

LAB 05

1. Swing components

1.1. AWTAccumulator

The screenshot shows an IDE with the file `AWTAccumulator.java` open. The code defines a class `AWTAccumulator` that extends `Frame`. It uses `TextField` and `Label` components from the `java.awt` package. The GUI is a simple window titled "AWT Accumulator" with two rows: "Enter an Integer:" followed by a text field, and "The Accumulated Sum is:" followed by a disabled text field.

```
1 package hust.soict.dsai.swing;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 public class AWTAccumulator extends Frame {
6
7     private TextField tfInput;
8     private TextField tfOutput;
9     private int sum = 0;
10
11     // Constructor to setup the GUI components and event handlers
12     public AWTAccumulator() {
13         setLayout(new GridLayout(2,2));
14
15         add(new Label("Enter an Integer: "));
16
17         tfInput = new TextField(10);
18         add(tfInput);
19         tfInput.addActionListener(new TfInputListener());
20
21         add(new Label("The Accumulated Sum is: "));
22
23         tfOutput = new TextField(10);
24         tfOutput.setEditable(false);
25         add(tfOutput);
26     }
27 }
```

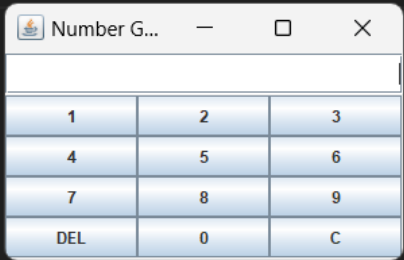
1.2. SwingAccumulator

The screenshot shows an IDE with the file `SwingAccumulator.java` open. The code defines a class `SwingAccumulator` that extends `JFrame`. It uses `JTextField` and `JLabel` components from the `javax.swing` package. The GUI is a window titled "Swing Accumula..." with two rows: "Enter an Integer:" followed by a text field, and "The Accumulated Sum is:" followed by a disabled text field.

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 import javax.swing.*;
8
9 public class SwingAccumulator extends JFrame {
10
11     private JTextField tfInput;
12     private JTextField tfOutput;
13     private int sum = 0;
14
15     public SwingAccumulator() {
16         Container cp = getContentPane();
17         cp.setLayout(new GridLayout(2,2));
18
19         cp.add(new JLabel("Enter an Integer: "));
20
21         tfInput = new JTextField(10);
22         cp.add(tfInput);
23         tfInput.addActionListener(new TfInputListener());
24
25         cp.add(new JLabel("The Accumulated Sum is: "));
26     }
27 }
```

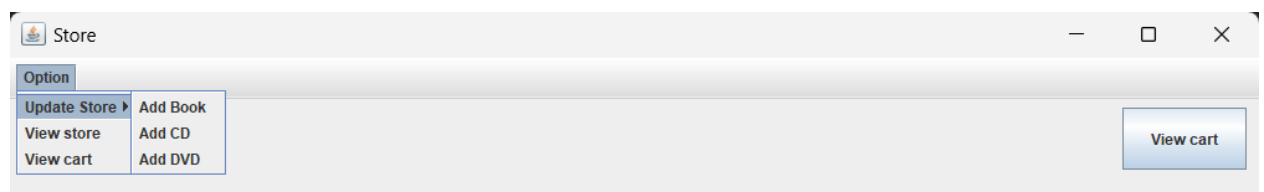
2. Organizing Swing components with Layout Managers

```
1 package hust.soict.dsai.swing;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 public class NumberGrid extends JFrame {
8     private JButton[] btnNumbers = new JButton[10];
9     private JButton btnDelete, btnReset;
10    private JTextField tfDisplay;
11
12    public NumberGrid() {
13
14        tfDisplay = new JTextField();
15        tfDisplay.setPreferredSize(new Dimension(200, 30));
16        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
17
18        JPanel panelButtons = new JPanel(new GridLayout(4,3));
19        addButtons(panelButtons);
20
21        Container cp = getContentPane();
22        cp.setLayout(new BorderLayout());
23        cp.add(tfDisplay, BorderLayout.NORTH);
24        cp.add(panelButtons, BorderLayout.CENTER);
25    }
26}
```



3. Create a graphical user interface for AIMS with Swing

3.1. View Store Screen



3.2. Adding more user interaction

```
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));

