

1. Swing components

```
import java.awt.*;
import java.awt.event.*;

public class AWTAccumulator extends Frame {
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator() {
        setLayout(new GridLayout(rows:2, cols:2));

        add(new Label(text:"Enter an Integer: "));
        tfInput = new TextField(columns:10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label(text:"The Accumulated Sum is: "));
        tfOutput = new TextField(columns:10);
        tfOutput.setEditable(b:false);
        add(tfOutput);

        setTitle(title:"AWT Accumulator");
        setSize(width:350, height:120);
        setVisible(b:true);

        addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(status:0); // Exit the program
            }
        });
    }
}
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

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

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

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

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

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

        tfOutput = new JTextField(columns:10);
        tfOutput.setEditable(b:false);
        cp.add(tfOutput);

        setTitle(title:"Swing Accumulator");
        setSize(width:350, height:120);
        setVisible(b:true);

        addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(status:0); // Exit the program
            }
        });
    }
}

```

```

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

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

```

```
package hust.soiict.dsai.swing;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

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

    public NumberGrid() {

        tfDisplay = new JTextField();
        tfDisplay.setPreferredSize(new Dimension(width:200, height:30));
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(rows:4,cols: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(title:"Number Grid");
        setSize(width:200, height:200);
        setVisible(b:true);
    }

    Run | Debug
    public static void main(String[] args) {
        new NumberGrid();
    }
}
```

```

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(text:"DEL");
    panelButtons.add(btnDelete);
    btnDelete.addActionListener(btnListener);

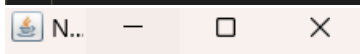
    btnNumbers[0] = new JButton(text:"0");
    panelButtons.add(btnNumbers[0]);
    btnNumbers[0].addActionListener(btnListener);

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

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 deleteString = tfDisplay.getText();
            if (deleteString.length() > 0) {
                deleteString = deleteString.substring(beginIndex:0, deleteString.length() - 1);
            }
            tfDisplay.setText(deleteString);
        } else {
            tfDisplay.setText(t:"");
        }
    }
}

```



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

2. Create a graphical user interface for AIMS with Swing

2.1

```
public class StoreScreen extends JFrame {
    private static Store store = new Store();
    private static Cart cart = new Cart();

    public static void initSetup() {

        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cost:19.95f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star War", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", director:null, length:0, cost:18.99f);
        store.addMedia(dvd1);
        store.addMedia(dvd2);
        store.addMedia(dvd3);

        Book book1 = new Book(title:"1984", category:"Dystopian Fiction", cost:15.99f);
        book1.addAuthor(authorName:"George Orwell");
        Book book2 = new Book(title:"Sapiens: A Brief History of Humankind", category:"History", cost:18.99f);
        book2.addAuthor(authorName:"Yuval Noah Harari");
        Book book3 = new Book(title:"The Great Gatsby", category:"Classic", cost:10.99f);
        book3.addAuthor(authorName:"F. Scott Fitzgerald");
        store.addMedia(book1);
        store.addMedia(book2);
        store.addMedia(book3);

        CompactDisc cd1 = new CompactDisc(title:"Random Access Memories", category:"Music", artist:"Daft Punk", cost:12.99f);
        cd1.addTrack(new Track(title:"Give Life Back to Music", length:272));
        cd1.addTrack(new Track(title:"Get Lucky", length:372));

        CompactDisc cd2 = new CompactDisc(title:"Divide", category:"Music", artist:"Ed Sheeran", cost:14.99f);
        cd2.addTrack(new Track(title:"Shape of You", length:234));
        cd2.addTrack(new Track(title:"Perfect", length:263));
        cd2.addTrack(new Track(title:"Castle on the Hill", length:261));

        CompactDisc cd3 = new CompactDisc(title:"Thriller", category:"Music", artist:"Michael Jackson", cost:19.99f);
        cd3.addTrack(new Track(title:"Beat It", length:258));
        cd3.addTrack(new Track(title:"Thriller", length:357));
        cd3.addTrack(new Track(title:"Billie Jean", length:293));
    }
}
```

```

        store.addMedia(cd1);
        store.addMedia(cd2);
        store.addMedia(cd3);
    }

    Run | Debug
    public static void main(String[] args) {
        initSetup();
        new StoreScreen(store);
    }

    public StoreScreen(Store store) {
        StoreScreen.store = store;
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());

        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(), BorderLayout.CENTER);

        setTitle(title:"Store");
        setSize(width:1024, height:768);
        setVisible(b:true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

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

