LAB 05

Name: Nguyễn Vũ Thủy

ID: 20235562

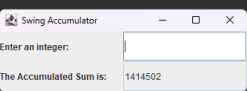# 1. Swing components

1.1. AWTAccumulator

1.2. SwingAccumulator

2. Organizing Swing components with Layout Managers

```java
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);
    }

    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")) {
                if (tfDisplay.getText().length() >= 1) {
                    tfDisplay.setText(tfDisplay.getText().substring(0, tfDisplay.getText().length()-1));
                }
            }else if (button.equals("C")) {
                tfDisplay.setText("");
            }else {
                tfDisplay.setText(tfDisplay.getText());
            }
        }
    }

    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);
```

3. Create a graphical user interface for AIMS with Swing

3.1.   View Store Screen
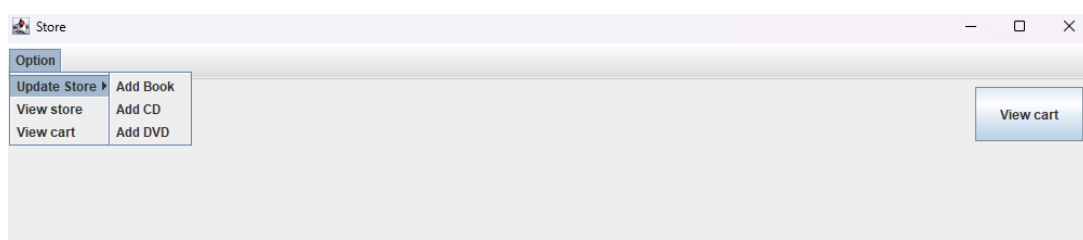
3.2. Adding more user interaction

```java
JButton addCartBtn = new JButton("Add to Cart");
addCartBtn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
container.add(addCartBtn);


if (media instanceof Playable) {
    JButton playBtn = new JButton("Play");
    playBtn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            JDialog playDialog = new JDialog();
            JPanel mainGui = new JPanel(new BorderLayout());
            mainGui.setBorder(new EmptyBorder(20, 20, 20, 20));

            // Display Playing Message
            mainGui.add(new JLabel("Playing... " + media.getTitle()), BorderLayout.CENTER);
            System.out.println(media.getTitle());
            // Close Button
            JPanel buttonPanel = new JPanel(new FlowLayout());
            JButton close = new JButton("Stop");
            close.addActionListener(ev -> {
                playDialog.setVisible(false);
                System.out.println("Stopped playing.");
            });
            buttonPanel.add(close);
            mainGui.add(buttonPanel, BorderLayout.SOUTH);

            playDialog.setContentPane(mainGui);
            playDialog.setLocationRelativeTo(playBtn);
            playDialog.pack();

            // Show Dialog
            playDialog.setVisible(true);
        }
    });
    container.add(playBtn);
```

4. JavaFX API
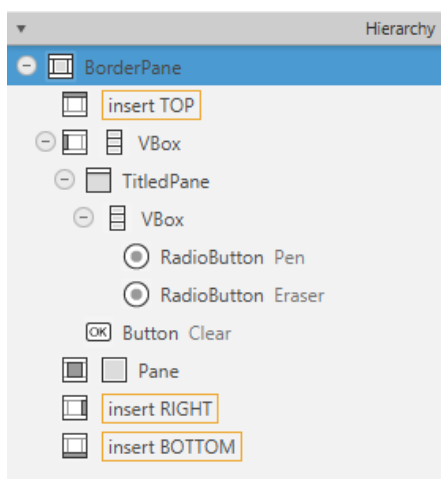
Inspector   🔍   ⚙▾

▶   Properties : BorderPane

▼   Layout : BorderPane

Internal

Padding	☐	☐	☐	☐
	8	❯ 8	8	8

Size

Min Width	USE_PREF_SIZE	▾
Min Height	USE_PREF_SIZE	▾
Pref Width	640	▾
Pref Height	480	▾
Max Width	USE_PREF_SIZE	▾
Max Height	USE_PREF_SIZE	▾
Width	640	
Height	480	

Position

Layout X	0
Layout Y	0

Transforms

Rotate	0	◯

	X	Y	Z
Rotation Axis	0	0	1
Scale X	1		
Scale Y	1		
Scale Z	1		
Translate X	0		
Translate Y	0		
Translate Z	0		

Bounds

Layout Bounds	0,0 640x480	▾
Bounds In Local	0,0 640x480	▾
Bounds In Parent	0,0 640x480	▾

Layout : VBox

Border Pane Constraints

Alignment	CENTER ▼

Margin: ☐ ☐ ☐ ☐
| 0 | > | 8 | 0 | 0 |

Internal

Padding: ☐ ☐ ☐ ☐
| 0 | > | 0 | 0 | 0 |

Spacing: 8

Specific

Fill Width ✓

Size

Min Width	USE_COMPUTED_SIZE ▼
Min Height	USE_COMPUTED_SIZE ▼
Pref Width	USE_COMPUTED_SIZE ▼
Pref Height	USE_COMPUTED_SIZE ▼
Max Width	USE_COMPUTED_SIZE ▼
Max Height	MAX_VALUE ▼
Width	75
Height	464

Position

Inspector 🔍 ⚙▾

▶ Properties : Pane
▶ Layout : Pane
▼ Code : Pane

Identity

fx:id | drawingAreaPane

DragDrop

On Drag Detected
| ▼

On Drag Done
| ▼

On Drag Dropped
| ▼

⊖ ▣ BorderPane
   □ insert TOP
  ⊖ ▣ 目 VBox
    ⊖ □ TitledPane
      ⊖ 目 VBox
        ◉ RadioButton Pen
        ◉ RadioButton Eraser
     [OK] Button Clear
   ▣ □ Pane
   □ insert RIGHT
   □ insert BOTTOM

Style

| -fx-background-color | white | + | ▼ |

Style Class

| | + | ▼ |

Stylesheets

| + |

Id

| |

Extras

On Context Menu Requested

\#

On Mouse Clicked

\#

On Mouse Dragged ⚙

\# drawingAreaMouseDragged

On Mouse Entered

\#

On Mouse Exited

BorderPane
    insert TOP
    VBox
        TitledPane
            VBox
                ⦿ RadioButton Pen
                ⦿ RadioButton Eraser
            OK Button Clear
        ▣ ▢ Pane
        insert RIGHT
        insert BOTTOM