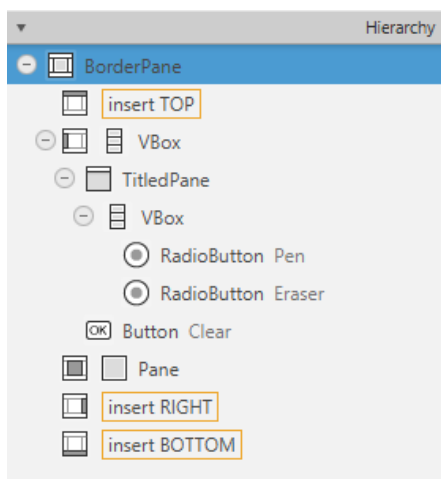
            // Display Playing Message
            mainGui.add(new JLabel("Playing... " + media.getTitle()), BorderLayout.CENTER);
            System.out.println(media.getTitle());
            // Close Button
            JPanel buttonPanel = new JPanel(new FlowLayout());
            JButton close = new JButton("Stop");
            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 : BorderLayout

Layout : BorderLayout

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

Rotation Axis

X

0

Y

0

Z

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

Inspector

Properties : Pane

Layout : Pane

Code : Pane

Identity

fx:id

DragDrop

On Drag Detected

#

On Drag Done

#

On Drag Dropped

#

Layout : VBox

Border Pane Constraints

Alignment

Margin >

Internal

Padding >

Spacing

Specific

Fill Width ☒

Size

Min Width

Min Height

Pref Width

Pref Height

Max Width

Max Height

Width

Height

Position

BorderPane

insert TOP

VBox

TitledPane

VBox

RadioButton Pen

RadioButton Eraser

OK Button Clear

Pane

insert RIGHT

insert BOTTOM

Style

-fx-background-color

Style Class

Stylesheets

Id

Extras

On Context Menu Requested

#

On Mouse Clicked

#

On Mouse Dragged

#

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

Font

Text Fill

