }
        });
        return btnFinish;
    }
}
```

For AddBookToStoreScreen class:

```
package hust.soict.globalict.aims.screen.manager;

import java.awt.BorderLayout;

public class AddBookToStoreScreen extends AddItemToStoreScreen {
    private JFrame frame = new JFrame();

    public AddBookToStoreScreen() {
        super();
        JPanel panel = new JPanel(new BorderLayout());

        panel.add(createNorth("Book"), BorderLayout.NORTH);
        getMainPanel().setBorder(new EmptyBorder(10, 200, 200, 200));
        panel.add(getMainPanel(), BorderLayout.CENTER);
        panel.add(createSouth(), BorderLayout.SOUTH);

        frame.add(panel);
        frame.setTitle("Adding book");
        frame.setSize(700, 500);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    @Override
    public JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");
```

```
        JMenuItem viewStore = new JMenuItem("View store");
        viewStore.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new StoreManagerScreen(Main.store);
            }
        });
        menu.add(viewStore);

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBook = new JMenuItem("Add Book");
        addBook.setEnabled(false);
        smUpdateStore.add(addBook);

        JMenuItem addCD = new JMenuItem("Add CD");
        addCD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddCompactDiscToStoreScreen();
            }
        });
        smUpdateStore.add(addCD);

        JMenuItem addDVD = new JMenuItem("Add DVD");
        addDVD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```

```
                frame.dispose();
                new AddDigitalVideoDiscToStoreScreen();
            }
        });
        smUpdateStore.add(addDVD);
        menu.add(smUpdateStore);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }
```

```
    @Override
    public JButton createButton() {
        JButton btnFinish = new JButton("Finish");
        btnFinish.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Main.store.addMedia(new Book(0, getTitle(),
                        getCategory(), Float.parseFloat(getCost())));
                btnFinish.setEnabled(false);
                setTitle("");
                setCategory("");
                setCost("");
            }
        });
        return btnFinish;
    }
}
```

Modifying MediaStore class:

```
        if(media instanceof Playable) {
            JButton playButton = new JButton("Play");
            playButton.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                     JDialog d = new JDialog();
                     d.setTitle("Playing media");
                     d.setLayout(new BorderLayout());
                     JLabel l = new JLabel(media.getTitle() + " is playing at the present");
                     d.add(l, BorderLayout.CENTER);
                     d.setSize(300, 150);
                     d.setLocationRelativeTo(null);
                     d.setVisible(true);
                }
            });
            container.add(playButton);
        }
```

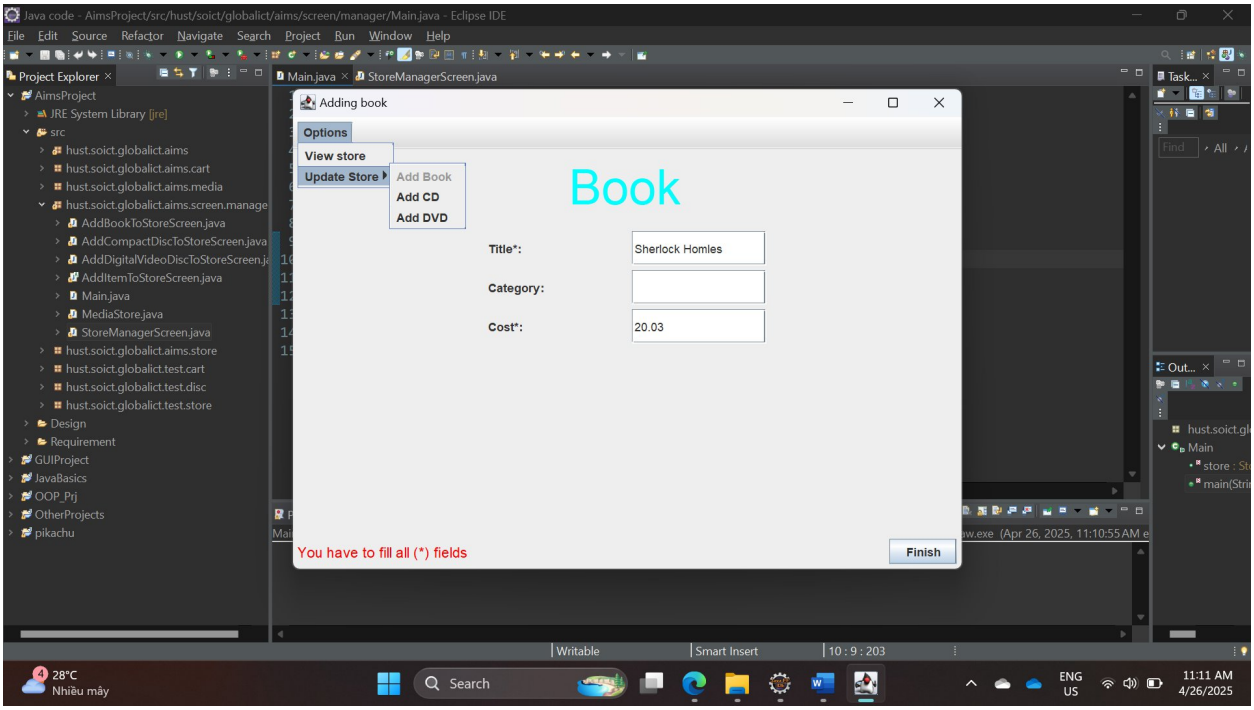
Modifying StoreManagerScreen class:

