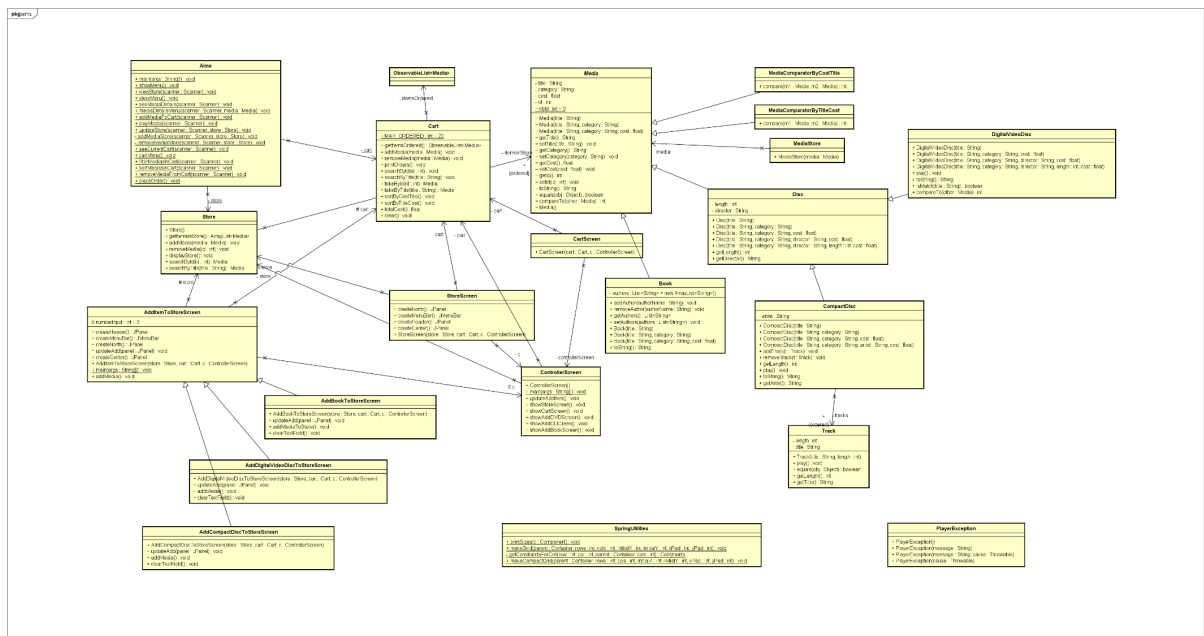
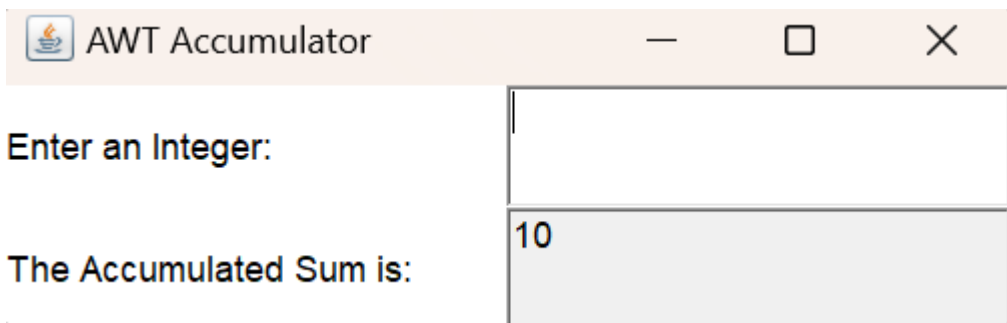