Wrap Text ☐

Text Alignment

Text Overrun

Ellipsis String

Underline ☐

Line Spacing

Layout : Button

VBox Constraints

Vgrow

Margin ☐ 0 ☐ 0 ☐ 0 ☐ 0

Internal

Padding ☐ 0 ☐ 0 ☐ 0 ☐ 0

Size

Min Width

Min Height

Pref Width

Pref Height

Max Width

Max Height

Width

Height

Code : Button

On Action

DragDrop

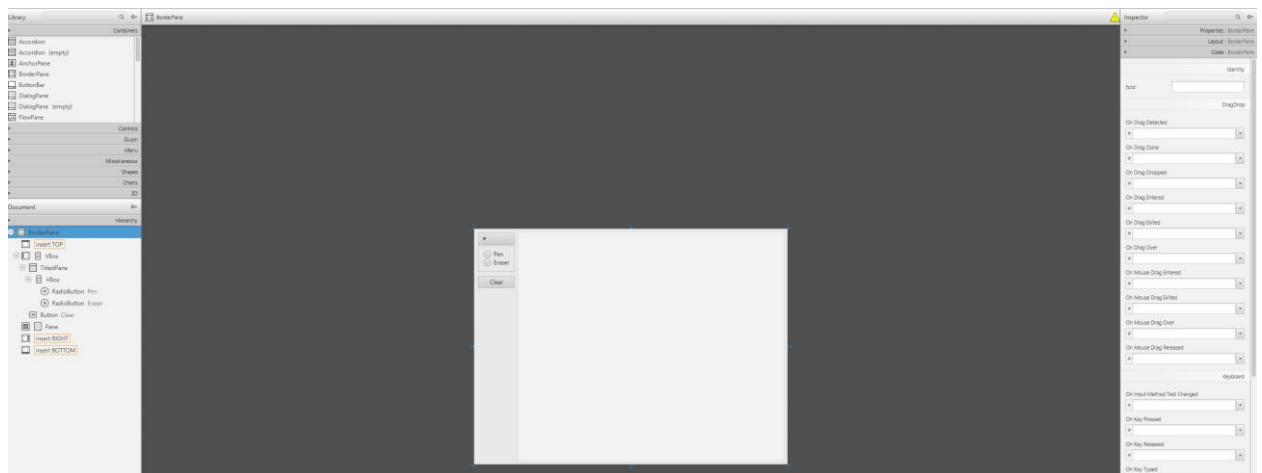
On Drag Detected

On Drag Done

On Drag Dropped

On Drag Entered

On Drag Exited



```

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

```

```

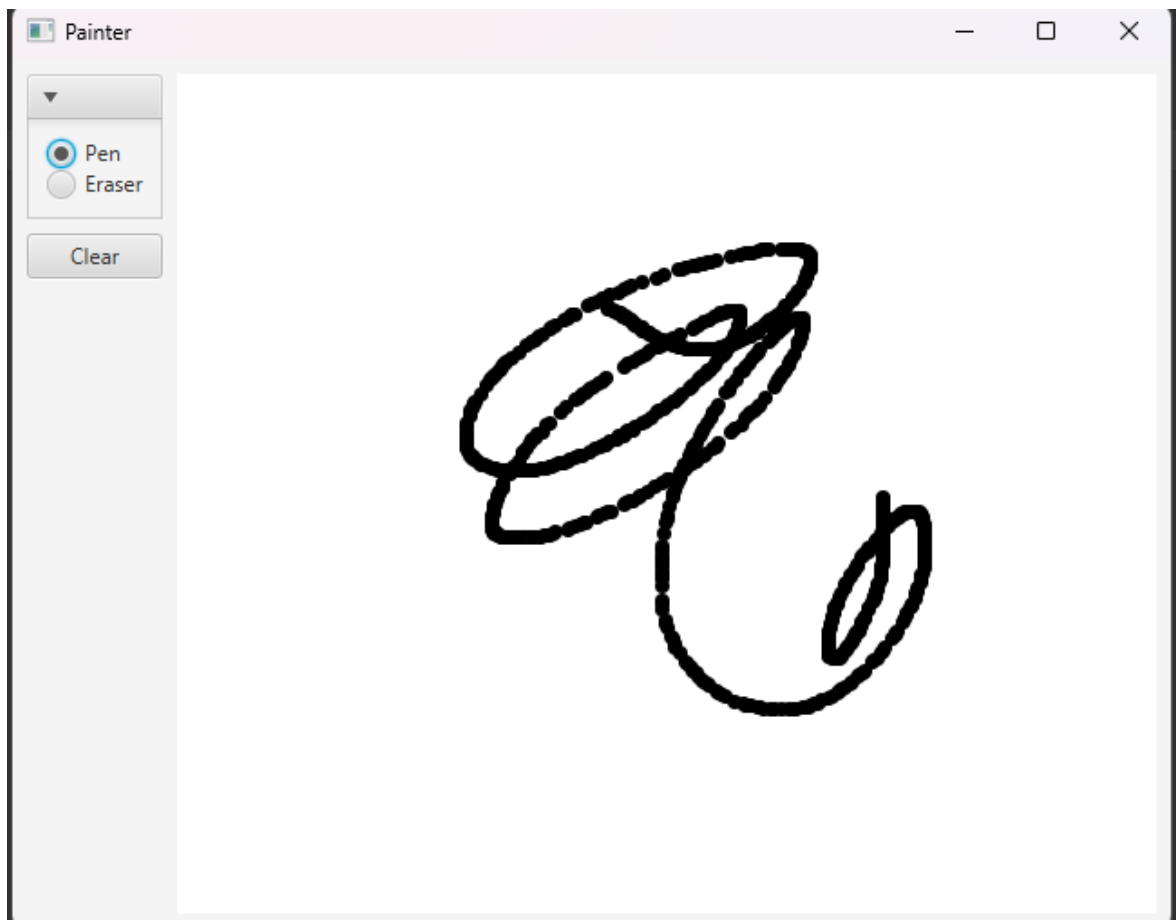
</> Painter.fxml M ● PainterController.java M Painter.java M X
GUIProject > hust > soict > dsai > javafx C:\Users\hoang le\Workspaces\GUIProject\hust\soict\dsai\javafx\PainterController.java
1  package hust.soict.dsai; (preview)
2
3  import javafx.application.Application;
4  import javafx.fxml.FXMLLoader;
5  import javafx.scene.Parent;
6  import javafx.scene.Scene;
7  import javafx.stage.Stage;
8
9  public class Painter extends Application{
10
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     Run | Debug
22     public static void main(String[] args) {
23         launch(args);
24     }

```



```
GUIProject > hust > soict > dsai > javafx > PainterController.java > PainterController

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 }
```