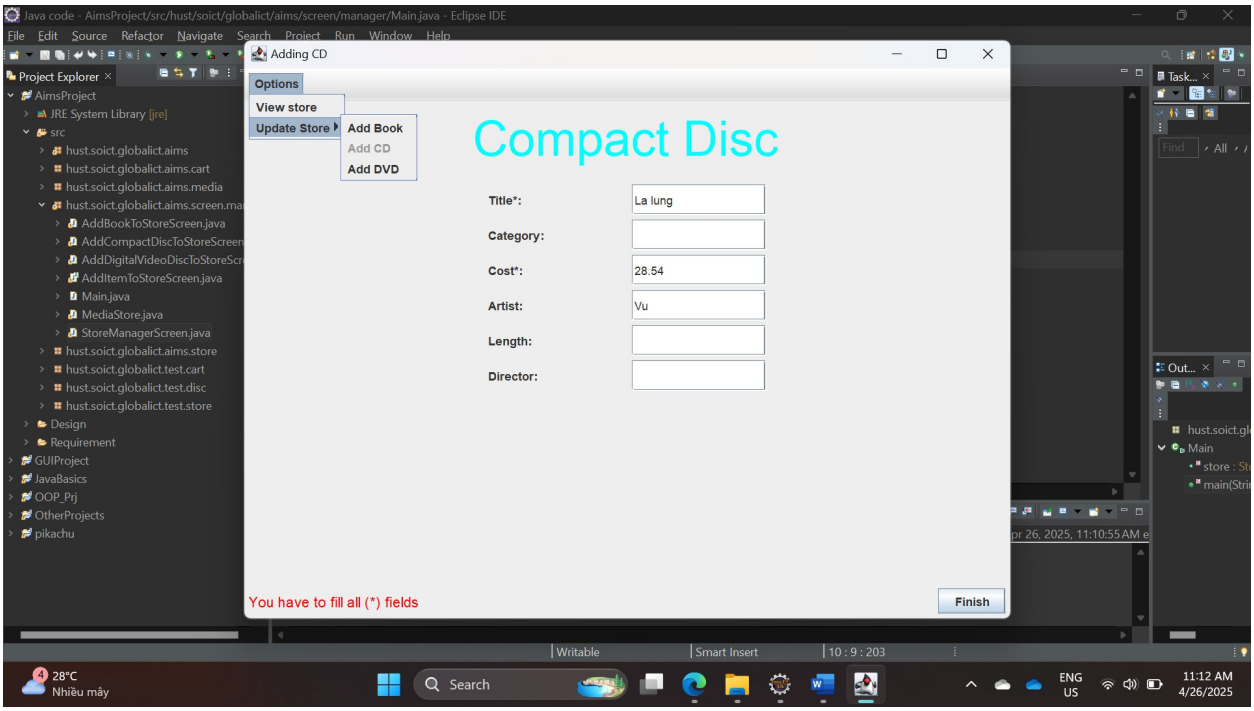
```
public class StoreManagerScreen extends JFrame {
    private Store store;
    private Container cp;
    private JFrame frame;

    public StoreManagerScreen(Store store) {
        this.store = store;
        frame = new JFrame();

        cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(), BorderLayout.CENTER);

        frame.add(cp);
        frame.setTitle("Store");
        frame.setSize(1024, 680);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
```