# LAB05

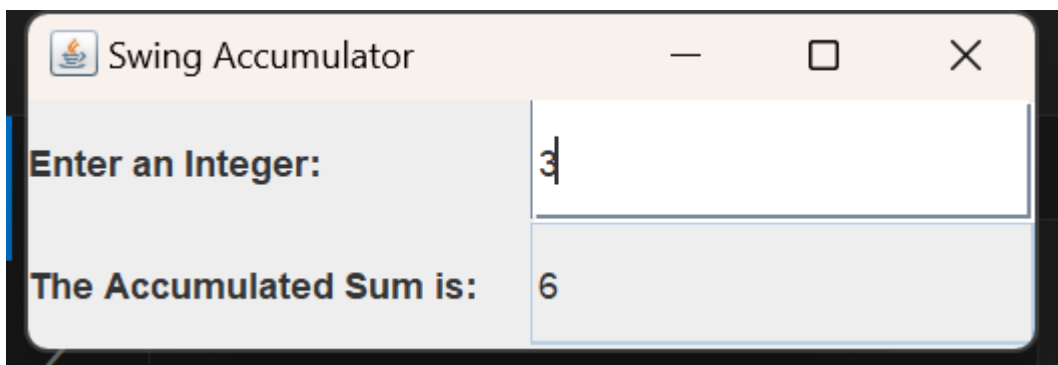
## class diagram



## 1.1



## 1.2

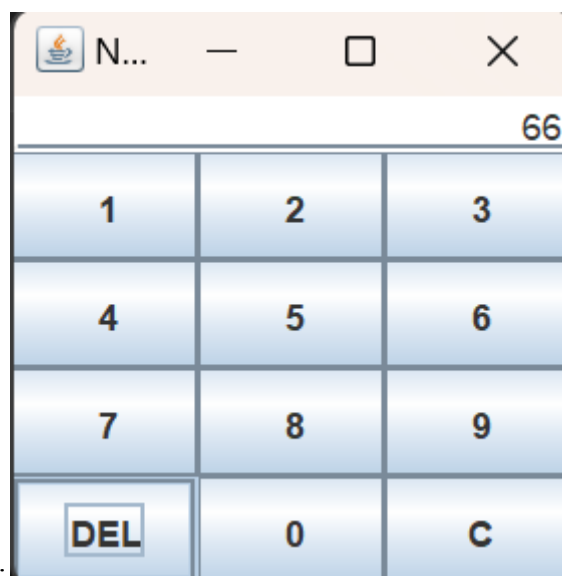
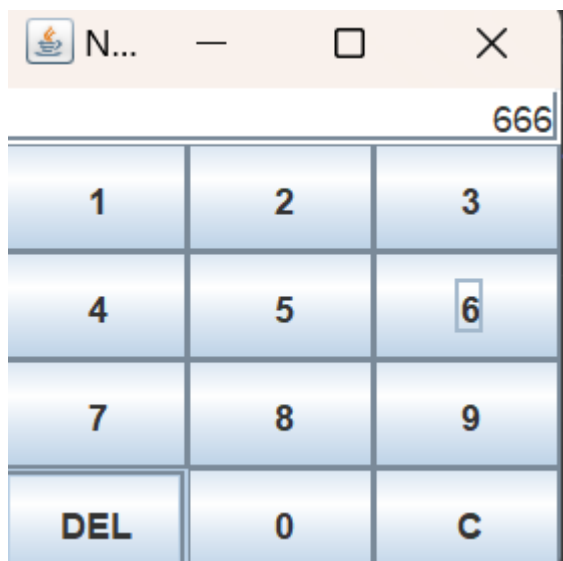


1.3

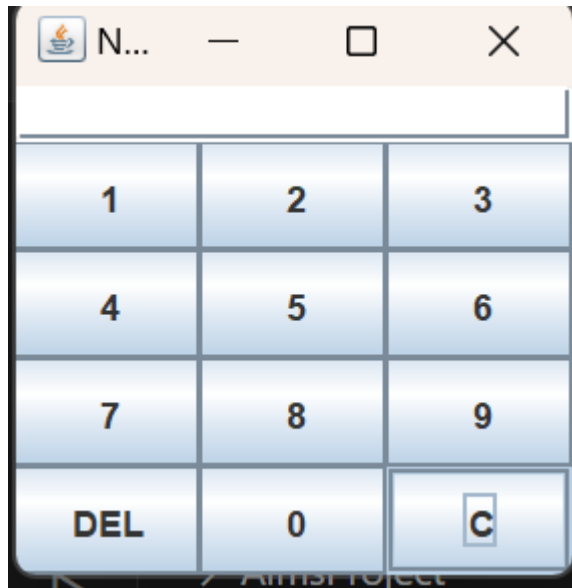
**Question:** What are the differences between the Swing and AWT implementation?

**Answer:** Swing and AWT both provide a way to create GUI applications in Java. However, Swing is an extension of AWT and offers more powerful and flexible features. Swing is more versatile, consistent, and flexible than AWT, making it the preferred choice for developing modern Java GUI applications. However, AWT can be used for lightweight or OS-specific applications that require native platform integration.

2.



- use DEL:



- use C:

```
private class ButtonListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String button = e.getActionCommand();
        if (button.charAt(index:0) >= '0' && button.charAt(index:0) <= '9') {
            tfDisplay.setText(tfDisplay.getText() + button);
        } else if (button.equals(anObject:"DEL")) {
            String currentText = tfDisplay.getText();
            if (!currentText.isEmpty()) {
                tfDisplay.setText(currentText.substring(beginIndex:0, currentText.length() - 1));
            }
        } else if (button.equals(anObject:"C")) {
            tfDisplay.setText(t:"");
        } else {
            System.out.println("Button pressed: " + button);
        }
    }
}
```

3.

**Question:** Note how the code checks if the Media implements the Playable interface to create a 'Play' button

**Answer:** If **media** is an instance of the **Video** class, it will create a "Play" button because **Video** implements the **Playable** interface. Otherwise, the "Play" button will not be created for **Media** objects that do not have playback capability.

2.2

```

public class MediaStore extends JPanel {
    public MediaStore(Media media, Cart cart) {

        JButton addCartBtn = new JButton(text:"Add to Cart");
        addCartBtn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                cart.addMedia(media);
            }
        });
        container.add(addCartBtn);

        if (media instanceof Playable) {
            JButton playBtn = new JButton(text:"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(top:20, left:20, bottom:20, right:20));










                    mainGui.add(new JLabel("Playing... " + media.getTitle()), BorderLayout.CENTER);
                    System.out.println(media.getTitle());

                    JPanel buttonPanel = new JPanel(new FlowLayout());
                    JButton close = new JButton(text:"Stop");
                    close.addActionListener(ev -> {
                        playDialog.setVisible(b:false);
                        System.out.println(x:"Stopped playing.");
                    });
                    buttonPanel.add(close);
                    mainGui.add(buttonPanel, BorderLayout.SOUTH);

                    playDialog.setContentPane(mainGui);
                    playDialog.setLocationRelativeTo(playBtn);
                    playDialog.pack();
                    playDialog.setVisible(b:true);
                }
            });
        }
    }
}

```

3.