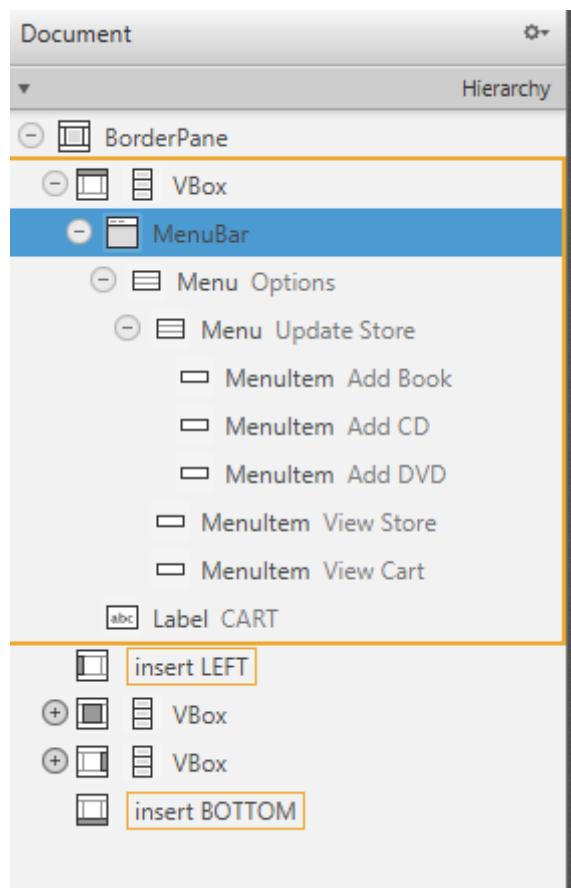


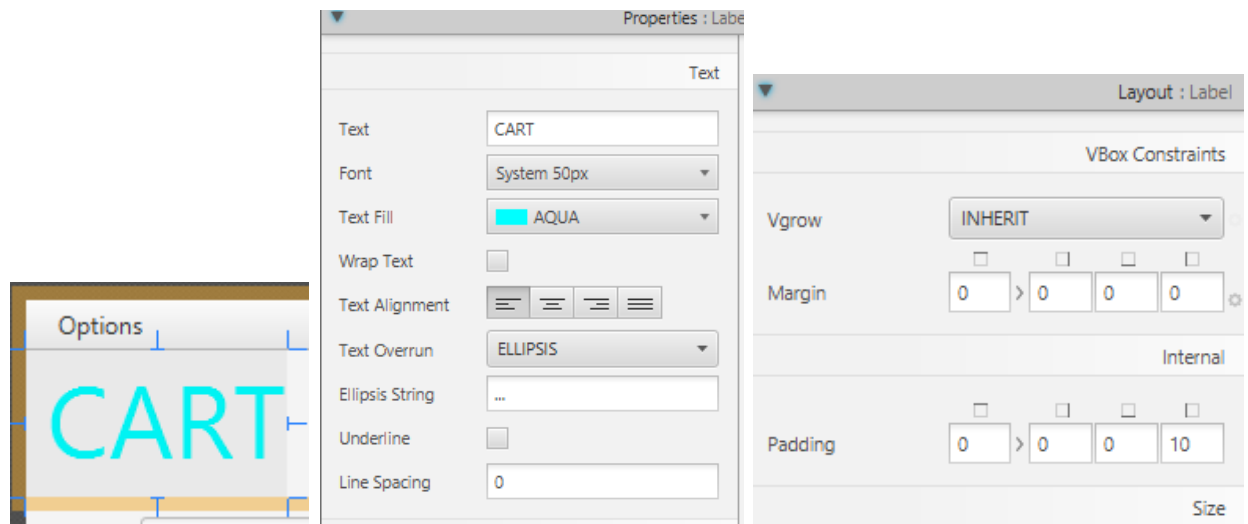
Handle Erase Function

```
@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);
    }
}
```

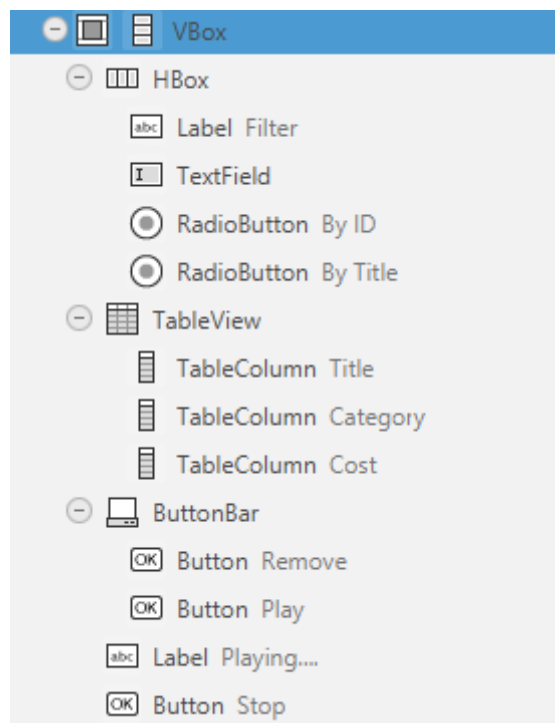
5. Setting up the View Cart Screen with ScreenBuilder

