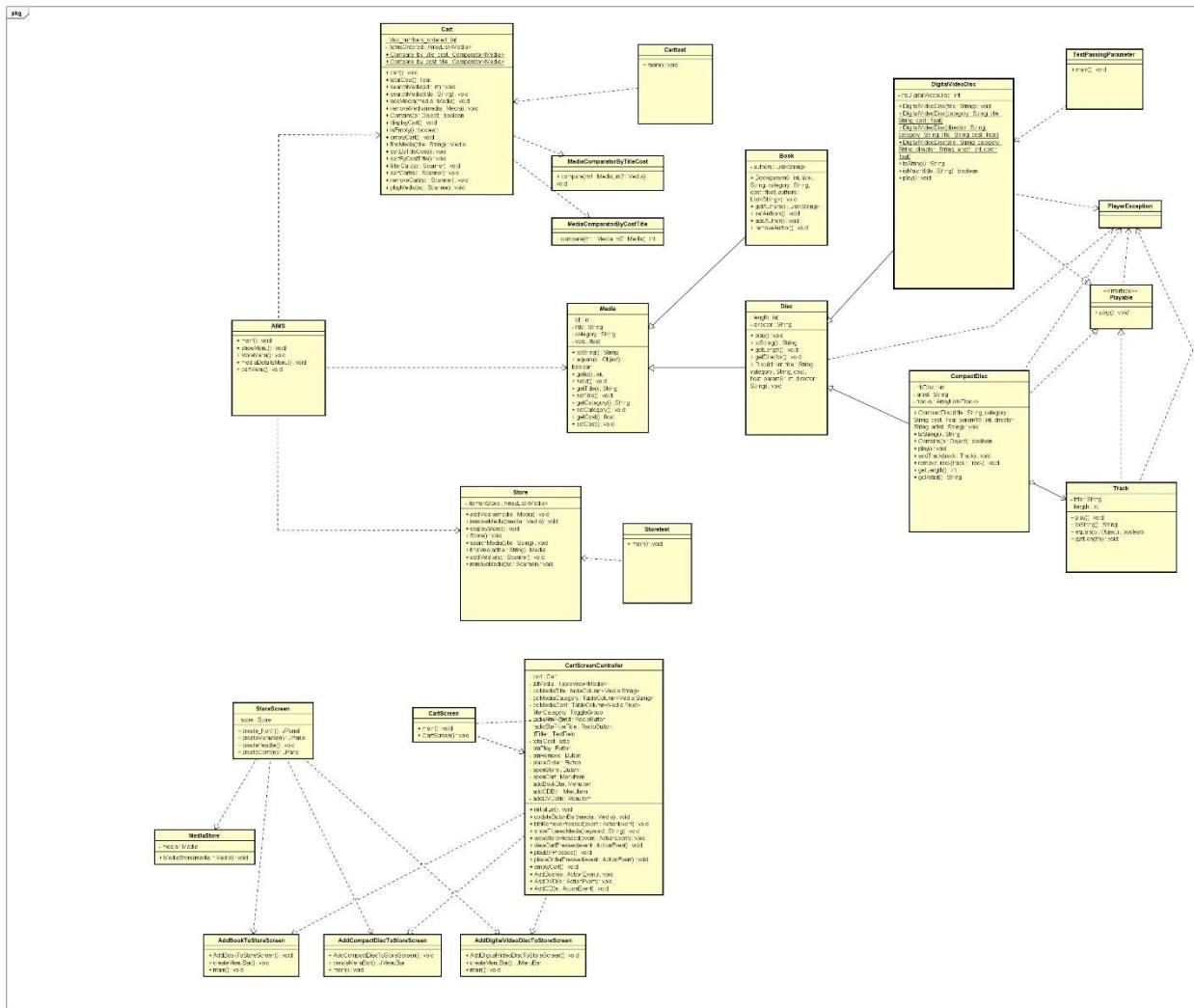


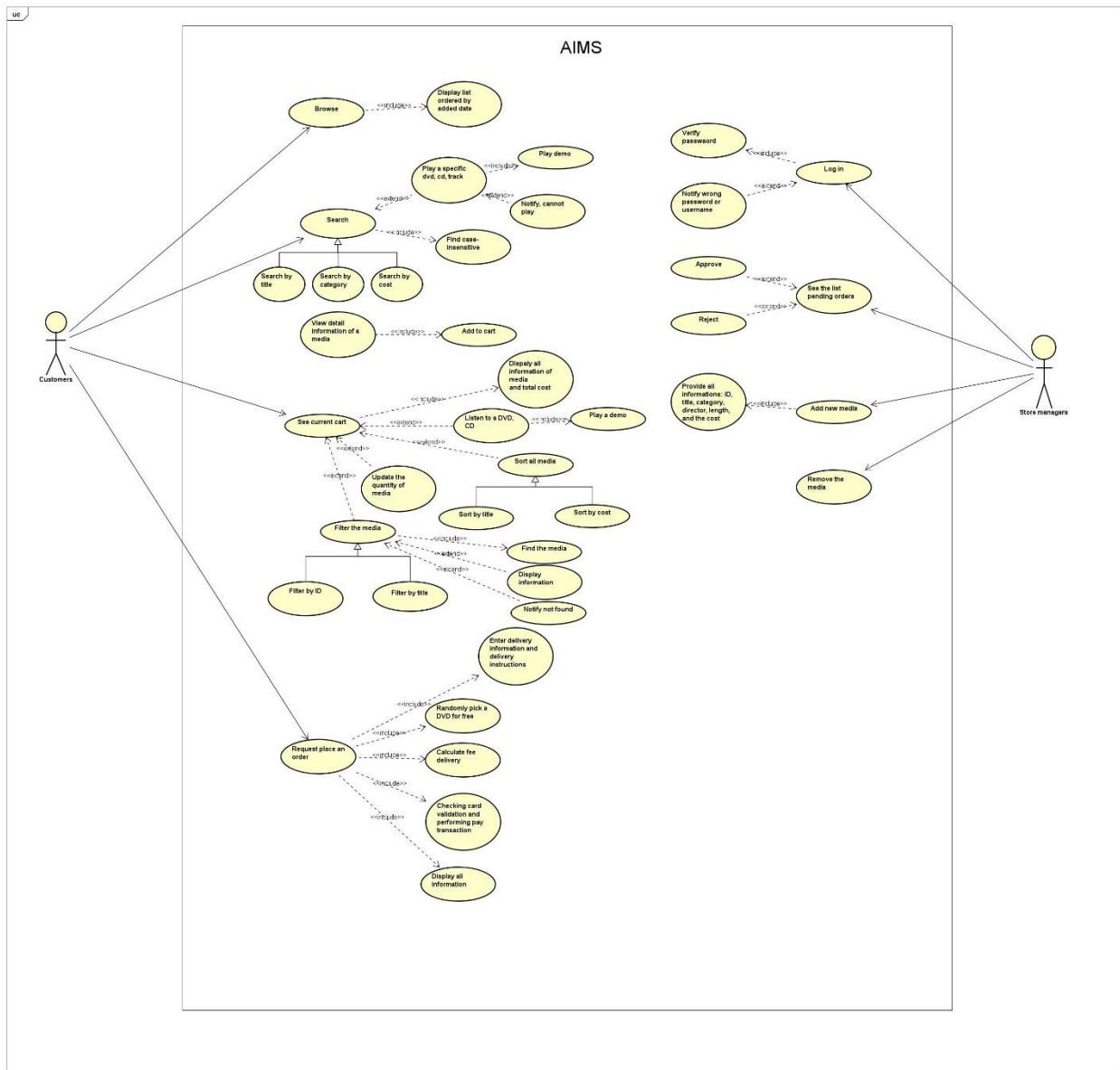
Report Lab05

Name: Nguyễn Đức Mạnh

ID: 20235525

I. Diagram





II. Source code

1. Swing components:
- a) Swing Accumulator

Eclipse - GUIProject/src/hust/soict/dsai/swing/SwingAccumulator.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Type Hierachy
src
  hust/soict/dsai/swing
    SwingAccumulator.java
  hust/soict/dsai/javafx
  hust/soict/dsai/swing
    AWTAccumulator.java
    NumberGrid.java
    SwingAccumulator.java
  Referenced Libraries
  JRE System Library [jdk-1.8]
  OtherProject.zip_expanded [Eclipse main]
  ShowTwoNumbers
  SolvingEquation
  SwingAccumulator
  Triangle
  Problems Javadoc Declaration Console
No consoles to display at this time.
```

SwingAccumulator.java

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.Container;
4
5 public class SwingAccumulator extends JFrame {
6     private JTextField tfInput;
7     private JTextField tfOutput;
8     private int s = 0;
9
10    public SwingAccumulator() {
11        Container cp = getContentPane();
12        cp.setLayout(new GridLayout(2, 2));
13
14        cp.add(new JLabel("Enter an Integer: "));
15
16        tfInput = new JTextField(10);
17        cp.add(tfInput);
18        tfInput.addActionListener(new TFIInputListener());
19
20        cp.add(new JLabel("The Accumulated Sum is: "));
21
22        tfOutput = new JTextField(10);
23        cp.add(tfOutput);
24        tfOutput.setEditable(false);
25        cp.add(tfOutput);
26
27        setTitle("Swing Accumulator");
28        setSize(350, 120);
29    }
30
31    public static void main(String[] args) {
32        new SwingAccumulator();
33    }
34
35    private class TFIInputListener implements ActionListener {
36        public void actionPerformed(ActionEvent evt) {
37            int numberIn = Integer.parseInt(tfInput.getText());
38            s += numberIn;
39            tfInput.setText("");
40            tfOutput.setText(s + "");
41        }
42    }
43
44    private class TFOutputListener implements ActionListener {
45        public void actionPerformed(ActionEvent evt) {
46            int numberIn = Integer.parseInt(tfInput.getText());
47            s += numberIn;
48            tfInput.setText("");
49            tfOutput.setText(s + "");
50        }
51    }
52}
```

Eclipse - GUIProject/src/hust/soict/dsai/swing/SwingAccumulator.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Type Hierachy
src
  hust/soict/dsai/swing
    SwingAccumulator.java
  hust/soict/dsai/javafx
  hust/soict/dsai/swing
    AWTAccumulator.java
    NumberGrid.java
    SwingAccumulator.java
  Referenced Libraries
  JRE System Library [jdk-1.8]
  OtherProject.zip_expanded [Eclipse main]
  ShowTwoNumbers
  SolvingEquation
  SwingAccumulator
  Triangle
  Problems Javadoc Declaration Console
No consoles to display at this time.
```

SwingAccumulator.java

```
25 tfInput.addActionListener(new TFIInputListener());
26
27 cp.add(new JLabel("The Accumulated Sum is: "));
28 tfOutput = new JTextField(10);
29 tfOutput.setEditable(false);
30 cp.add(tfOutput);
31
32 setTitle("Swing Accumulator");
33 setSize(350, 120);
34 setVisible(true);
35
36 public static void main(String[] args) {
37     new SwingAccumulator();
38 }
39
40 private class TFIInputListener implements ActionListener {
41     public void actionPerformed(ActionEvent evt) {
42         int numberIn = Integer.parseInt(tfInput.getText());
43         s += numberIn;
44         tfInput.setText("");
45         tfOutput.setText(s + "");
46     }
47 }
48
49 private class TFOutputListener implements ActionListener {
50     public void actionPerformed(ActionEvent evt) {
51         int numberIn = Integer.parseInt(tfInput.getText());
52         s += numberIn;
53         tfInput.setText("");
54         tfOutput.setText(s + "");
55     }
56 }
```

Eclipse - GUIProject/src/hust/soict/dsai/swing/SwingAccumulator.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Type Hierachy
src
  hust/soict/dsai/swing
    SwingAccumulator.java
  hust/soict/dsai/javafx
  hust/soict/dsai/swing
    AWTAccumulator.java
    NumberGrid.java
    SwingAccumulator.java
  Referenced Libraries
  JRE System Library [jdk-1.8]
  OtherProject.zip_expanded [Eclipse main]
  ShowTwoNumbers
  SolvingEquation
  SwingAccumulator
  Triangle
  Problems Javadoc Declaration Console
SwingAccumulator [Java Application] C:\Program Files\Java\jdk-1.8\bin\javaw.exe (14:05:16, 20 thg 12, 2024) [pid: 7068]
```