- ☐  BorderLayout
  -  insert TOP
- ☐  VBox
- ☐  TitledPane
  - ☐  VBox
    - ☐ RadioButton Pen
    - ☐ RadioButton Eraser
  - Button Clear
-   Pane
-  insert RIGHT
-  insert BOTTOM

▶

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

Identity

fx:id

drawingArenaPane






DragDrop

Alignment

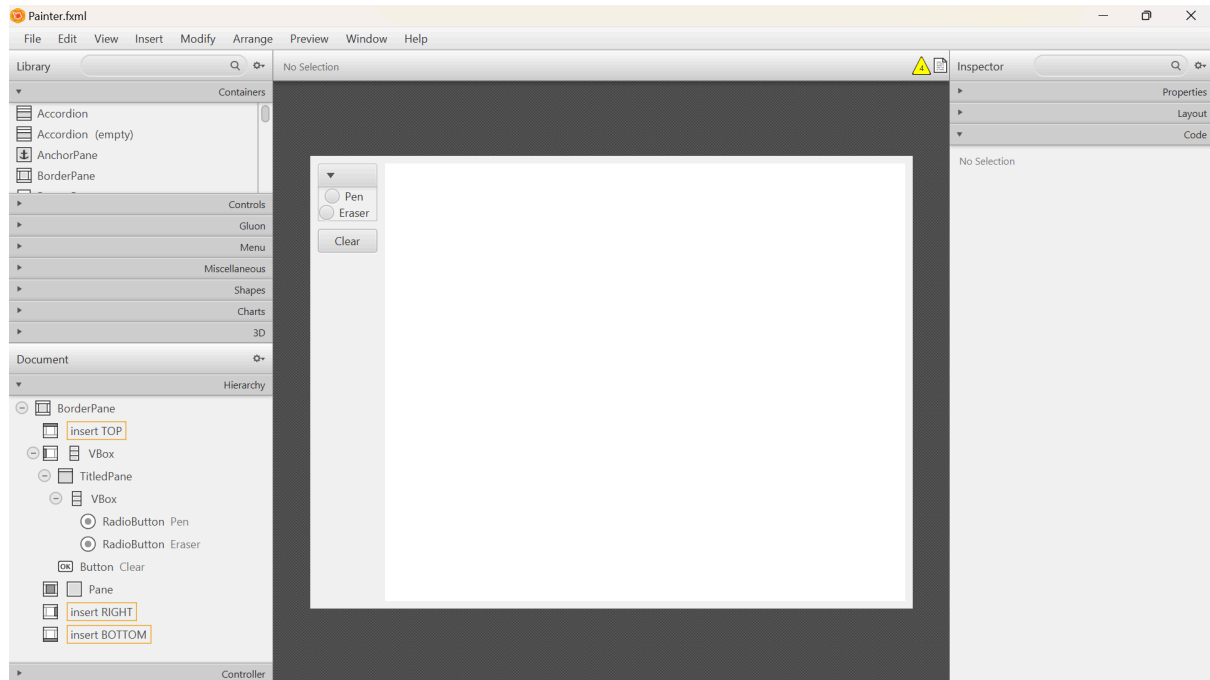
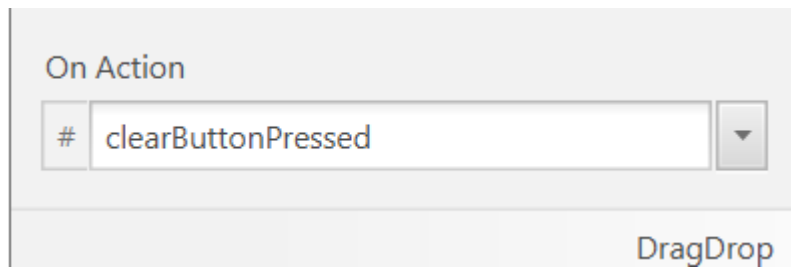
CENTER

▼

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	▼

Text	
Text	Clear
Font	System 12px ▼
Text Fill	 BLACK ▼
Wrap Text	<input type="checkbox"/>
Text Alignment	   
Text Overrun	ELLIPSIS ▼
Ellipsis String	...
Underline	<input type="checkbox"/>

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	▼



```

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

```



```

0  import javafx.scene.shape.Circle;
1
2
3  public class PainterController {
4
5      private int color;
6
7      @FXML
8      private ToggleGroup identity;
9
10     @FXML
11     void chooseOption(ActionEvent event) {
12         String button = ((RadioButton)event.getSource()).getText();
13         if (button.equals(anObject:"Pen")) {
14             System.out.println(x:"1");
15             color = 1;
16         }else {
17             System.out.println(x:"0");
18             color = 0;
19         }
20     }
21
22     @FXML
23     private Pane drawingAreaPane;
24
25     @FXML
26     void clearButtonPressed(ActionEvent event) {
27         drawingAreaPane.getChildren().clear();
28     }
29
30     @FXML
31     void drawingAreaMouseDragged(MouseEvent event) {
32         if (color == 1 && event.getX()>=0) {
33             Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.BLACK);
34             drawingAreaPane.getChildren().add(newCircle);
35         }else if(color == 0 && event.getX()>=0){
36             Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.WHITE);
37             drawingAreaPane.getChildren().add(newCircle);
38         }
39     }
40
41 }