Properties : Button

Text

Text	Clear
Font	System 12px
Text Fill	■ BLACK
Wrap Text	☐
Text Alignment	
Text Overrun	ELLIPSIS
Ellipsis String	...
Underline	☐
Line Spacing	0

Specific

Layout : Button

VBox Constraints

Vgrow INHERIT

Margin ☐ ☐ ☐ ☐
0 › 0 0 0

Internal

Padding ☐ ☐ ☐ ☐
0 › 0 0 0

Size

Min Width USE_COMPUTED_SIZE
Min Height USE_COMPUTED_SIZE
Pref Width USE_COMPUTED_SIZE
Pref Height USE_COMPUTED_SIZE
Max Width MAX_VALUE
Max Height USE_COMPUTED_SIZE
Width 75
Height 25

Code : Button

On Action
clearButtonPressed

DragDrop

On Drag Detected
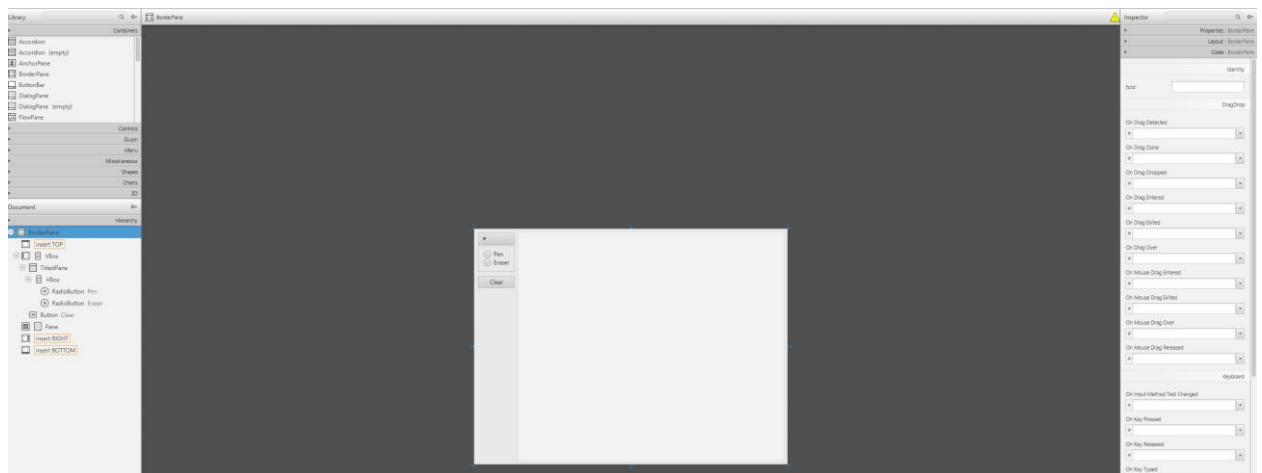#

On Drag Done
#

On Drag Dropped
#

On Drag Entered
#

On Drag Exited

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.RadioButton?>
6  <?import javafx.scene.control.TitledPane?>
7  <?import javafx.scene.control.ToggleGroup?>
8  <?import javafx.scene.layout.BorderPane?>
9  <?import javafx.scene.layout.Pane?>
10 <?import javafx.scene.layout.VBox?>
11
   Bind to grammar/schema...
12 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xmlns="http://javafx.com/javaf
13    <left>
14       <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
15          <BorderPane.margin>
16             <Insets right="8.0" />
17          </BorderPane.margin>
18          <children>
19             <TitledPane>
20                <content>
21                   <VBox>
22                      <children>
23                         <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Pen">
24                            <toggleGroup>
25                               <ToggleGroup fx:id="identity" />
26                            </toggleGroup>
27                         </RadioButton>
28                         <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Eraser" toggleGroup="$identity" />
29                      </children>
30                   </VBox>
31                </content>
32             </TitledPane>
33             <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />
34          </children>
35       </VBox>
36    </left>
37    <center>
38       <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
39    </center>
40    <padding>
41       <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
42    </padding>
43 </BorderPane>
44
```

```java
1  package hust.soict.dsai.javafx;
2
3  import javafx.event.ActionEvent;
11
12
13 public class PainterController {
14
15    private int color;
16
17    @FXML
18    private ToggleGroup identity;
19
20    @FXML
21    void chooseOption(ActionEvent event) {
22       String button = ((RadioButton)event.getSource()).getText();
23       if (button.equals("Pen")) {
24          System.out.println("1");
25          color = 1;
26       }else {
27          System.out.println("0");
28          color = 0;
29       }
30    }
31
32    @FXML
33    private Pane drawingAreaPane;
34
35    @FXML
36    void clearButtonPressed(ActionEvent event) {
37       drawingAreaPane.getChildren().clear();
38    }
39
40    @FXML
41    void drawingAreaMouseDragged(MouseEvent event) {
42       if (color == 1 && event.getX()>=0) {
43          Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.BLACK);
44          drawingAreaPane.getChildren().add(newCircle);
45       }else if(color == 0 && event.getX()>=0){
46          Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.WHITE);
47          drawingAreaPane.getChildren().add(newCircle);
48       }
49    }
50
51 }
```

```
1 package hust.soict.dsai.javafx;
2
3  import javafx.application.Application;
8
9 public class Painter extends Application{
10
11     @Override
12     public void start(Stage stage) throws Exception{
13         Parent root = FXMLLoader.load(getClass().getResource("/hust/soict/dsai/javafx/Painter.fxml"));
14
15         Scene scene = new Scene(root);
16         stage.setTitle("Painter");
17         stage.setScene(scene);
18         stage.show();
19     }
20
21     public static void main(String[] args) {
22         launch(args);
23     }
24 }
25
```


Handle Erase Function