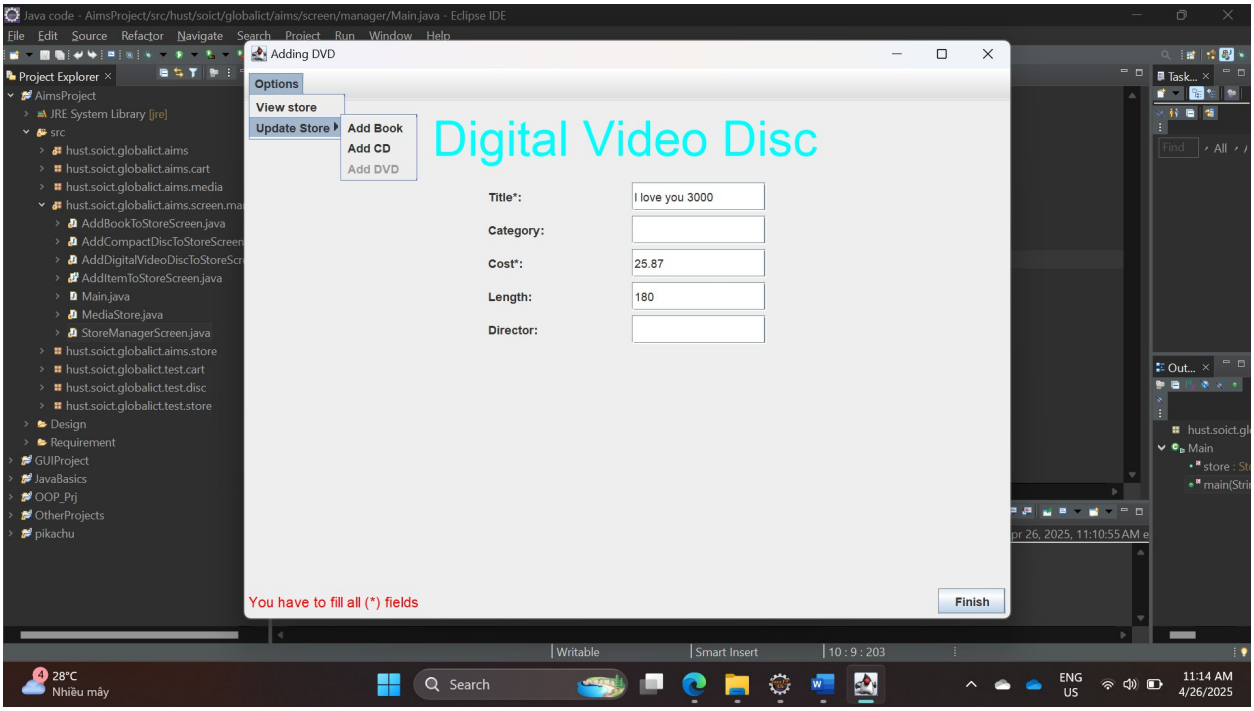
```
    JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");

        JMenuItem viewStore = new JMenuItem("View store");
        viewStore.setEnabled(false);
        menu.add(viewStore);

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBook = new JMenuItem("Add Book");
        addBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddBookToStoreScreen();
            }
        });
        smUpdateStore.add(addBook);

        JMenuItem addCD = new JMenuItem("Add CD");
        addCD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddCompactDiscToStoreScreen();
            }
        });
        smUpdateStore.add(addCD);
```

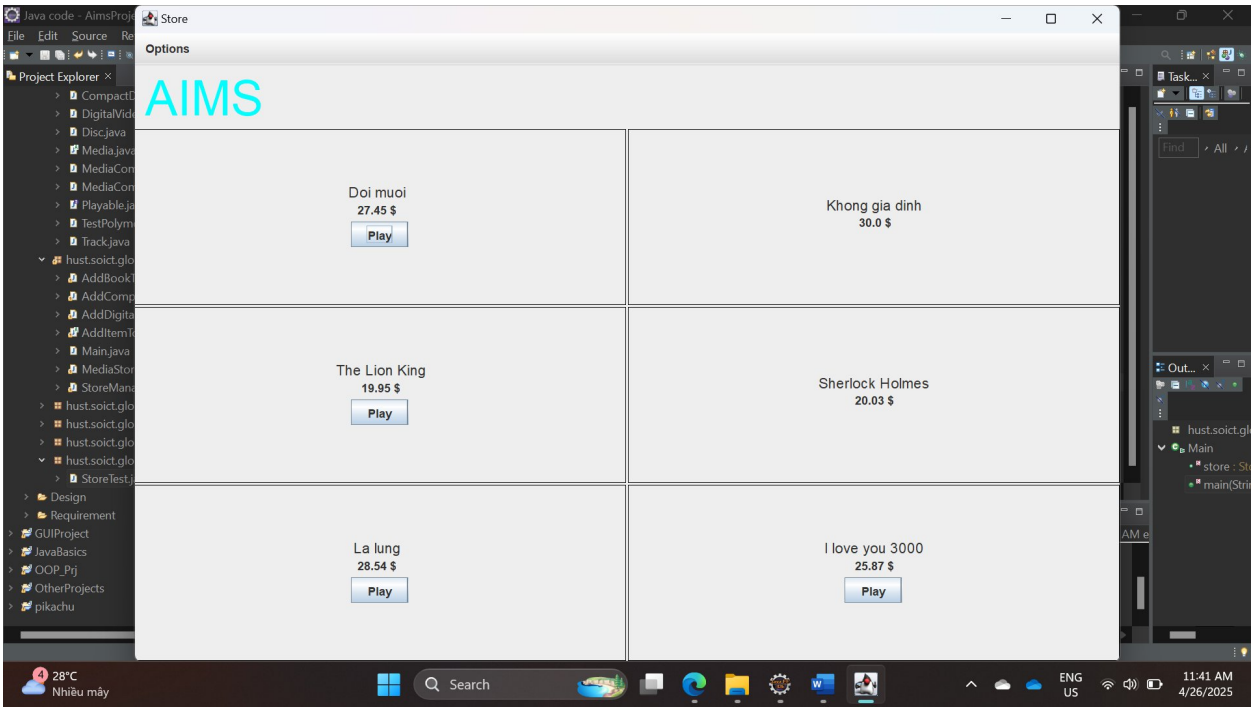
```
        JMenuItem addDVD = new JMenuItem("Add DVD");
        addDVD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
                new AddDigitalVideoDiscToStoreScreen();
            }
        });
        smUpdateStore.add(addDVD);
        menu.add(smUpdateStore);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }
```