```

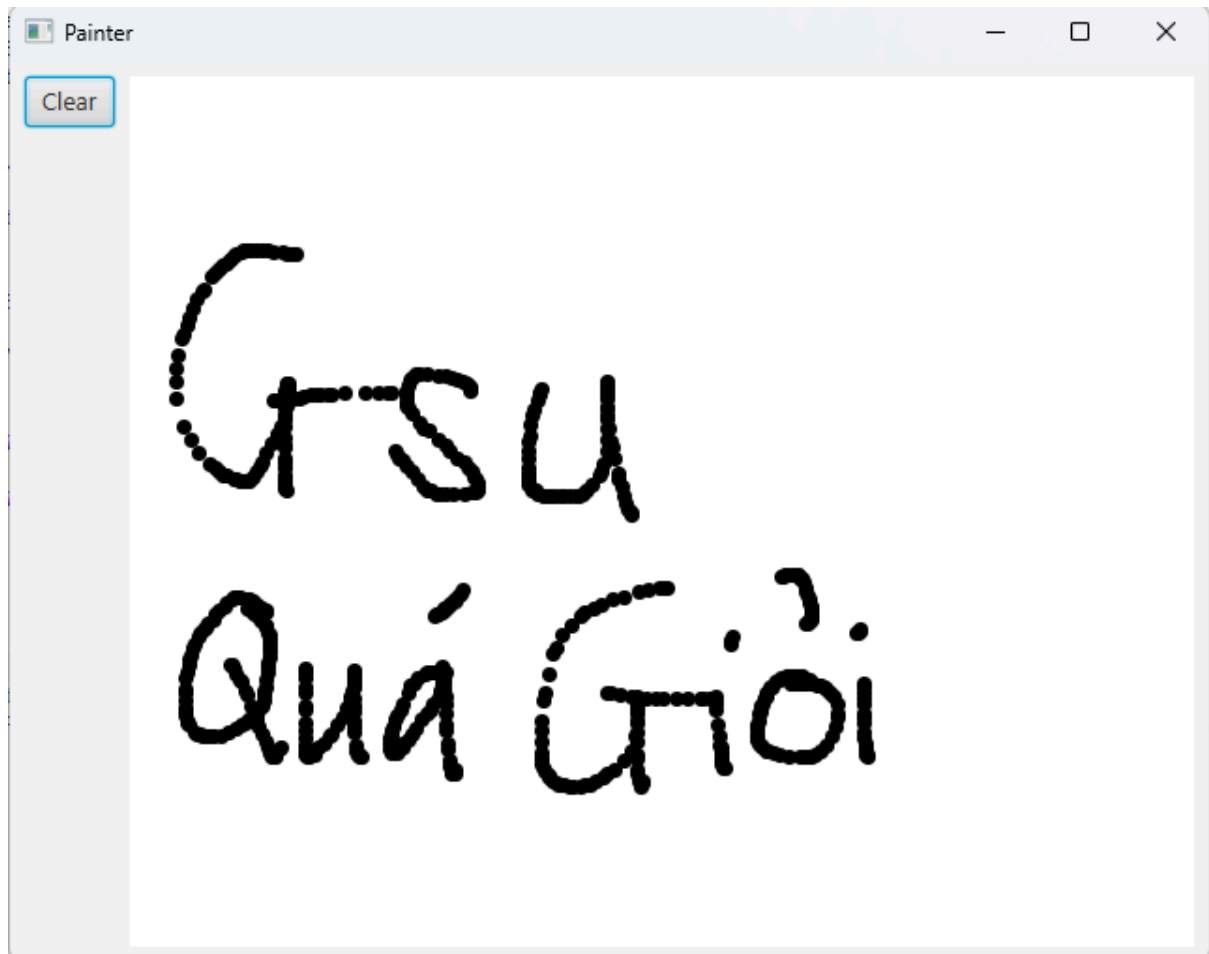
Java: Ready

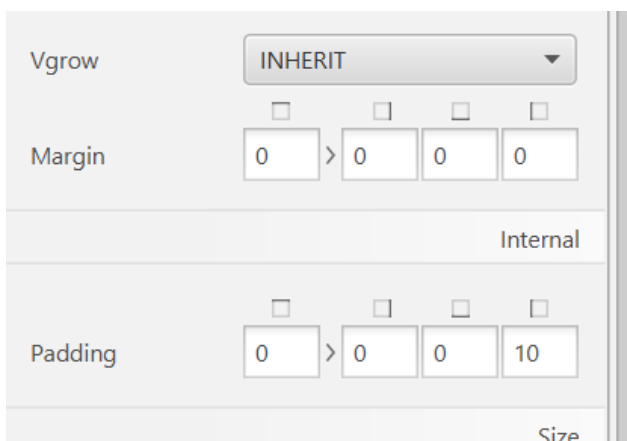
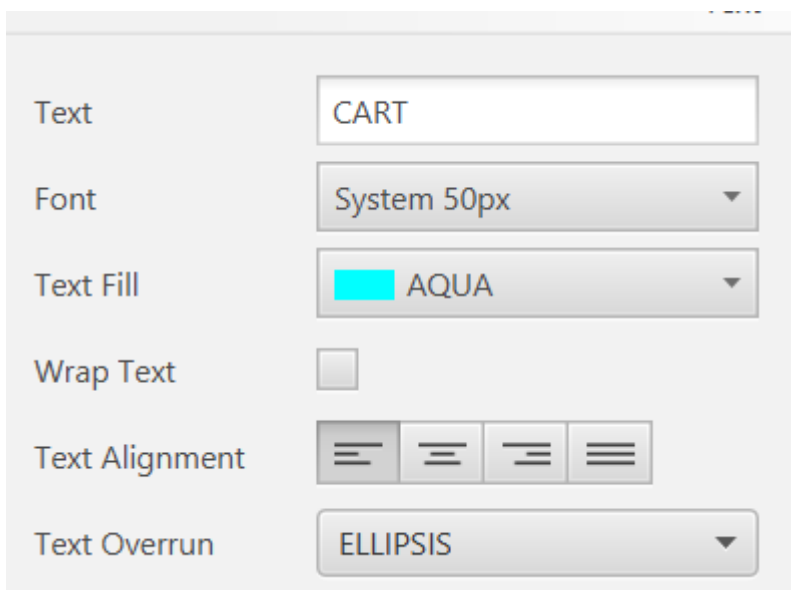
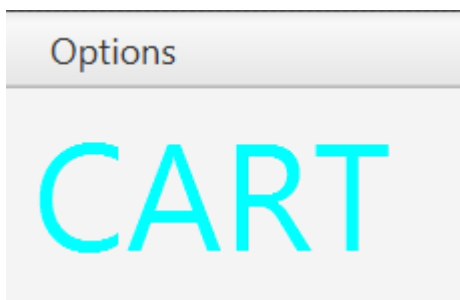
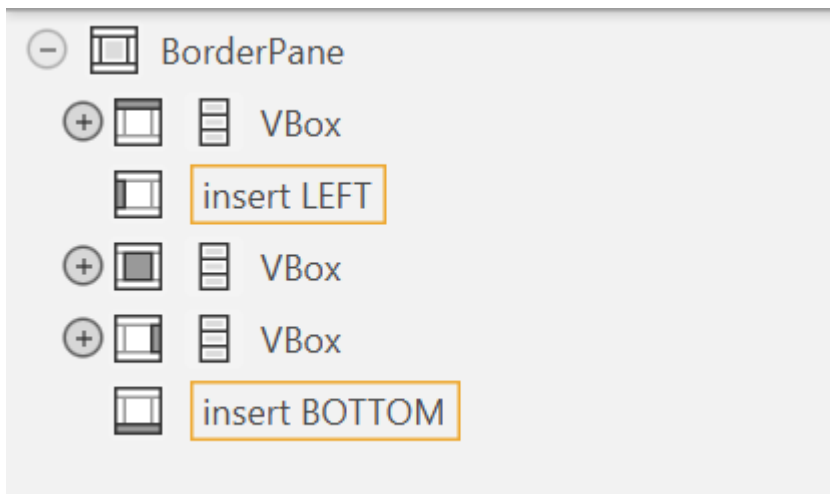
```

UIProject > hust > soict > globalict > javafx > Painter.java
1
2  import javafx.application.Application;
3  import javafx.fxml.FXMLLoader;
4  import javafx.scene.Parent;
5  import javafx.scene.Scene;
6  import javafx.stage.Stage;
7
8  public class Painter extends Application{