SwingAccumulator.java

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.Container;
4
5 public class SwingAccumulator extends JFrame {
6     private JTextField tfInput;
7     private JTextField tfOutput;
8     private int s = 0;
9
10    public SwingAccumulator() {
11        Container cp = getContentPane();
12        cp.setLayout(new GridLayout(2, 2));
13
14        cp.add(new JLabel("Enter an Integer: "));
15
16        tfInput = new JTextField(10);
17        cp.add(tfInput);
18        tfInput.addActionListener(new TFIInputListener());
19
20        cp.add(new JLabel("The Accumulated Sum is: "));
21
22        tfOutput = new JTextField(10);
23        cp.add(tfOutput);
24        tfOutput.setEditable(false);
25        cp.add(tfOutput);
26
27        setTitle("Swing Accumulator");
28        setSize(350, 120);
29        setVisible(true);
30    }
31
32    public static void main(String[] args) {
33        new SwingAccumulator();
34    }
35
36    private class TFIInputListener implements ActionListener {
37        public void actionPerformed(ActionEvent evt) {
38            int numberIn = Integer.parseInt(tfInput.getText());
39            s += numberIn;
40            tfInput.setText("");
41            tfOutput.setText(s + "");
42        }
43    }
44
45    private class TFOutputListener implements ActionListener {
46        public void actionPerformed(ActionEvent evt) {
47            int numberIn = Integer.parseInt(tfInput.getText());
48            s += numberIn;
49            tfInput.setText("");
50            tfOutput.setText(s + "");
51        }
52    }
53}
```

Swing Accumulator

Enter an Integer: 20

The Accumulated Sum is: 20

b) AWT Accumulator

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Eclipse - GUIProject/src/hust/soict/dsai/swing/AWTAccumulator.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, search, and project navigation.
- Left Sidebar:** Package Explorer showing the project structure. It includes nodes for AIMSPProject, BankAccount, Calculating, ChoosingOption, DaysOfAMonth, FirstDialog, GUIProject [Eclipse main], JRE System Library [jdk-1.8], src (containing hust.soict.dsai.javafx and hust.soict.dsai.swing), Referenced Libraries, HelloNameDialog, HelloWorld, InputFromKeyboard, MatrixAddition, NumSort, OtherProject.zip_expanded [Eclipse], ShowTwoNumbers, SolvingEquation, SwingAccumulator, and Triangle.
- Central Area:** Editor tab for "AWTAccumulator.java". The code is as follows:

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.Frame;
4 import java.awt.GridLayout;
5 import java.awt.Label;
6 import java.awt.TextField;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 public class AWTAccumulator extends Frame {
11     private TextField tfInput;
12     private TextField tfOutput;
13     private int s = 0;
14
15     public AWTAccumulator() {
16         setLayout(new GridLayout(2, 2));
17         add(new Label("Enter an Integer: "));
18
19         tfInput = new TextField(10);
20         add(tfInput);
21         tfInput.addActionListener(new TFIputListener());
22
23         add(new Label("The Accumulated Sum is: "));
24         tfOutput = new TextField(10);
25         tfOutput.setEditable(false);
26         add(tfOutput);
27     }
28
29     static void main(String[] args) {
30         new AWTAccumulator();
31     }
32
33     private class TFIputListener implements ActionListener {
34         public void actionPerformed(ActionEvent evt) {
35             int numberIn = Integer.parseInt(tfInput.getText());
36             s += numberIn;
37             tfInput.setText("");
38             tfOutput.setText(s + "");
39         }
40     }
41
42 }
43
44 }
```

- Bottom Status Bar:** Writable, Smart Insert, 3:23:56
- Bottom Buttons:** Problems, Javadoc, Declaration, Console (disabled)

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Eclipse - GUIProject/src/hust/soict/dsai/swing/AWTAccumulator.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, search, and project navigation.
- Left Sidebar:** Package Explorer showing the project structure. It includes nodes for AIMSPProject, BankAccount, Calculating, ChoosingOption, DaysOfAMonth, FirstDialog, GUIProject [Eclipse main], JRE System Library [jdk-1.8], src (containing hust.soict.dsai.javafx and hust.soict.dsai.swing), Referenced Libraries, HelloNameDialog, HelloWorld, InputFromKeyboard, MatrixAddition, NumSort, OtherProject.zip_expanded [Eclipse], ShowTwoNumbers, SolvingEquation, SwingAccumulator, and Triangle.
- Central Area:** Editor tab for "AWTAccumulator.java". The code is now complete with the main method and the TFIputListener implementation:

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.Frame;
4 import java.awt.GridLayout;
5 import java.awt.Label;
6 import java.awt.TextField;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 public class AWTAccumulator extends Frame {
11     private TextField tfInput;
12     private TextField tfOutput;
13     private int s = 0;
14
15     public AWTAccumulator() {
16         setLayout(new GridLayout(2, 2));
17         add(new Label("Enter an Integer: "));
18
19         tfInput = new TextField(10);
20         add(tfInput);
21         tfInput.addActionListener(new TFIputListener());
22
23         add(new Label("The Accumulated Sum is: "));
24         tfOutput = new TextField(10);
25         tfOutput.setEditable(false);
26         add(tfOutput);
27     }
28
29     static void main(String[] args) {
30         new AWTAccumulator();
31     }
32
33     private class TFIputListener implements ActionListener {
34         public void actionPerformed(ActionEvent evt) {
35             int numberIn = Integer.parseInt(tfInput.getText());
36             s += numberIn;
37             tfInput.setText("");
38             tfOutput.setText(s + "");
39         }
40     }
41
42 }
43
44 }
```

- Bottom Status Bar:** Writable, Smart Insert, 3:23:56
- Bottom Buttons:** Problems, Javadoc, Declaration, Console (disabled)

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like ALMSPProject, BankAccount, Calculating, ChoosingOption, DaysOfMonth, FirstDialog, GUIProject (selected), JRE System Library [jdk-1.8], NumberGrid.java, SwingAccumulator.java, and Triangle.
- Code Editor:** Displays the `AWTAccumulator.java` file with the following code:

```
21     tfInput.addActionListener(new TFIinputListener());
22
23     add(new Label("The Accumulated Sum is: "));
24     tfOutput = new TextField(10);
25     tfOutput.setEditable(false);
26     add(tfOutput);
27
28     setTitle("AWT Accumulator");
29     setSize(350, 120);
30     setVisible(true);
31 }
32
33 public static void main(String[ s
34     new AWTAccumulator();
35 }
36
37 private class TFIinputListener implements ActionListener {
38     public void actionPerformed(ActionEvent evt) {
39         int numberIn = Integer.parseInt(tfInput.getText());
40         s += numberIn;
41         tfInput.setText("");
42         tfOutput.setText(s + "");
43     }
44 }
45 }
```

- Run View:** Shows a window titled "AWT Accumulator" with the text "Enter an Integer:" and a text field containing "24". Below it, the text "The Accumulated Sum is:" is followed by a blank space.
- Console:** Displays the output "AWTAccumulator [Java Application] C:\Program Files\Java\jdk-1.8\bin\javaw.exe (16:04:40, 20 thg 12, 2024) [pid: 8996]".

c) Number Grid

Eclipse - GUIProject/src/hust/soict/dsa1/swing/NumberGrid.java - Eclipse IDE

```

1 package hust.soict.dsa1.swing;
2
3 import java.awt.BorderLayout;
4 import java.awt.ComponentOrientation;
5 import java.awt.Container;
6 import java.awt.GridLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 import javax.swing.JButton;
11 import javax.swing.JFrame;
12 import javax.swing.JPanel;
13 import javax.swing.JTextField;
14
15 public class NumberGrid extends JFrame{
16     private JButton[] btnNumbers = new JButton[10];
17     private JButton btnDelete, btnReset;
18     private JTextField tfDisplay;
19
20     public NumberGrid() {
21
22         tfDisplay = new JTextField();
23         tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
24
25         JPanel panelButtons = new JPanel(new GridLayout(4, 3));
26         addButtons(panelButtons);
27
28         Container cp = getContentPane();
29         cp.setLayout(new BorderLayout());
30         cp.add(tfDisplay, BorderLayout.NORTH);
31         cp.add(panelButtons, BorderLayout.CENTER);
32
33         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34         setTitle("Number Grid");
35         setSize(200, 200);
36         setVisible(true);
37     }
38
39     private class ButtonListener implements ActionListener{
40
41         @Override
42         public void actionPerformed(ActionEvent e) {
43             String button = e.getActionCommand();
44             if (button.charAt(0) >='0' & button.charAt(0) <='9') {
45                 tfDisplay.setText(tfDisplay.getText() + button);
46             }else if (button.equals("DEL")) {
47                 if (tfDisplay.getText().length() >= 1) {
48                     tfDisplay.setText(tfDisplay.getText().substring(0, tfDisplay.getText().length()-1));
49                 }
50             }else if (button.equals("C")) {
51                 tfDisplay.setText("");
52             }
53         }
54     }
55
56     void addButtons(JPanel panelButtons) {
57         ButtonListener btnListener = new ButtonListener();
58         for (int i = 1; i<=9; i++) {
59             btnNumbers[i] = new JButton(""+i);
60             panelButtons.add(btnNumbers[i]);
61             btnNumbers[i].addActionListener(btnListener);
62         }
63         btnDelete = new JButton("DEL");
64         panelButtons.add(btnDelete);
65         btnDelete.addActionListener(btnListener);
66     }

```

Eclipse - GUIProject/src/hust/soict/dsa1/swing/NumberGrid.java - Eclipse IDE

```

31         cp.add(panelButtons, BorderLayout.CENTER);
32
33         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34         setTitle("Number Grid");
35         setSize(200, 200);
36         setVisible(true);
37     }
38
39     private class ButtonListener implements ActionListener{
40
41         @Override
42         public void actionPerformed(ActionEvent e) {
43             String button = e.getActionCommand();
44             if (button.charAt(0) >='0' & button.charAt(0) <='9') {
45                 tfDisplay.setText(tfDisplay.getText() + button);
46             }else if (button.equals("DEL")) {
47                 if (tfDisplay.getText().length() >= 1) {
48                     tfDisplay.setText(tfDisplay.getText().substring(0, tfDisplay.getText().length()-1));
49                 }
50             }else if (button.equals("C")) {
51                 tfDisplay.setText("");
52             }
53         }
54     }
55
56     void addButtons(JPanel panelButtons) {
57         ButtonListener btnListener = new ButtonListener();
58         for (int i = 1; i<=9; i++) {
59             btnNumbers[i] = new JButton(""+i);
60             panelButtons.add(btnNumbers[i]);
61             btnNumbers[i].addActionListener(btnListener);
62         }
63         btnDelete = new JButton("DEL");
64         panelButtons.add(btnDelete);
65         btnDelete.addActionListener(btnListener);
66     }
67
68     btnNumbers[0] = new JButton("0");
69     panelButtons.add(btnNumbers[0]);
70     btnNumbers[0].addActionListener(btnListener);
71
72     btnReset = new JButton("C");
73     panelButtons.add(btnReset);
74     btnReset.addActionListener(btnListener);
75 }
76
77 public static void main(String[] args) {
78     new NumberGrid();
79 }
80

```

Eclipse - GUIProject/src/hust/soict/dsa1/swing/NumberGrid.java - Eclipse IDE