```
    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        if (color == 1 && event.getX()>=0) {
            Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.BLACK);
            drawingAreaPane.getChildren().add(newCircle);
        }else if(color == 0 && event.getX()>=0){
            Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.WHITE);
            drawingAreaPane.getChildren().add(newCircle);
        }
    }
}
```

5. Setting up the View Cart Screen with ScreenBuilder

5.2. Setting up the TOP area
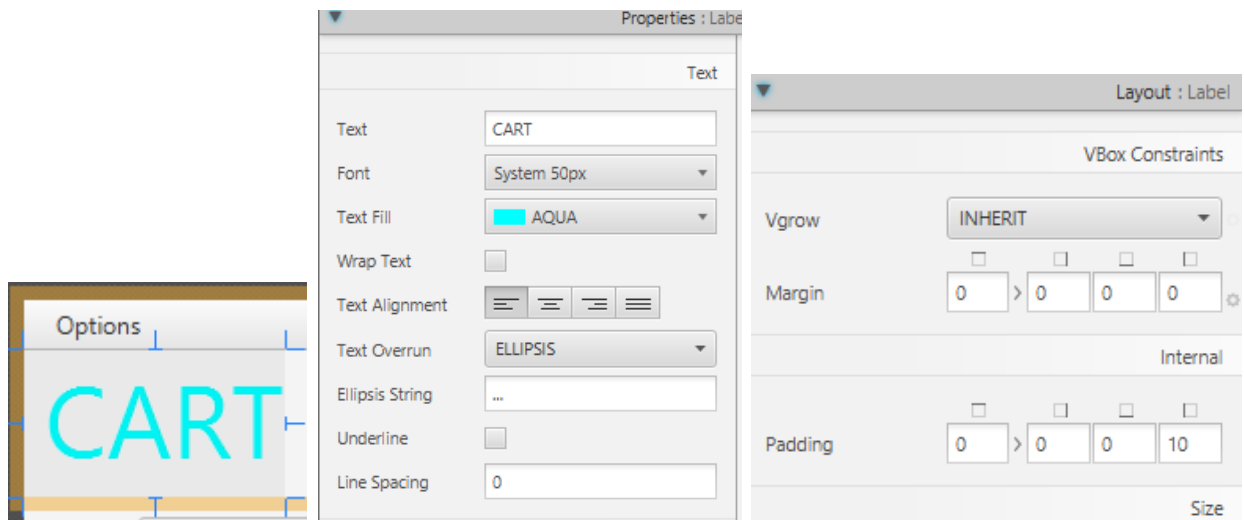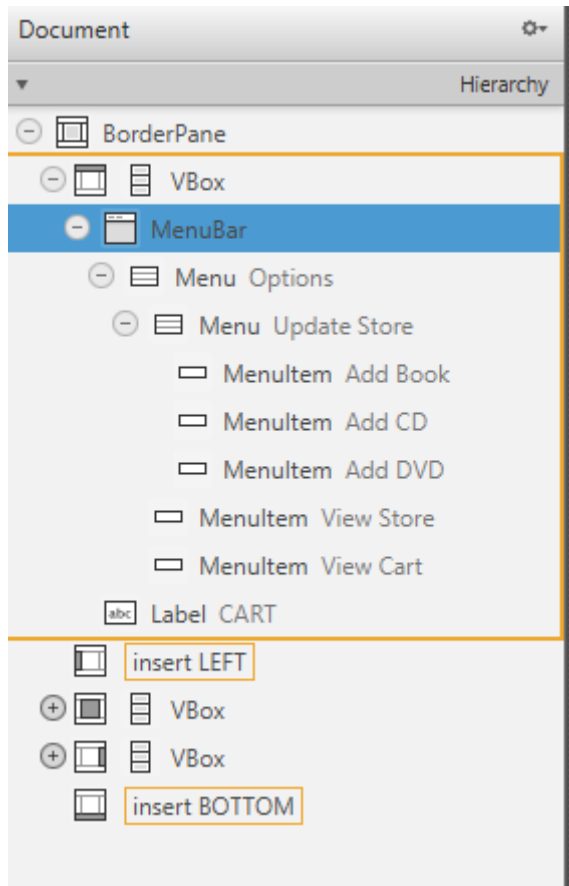
5.3. Setting up the CENTER area
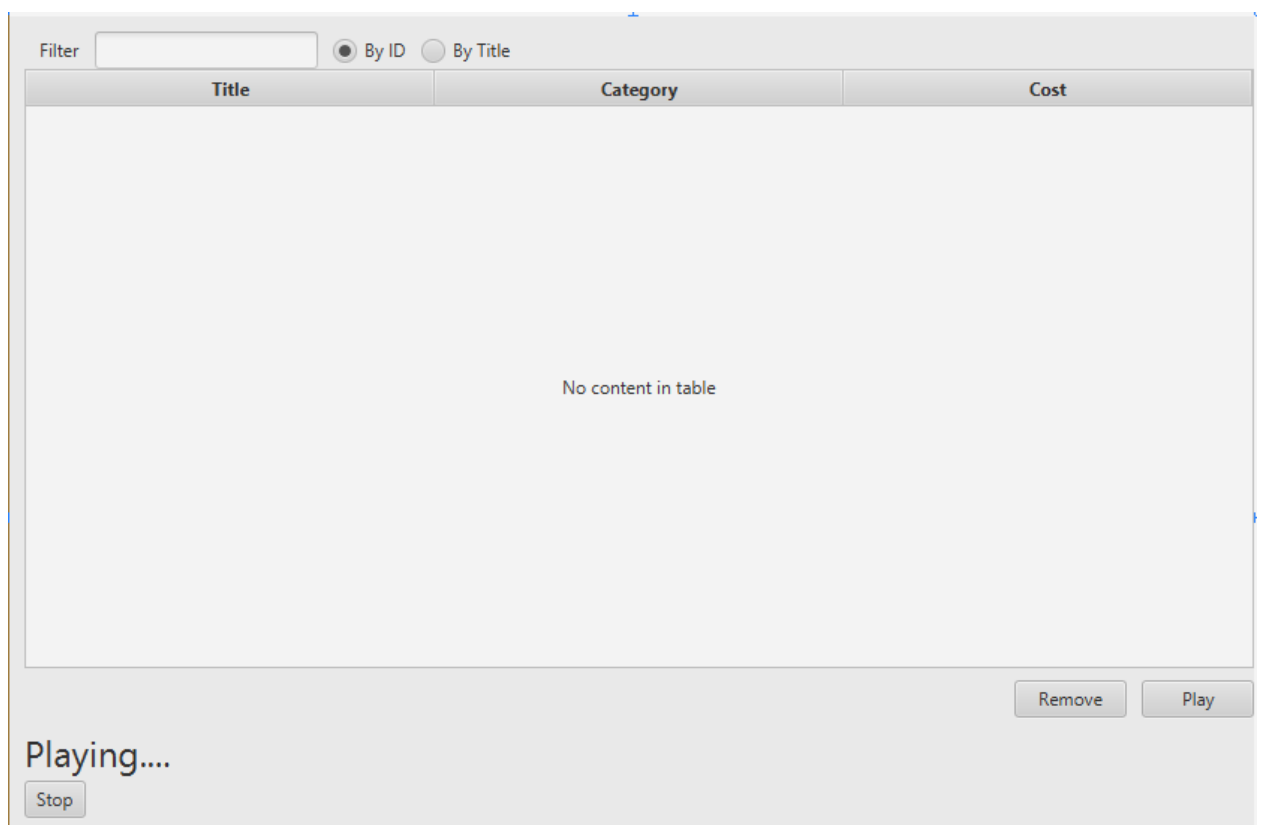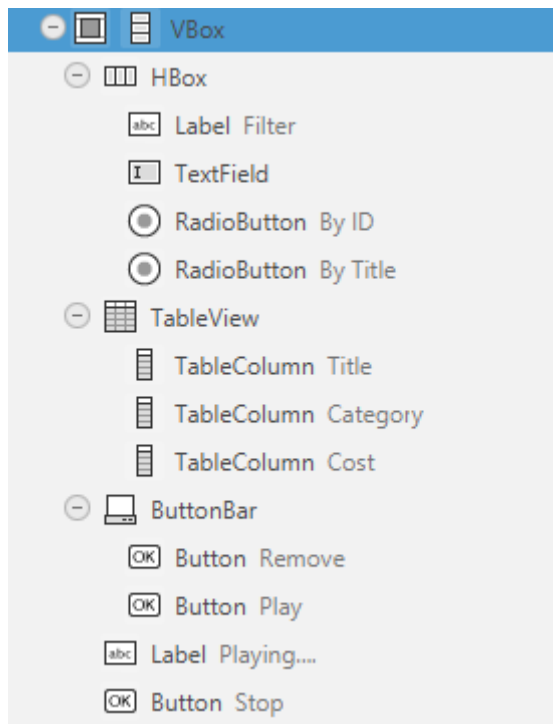
5.4. Setting up the RIGHT area
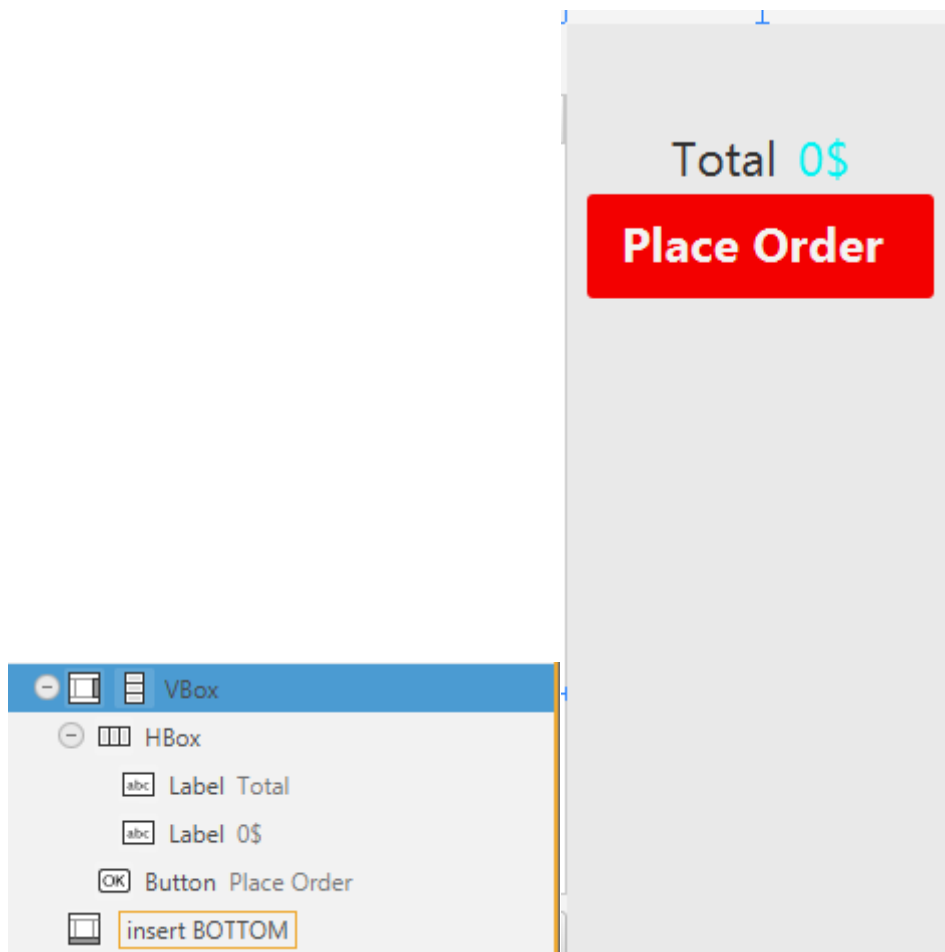
6. Integrating JavaFX into Swing application – The JFXPanel class