9
10     @Override
11     public void start(Stage stage) throws Exception{
12         Parent root = FXMLLoader.load(getClass().getResource(name:"/hust/soict/dsai/javafx/Painter.fxml"));
13
14         Scene scene = new Scene(root);
15         stage.setTitle("Painter");
16         stage.setScene(scene);
17         stage.show();
18     }
19
20     public static void main(String[] args) {
21         launch(args);
22     }
23 }

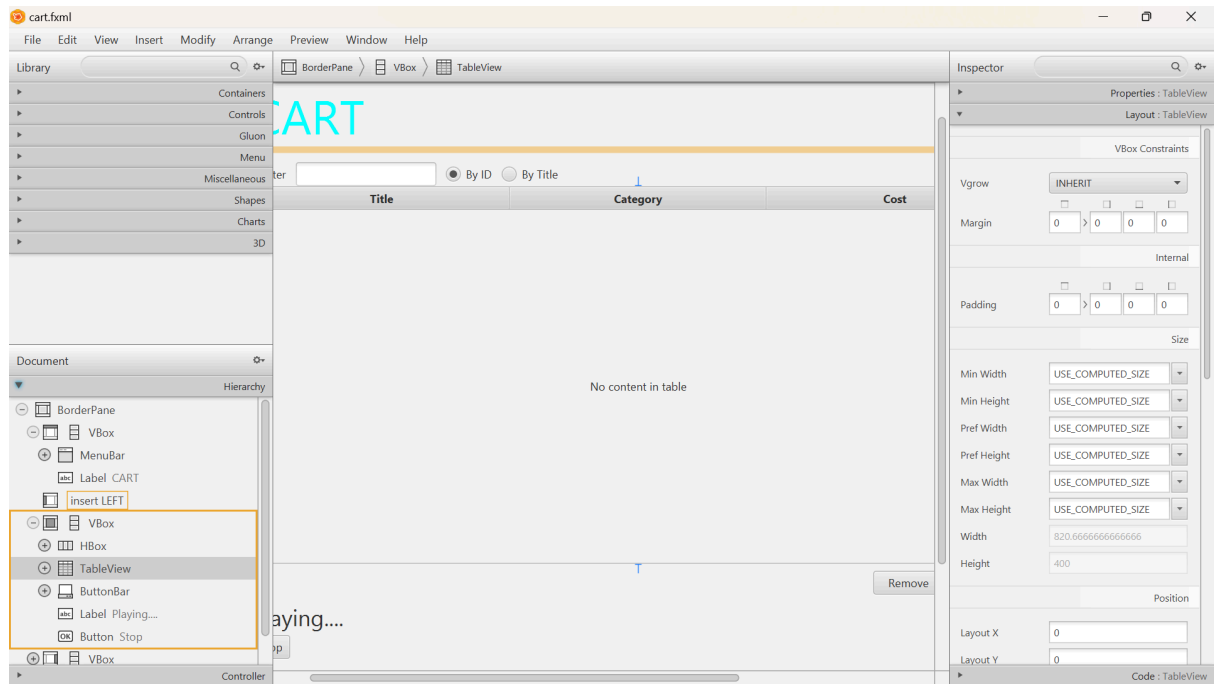
```