```

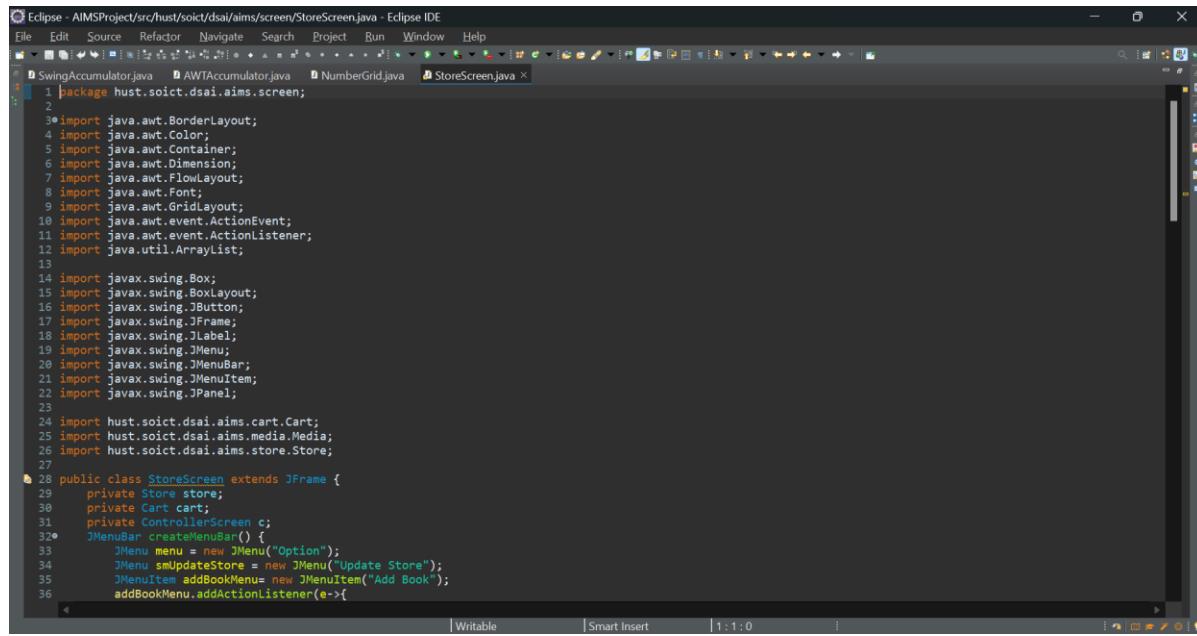
45         }else if (button.equals("DEL")) {
46             if (tfDisplay.getText().length() >= 1) {
47                 tfDisplay.setText(tfDisplay.getText().substring(0, tfDisplay.getText().length()-1));
48             }
49         }else if (button.equals("C")) {
50             tfDisplay.setText("");
51         }
52     }
53 }
54
55 void addButtons(JPanel panelButtons) {
56     ButtonListener btnListener = new ButtonListener();
57     for (int i = 1; i<=9; i++) {
58         btnNumbers[i] = new JButton(""+i);
59         panelButtons.add(btnNumbers[i]);
60         btnNumbers[i].addActionListener(btnListener);
61     }
62     btnDelete = new JButton("DEL");
63     panelButtons.add(btnDelete);
64     btnDelete.addActionListener(btnListener);
65
66     btnNumbers[0] = new JButton("0");
67     panelButtons.add(btnNumbers[0]);
68     btnNumbers[0].addActionListener(btnListener);
69
70     btnReset = new JButton("C");
71     panelButtons.add(btnReset);
72     btnReset.addActionListener(btnListener);
73 }
74
75 public static void main(String[] args) {
76     new NumberGrid();
77 }
78
79 }
80

```

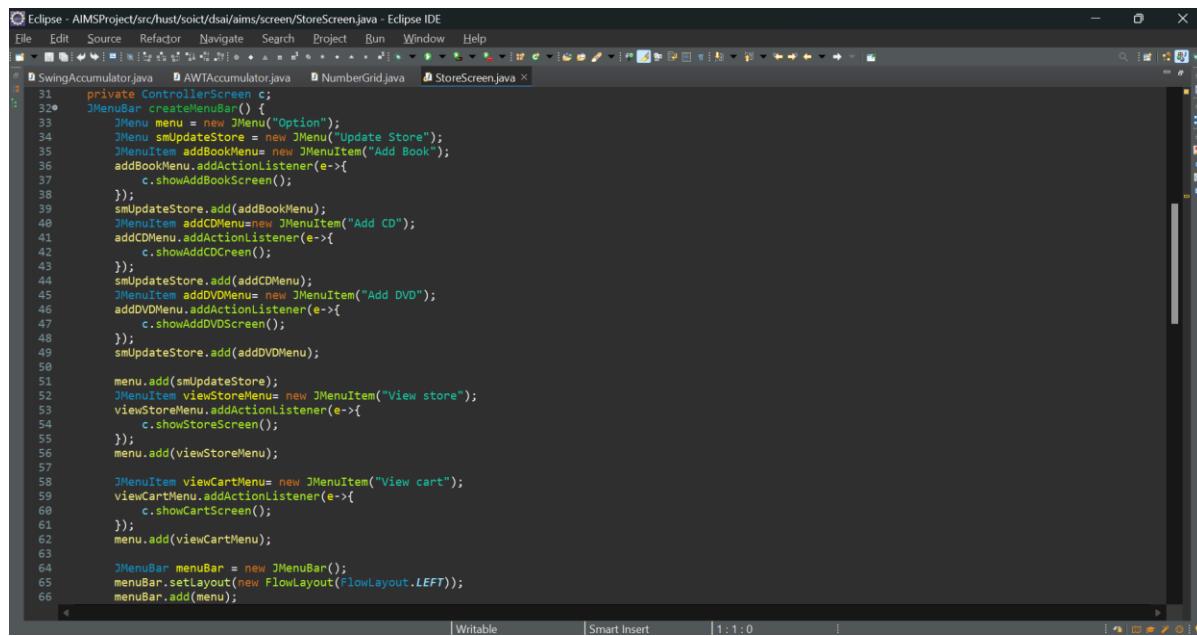
1	2	3
4	5	6
7	8	9
DEL	0	C

2. Create a graphical user interface for AIMS with Swing:

a) Store Screen



```
1 package hust.soict.dsai.aims.screen;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Container;
6 import java.awt.Dimension;
7 import java.awt.FlowLayout;
8 import java.awt.Font;
9 import java.awt.GridLayout;
10 import java.awt.event.ActionEvent;
11 import java.awt.event.ActionListener;
12 import java.util.ArrayList;
13
14 import javax.swing.Box;
15 import javax.swing.BoxLayout;
16 import javax.swing.JButton;
17 import javax.swing.JFrame;
18 import javax.swing.JLabel;
19 import javax.swing.JMenu;
20 import javax.swing.JMenuBar;
21 import javax.swing.JMenuItem;
22 import javax.swing.JPanel;
23
24 import hust.soict.dsai.aims.cart.Cart;
25 import hust.soict.dsai.aims.media.Media;
26 import hust.soict.dsai.aims.store.Store;
27
28 public class StoreScreen extends JFrame {
29     private Store store;
30     private Cart cart;
31     private ControllerScreen c;
32     JMenuBar createMenuBar() {
33         JMenu menu = new JMenu("Option");
34         JMenu smUpdateStore = new JMenu("Update Store");
35         JMenuItem addBookMenu= new JMenuItem("Add Book");
36         addBookMenu.addActionListener(e->{
37             c.showAddBookScreen();
38         });
39         smUpdateStore.add(addBookMenu);
40         JMenuItem addCDMenu=new JMenuItem("Add CD");
41         addCDMenu.addActionListener(e->{
42             c.showAddCDScreen();
43         });
44         smUpdateStore.add(addCDMenu);
45         JMenuItem addDVDMenu= new JMenuItem("Add DVD");
46         addDVDMenu.addActionListener(e->{
47             c.showAddDVDScreen();
48         });
49         smUpdateStore.add(addDVDMenu);
50
51         menu.add(smUpdateStore);
52         JMenuItem viewStoreMenu= new JMenuItem("View store");
53         viewStoreMenu.addActionListener(e->{
54             c.showStoreScreen();
55         });
56         menu.add(viewStoreMenu);
57
58         JMenuItem viewCartMenu= new JMenuItem("View cart");
59         viewCartMenu.addActionListener(e->{
60             c.showCartScreen();
61         });
62         menu.add(viewCartMenu);
63
64         JMenuBar menuBar = new JMenuBar();
65         menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
66         menuBar.add(menu);
67 }
```



```
31     private ControllerScreen c;
32     JMenuBar createMenuBar() {
33         JMenu menu = new JMenu("Option");
34         JMenu smUpdateStore = new JMenu("Update Store");
35         JMenuItem addBookMenu= new JMenuItem("Add Book");
36         addBookMenu.addActionListener(e->{
37             c.showAddBookScreen();
38         });
39         smUpdateStore.add(addBookMenu);
40         JMenuItem addCDMenu=new JMenuItem("Add CD");
41         addCDMenu.addActionListener(e->{
42             c.showAddCDScreen();
43         });
44         smUpdateStore.add(addCDMenu);
45         JMenuItem addDVDMenu= new JMenuItem("Add DVD");
46         addDVDMenu.addActionListener(e->{
47             c.showAddDVDScreen();
48         });
49         smUpdateStore.add(addDVDMenu);
50
51         menu.add(smUpdateStore);
52         JMenuItem viewStoreMenu= new JMenuItem("View store");
53         viewStoreMenu.addActionListener(e->{
54             c.showStoreScreen();
55         });
56         menu.add(viewStoreMenu);
57
58         JMenuItem viewCartMenu= new JMenuItem("View cart");
59         viewCartMenu.addActionListener(e->{
60             c.showCartScreen();
61         });
62         menu.add(viewCartMenu);
63
64         JMenuBar menuBar = new JMenuBar();
65         menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
66         menuBar.add(menu);
67 }
```

The screenshot shows the Eclipse IDE interface with the title bar "Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/StoreScreen.java - Eclipse IDE". The main area displays the Java code for the `StoreScreen.java` file. The code defines a `createHeader` method that creates a `JPanel` header with a `JMenuBar`, a `JLabel` titled "AIMS", and a `JButton` labeled "View cart". The button's action listener is annotated with `@Override` and `actionPerformed`. The code uses `BoxLayout` and `FlowLayout` for layout management. The code is color-coded for syntax highlighting.