```java
package hust.soict.dsai.aims.screen;

import java.awt.Dimension;

public class CartScreen extends JFrame{

    private Cart cart;
    private ControllerScreen controllerScreen;

    public CartScreen(Cart cart, ControllerScreen c) {
        super();

        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        this.setVisible(true);
        this.setPreferredSize(new Dimension(1024, 768));
        pack();
        Platform.runLater(new Runnable() {

            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass()
                            .getResource("/hust/soict/dsai/aims/screen/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart,c);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root,1024,768));
                } catch (IOException e) {
                    e.printStackTrace();
                }

            }
        });
    }
```

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

Code : TableView

▼

Identity

fx:id   tblMedia

Main

On Sort

#

DragDrop

Code : TableColumn

▼

Identity

fx:id   colMediaTitle

Edit

On Edit Start

Properties : TableColumn

▶

Layout : TableColumn

▶

Code : TableColumn

▼

Identity

fx:id   colMediaCategory

Edit

On Edit Start

#

Code : TableColumn

▼

Identity

fx:id   colMediaCost

Edit

On Edit Start

#

On Edit Commit
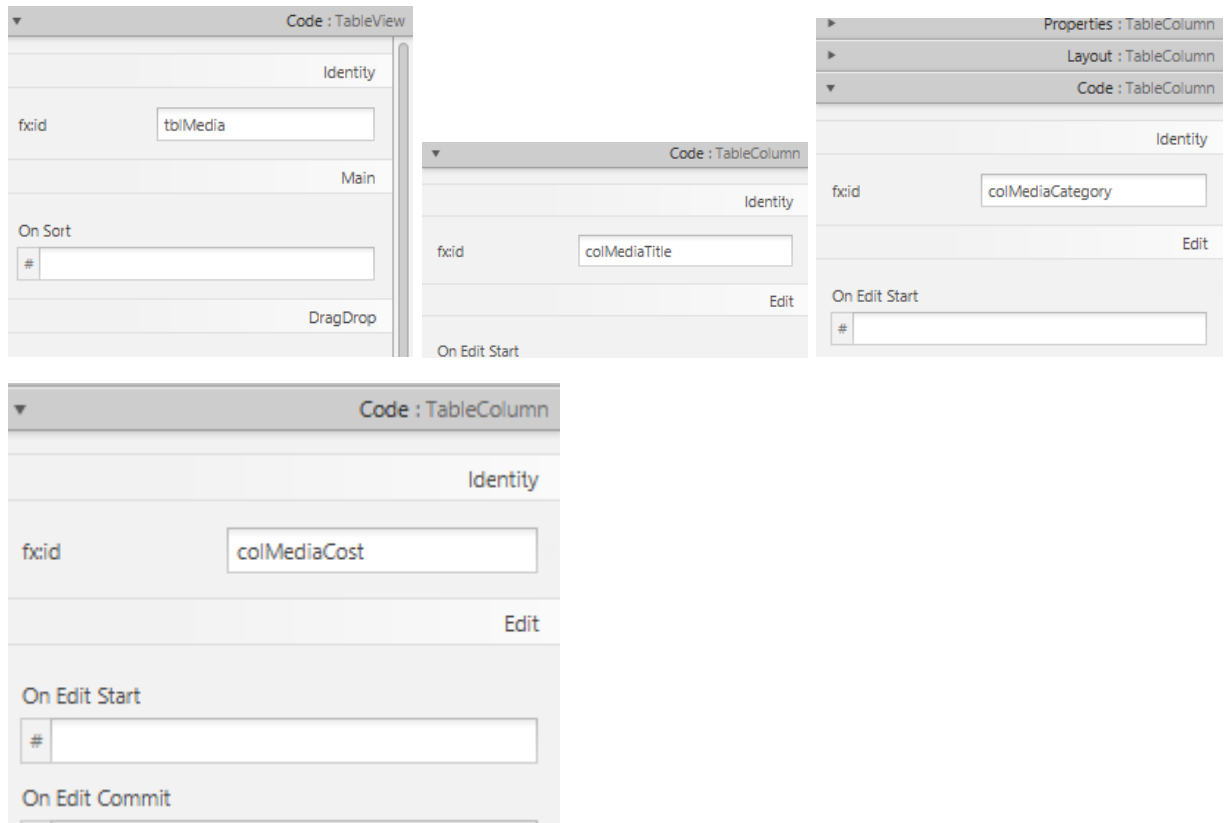
- Update itemsOrdered to ObservableList

Cart

Options

CART

Filter [＿＿＿＿＿]   ● By ID   ○ By Title

Title	Category	Cost
Test Book	Education	4214.0
Arcane Music	Fun	2141.0
Artic Monkey	Pop	2141.0

Total 8496.0$

Place Order

Remove   Play
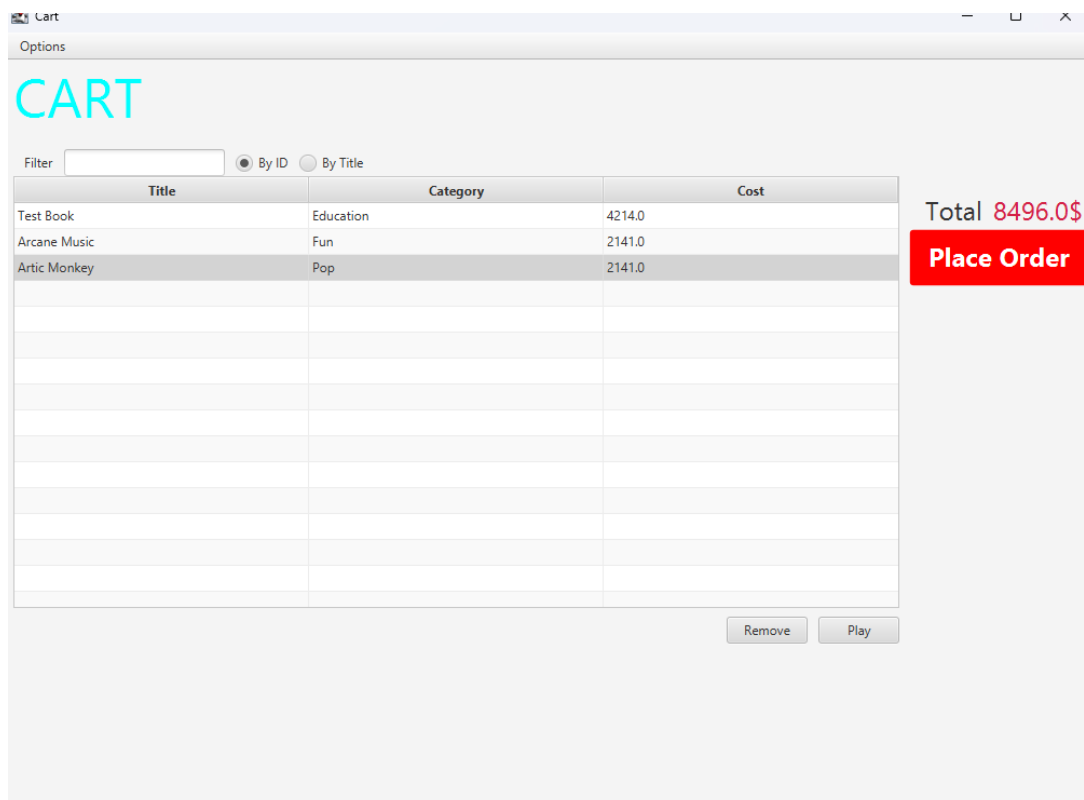
8. Updating buttons based on selected item in TableView – ChangeListener

Code : Button

Identity

fx:id `btnPlay`

Main

On Action
`btnPlayPressed`

DragDrop

Code : Button

Identity

fx:id `btnRemove`

Main

On Action
`btnRemovePressed`

DragDrop

CART

Filter		● By ID ○ By Title	

Title	Category	Cost
Son Tung MTP	Music	21321.0
Arcane Tracks	Music	22511.0
Test Book	Education	21421.0

Total 65253.0$

Place Order

[Remove] [Play]

Playing Son Tung MTP....

[Stop]

9. Deleting a media