## Painter

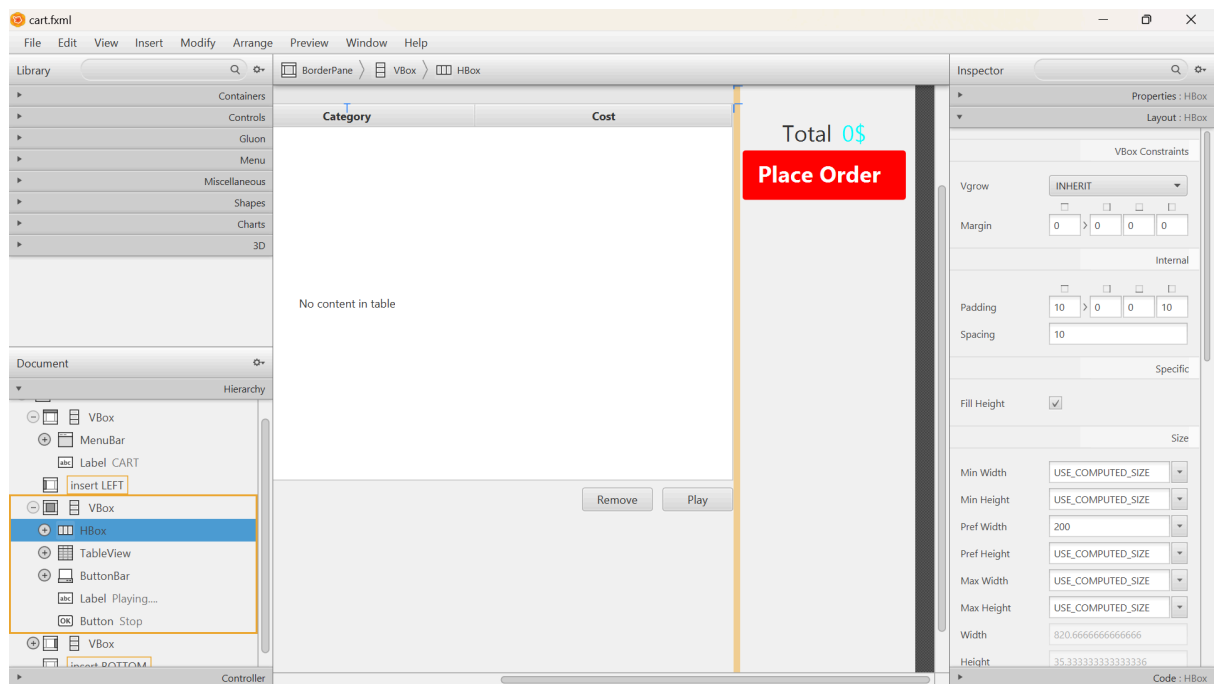




## 4.3



## 4.4



## 5.

```

public class CartScreen extends JFrame {
    public CartScreen(Cart cart, ControllerScreen c) {
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle(title:"Cart");
        this.setPreferredSize(new Dimension(width:1024, height:768));
        pack();
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass()
                        .getResource(name:"/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();
                }
            }
        });
    }
}

```

6.

Layout : TableView

Code : TableView

Identity	
fx:id	<input type="text" value="tblMedia"/>
Main	
On Sort	
#	<input type="text"/>
DragDrop	
On Drag Detected	