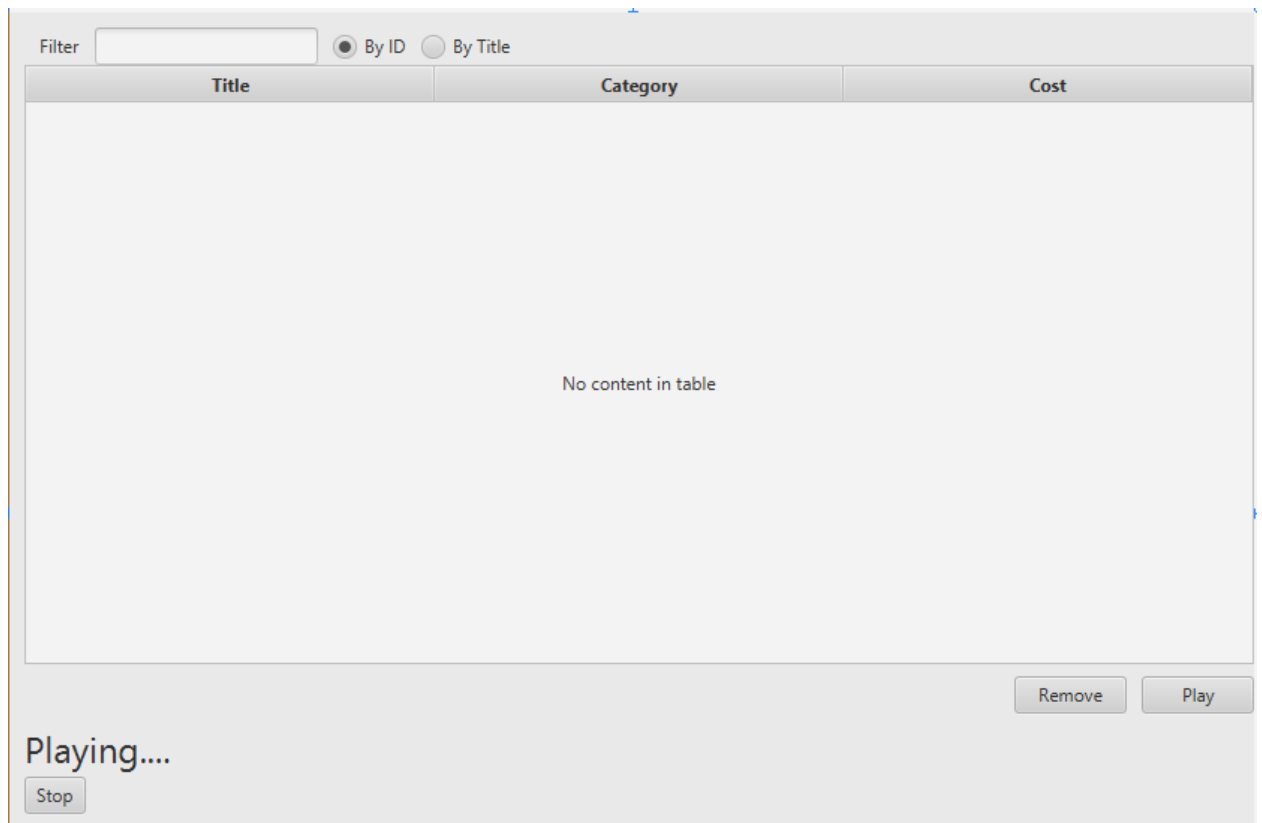
5.2. Setting up the TOP area



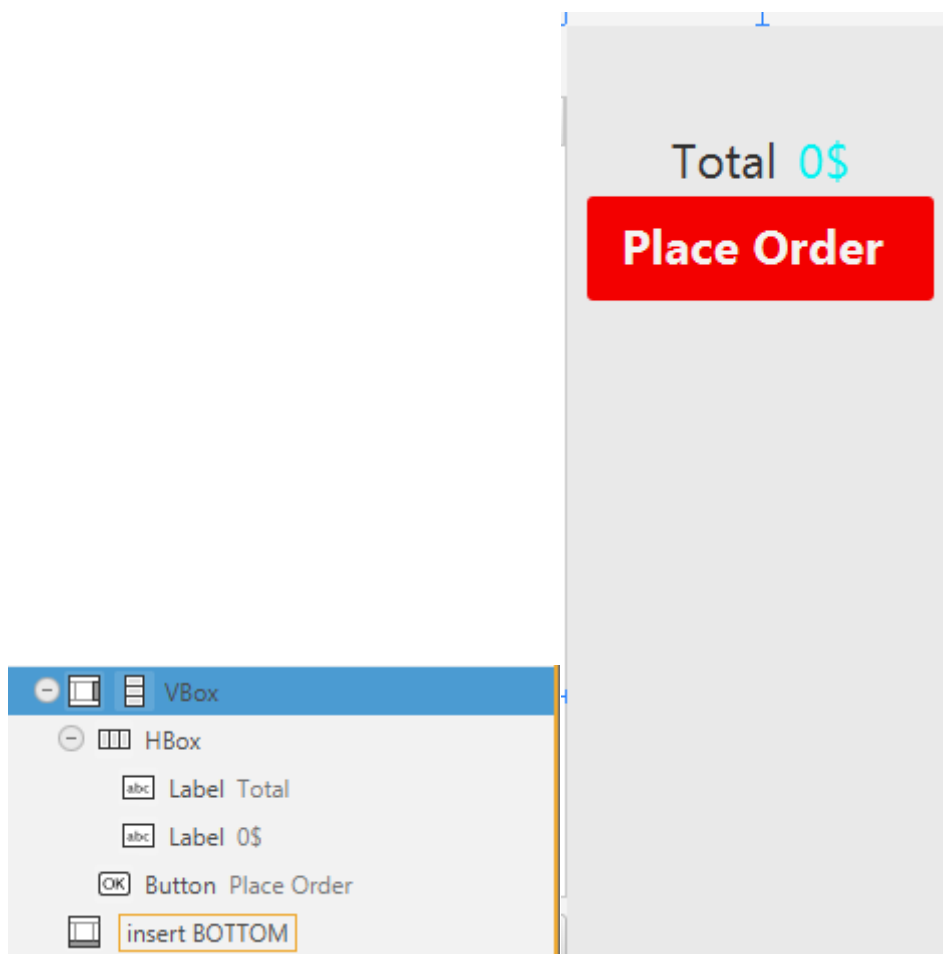


5.3. Setting up the CENTER area





5.4. Setting up the RIGHT area



6. Integrating JavaFX into Swing application – The JFXPanel class

```
public class CartScreen extends JFrame {

    private Cart cart;
    private ControllerScreen controllersScreen;

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

        this.cart = cart;