```java
136    @FXML
137    void btnRemovePressed(ActionEvent event) {
138        Media media = tblMedia.getSelectionModel().getSelectedItem();
139        cart.removeMedia(media);
140        totalCost.setText(cart.totalCost()+"$");
141    }
142
```

10. Filter items in cart – FilteredList

```
1140      tfFilter.textProperty().addListener(new ChangeListener<String>() {
115
1160          @Override
117          public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
118
119              showFilterMedia(newValue);
120          }
121      });|
122      tblMedia.getSelectionModel().selectedItemProperty().addListener(
```

```
870  void showFilterMedia(String searchString) {
88      if(searchString.isEmpty()) {
89          tblMedia.setItems(this.cart.getItemsOrdered());
90      } else {
91          if(radioBtnFilterId.isSelected()) {
92              tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getId()==Integer.parseInt(searchString)));
93          }else
94              tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getTitle().contains(searchString)));
95      }
96  }
97
```

11. Complete the Aims GUI application
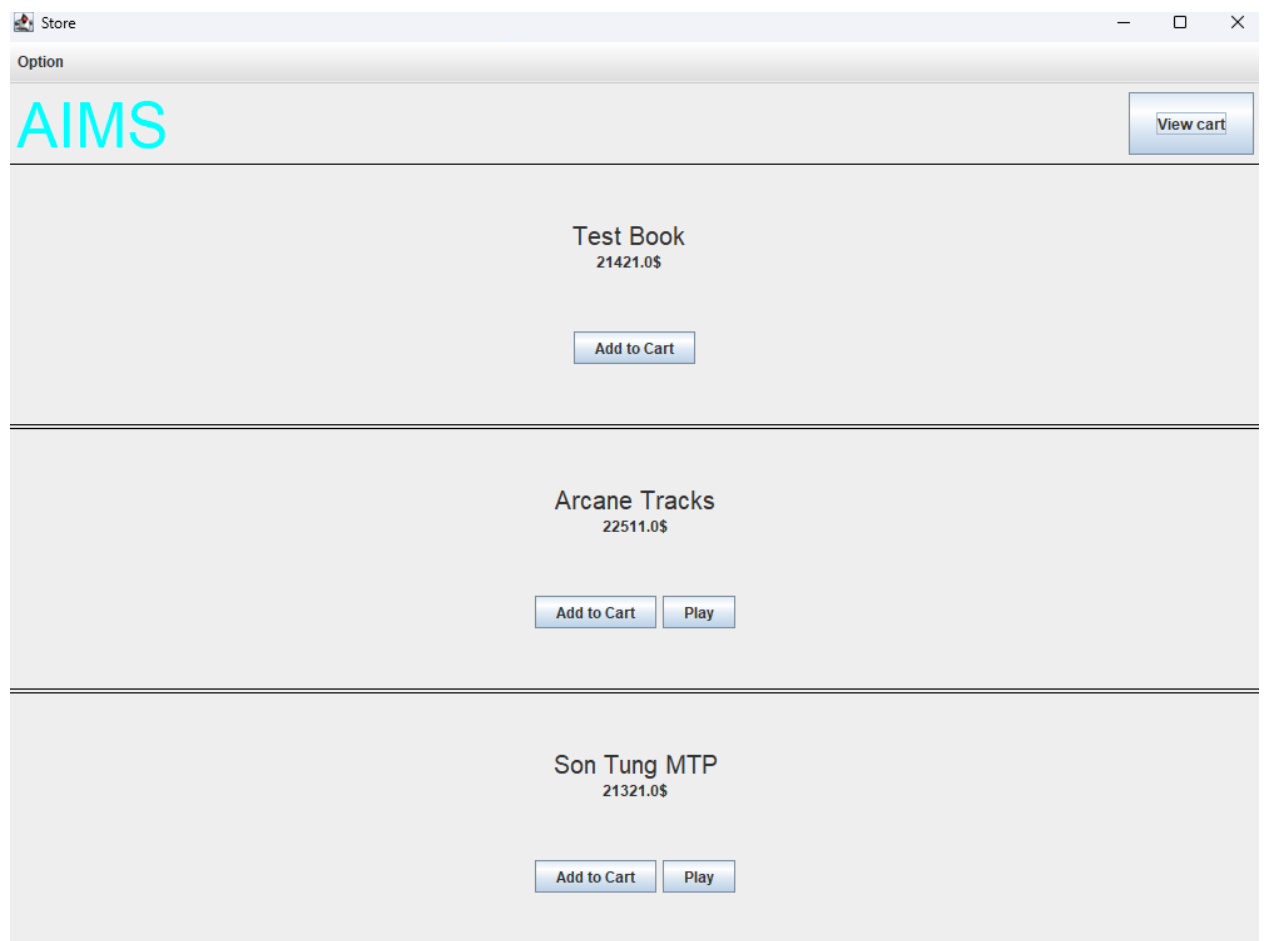
Options

CART

Filter [   ]   ○ By ID   ● By Title

Title	Category	Cost
Test Book	Education	21421.0
Arcane Tracks	Music	22511.0
Son Tung MTP	Music	21321.0

Total 65253.0

Order

[Remove]   [Play]

Add Book

Add Book

Option

AIMS

[View cart]

title [   ]

category [   ]

cost [   ]

Authors(Names are separated by a comma) [   ]

add

Add CD

Add DVD

Filter by Title

Filter by Id

AddItemToStoreScreen