```
64     JMenuBar menuBar = new JMenuBar();
65     menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
66     menuBar.add(menu);
67 
68     return menuBar;
69 }
70 
71 JPanel createHeader() {
72     JPanel header = new JPanel();
73     header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));
74 
75     JLabel title = new JLabel("AIMS");
76     title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
77     title.setForeground(color.CYAN);
78 
79     JButton btnCart = new JButton("View cart");
80 
81     btnCart.setPreferredSize(new Dimension(100, 50));
82     btnCart.setMaximumSize(new Dimension(100, 50));
83     btnCart.addActionListener(new ActionListener() {
84         @Override
85         public void actionPerformed(ActionEvent e) {
86             c.showCartScreen();
87         }
88     });
89 }
90 
91 header.add(Box.createRigidArea(new Dimension(10, 10)));
92 header.add(title);
93 header.add(Box.createHorizontalGlue());
94 header.add(btnCart);
95 header.add(Box.createRigidArea(new Dimension(10, 10)));
96 
97 return header;
98 }
99 }
```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/StoreScreen.java - Eclipse IDE

```

100+     JPanel createNorth() {
101         JPanel north = new JPanel();
102         north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
103         north.add(createMenuBar());
104         north.add(createHeader());
105         return north;
106     }
107
108+    JPanel createCenter() {
109         JPanel center = new JPanel();
110         center.setLayout(new GridLayout(3, 3, 2, 2));
111
112         ArrayList<Media> mediaInStore = store.getItemsInStore();
113         for (int i=0; i< mediaInStore.size(); i++) {
114             MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
115             center.add(cell);
116         }
117     }
118
119     return center;
120 }
121
122+ public StoreScreen(Store store, Cart cart, ControllerScreen c) {
123     this.store = store;
124     this.cart = cart;
125     this.c = c;
126     Container cp = getContentPane();
127     cp.setLayout(new BorderLayout());
128
129     cp.add(createNorth(), BorderLayout.NORTH);
130     cp.add(createCenter(), BorderLayout.CENTER);
131
132 //    setVisible(true);
133 //    setTitle("Store");
134 //    setSize(1024, 768);
135
136 }
137
138+ public static void main(String[] args) {
139     Cart cart = new Cart();
140     Store store = new Store();
141     ControllerScreen c = new ControllerScreen();
142     new StoreScreen(store, cart, c);
143 }
144
145 }

```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/StoreScreen.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Type Hier...

SwingAccumulator.java AWTAccumulator.java NumberGrid.java StoreScreen.java

```

120 }
121
122+ public StoreScreen(Store store, Cart cart) {
123     this.store = store;
124     this.cart = cart;
125     this.c = c;
126     Container cp = getContentPane();
127     cp.setLayout(new BorderLayout());
128
129     cp.add(createNorth(), BorderLayout.NORTH);
130     cp.add(createCenter(), BorderLayout.CENTER);
131
132 //    setVisible(true);
133 //    setTitle("Store");
134 //    setSize(1024, 768);
135
136 }
137
138+ public static void main(String[] args) {
139     Cart cart = new Cart();
140     Store store = new Store();
141     ControllerScreen c = new ControllerScreen();
142     new StoreScreen(store, cart, c);
143 }
144
145 }

```

Store

AIMS

	A	B
A	10000.0\$	10000.0\$
B		
C	100000.0\$	1000.0\$
D		
E	10000.0\$	

Add to Cart Add to Cart Add to Cart Add to Cart Add to Cart

Problems Javadoc Declaration Console

StoreScreen [Java Application] C:\Program Files\Java\jdk-1.8\bin\java -jar 12_20_2024 5:27:48 CH javaFX.fxml.Loader WARNING: Loading FXML document with JavaFX API of

b) Media Store

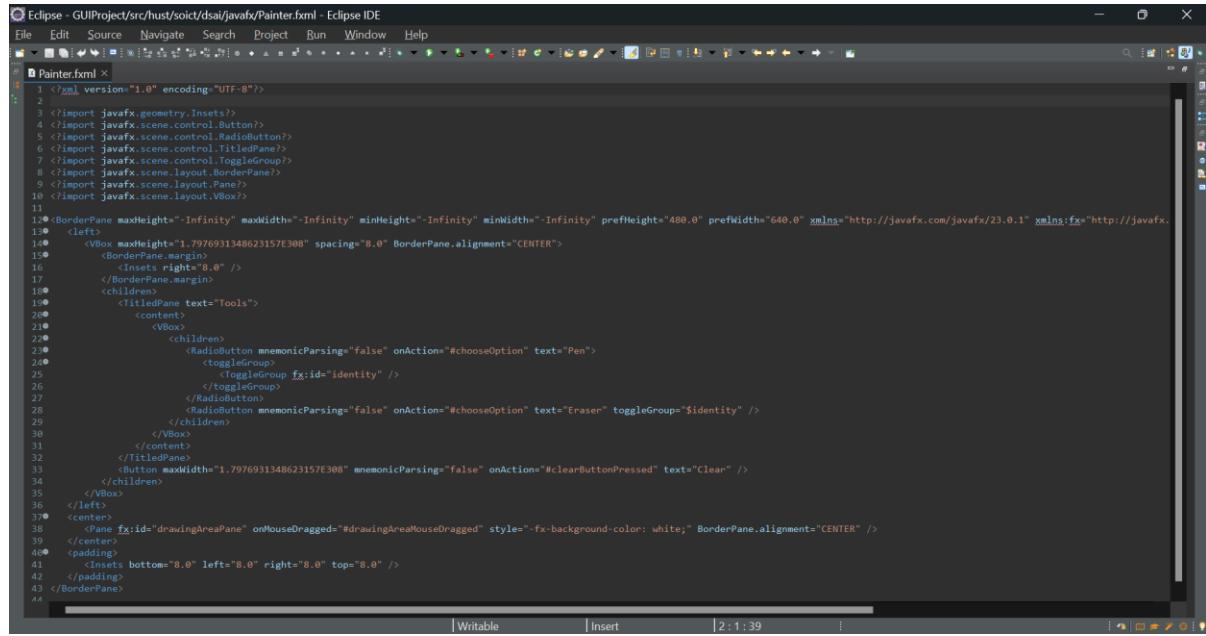
```

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/MediaStore.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MediaStore.java X
1 package hust.soict.dsai.aims.screen;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.FlowLayout;
6 import java.awt.Font;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 import javax.naming.LimitExceededException;
11 import javax.swing.BorderFactory;
12 import javax.swing.Box;
13 import javax.swing.BoxLayout;
14 import javax.swing.JButton;
15 import javax.swing.JDialog;
16 import javax.swing.JLabel;
17 import javax.swing.JPanel;
18 import javax.swing.border.EmptyBorder;
19
20 import hust.soict.dsai.aims.cart.Cart;
21 import hust.soict.dsai.aims.media.Media;
22 import hust.soict.dsai.aims.media.Playable;
23
24 public class MediaStore extends JPanel {
25
26     private Media media;
27
28     public MediaStore(Media media, Cart cart) {
29         this.media = media;
30         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
31
32         JLabel title = new JLabel(media.getTitle());
33         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 20));
34         title.setAlignmentX(CENTER_ALIGNMENT);
35
36         JLabel cost = new JLabel(" " + media.getCost() + "$");
37
38         title.setAlignmentX(CENTER_ALIGNMENT);
39
36     }
37
38     JPanel container = new JPanel();
39     container.setLayout(new FlowLayout(FlowLayout.CENTER));
40
41     JButton addCartBtn = new JButton("Add to Cart");
42     addCartBtn.addActionListener(new ActionListener() {
43
44         @Override
45         public void actionPerformed(ActionEvent e) {
46             try {
47                 cart.addMedia(media);
48             } catch (LimitExceededException e1) {
49                 // TODO Auto-generated catch block
50                 e1.printStackTrace();
51             }
52         }
53     });
54
55     container.add(addCartBtn);
56
57
58     if (media instanceof Playable) {
59         JButton playBtn = new JButton("Play");
60         playBtn.addActionListener(new ActionListener() {
61
62             @Override
63             public void actionPerformed(ActionEvent e) {
64                 JDialog playDialog = new JDialog();
65                 JPanel mainGui = new JPanel(new BorderLayout());
66                 mainGui.setBorder(new EmptyBorder(20, 20, 20, 20));
67
68                 // Display Playing Message
69                 mainGui.add(new JLabel("Playing... " + media.getTitle()), BorderLayout.CENTER);
70                 System.out.println(media.getTitle());
71
72                 JPanel buttonPanel = new JPanel(new FlowLayout());
73                 JButton close = new JButton("Stop");
74                 close.addActionListener(ev -> {
75                     playDialog.setVisible(false);
76                     System.out.println("Stopped playing.");
77                 });
78                 buttonPanel.add(close);
79                 mainGui.add(buttonPanel, BorderLayout.SOUTH);
80
81                 playDialog.setContentPane(mainGui);
82                 playDialog.setLocationRelativeTo(playBtn);
83                 playDialog.pack();
84
85                 // Show Dialog
86                 playDialog.setVisible(true);
87             }
88         });
89         container.add(playBtn);
90     }
91
92     this.add(Box.createVerticalGlue());
93     this.add(title);
94     this.add(cost);
95     this.add(Box.createVerticalGlue());
96     this.add(container);
97
98     this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
99
100 }

```

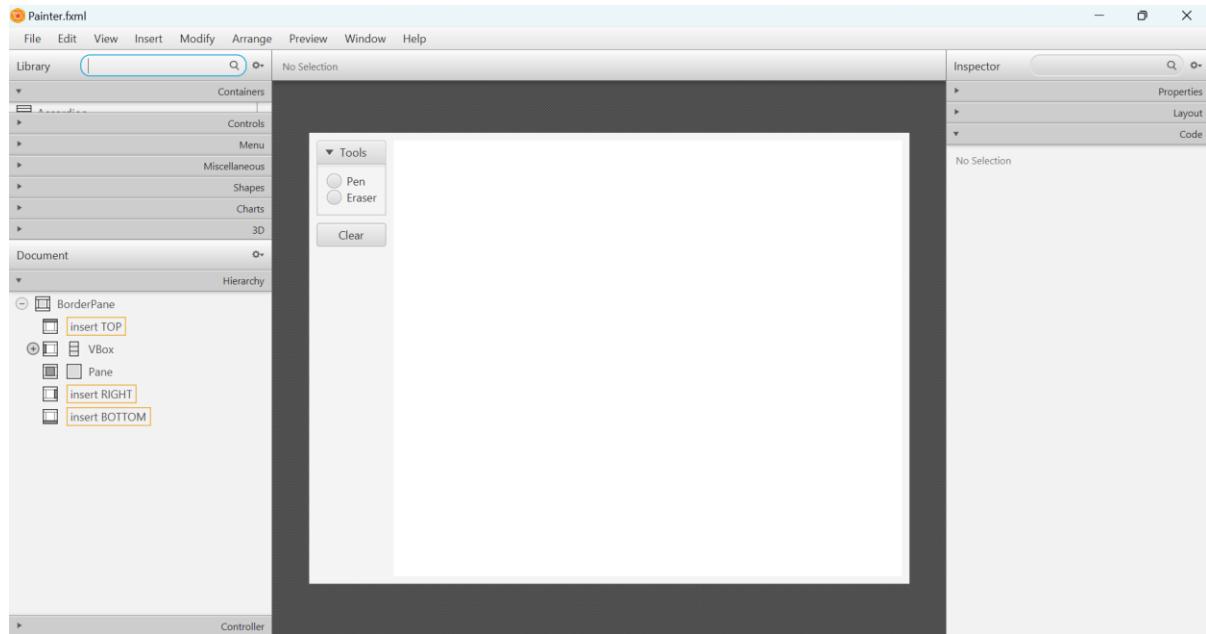
3. JavaFX API:

a) Painter.fxml

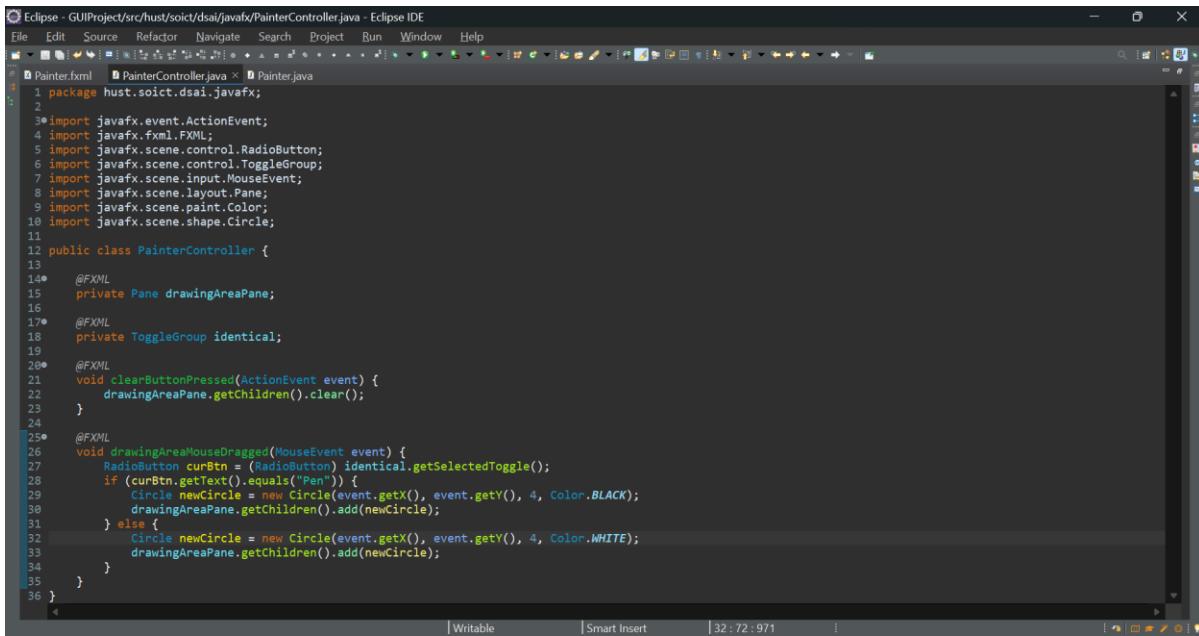


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BorderPane>
<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1">
    <imports>
        <import javafx.geometry.Insets>
        <import javafx.scene.control.Button>
        <import javafx.scene.control.RadioButton>
        <import javafx.scene.control.TitledPane>
        <import javafx.scene.control.ToggleGroup>
        <import javafx.scene.layout.BorderPane>
        <import javafx.scene.layout.Pane>
        <import javafx.scene.layout.VBox>
    </imports>
    <left>
        <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
            <BorderPane.margin>
                <Insets right="8.0" />
            </BorderPane.margin>
            <children>
                <titledpane text="Tools">
                    <content>
                        <VBox>
                            <children>
                                <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Pen">
                                    <toggleGroup fx:id="identity" />
                                </RadioButton>
                                <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Eraser" toggleGroup="$identity" />
                            </children>
                        </VBox>
                    </content>
                </TitledPane>
                <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />
            </children>
        </VBox>
    </left>
    <center>
        <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
        <padding>
            <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
        </padding>
    </center>
</BorderPane>
```