```

```
JMenuBar createMenuBar() {  
  
    JMenu menu = new JMenu(s:"Options");  
  
    JMenu smUpdateStore = new JMenu(s:"Update Store");  
    JMenuItem smAddBook = new JMenuItem(text:"Add Book");  
    JMenuItem smAddCD = new JMenuItem(text:"Add CD");  
    JMenuItem smAddDVD = new JMenuItem(text:"Add DVD");  
  
    smUpdateStore.add(smAddBook);  
    smUpdateStore.add(smAddCD);  
    smUpdateStore.add(smAddDVD);  
  
    smAddBook.addActionListener(new btnMenuListener());  
    smAddCD.addActionListener(new btnMenuListener());  
    smAddDVD.addActionListener(new btnMenuListener());  
  
    menu.add(smUpdateStore);  
  
    JMenuItem viewStoreMenu = new JMenuItem(text:"View store");  
    JMenuItem viewCartMenu = new JMenuItem(text:"View cart");  
    menu.add(viewStoreMenu);  
    menu.add(viewCartMenu);  
    viewStoreMenu.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            new StoreScreen(store);  
        }  
    });  
    viewCartMenu.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            new CartScreen(cart);  
        }  
    });  
}
```

```

private class btnMenuListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
        if (command.equals(anObject:"Add Book")) {
            new AddBookToStoreScreen(store);
        } else if (command.equals(anObject:"Add CD")) {
            new AddCompactDiscToStoreScreen(store);
        } else if (command.equals(anObject:"Add DVD")) {
            new AddDigitalVideoDiscToStoreScreen(store);
        }
    }
}

JPanel createHeader() {

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

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

    JButton cart = new JButton(text:"View cart");
    cart.setPreferredSize(new Dimension(width:100, height:50));
    cart.setMaximumSize(new Dimension(width:100, height:50));
    cart.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            new CartScreen(cart);
        }
    });
}

```



```

        header.add(Box.createRigidArea(new Dimension(width:10, height:10)));
        header.add(title);
        header.add(Box.createHorizontalGlue());
        header.add(cart);
        header.add(Box.createRigidArea(new Dimension(width:10, height:10)));

        return header;
    }

    JPanel createCenter() {

        JPanel center = new JPanel();
        center.setLayout(new GridLayout(rows:3, cols:3, hgap:2, vgap:2));

        ArrayList<Media> mediaInStore = store.getItemsInStore();
        for (int i = 0; i < mediaInStore.size(); i++) {
            MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
            center.add(cell);
        }

        return center;
    }
}

```

2.1.4

```

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

        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:20));
        title.setAlignmentX(CENTER_ALIGNMENT);

        JLabel cost = new JLabel(""+media.getCost()+"$");
        cost.setAlignmentX(CENTER_ALIGNMENT);

        JPanel container = new JPanel();
        container.setLayout(new FlowLayout(FlowLayout.CENTER));

        JButton addToCartButton = new JButton(text:"Add to cart");
        addToCartButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    String message = cart.addMedia(media);
                    JOptionPane.showMessageDialog(parentComponent:null, message);
                } catch (LimitExceededException ex) {
                    JOptionPane.showMessageDialog(parentComponent:null, ex.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
        container.add(addToCartButton);
    }
}