Test program:

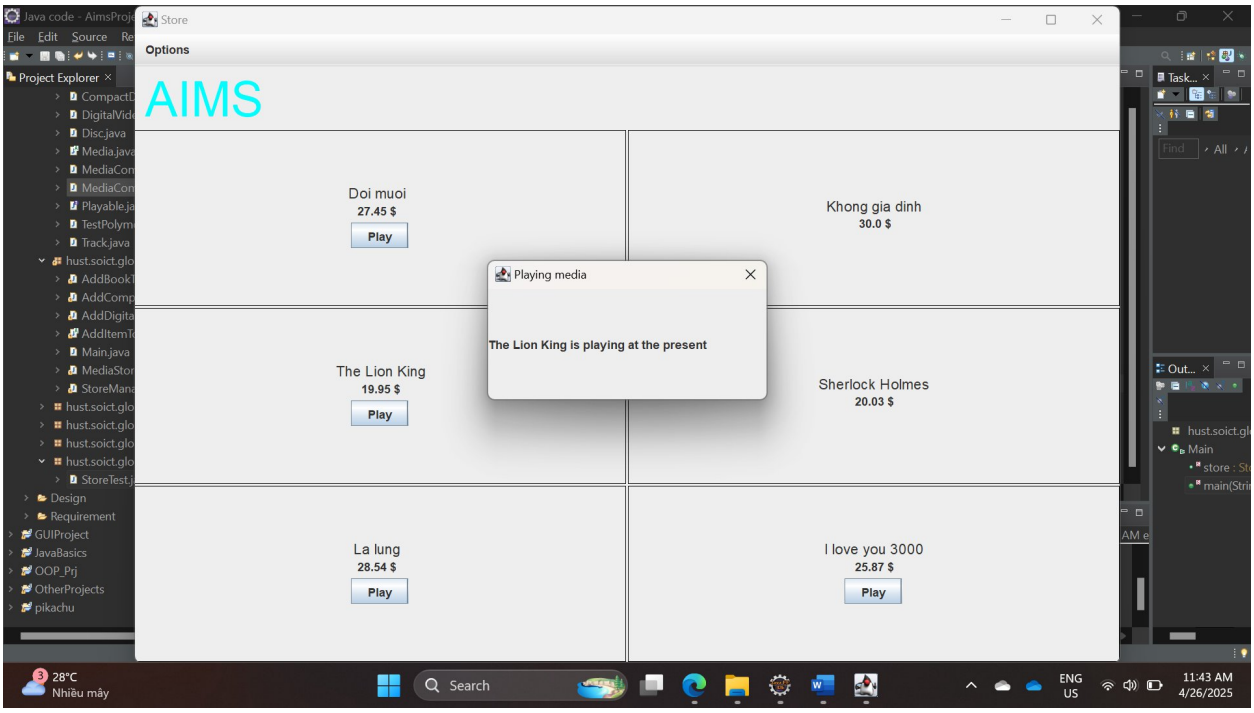



Java code - AimsProject/src/hust/soict/globalict/aims/screen/manager/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
AimsProject
JRE System Library [jre]
src
hust.soict.globalict.aims
hust.soict.globalict.aims.cart
hust.soict.globalict.aims.media
hust.soict.globalict.aims.screen.ma
AddBookToStoreScreen.java
AddCompactDiscToStoreScreen
AddDigitalVideoDiscToStoreScr
AddItemToStoreScreen.java
Main.java
MediaStore.java
StoreManagerScreen.java
hust.soict.globalict.aims.store
hust.soict.globalict.test.cart
hust.soict.globalict.test.disc
hust.soict.globalict.test.store
Design
Requirement
GUIProject
JavaBasics
OOP_Prj
OtherProjects
pikachu
Adding CD
Options
View store
Update Store
Add Book
Add CD
Add DVD
Compact Disc
Title*:
La lung
Category:
Cost*:
28.54
Artist:
Vu
Length:
Director:
You have to fill all (*) fields
Finish
Writable
Smart Insert
10 : 9 : 203
28°C
Nhiều mây
Search
ENG
US
11:12 AM
4/26/2025

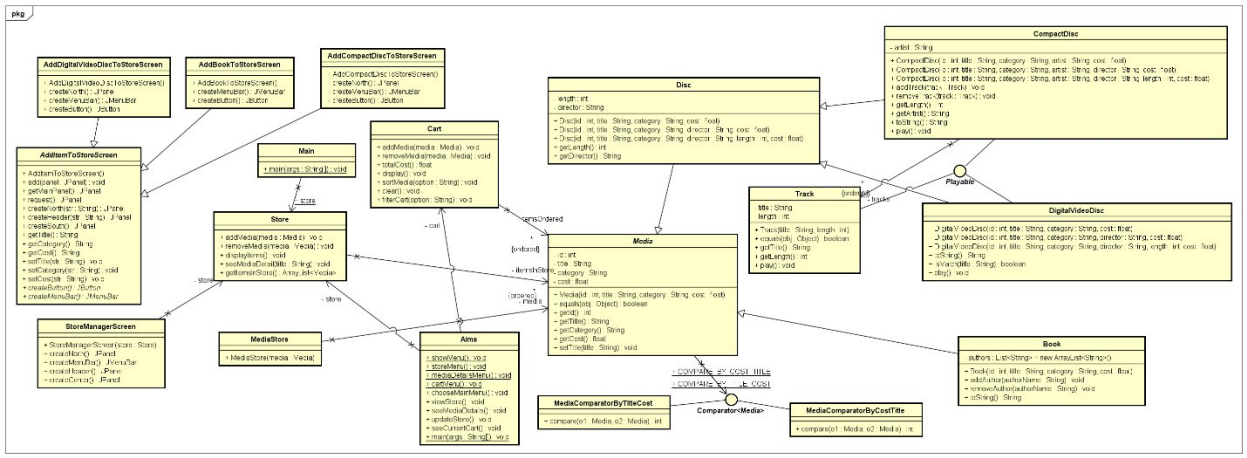

Java code - AimsProject/src/hust/soict/globalict/aims/screen/manager/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
AimsProject