b) Screen Builder



c) Painter Controller



The screenshot shows the Eclipse IDE interface with the title "Eclipse - GUIProject/src/hust/soict/dsai/javafx/PainterController.java - Eclipse IDE". The code editor displays Java code for a PainterController class. The code includes imports for javafx.event.ActionEvent, javafx.fxml.FXML, javafx.scene.control.RadioButton, javafx.scene.control.ToggleGroup, javafx.scene.input.MouseEvent, javafx.scene.layout.Pane, javafx.scene.paint.Color, and javafx.scene.shape.Circle. It defines a private Pane drawingAreaPane and a private ToggleGroup identical. The class contains methods for handling button presses and mouse drag events to draw circles on the pane.

```
1 package hust.soict.dsai.javafx;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.RadioButton;
6 import javafx.scene.control.ToggleGroup;
7 import javafx.scene.input.MouseEvent;
8 import javafx.scene.layout.Pane;
9 import javafx.scene.paint.Color;
10 import javafx.scene.shape.Circle;
11
12 public class PainterController {
13
14     @FXML
15     private Pane drawingAreaPane;
16
17     @FXML
18     private ToggleGroup identical;
19
20     @FXML
21     void clearButtonPressed(ActionEvent event) {
22         drawingAreaPane.getChildren().clear();
23     }
24
25     @FXML
26     void drawingAreaMouseDragged(MouseEvent event) {
27         RadioButton curBtn = (RadioButton) identical.getSelectedToggle();
28         if (curBtn.getText().equals("Pen")) {
29             Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
30             drawingAreaPane.getChildren().add(newCircle);
31         } else {
32             Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.WHITE);
33             drawingAreaPane.getChildren().add(newCircle);
34         }
35     }
36 }
```

d) Painter

The screenshot shows two windows of the Eclipse IDE. The top window displays the code for `Painter.java`:

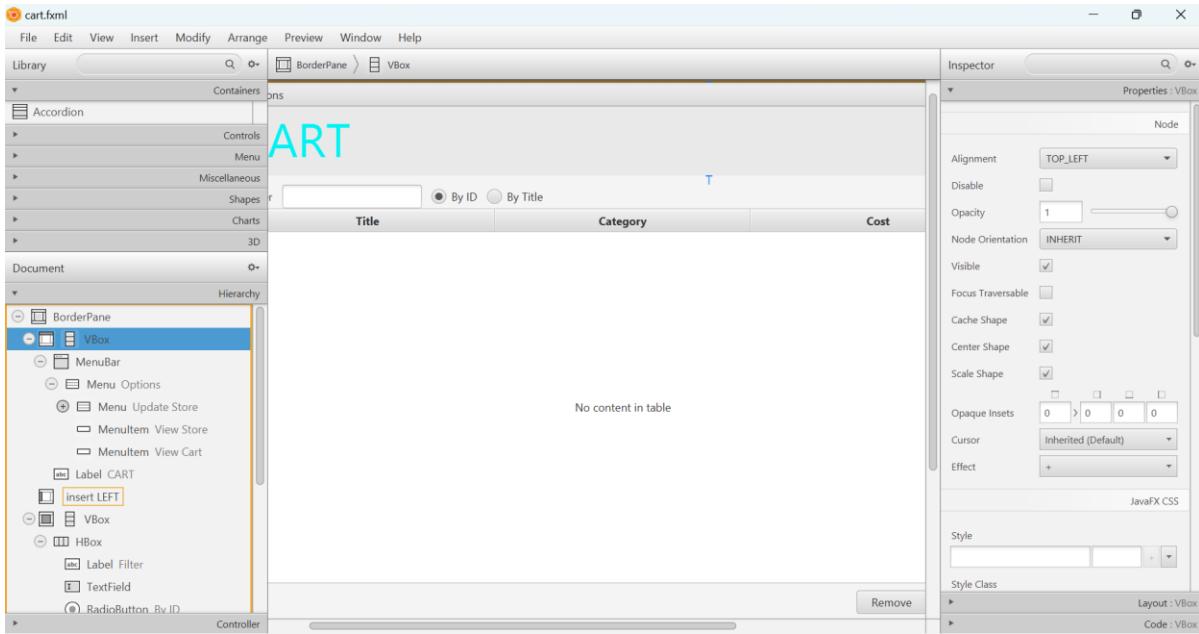
```

1 package hust.soict.dsai.javafx;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Painter extends Application {
10     public void start(Stage stage) throws Exception {
11         Parent root = FXMLLoader.load(getClass().getResource("/hust/soict/dsai/javafx/Painter.fxml"));
12         Scene scene = new Scene(root);
13         stage.setTitle("Painter");
14         stage.setScene(scene);
15         stage.show();
16     }
17
18    public static void main(String[] args) {
19        Launch(args);
20    }
21
22 }

```

The bottom window shows a JavaFX application window titled "Painter". On the left, there is a sidebar with tools: "Clear", "Tools", "Pen" (which is selected), and "Eraser". The main area contains a drawing canvas with some small black dots.

4. Setting up the View Cart Screen with ScreenBuilder:



5. Integrating JavaFX into Swing application – The JFXPanel class:

```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/CartScreen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CartPainter.fxml CartPainterController.java Painter.java CartScreen.java
1 import java.awt.BorderLayout;
2 import java.awt.Container;
3 import java.awt.Dimension;
4 import java.awt.GridLayout;
5 import java.awt.Window;
6 import javax.swing.JFrame;
7 import javax.swing.JPanel;
8 import javax.swing.JScrollPane;
9 import javax.swing.JTable;
10 import javax.swing.JTable;
11 import javax.swing.table.TableModel;
12
13 public class CartScreen extends JFrame {
14
15     private Cart cart;
16     private ControllerScreen controllerScreen;
17
18     public CartScreen(Cart cart, ControllerScreen c) {
19         super();
20
21         this.cart = cart;
22
23         JFXPanel fxPanel = new JFXPanel();
24         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25         this.setLayout(new BorderLayout());
26         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27         Platform.runLater(new Runnable() {
28             @Override
29             public void run() {
30                 try {
31                     FXMLLoader loader = new FXMLLoader(getClass()
32                         .getResource("hust/soict/dsai/aims/screen/cart.fxml"));
33                     CartScreenController controller = new CartScreenController(cart, c);
34                     loader.setController(controller);
35                     Parent root = loader.load();
36                     fxPanel.setScene(new Scene(root, 1024, 768));
37                 } catch (IOException e) {
38                     e.printStackTrace();
39                 }
40             }
41         });
42     }
43
44     public static void main(String[] args) {
45         Cart cart = new Cart();
46         ControllerScreen c = new ControllerScreen();
47         new CartScreen(cart, c);
48     }
49 }