▼

Code : TableColumn

Identity

fx:id

colMediaTitle

Edit

▼

Code : TableColumn

Identity

fx:id

colMediaCategory

Edit

▼

Code : TableColumn

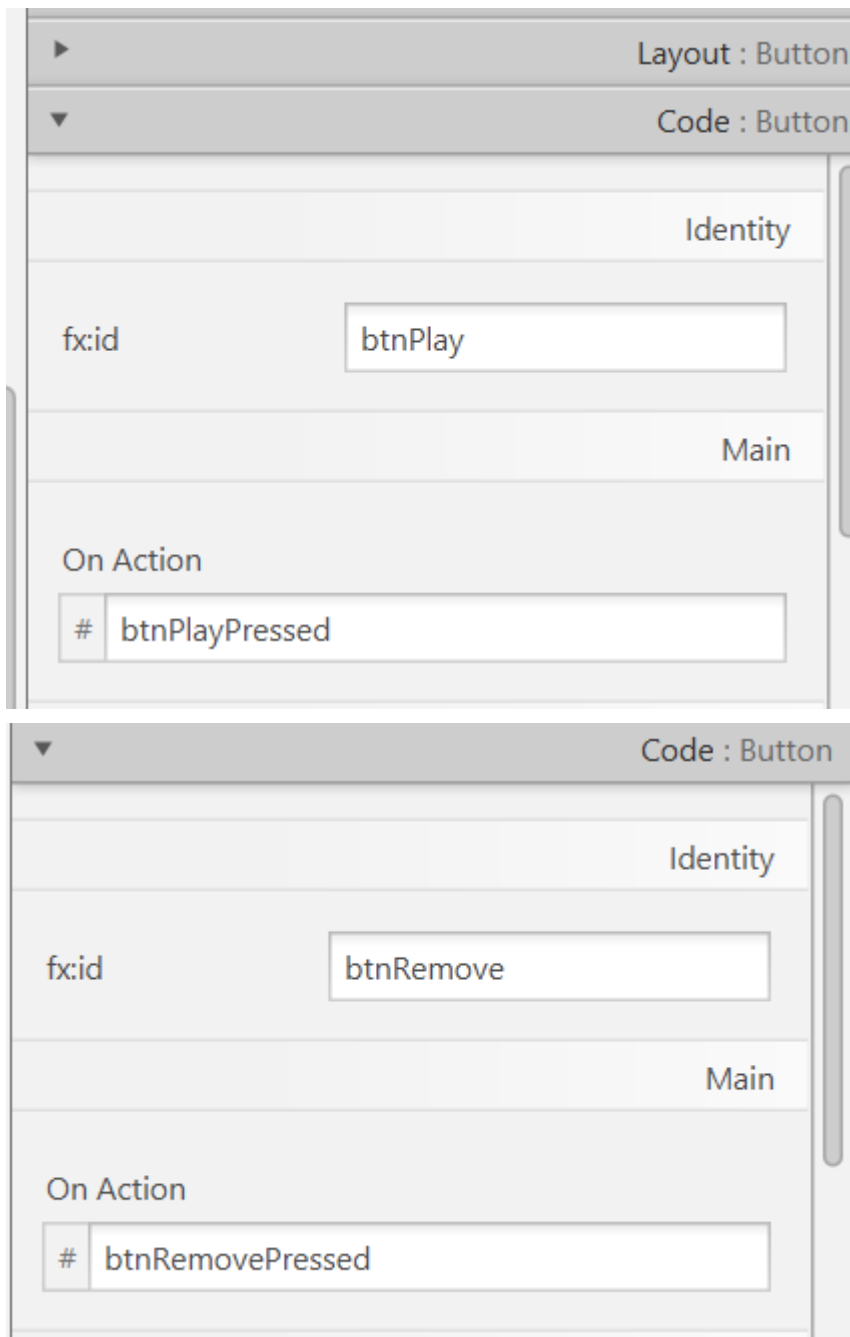
Identity

fx:id

colMediaCost

Edit

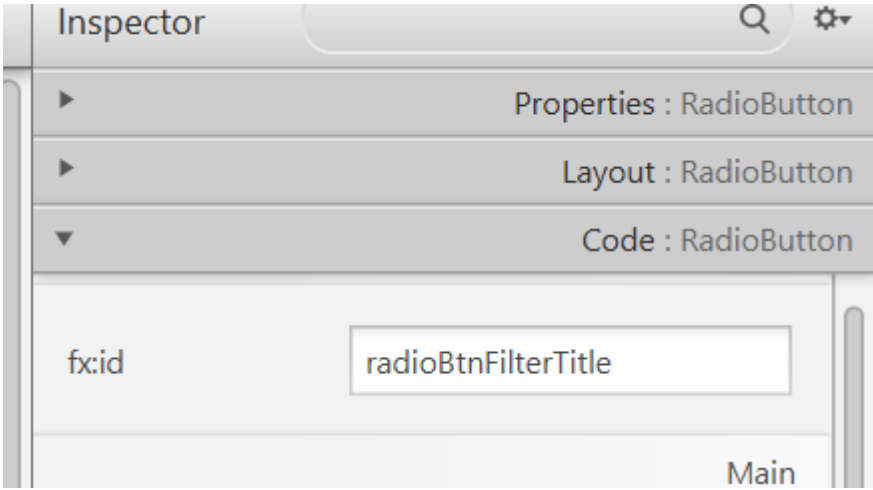
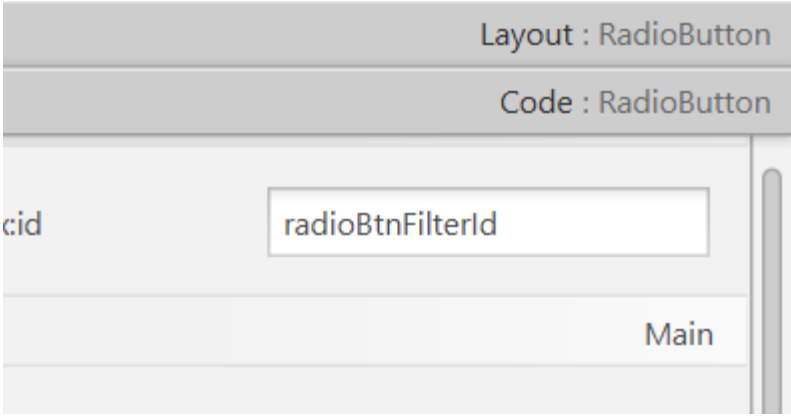
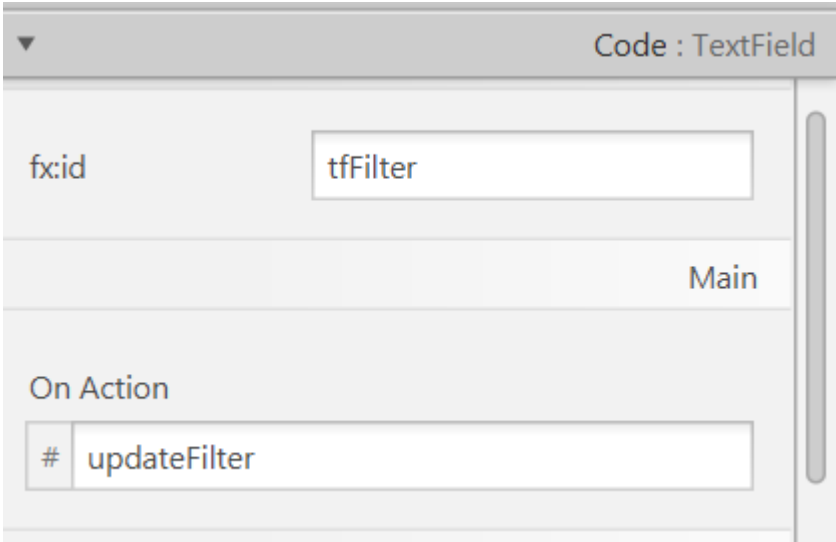
7.



8.

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

9.





```

@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)));
        }
    }
}

```

## AddItemToStoreScreen

```

public class AddItemToStoreScreen extends JFrame {

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

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

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

        JMenuItem viewCartMenu = new JMenuItem(text:"View cart");
        viewCartMenu.addActionListener(e -> {
            c.showCartScreen();
        });
    }
}

```

## AddDigitalVideoDiscToStoreScreen

```

public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
    void updateAdd(JPanel panel) {
        panel.add(lengthLabel);
        length = new JTextField(columns:2);
        lengthLabel.setLabelFor(length);
        panel.add(length);
        JButton tes = new JButton(text:"add");
        tes.setVisible(aFlag:false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(width:100, height:300));
        addBtn = new JButton(text:"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);
        this.store.addMedia(dvd);
        JOptionPane.showMessageDialog(parentComponent:null, message:"add DVD successfully!");
        clearTextField();
    }

    public void clearTextField(){
        this.title.setText(t:"");
        this.director.setText(t:"");
    }
}

```

Ln 25, Col 30 Ta

## AddCompactDiscToStoreScreen

```

public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
    void updateAdd(JPanel panel) {
        this.numberInput = 6;

        JLabel artistLabel = new JLabel(text:"artist", JLabel.TRAILING);
        panel.add(artistLabel);
        artist = new JTextField(columns:2);
        artist.setPreferredSize(new Dimension(width:150, height:20));
        artistLabel.setLabelFor(artist);