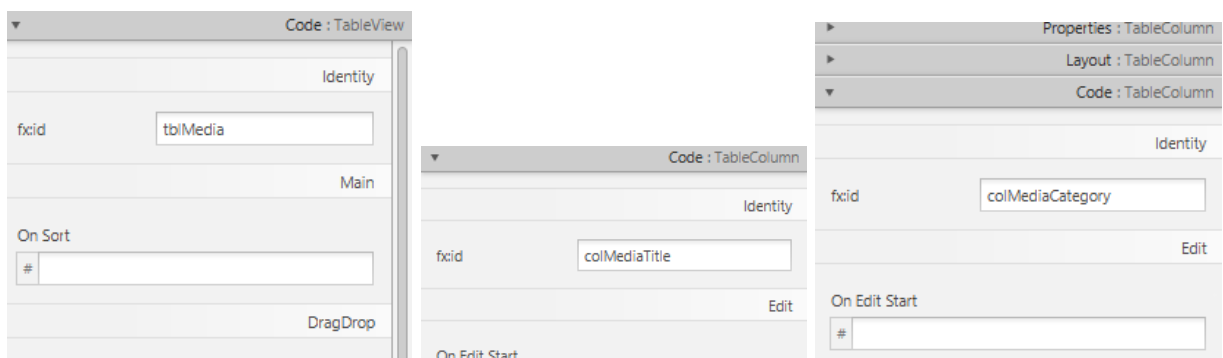
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        // this.setVisible(true);
        this.setPreferredSize(new Dimension(1024, 768));
        pack();
        Platform.runLater(new Runnable() {

            @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();
                }
            }
        });
    }

    Run | Debug
    public static void main(String[] args) {
    }

}
```