```java
1  package hust.soict.dsai.aims.screen;
2
3  import java.awt.BorderLayout;
28
29  public class AddItemToStoreScreen extends JFrame {
30      protected Store store;
31      protected Cart cart;
32      protected JTextField title;
33      protected JTextField category;
34      protected JTextField cost;
35      protected JButton addBtn;
36      protected ControllerScreen c;
37      protected JPanel center;
38      protected int numberInput=3;
39      JMenuBar createMenuBar() {
40          JMenu menu = new JMenu("Option");
41
42          JMenu smUpdateStore = new JMenu("Update Store");
43          JMenuItem addBookMenu= new JMenuItem("Add Book");
44          addBookMenu.addActionListener(e->{
45              c.showAddBookScreen();
46          });
47          smUpdateStore.add(addBookMenu);
48          JMenuItem addCDMenu=new JMenuItem("Add CD");
49          addCDMenu.addActionListener(e->{
50              c.showAddCDCreen();
51          });
52          smUpdateStore.add(addCDMenu);
53          JMenuItem addDVDMenu= new JMenuItem("Add DVD");
54          addDVDMenu.addActionListener(e->{
55              c.showAddDVDScreen();
56          });
57          smUpdateStore.add(addDVDMenu);
58
59          menu.add(smUpdateStore);
60          JMenuItem viewStoreMenu= new JMenuItem("View store");
61          viewStoreMenu.addActionListener(e->{
62              c.showStoreScreen();
63          });
64          menu.add(viewStoreMenu);
65
66          JMenuItem viewCartMenu= new JMenuItem("View cart");
67          viewCartMenu.addActionListener(e->{
68              c.showCartScreen();
69          });
70          menu.add(viewCartMenu);
71
72          JMenuBar menuBar = new JMenuBar();
73          menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
74          menuBar.add(menu);
75
76          return menuBar;
77      }
78
79      JPanel createHeader() {
80          JPanel header = new JPanel();
81          header.setLayout(new BoxLayout(header, BoxLayou
```

AddDigitalVideoDiscToStoreScreen

```java
1  package hust.soict.dsai.aims.screen;
2
3  import java.awt.Dimension;
14
15 public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
16     private JTextField director;
17     private JTextField length;
18
19     public AddDigitalVideoDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
20         super(store, cart, c);
21
22     }
23     @Override
24     void updateAdd(JPanel panel) {
25         this.numberInput = 6;
26         // SpringLayout layout =new SpringLayout();
27         // JPanel p = new JPanel(layout);
28         JLabel directorLabel = new JLabel("director", JLabel.TRAILING);
29         panel.add(directorLabel);
30         director = new JTextField(2);
31         director.setPreferredSize(new Dimension(150, 20));
32         directorLabel.setLabelFor(director);
33         panel.add(director);
34         JLabel lengthLabel = new JLabel("length", JLabel.TRAILING);
35         panel.add(lengthLabel);
36         length = new JTextField(2);
37         lengthLabel.setLabelFor(length);
38         panel.add(length);
39         JButton tes = new JButton("add");
40         tes.setVisible(false);
41         panel.add(tes);
42         panel.setPreferredSize(new Dimension(100, 300));
43         addBtn = new JButton("add");
44         addBtn.addActionListener(e -> {
45             addMedia();
46         });
47         panel.add(addBtn);
48     }
49
50     public void addMedia() {
51         String title = this.title.getText();
52         String director = this.director.getText();
53         String category = this.category.getText();
54         float cost = Float.parseFloat(this.cost.getText());
55         int length = Integer.parseInt(this.length.getText());
56         DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
57         this.store.addMedia(dvd);
58         JOptionPane.showMessageDialog(null, "add DVD successfully!");
59         clearTextField();
60
61     }
62     public void clearTextField(){
63         this.title.setText("");
64         this.director.setText("");
65         this.cost.setText("");
66         this.length.setText("");
67         this.category.setText("");
```

AddCompactDiscToStoreScreen

```java
package hust.soict.dsai.aims.screen;

import java.awt.Dimension;

public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
    private JTextField artist;
    private JTextField listTrack;

    public AddCompactDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
        super(store, cart, c);

    }

    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 6;

        JLabel artistLabel = new JLabel("artist", JLabel.TRAILING);
        panel.add(artistLabel);
        artist = new JTextField(2);
        artist.setPreferredSize(new Dimension(150, 20));
        artistLabel.setLabelFor(artist);
        panel.add(artist);
        JLabel listTrackLabel = new JLabel("List track (each track separated by a comma Ex: track-length)",
                JLabel.TRAILING);

        panel.add(listTrackLabel);
        listTrack = new JTextField(2);
        listTrackLabel.setLabelFor(listTrack);
        panel.add(listTrack);
        JButton tes = new JButton("add");
        tes.setVisible(false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(100, 300));
        addBtn = new JButton("add");
        addBtn.addActionListener(e -> {
            addMedia();
        });
        panel.add(addBtn);
    }

    public void addMedia() {
        String title = this.title.getText();
        String listTrack = this.listTrack.getText();
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
        String artist = (this.artist.getText());
        String[] arrayTrack = listTrack.trim().split(",");

        CompactDisc cd = new CompactDisc(artist, title, category, "", cost);
        ;
        for (String track : arrayTrack) {
            String titleTrack = track.split("-")[0].trim();
            int lengthTrack = Integer.parseInt(track.split("-")[1].trim());
            Track newTrack = new Track(titleTrack, lengthTrack);
            cd.addTrack(newTrack);
```

AddBookToStoreScreen

```java
package hust.soict.dsai.aims.screen;

import java.awt.Dimension;

public class AddBookToStoreScreen extends AddItemToStoreScreen {
    private JTextField listAuthor;

    public AddBookToStoreScreen(Store store, Cart cart, ControllerScreen c) {
        super(store, cart, c);