        panel.add(artist);
        JLabel listTrackLabel = new JLabel(text:"List track (each track separated by a comma)", JLabel.TRAILING);
        panel.add(listTrackLabel);
        listTrack = new JTextField(columns:2);
        listTrackLabel.setLabelFor(listTrack);
        panel.add(listTrack);
        JButton tes = new JButton(text:"add");
        tes.setVisible(aFlag:false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(width:100, height:300));
        addBtn = new JButton(text:"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());
    }
}

```

Ln 50

## AddBookToStoreScreen

```

public class AddBookToStoreScreen extends AddItemToStoreScreen {
    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 5;

        JLabel listAuthorLabel = new JLabel(text:"Authors(Names are separated by a comma)", JLabel.TRAILING);
        panel.add(listAuthorLabel);
        listAuthor = new JTextField(columns:2);
        listAuthor.setPreferredSize(new Dimension(width:150, height:20));
        listAuthorLabel.setLabelFor(listAuthor);
        panel.add(listAuthor);

        JButton tes = new JButton(text:"add");
        tes.setVisible(aFlag:false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(width:100, height:300));
        addBtn = new JButton(text:"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(regex:",");
        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);
        }
    }
}

```

Ln 1, Col 1 Tab Size: 4

## Exception:

```

public class PlayerException extends RuntimeException {

    public PlayerException() {
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message) {
        super(message);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause) {
        super(message, cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
        // TODO Auto-generated constructor stub
    }
}

```

## Store

Store

Option

Update Store ▾

View store

View cart

Add Book

Add CD

Add DVD

View cart

Option

AIMS

View cart

title

category

cost

Authors(Names are separated by a comma)

add

## Cart

Cart

Options

CART

Filter

☒ By ID ☐ By Title

Title

Category

Cost

No content in table

Total 0.00\$

Place Order