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



Code : TableColumn

Identity

fxid: colMediaCost

Edit

On Edit Start

#

On Edit Commit

- Update itemsOrdered to ObservableList

Cart

Options

CART

Filter:

☒ By ID ☐ By Title

Title	Category	Cost
sgk tieng anh 9	Text Book	5.0
Dac Nhan Tam	Selfhelp	6.5

Total 0.0\$

Place Order

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

Code : Button

Identity

fxid: btnPlay

Main

On Action

btnPlayPressed

DragDrop

Code : Button

Identity

fxid: btnRemove

Main

On Action

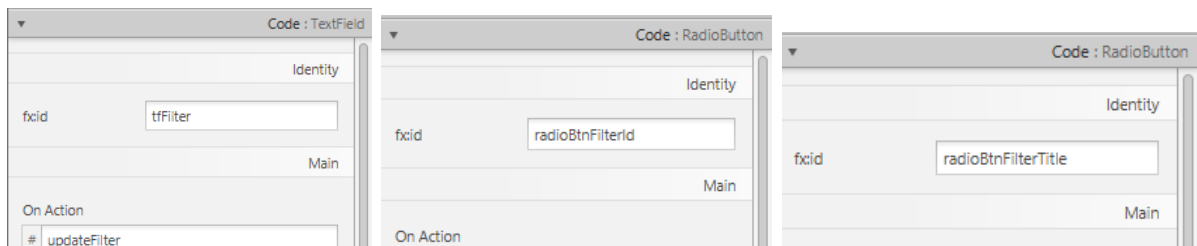
btnRemovePressed

DragDrop

9. Deleting a media

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

10. Filter items in cart – FilteredList



```
@Override
public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
    showFilterMedia(newValue);
}

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)));
        }
    }
}
```

11. Complete the Aims GUI application

Store

Option

AIMS

View cart

Interstellar

15.0\$

Add to Cart

Play

tieng anh 9

5.0\$

Add to Cart

Dac Nhan Tam

6.5\$

Add to Cart

Options

CART

Filter

☒ By ID ☐ By Title

Title	Category	Cost
Interstellar	Film	15.0
tieng anh 9	Text Book	5.0
Dac Nhan Tam	Selfhelp	6.5

Total 0.0\$

Place Order

Add Book

AIMS

Option

View cart

title

category

cost

Authors(Names are separated by a comma)

add

Add CD

AIMS

Option

View cart

title

category

cost

artist

List track (each track separated by a comma Ex: track-length)

add

Add DVD

Add DVD

Option

AIMS

View cart

title

category

cost

director

length

add

Filter by Title

Cart

Options

CART

Filter
☐ By ID
☒ By Title

Title	Category	Cost
Interstellar	Film	15.0

Total 0.0\$

Place Order

Filter by Id

CART

Filter
☒ By ID
☐ By Title

Title	Category	Cost
tieng anh 9	Text Book	5.0

Total 0.0\$

Place Order

AddItemToStoreScreen

```
public class AddItemToStoreScreen extends JFrame {
    protected Store store;
    protected Cart cart;
    protected JTextField title;
    protected JTextField category;
    protected JTextField cost;
    protected JButton addBtn;
    protected ControllerScreen c;
    protected JPanel center;
    protected int numberInput = 3;

    JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Option");

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBookMenu = new JMenuItem("Add Book");
        addBookMenu.addActionListener(e -> {
            c.showAddBookScreen();
        });
        smUpdateStore.add(addBookMenu);
        JMenuItem addCDMenu = new JMenuItem("Add CD");
        addCDMenu.addActionListener(e -> {
            c.showAddCDCreen();
        });
        smUpdateStore.add(addCDMenu);
        JMenuItem addDVDMenu = new JMenuItem("Add DVD");
        addDVDMenu.addActionListener(e -> {
            c.showAddDVDScreen();
        });
        smUpdateStore.add(addDVDMenu);

        menu.add(smUpdateStore);
        JMenuItem viewStoreMenu = new JMenuItem("View store");
        viewStoreMenu.addActionListener(e -> {
            c.showStoreScreen();
        });
        menu.add(viewStoreMenu);

        JMenuItem viewCartMenu = new JMenuItem("View cart");
        viewCartMenu.addActionListener(e -> {
            c.showCartScreen();
        });
        menu.add(viewCartMenu);
    }
}
```