```

6. View the items in cart – JavaFX's data-driven UI:

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```
1 package hust.soict.dsai.aims.screen;
2
3 import hust.soict.dsai.aims.cart.Cart;
4 import hust.soict.dsai.aims.media.Media;
5 import hust.soict.dsai.aims.media.Playable;
6 import javafx.animation.PauseTransition;
7 import javafx.beans.value.ChangeListener;
8 import javafx.beans.value.ObservableValue;
9 import javafx.collections.transformation.FilteredList;
10 import javafx.event.ActionEvent;
11 import javafx.fxml.FXML;
12 import javafx.scene.control.Button;
13 import javafx.scene.control.Label;
14 import javafx.scene.control.RadioButton;
15 import javafx.scene.control.TableColumn;
16 import javafx.scene.control.TableColumn.CellEditEvent;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.ToggleGroup;
19 import javafx.scene.control.cell.PropertyValueFactory;
20 import javafx.scene.input.InputMethodEvent;
21 import javafx.util.Duration;
22
23 public class CartScreenController {
24
25     private Cart cart;
26     private ControllerScreen controllerScreen;
27
28     @FXML
29     private TableView<Media> tblMedia;
30
31     @FXML
32     private TableColumn<Media, String> colMediaCategory;
33
34     @FXML
35     private TableColumn<Media, Float> colMediaCost;
36
37     @FXML
38     private TableColumn<Media, String> colMediaTitle;
39
40     @FXML
41     private ToggleGroup filterCategory;
42
43     @FXML
44     private Label totalCost;
45
46     @FXML
47     private Button btnPlay;
48
49     @FXML
50     private Button btnRemove;
51
52     @FXML
53     private Label playingMedia;
54
55     @FXML
56     private Button btnStop;
57
58     @FXML
59     private Button btnOrder;
60
61     @FXML
62     private RadioButton radioBtnFilterId;
63
64     @FXML
65     private TextField tfFilter;
66
67     public CartScreenController(Cart cart , ControllerScreen controllerScreen) {
68
69         super();
70         this.cart = cart;
71         this.controllerScreen = controllerScreen;
72     }
73
74     void updateButtonBar(Media media) {
75
76         btnRemove.setVisible(true);
77         if(media instanceof Playable) {
78             btnPlay.setVisible(true);
79         } else {
80             btnPlay.setVisible(false);
81         }
82     }
83
84     void showFilterMedia(String searchString) {
85
86         if(searchString.isEmpty()) {
87             tblMedia.setItems(this.cart.getItemsOrdered());
88         } else {
89             if(radioBtnFilterId.isSelected()) {
90                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getId()==Integer.parseInt(searchString)));
91             } else {
92                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getTitle().contains(searchString)));
93             }
94         }
95     }
96
97     @FXML
98     private void initialize() {
99
100         colMediaTitle.setCellValueFactory(
101             new PropertyValueFactory<Media, String>("Title"));
102
103         colMediaCategory.setCellValueFactory(
104             new PropertyValueFactory<Media, String>("Category"));
105         colMediaCost.setCellValueFactory(
106             new PropertyValueFactory<Media, Float>("Cost"));
107
108         tblMedia.setItems(this.cart.getItemsOrdered());
109         totalCost.setText(cart.getTotalCost()+"$");
110
111         btnPlay.setDisable(true);
112         btnRemove.setDisable(true);
113         playingMedia.setDisable(true);
114         btnStop.setDisable(true);
115
116         tfFilter.textProperty().addListener(new ChangeListener<String>() {
117
118             @Override
119             public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
120
121                 showFilterMedia(newValue);
122             }
123         });
124         tblMedia.getSelectionModel().selectedItemProperty().addListener(
125             new ChangeListener<Media>() {
126
127                 @Override
128                 public void changed(ObservableValue<? extends Media> observable, Media oldValue,
129                                     Media newValue) {
130
131                     if(newValue != null) {
132                         updateButtonBar(newValue);
133                     }
134                 }
135             });
136     }
137 }
```

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```
1 package hust.soict.dsai.aims.screen;
2
3 import hust.soict.dsai.aims.cart.Cart;
4 import hust.soict.dsai.aims.media.Media;
5 import hust.soict.dsai.aims.media.Playable;
6 import javafx.animation.PauseTransition;
7 import javafx.beans.value.ChangeListener;
8 import javafx.beans.value.ObservableValue;
9 import javafx.collections.transformation.FilteredList;
10 import javafx.event.ActionEvent;
11 import javafx.fxml.FXML;
12 import javafx.scene.control.Button;
13 import javafx.scene.control.Label;
14 import javafx.scene.control.RadioButton;
15 import javafx.scene.control.TableColumn;
16 import javafx.scene.control.TableColumn.CellEditEvent;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.ToggleGroup;
19 import javafx.scene.control.cell.PropertyValueFactory;
20 import javafx.scene.input.InputMethodEvent;
21 import javafx.util.Duration;
22
23 public class CartScreenController {
24
25     private Cart cart;
26     private ControllerScreen controllerScreen;
27
28     @FXML
29     private Label totalCost;
30
31     @FXML
32     private Button btnPlay;
33
34     @FXML
35     private Button btnRemove;
36
37     @FXML
38     private Label playingMedia;
39
40     @FXML
41     private Button btnStop;
42
43     @FXML
44     private Button btnOrder;
45
46     @FXML
47     private RadioButton radioBtnFilterId;
48
49     @FXML
50     private TextField tfFilter;
51
52     public CartScreenController(Cart cart , ControllerScreen controllerScreen) {
53
54         super();
55         this.cart = cart;
56         this.controllerScreen = controllerScreen;
57     }
58
59     void updateButtonBar(Media media) {
60
61         btnRemove.setVisible(true);
62         if(media instanceof Playable) {
63             btnPlay.setVisible(true);
64         } else {
65             btnPlay.setVisible(false);
66         }
67     }
68
69     void showFilterMedia(String searchString) {
70
71         if(searchString.isEmpty()) {
72             tblMedia.setItems(this.cart.getItemsOrdered());
73         } else {
74             if(radioBtnFilterId.isSelected()) {
75                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getId()==Integer.parseInt(searchString)));
76             } else {
77                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getTitle().contains(searchString)));
78             }
79         }
80     }
81
82     @FXML
83     private void initialize() {
84
85         colMediaTitle.setCellValueFactory(
86             new PropertyValueFactory<Media, String>("Title"));
87
88         colMediaCategory.setCellValueFactory(
89             new PropertyValueFactory<Media, String>("Category"));
90         colMediaCost.setCellValueFactory(
91             new PropertyValueFactory<Media, Float>("Cost"));
92
93         tblMedia.setItems(this.cart.getItemsOrdered());
94         totalCost.setText(cart.getTotalCost()+"$");
95
96         btnPlay.setDisable(true);
97         btnRemove.setDisable(true);
98         playingMedia.setDisable(true);
99         btnStop.setDisable(true);
100
101         tfFilter.textProperty().addListener(new ChangeListener<String>() {
102
103             @Override
104             public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
105
106                 showFilterMedia(newValue);
107             }
108         });
109         tblMedia.getSelectionModel().selectedItemProperty().addListener(
110             new ChangeListener<Media>() {
111
112                 @Override
113                 public void changed(ObservableValue<? extends Media> observable, Media oldValue,
114                                     Media newValue) {
115
116                     if(newValue != null) {
117                         updateButtonBar(newValue);
118                     }
119                 }
120             });
121
122         @Override
123         public void changed(ObservableValue<? extends Media> observable, Media oldValue,
124                             Media newValue) {
125
126             if(newValue != null) {
127                 updateButtonBar(newValue);
128             }
129         }
130     }
131 }
```

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```
1 package hust.soict.dsai.aims.screen;
2
3 import hust.soict.dsai.aims.cart.Cart;
4 import hust.soict.dsai.aims.media.Media;
5 import hust.soict.dsai.aims.media.Playable;
6 import javafx.animation.PauseTransition;
7 import javafx.beans.value.ChangeListener;
8 import javafx.beans.value.ObservableValue;
9 import javafx.collections.transformation.FilteredList;
10 import javafx.event.ActionEvent;
11 import javafx.fxml.FXML;
12 import javafx.scene.control.Button;
13 import javafx.scene.control.Label;
14 import javafx.scene.control.RadioButton;
15 import javafx.scene.control.TableColumn;
16 import javafx.scene.control.TableColumn.CellEditEvent;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.ToggleGroup;
19 import javafx.scene.control.cell.PropertyValueFactory;
20 import javafx.scene.input.InputMethodEvent;
21 import javafx.util.Duration;
22
23 public class CartScreenController {
24
25     private Cart cart;
26     private ControllerScreen controllerScreen;
27
28     @FXML
29     private Label totalCost;
30
31     @FXML
32     private Button btnPlay;
33
34     @FXML
35     private Button btnRemove;
36
37     @FXML
38     private Label playingMedia;
39
40     @FXML
41     private Button btnStop;
42
43     @FXML
44     private Button btnOrder;
45
46     @FXML
47     private RadioButton radioBtnFilterId;
48
49     @FXML
50     private TextField tfFilter;
51
52     public CartScreenController(Cart cart , ControllerScreen controllerScreen) {
53
54         super();
55         this.cart = cart;
56         this.controllerScreen = controllerScreen;
57     }
58
59     void updateButtonBar(Media media) {
60
61         btnRemove.setVisible(true);
62         if(media instanceof Playable) {
63             btnPlay.setVisible(true);
64         } else {
65             btnPlay.setVisible(false);
66         }
67     }
68
69     void showFilterMedia(String searchString) {
70
71         if(searchString.isEmpty()) {
72             tblMedia.setItems(this.cart.getItemsOrdered());
73         } else {
74             if(radioBtnFilterId.isSelected()) {
75                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getId()==Integer.parseInt(searchString)));
76             } else {
77                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getTitle().contains(searchString)));
78             }
79         }
80     }
81
82     @FXML
83     private void initialize() {
84
85         colMediaTitle.setCellValueFactory(
86             new PropertyValueFactory<Media, String>("Title"));
87
88         colMediaCategory.setCellValueFactory(
89             new PropertyValueFactory<Media, String>("Category"));
90         colMediaCost.setCellValueFactory(
91             new PropertyValueFactory<Media, Float>("Cost"));
92
93         tblMedia.setItems(this.cart.getItemsOrdered());
94         totalCost.setText(cart.getTotalCost()+"$");
95
96         btnPlay.setDisable(true);
97         btnRemove.setDisable(true);
98         playingMedia.setDisable(true);
99         btnStop.setDisable(true);
100
101         tfFilter.textProperty().addListener(new ChangeListener<String>() {
102
103             @Override
104             public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
105
106                 showFilterMedia(newValue);
107             }
108         });
109         tblMedia.getSelectionModel().selectedItemProperty().addListener(
110             new ChangeListener<Media>() {
111
112                 @Override
113                 public void changed(ObservableValue<? extends Media> observable, Media oldValue,
114                                     Media newValue) {
115
116                     if(newValue != null) {
117                         updateButtonBar(newValue);
118                     }
119                 }
120             });
121
122         @Override
123         public void changed(ObservableValue<? extends Media> observable, Media oldValue,
124                             Media newValue) {
125
126             if(newValue != null) {
127                 updateButtonBar(newValue);
128             }
129         }
130     }
131 }
```

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```

124         if(newValue != null) {
125             updateButtonBar(newValue);
126         }
127         totalCost.setText(cart.totalCost()+"$");
128     });
129 }
130 }
131
132● @FXML
133 void btnRemovePressed(ActionEvent event) {
134     Media media = tbhMedia.getSelectionModel().getSelectedItem();
135     cart.removeMedia(media);
136     totalCost.setText(cart.totalCost()+"$");
137 }
138
139● @FXML
140 void btnPlayPressed(ActionEvent event) {
141     Media media = tbhMedia.getSelectionModel().getSelectedItem();
142     playingMedia.setText("Playing "+media.getTitle()+"....");
143     playingMedia.setVisible(true);
144     btnStop.setVisible(true);
145 }
146
147● @FXML
148 void btnStopPressed(ActionEvent event) {
149     playingMedia.setVisible(false);
150     btnStop.setVisible(false);
151 }
152
153● @FXML
154 void btnOrderPressed(ActionEvent event) {
155     System.out.println("Order");
156     btnOrder.setText("Success!!!");
157     btnOrder.setDisable(true);
158     cart.getItemsOrdered().removeAll(cart.getItemsOrdered());
159     totalCost.setText("0.0$");
160     pauseTransition pt = new PauseTransition(Duration.seconds(1));
161     pt.setOnFinished(e ->{
162         btnOrder.setDisable(false);
163         playingMedia.setVisible(false);
164         btnPlay.setVisible(true);
165         btnOrder.setText("Order");
166     });
167 }
168
169
170● @FXML
171 void changeToStoreScreen(ActionEvent event) {
172     this.controllerScreen.showStoreScreen();
173 }
174● @FXML
175 void changeToAddBookScreen(ActionEvent event) {
176     this.controllerScreen.showAddBookScreen();
177 }
178
179● @FXML
180 void changeToAddCDScreen(ActionEvent event) {
181     this.controllerScreen.showAddCDScreen();
182 }
183
184● @FXML
185 void changeToAddDVDScreen(ActionEvent event) {
186     this.controllerScreen.showAddDVDScreen();
187 }
188
189● @FXML
190 void changeToCartScreen(ActionEvent event) {
191     this.controllerScreen.showCartScreen();
192 }
193
194● @FXML
195 void updateFiller(InputMethodEvent event) {
196     System.out.println(event.toString());
197 }
198
199
200 }