JRE System Library [jre]
src
hust.soict.globalict.aims
hust.soict.globalict.aims.cart
hust.soict.globalict.aims.media
hust.soict.globalict.aims.screen.ma
AddBookToStoreScreen.java
AddCompactDiscToStoreScreen
AddDigitalVideoDiscToStoreScr
AddItemToStoreScreen.java
Main.java
MediaStore.java
StoreManagerScreen.java
hust.soict.globalict.aims.store
hust.soict.globalict.test.cart
hust.soict.globalict.test.disc
hust.soict.globalict.test.store
Design
Requirement
GUIProject
JavaBasics
OOP_Prj
OtherProjects
pikachu
Adding DVD
Options
View store
Update Store
Add Book
Add CD
Add DVD
Digital Video Disc
Title*:
I love you 3000
Category:
Cost*:
25.87
Length:
180
Director:
You have to fill all (*) fields
Finish
Writable
Smart Insert
10 : 9 : 203
28°C
Nhiều mây
Search
ENG
US
11:14 AM
4/26/2025



Play a media:



- Update the UML class diagram for the **`AimsProject`**.


pkg
AddDigitalVideoDiscToStoreScreen
AddBookToStoreScreen
AddCompactDiscToStoreScreen
AddItemToStoreScreen
Main
Cart
Disc
CompactDisc
Playable
Track
DigitalVideoDisc
Store
Media
StoreManagerScreen
MediaStore
Aims
Book
MediaComparatorByTitleCost
MediaComparatorByCostTitle
Comparator<Media>