    }
    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 5;

        JLabel listAuthorLabel = new JLabel("Authors(Names are separated by a comma)", JLabel.TRAILIN
        panel.add(listAuthorLabel);
        listAuthor = new JTextField(2);
        listAuthor.setPreferredSize(new Dimension(150, 20));
        listAuthorLabel.setLabelFor(listAuthor);
        panel.add(listAuthor);

        JButton tes = new JButton("add");
        tes.setVisible(false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(100, 300));
        addBtn = new JButton("add");
        addBtn.addActionListener(e -> {
            addMediaToStore();
        });
        panel.add(addBtn);
    }

    public void addMediaToStore() {
        String title = this.title.getText();
        String listAuthor = this.listAuthor.getText();
        String[] arrayAuthor=listAuthor.split(",");
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
        Book book = new Book(title, category, cost);
        for(String author:arrayAuthor) {
        book.addAuthor(author);
        }
        this.store.addMedia(book);
        JOptionPane.showMessageDialog(null, "add Book successfully!");
        clearTextField();

    }
    public void clearTextField(){
        this.title.setText("");
        this.listAuthor.setText("");
        this.cost.setText("");
        this.category.setText("");
    }



}
```

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

```java
    public void addMedia(Media media) throws LimitExceededException {
        if ((itemsOrdered.size()) >= MAX_ORDERED) {
            throw new LimitExceededException("Full");
        }
        else if (itemsOrdered.contains(media)) {
            System.out.println("This is already in your order!");
        }
        else {
            itemsOrdered.add(media);
            System.out.println("Media added!");
        }
    }
}
```

13. Create a class which inherits from Exception

```
1
2  package hust.soict.dsai.aims.exception;
3
4  public class PlayerException extends Exception {
5
6      public PlayerException() {
7          // TODO Auto-generated constructor stub
8      }
9
10     public PlayerException(String message) {
11         super(message);
12         // TODO Auto-generated constructor stub
13     }
14
15     public PlayerException(Throwable cause) {
16         super(cause);
17         // TODO Auto-generated constructor stub
18     }
19
20     public PlayerException(String message, Throwable cause) {
21         super(message, cause);
22         // TODO Auto-generated constructor stub
23     }
24
25     public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
26         super(message, cause, enableSuppression, writableStackTrace);
27         // TODO Auto-generated constructor stub
28     }
29
30 }
```

CompactDisc

```
    @Override
    public void play() throws PlayerException{
        System.out.println("CompactDisc Artist: " + this.getArtist());
        System.out.println("Total length: " + this.getLength());

        if (this.getLength() > 0) {
            System.out.println("Compactdisc: " + this.getTitle());
            System.out.println("CompactDisc Artist: " + this.getArtist());
            System.out.println("Total length: " + this.getLength());
            java.util.Iterator iter = tracks.iterator();
            Track nextTrack;
            while (iter.hasNext()) {
                nextTrack = (Track) iter.next();
                try {
                    nextTrack.play();
                }
                catch(PlayerException e ) {
                    throw e;
                }
            }
        }
        else {
            throw new PlayerException("Error: DVD length is non-positive!");
        }

        System.out.println("----------------Play All Tracks----------------");
        for (Track track: tracks) {
            track.play();
        }
    }
```

DVD

```java
3    @Override
4    public void play() throws PlayerException{
5        if (this.getLength() > 0) {
6            System.out.println("Playing DVD: " + this.getTitle());
7            System.out.println("DVD length: " + this.getLength());
8        }
9        else {throw new PlayerException("Error: DVD length is non-positive!");}
0    }
```

Track

```java
5    @Override
6    public void play() throws PlayerException {
7        if (this.getLength() > 0) {
8            System.out.println("Playing track: " + this.getTitle());
9            System.out.println("Track length: " + this.getLength());
0        }
1        else {throw new PlayerException("Error: DVD length is non-positive!");}
2    }
```

14. Update the Aims class

```java
                switch (choice) {
                    case 1:
                        cart.addMedia(media);
                        System.out.println("Media added to cart.");
                        break;
                    case 2:
                        if (media instanceof Playable) {
                            if(media instanceof DigitalVideoDisc){
                                try {
                                    ((DigitalVideoDisc) media).play();
                                } catch (PlayerException e) {
                                    // TODO Auto-generated catch block
                                    e.getMessage();
                                    e.toString();
                                    e.printStackTrace();
                                }
                            }
                            else if(media instanceof CompactDisc){
                                try {
                                    ((CompactDisc)media).play();
                                } catch (PlayerException e) {
                                    // TODO Auto-generated catch block
                                    e.getMessage();
                                    e.toString();
                                    e.printStackTrace();
                                }
                            }
                        } else {
                            System.out.println("This media cannot be played.");
                        }
                        break;
                    case 0:
                        break;
                    default:
                        System.out.println("Invalid choice! Please choose a number between 0-2.");
                }
            } while (choice != 0);
    }
```

```java
    public static void playMedia(Scanner scanner) {
        System.out.print("Enter the title of the media to play: ");
        String title = scanner.nextLine();
        Media media = store.searchByTitle(title);

        if (media != null) {
            if (media instanceof Playable) {
                if(media instanceof DigitalVideoDisc){
                    try {
                        ((DigitalVideoDisc) media).play();
                    } catch (PlayerException e) {
                        // TODO Auto-generated catch block
                        e.getMessage();
                        e.toString();
                        e.printStackTrace();
                    }
                }
                else if(media instanceof CompactDisc){
                    try {
                        ((CompactDisc)media).play();
                    } catch (PlayerException e) {
                        // TODO Auto-generated catch block
                        e.getMessage();
                        e.toString();
                        e.printStackTrace();
                    }
                }
            } else {
                System.out.println("This media cannot be played.");
            }
        } else {
            System.out.println("Media not found in the store.");
        }
    }
}
```

15. Modify the equals() method of Media class

```java
44
45      @Override
46      public boolean equals(Object o) {
47          if (o instanceof Media) {
48              Media media = (Media) o;
49              if (this.title.equals(media.title)) {
50                  return true;
51              }else {
52                  return false;
53              }
54          }
55          return false;
56      }
57
```

17. Update Aims class diagram