```

Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```

1● @Override
2 public void start(Stage stage) {
3     cart.getItemsOrdered().removeAll(cart.getItemsOrdered());
4     totalCost.setText("0.0$");
5     pauseTransition pt = new PauseTransition(Duration.seconds(1));
6     pt.setOnFinished(e ->{
7         btnOrder.setDisable(false);
8         playingMedia.setVisible(false);
9         btnPlay.setVisible(true);
10        btnOrder.setText("Order");
11    });
12    pt.playFromStart();
13 }
14
15● @FXML
16 void changeToStoreScreen(ActionEvent event) {
17     this.controllerScreen.showStoreScreen();
18 }
19● @FXML
20 void changeToAddBookScreen(ActionEvent event) {
21     this.controllerScreen.showAddBookScreen();
22 }
23
24● @FXML
25 void changeToAddCDScreen(ActionEvent event) {
26     this.controllerScreen.showAddCDScreen();
27 }
28
29● @FXML
30 void changeToAddDVDScreen(ActionEvent event) {
31     this.controllerScreen.showAddDVDScreen();
32 }
33
34● @FXML
35 void changeToCartScreen(ActionEvent event) {
36     this.controllerScreen.showCartScreen();
37 }
38
39● @FXML
40 void updateFiller(InputMethodEvent event) {
41     System.out.println(event.toString());
42 }
43
44
45 }

```

Cart

Options

CART

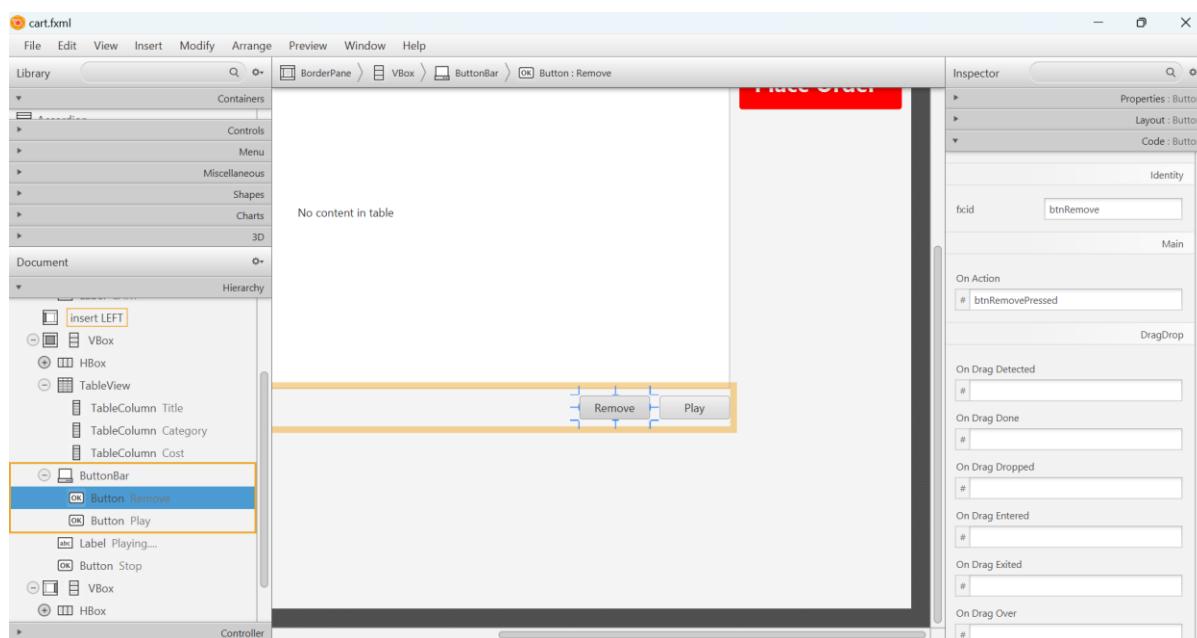
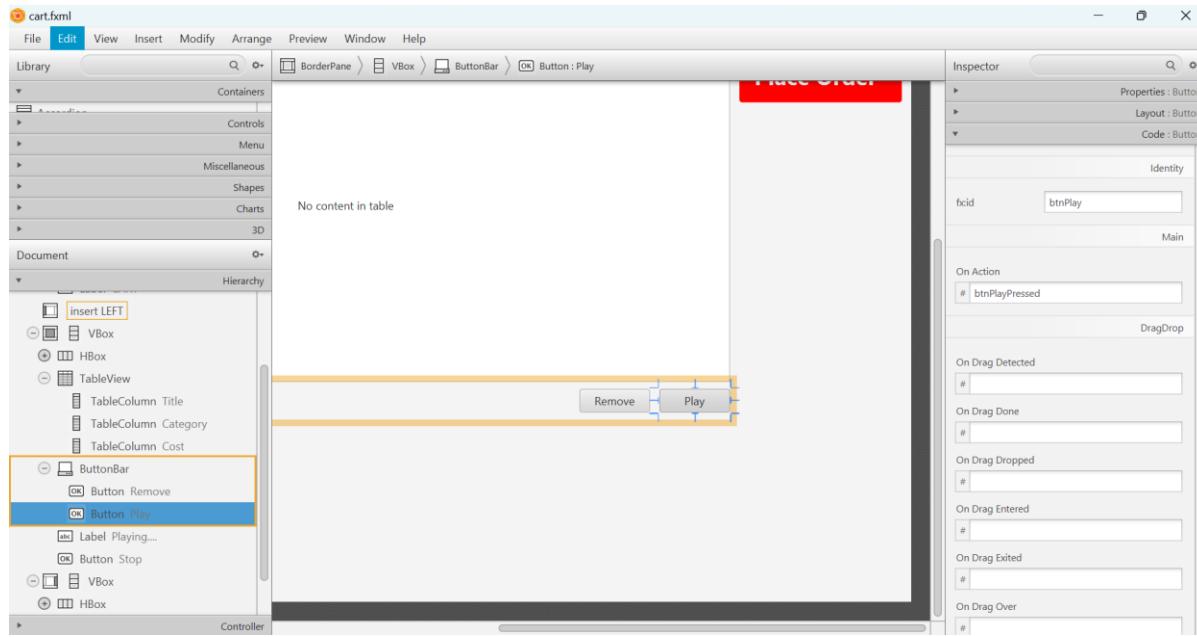
Filter By ID By Title

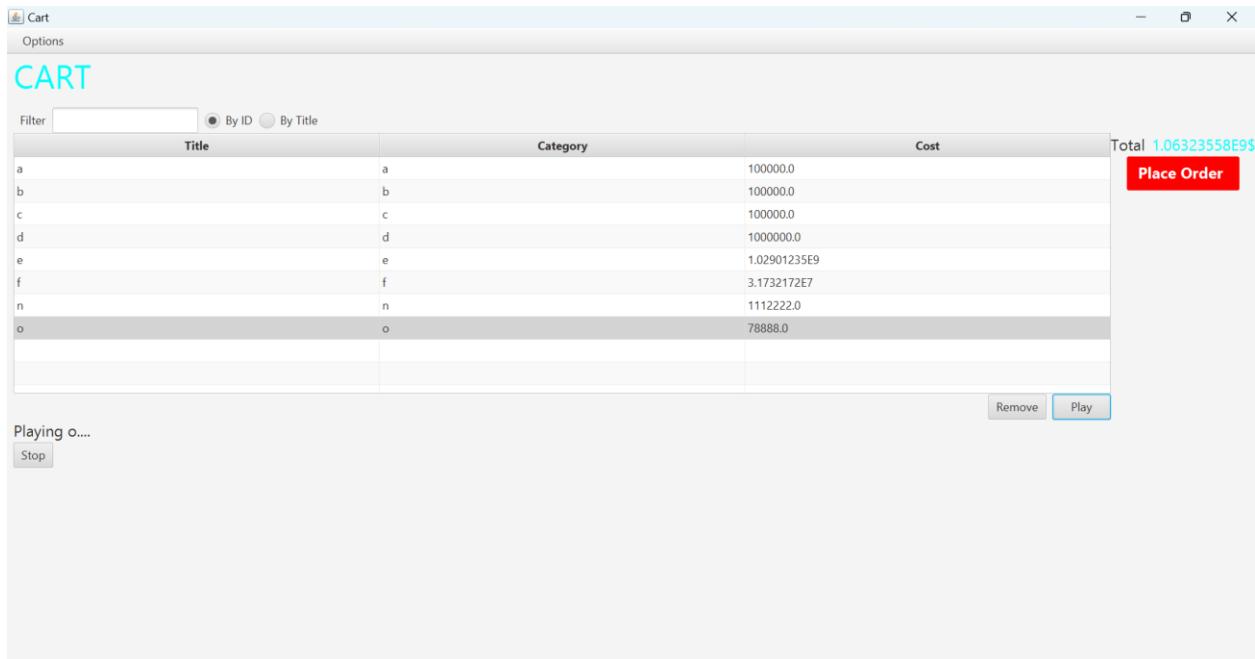
Title	Category	Cost
a	a	100000.0
b	b	100000.0
c	c	100000.0
d	d	1000000.0
e	e	1.02901235E9
f	f	3.1732172E7
n	n	1112222.0
o	o	78888.0
		Total 1.06323558E9\$

Place Order

Remove Play

7. Updating buttons based on selected item in TableView – ChangeListener:





8. Deleting a media:

```
@FXML
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    totalCost.setText(cart.totalCost()+"$");
}
```

9. Filter items in cart – FilteredList:

Code : TextField	Code : RadioButton	Code : RadioButton
Identity	Identity	Identity
fx:id tffilter	fx:id radioBtnFilterId	fx:id radioBtnFilterTitle
Main	Main	Main
On Action	On Action	On Action
# updateFilter	#	#

Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```

    void updateButtonBar(Media media) {
        btnNew.setsetVisible(true);
        if (media instanceof Playable) {
            btnPlay.setsetVisible(true);
        } else {
            btnPlay.setsetVisible(false);
        }
    }

    void showFilterMedia(String searchString) {
        if (searchString.isEmpty()) {
            tblMedia.setItems(this.cart.getItemsOrdered());
        } else {
            if (radioBtnFilterId.isSelected()) {
                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(), item -> item.getId() == Integer.parseInt(searchString)));
            } else
                tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(), item -> item.getTitle().contains(searchString)));
        }
    }

    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(
            new PropertyValueFactory<Media, String>("title"));

        colMediaCategory.setCellValueFactory(
            new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(
            new PropertyValueFactory<Media, Float>("Cost"));
    }
}

    Media added!
    Media added!
    Media added!
    Removed
    Newword
    Removed

```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/screen/CartScreenController.java - Eclipse IDE

```

    tblMedia.setItems(this.cart.getItemsOrdered());
    totalCost.setText(cart.totalCost() + "$");
    btnPlay.setsetVisible(false);
    playingMedia.setVisible(false);
    btnStop.setsetVisible(false);
    tfFilter.textProperty().addListener(new ChangeListener<String>() {
        @Override
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
            showFilterMedia(newValue);
        }
    });

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
                totalCost.setText(cart.totalCost() + "$");
            }
        });
}

    @FXML
    void btnRemovePressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        if (media != null)
            cart.removeItem(media);
    }
}

    Media added!
    Media added!
    Media added!
    Removed
    Removed
    Removed

```

10. Complete the Aims GUI application:

- Add Book to Store Screen:

The screenshot shows the Eclipse IDE interface with the file `AddBookToStoreScreen.java` open. The code implements a screen for adding books to a store. It includes methods for updating the panel, adding media to the store, and handling author input. The code uses Java Swing components like `JLabel`, `JTextField`, and `JButton`.

```
1 package hust.soict.dsai.aims.screen;
2
3 import java.awt.Dimension;
4
5 import javax.swing.*;
6
7 import hust.soict.dsai.aims.cart.Cart;
8 import hust.soict.dsai.aims.media.Book;
9 import hust.soict.dsai.aims.store.Store;
10
11 public class AddBookToStoreScreen extends AddItemToStoreScreen {
12     private JTextField listAuthor;
13
14     public AddBookToStoreScreen(Store store, Cart cart, ControllerScreen c) {
15         super(store, cart, c);
16     }
17
18     @Override
19     void updateAdd	JPanel panel) {
20         this.numberInput = 5;
21
22         JLabel listAuthorLabel = new JLabel("Authors Names are separated by a comma", JLabel.TRAILING);
23         panel.add(listAuthorLabel);
24         listAuthor = new JTextField();
25         listAuthor.setPreferredSize(new Dimension(150, 20));
26         listAuthorLabel.setLabelFor(listAuthor);
27         panel.add(listAuthor);
28
29         JButton tes = new JButton("Add");
30         tes.setVisible(false);
31         panel.add(tes);
32         panel.setPreferredSize(new Dimension(100, 300));
33         addBtn = new JButton("Add");
34         addBtn.addActionListener(e -> {
35             addMediaToStore();
36         });
37         panel.add(addBtn);
38     }
39
40     public void addMediaToStore() {
41         String title = this.title.getText();
42         String listAuthor = this.listAuthor.getText();
43         String[] arrayAuthor = listAuthor.split(" ");
44         String[] arrayAuthor = listAuthor.split(" ");
45         String category = this.category.getText();
46         float cost = Float.parseFloat(this.cost.getText());
47         Book book = new Book(title, category, cost);
48         for(String author:arrayAuthor) {
49             book.addAuthor(author);
50         }
51         this.store.addMedia(book);
52         JOptionPane.showMessageDialog(null, "Add Book successfully!");
53         clearTextField();
54     }
55
56     public void clearTextField(){
57         this.title.setText("");
58         this.listAuthor.setText("");
59         this.cost.setText("");
60         this.category.setText("");
61     }
62
63 }
64
```