AddDigitalVideoDiscToStoreScreen

```
public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
    private JTextField director;
    private JTextField length;

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

    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 6;
        // SpringLayout layout = new SpringLayout();
        // JPanel p = new JPanel(layout);
        JLabel directorLabel = new JLabel("director", JLabel.TRAILING);
        panel.add(directorLabel);
        director = new JTextField(2);
        director.setPreferredSize(new Dimension(150, 20));
        directorLabel.setLabelFor(director);
        panel.add(director);
        JLabel lengthLabel = new JLabel("length", JLabel.TRAILING);
        panel.add(lengthLabel);
        length = new JTextField(2);
        lengthLabel.setLabelFor(length);
        panel.add(length);
        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 director = this.director.getText();
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
        int length = Integer.parseInt(this.length.getText());
        DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
    }
}
```

AddCompactDiscToStoreScreen

```
16 public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
17     private JTextField artist;
18     private JTextField listTrack;
19
20     public AddCompactDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
21         super(store, cart, c);
22     }
23
24
25     @Override
26     void updateAdd(JPanel panel) {
27         this.numberInput = 6;
28
29         JLabel artistLabel = new JLabel("artist", JLabel.TRAILING);
30         panel.add(artistLabel);
31         artist = new JTextField(2);
32         artist.setPreferredSize(new Dimension(150, 20));
33         artistLabel.setLabelFor(artist);
34         panel.add(artist);
35         JLabel listTrackLabel = new JLabel("List track (each track separated by a comma Ex: track-length)",
36             JLabel.TRAILING);
37
38         panel.add(listTrackLabel);
39         listTrack = new JTextField(2);
40         listTrackLabel.setLabelFor(listTrack);
41         panel.add(listTrack);
42         JButton tes = new JButton("add");
43         tes.setVisible(false);
44         panel.add(tes);
45         panel.setPreferredSize(new Dimension(100, 300));
46         addBtn = new JButton("add");
47         addBtn.addActionListener(e -> {
48             addMedia();
49         });
50         panel.add(addBtn);
51     }
52
53     public void addMedia() {
54         String title = this.title.getText();
55         String listTrack = this.listTrack.getText();
56         String category = this.category.getText();
57         float cost = Float.parseFloat(this.cost.getText());
```

AddBookToStoreScreen

```

15 public class AddBookToStoreScreen extends AddItemToStoreScreen {
16     private JTextField listAuthor;
17
18     public AddBookToStoreScreen(Store store, Cart cart, ControllerScreen c) {
19         super(store, cart, c);
20     }
21
22     @Override
23     void updateAdd(JPanel panel) {
24         this.numberInput = 5;
25
26         JLabel listAuthorLabel = new JLabel("Authors(Names are separated by a comma)", JLabel.TRAILING);
27         panel.add(listAuthorLabel);
28         listAuthor = new JTextField(2);
29         listAuthor.setPreferredSize(new Dimension(150, 20));
30         listAuthorLabel.setLabelFor(listAuthor);
31         panel.add(listAuthor);
32
33         JButton tes = new JButton("add");
34         tes.setVisible(false);
35         panel.add(tes);
36         panel.setPreferredSize(new Dimension(100, 300));
37         addBtn = new JButton("add");
38         addBtn.addActionListener(e -> {
39             addMediaToStore();
40         });
41         panel.add(addBtn);
42     }
43
44     public void addMediaToStore() {
45         String title = this.title.getText();
46         String listAuthor = this.listAuthor.getText();
47         String[] arrayAuthor=listAuthor.split(",");
48         String category = this.category.getText();
49         float cost = Float.parseFloat(this.cost.getText());
50         Book book = new Book(title, category, cost);
51         for(String author:arrayAuthor) {
52             book.addAuthor(author);
53         }
54         this.store.addMedia(book);
55         JOptionPane.showMessageDialog(null, "add Book successfully!");
56         clearTextField();

```

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

```

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 RuntimeException {
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(message:"Error: CD length is non-positive!");
    }

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

DVD

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

Track

```
@Override
public void play() throws PlayerException {
    if (this.getLength() > 0) {
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }
    else {throw new PlayerException(message:"Error: Track length is non-positive!");}
}
```

14. Update the Aims class

```
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);
```



```

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

```

@Override
public boolean equals(Object o) {
    if (o instanceof Media) {
        Media media = (Media) o;
        if (this.title.equals(media.title)) {
            return true;
        } else {
            return false;
        }
    }
    return false;
}

```

17. Update Aims class diagram