- Add CD to Store Screen:

The screenshot shows the Eclipse IDE interface with the file `AddCompactDiscToStoreScreen.java` open. The code implements a screen for adding compact discs to a store. It includes methods for updating the panel, adding media to the store, and handling artist and track input. The code uses Java Swing components like `JLabel`, `JTextField`, and `JButton`.

```
1 package hust.soict.dsai.aims.screen;
2
3 import java.awt.Dimension;
4
5 import javax.swing.*;
6
7 import hust.soict.dsai.aims.cart.Cart;
8 import hust.soict.dsai.aims.media.CompactDisc;
9 import hust.soict.dsai.aims.media.Track;
10 import hust.soict.dsai.aims.store.Store;
11
12 public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
13     private JTextField artist;
14     private JTextField listTrack;
15
16     public AddCompactDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
17         super(store, cart, c);
18     }
19
20     @Override
21     void updateAdd	JPanel panel) {
22         this.numberInput = 6;
23
24         JLabel artistLabel = new JLabel("Artist", JLabel.TRAILING);
25         panel.add(artistLabel);
26         artist = new JTextField();
27         artist.setPreferredSize(new Dimension(150, 20));
28         artistLabel.setLabelFor(artist);
29         artistLabel.setLabelFor(artist);
30         panel.add(artist);
31         JLabel listTrackLabel = new JLabel("List track (each track separated by a comma Ex: track-length)", JLabel.TRAILING);
32         panel.add(listTrackLabel);
33         listTrack = new JTextField();
34         listTrackLabel.setLabelFor(listTrack);
35         panel.add(listTrack);
36         JButton tes = new JButton("Add");
37         tes.setVisible(false);
38         panel.add(tes);
39         panel.setPreferredSize(new Dimension(100, 300));
40         addBtn = new JButton("Add");
41         addBtn.addActionListener(e -> {
42             addMedia();
43         });
44     }
45 }
```

```

42     addBtn = new JButton("Add");
43     addBtn.addActionListener(e -> {
44         addMedia();
45     });
46     panel.add(addBtn);
47 }
48
49 public void addMedia() {
50     String title = this.title.getText();
51     String listTrack = this.listTrack.getText();
52     String category = this.category.getText();
53     float cost = Float.parseFloat(this.cost.getText());
54     String artist = this.artist.getText();
55     String[] arrayTrack = listTrack.trim().split(",");
56
57     CompactDisc cd = new CompactDisc(artist, title, category, "", cost);
58
59     for (String track : arrayTrack) {
60         String titleTrack = track.split("-")[0].trim();
61         int lengthTrack = Integer.parseInt(track.split("-")[1].trim());
62         Track newTrack = new Track(titleTrack, lengthTrack);
63         cd.addTrack(newTrack);
64     }
65
66     this.store.addMedia(cd);
67     JOptionPane.showMessageDialog(null, "Add CD successfully!");
68     clearTextField();
69 }
70
71 public void clearTextField() {
72     this.title.setText("");
73     this.listTrack.setText("");
74     this.cost.setText("");
75     this.artist.setText("");
76     this.category.setText("");
77 }
78 }
79
80 }

```

- Add DVD to Store Screen:

```

1 Eclipse - AIMSPProject/src/hust/soict/dsai/aims/screen/AddDigitalVideoDiscToStoreScreen.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
2 AddBookToStoreScreen.java AddCompactDiscToStoreScreen.java AddDigitalVideoDiscToStoreScreen.java CartScreen.java AddItemToStoreScreen.java
3 package hust.soict.dsai.aims.screen;
4
5 import java.awt.Dimension;
6
7 import hust.soict.dsai.aims.cart.Cart;
8 import hust.soict.dsai.aims.media.DigitalVideoDisc;
9 import hust.soict.dsai.aims.store.Store;
10
11 public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
12     private JTextField director;
13     private JTextField length;
14
15     public AddDigitalVideoDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
16         super(store, cart, c);
17     }
18
19     @Override
20     void updateUI(JPanel panel) {
21         panel.setLayout(new GridLayout(0, 2));
22         // SpringLayout layout = new SpringLayout();
23         // JPanel p = new JPanel(layout);
24         JLabel directorLabel = new JLabel("Director", JLabel.TRAILING);
25         panel.add(directorLabel);
26         director = new JTextField(20);
27         director.setPreferredSize(new Dimension(150, 20));
28         director.addActionListener(e -> {
29             directorLabel.setLabelFor(director);
30         });
31         panel.add(director);
32         JLabel lengthLabel = new JLabel("Length", JLabel.TRAILING);
33         panel.add(lengthLabel);
34         length = new JTextField(20);
35         length.setPreferredSize(new Dimension(100, 30));
36         length.addActionListener(e -> {
37             length.setText(e.getActionCommand());
38         });
39         panel.add(length);
40         addBtn = new JButton("Add");
41         addBtn.addActionListener(e -> {
42             addMedia();
43         });
44         panel.add(addBtn);
45
46     public void addMedia() {
47         String title = this.title.getText();
48         String director = this.director.getText();
49         String category = this.category.getText();
50         float cost = Float.parseFloat(this.cost.getText());
51         int length = Integer.parseInt(this.length.getText());
52         DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
53         this.store.addMedia(dvd);
54         JOptionPane.showMessageDialog(null, "Add DVD successfully!");
55         clearTextField();
56     }
57
58     public void clearTextField(){
59         this.title.setText("");
60         this.director.setText("");
61         this.cost.setText("");
62         this.length.setText("");
63         this.category.setText("");
64     }
65
66
67 }

```

11. Check all the previous source codes to catch/handle/delegate runtime exceptions:

Eclipse - AIMSProject/src/hust/soict/dsai/aims/media/CompactDisc.java - Eclipse IDE

```

43     }
44 
45     return totalLength;
46 }
47 
48 @Override
49 public void play() throws PlayerException{
50     if (this.getLength() > 0) {
51         System.out.println("Compactdisc: " + this.getTitle());
52         System.out.println("CompactDisc Artist: " + this.getArtist());
53         System.out.println("Total length: " + this.getLength());
54         java.util.Iterator<Track> iter = tracks.iterator();
55         Track nextTrack;
56         while (iter.hasNext()) {
57             nextTrack = (Track) iter.next();
58             try {
59                 nextTrack.play();
60             } catch(PlayerException e) {
61                 throw e;
62             }
63         }
64     }
65     else {
66         throw new PlayerException("Error: CD length is non-positive!");
67     }
68 }
69 
70 System.out.println("-----Play All Tracks-----");
71 for (Track track: tracks) {
72     track.play();
73 }
74 }
```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/media/DigitalVideoDisc.java - Eclipse IDE

```

21     }
22     public DigitalVideoDiscs(String title, String category, String director, int length, float cost) {
23         super(nbDigitalVideoDiscs + 1, title, category, cost, director, length);
24         nbDigitalVideoDiscs += 1;
25     }
26     public boolean isMatch(String title) {
27         return this.getTitle().equals(title);
28     }
29     @Override
30     public String toString() {
31         return "DVD " + this.getTitle() + " " + this.getCategory() + " " + this.getDirector() + " " + String.valueOf(this.getLength()) + " " + String.valueOf(this.getCost());
32     }
33     @Override
34     public void play() throws PlayerException{
35         if (this.getLength() > 0) {
36             System.out.println("Playing DVD: " + this.getTitle());
37             System.out.println("DVD length: " + this.getLength());
38         } else {
39             throw new PlayerException("Error: DVD length is non-positive!");
40         }
41     }
42     @Override
43     public int compareTo(Media other) {
44         if (!other instanceof DigitalVideoDisc) {
45             return super.compareTo(other);
46         }
47         DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;
48         int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());
49         if (titleComparison != 0) {
50             return titleComparison;
51         }
52     }
53 }
```

Eclipse - AIMSProject/src/hust/soict/dsai/aims/media/Track.java - Eclipse IDE

```

25     }
26     public String getTitle() {
27         return title;
28     }
29     public int getLength() {
30         return length;
31     }
32     @Override
33     public boolean equals(Object obj) {
34         if (obj == null || !(obj instanceof Track)) {
35             return false;
36         }
37         Track other = (Track) obj;
38         return (this.title != null && this.title.equals(other.title)) && this.length == other.length;
39     }
40     @Override
41     public void play() throws PlayerException {
42         if (this.getLength() > 0) {
43             System.out.println("Playing track: " + this.getTitle());
44             System.out.println("Track length: " + this.getLength());
45         } else {
46             throw new PlayerException("Error: Track length is non-positive!");
47         }
48     }
49 }
```

12. Create a class which inherits from Exception:

The screenshot shows the Eclipse IDE interface with the title bar "Eclipse - AIMSProject/src/hust/soict/dsai/aims/exception/PlayerException.java - Eclipse IDE". The code editor displays the following Java code:

```
1 package hust.soict.dsai.aims.exception;
2
3 public class PlayerException extends Exception {
4
5     public PlayerException() {
6         // TODO Auto-generated constructor stub
7     }
8
9     public PlayerException(String message) {
10        super(message);
11        // TODO Auto-generated constructor stub
12    }
13
14     public PlayerException(Throwable cause) {
15        super(cause);
16        // TODO Auto-generated constructor stub
17    }
18
19     public PlayerException(String message, Throwable cause) {
20        super(message, cause);
21        // TODO Auto-generated constructor stub
22    }
23
24     public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
25        super(message, cause, enableSuppression, writableStackTrace);
26        // TODO Auto-generated constructor stub
27    }
28
29 }
```

The code editor has tabs for "Problems", "Javadoc", "Declaration", and "Console". The "Console" tab shows the output: "Media added! Media added! Media added! Removed Removed Removed". The status bar at the bottom shows "Writable", "Smart Insert", and "1:1:0".

13. Update the Aims class:

The screenshot shows the Eclipse IDE interface with the title bar "Eclipse - AIMSProject/src/hust/soict/dsai/aims/Aims.java - Eclipse IDE". The code editor displays the following Java code:

```
19     public static void main(String[] args) throws PlayerException, LimitExceededException {
20         Scanner scanner = new Scanner(System.in);
21         int choice;
22         do {
23             showMenu();
24             choice = scanner.nextInt();
25             scanner.nextLine();
26
27             switch (choice) {
28                 case 1:
29                     viewStore(scanner);
30                     break;
31                 case 2:
32                     updateStore(scanner, store);
33                     break;
34                 case 3:
35                     setCurrentCart(scanner);
36                     break;
37                 case 0:
38                     System.out.println("Exiting AIMS. Goodbye!");
39                     break;
40                 default:
41                     System.out.println("Invalid choice! Please choose a number between 0-3.");
42             }
43         } while (choice != 0);
44         scanner.close();
45     }
46
47     public static void showMenu() {
48         System.out.println("1. View Store");
49         System.out.println("2. Update Store");
50         System.out.println("3. Set Current Cart");
51         System.out.println("0. Exit");
52     }
53 }
```

The code editor has tabs for "Problems", "Javadoc", "Declaration", and "Console". The "Console" tab shows the output: "Media added! Media added! Media added! Removed Removed Removed". The status bar at the bottom shows "Writable", "Smart Insert", and "47 : 1 : 1279".