```

```

        if (media instanceof Playable) {
            JButton playButton = new JButton(text:"Play");
            playButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {

                    JDialog dialog = new JDialog();
                    dialog.setTitle(media.getTitle());
                    dialog.setSize(width:400, height:300);

                    String mediaInfo = "";
                    try {
                        mediaInfo = "<html>" + media.playGUI().replace("\n", "<br/>") + "</html>";
                        JLabel mediaLabel = new JLabel(mediaInfo);
                        mediaLabel.setVerticalAlignment(JLabel.CENTER);
                        mediaLabel.setHorizontalAlignment(JLabel.CENTER);
                        JScrollPane scrollPane = new JScrollPane(mediaLabel);
                        scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

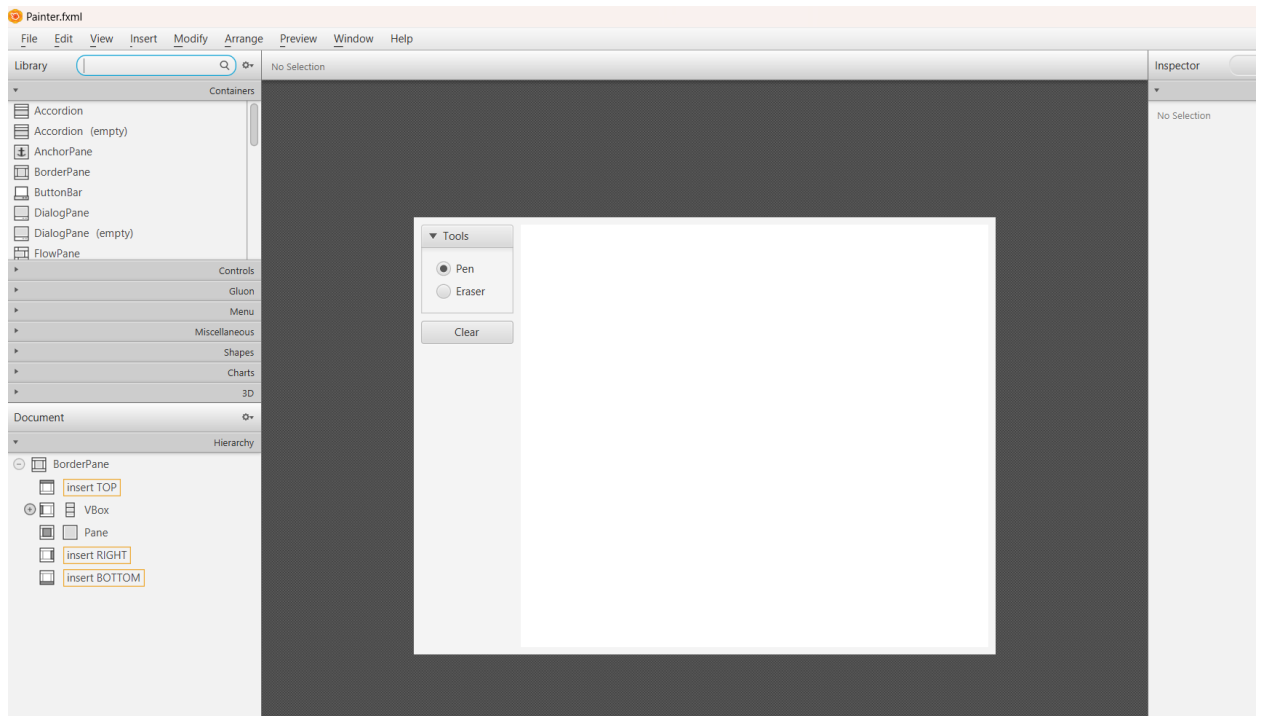
                        dialog.add(scrollPane);
                        dialog.setVisible(b:true);
                    } catch (PlayerException e1) {
                        JOptionPane.showMessageDialog(parentComponent:null, e1.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
                    }
                }
            });
            container.add(playButton);
        }

        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());
        this.add(container);

        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }
}

```

3. JavaFX API



```
public class PainterController {

    @FXML
    private RadioButton eraser;

    @FXML
    private RadioButton pen;

    @FXML
    private ResourceBundle resources;

    @FXML
    private URL location;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    void clearButtonPressed(ActionEvent event) {
        drawingAreaPane.getChildren().clear();
    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Rectangle clipArea = new Rectangle(0, 0, drawingAreaPane.getWidth(), drawingAreaPane.getHeight());
        drawingAreaPane.setClip(clipArea);
        Color inkColor = Color.BLACK;
        if (eraser.isSelected()) {
            inkColor = Color.WHITE;
        }
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, inkColor);
        drawingAreaPane.getChildren().add(newCircle);
    }

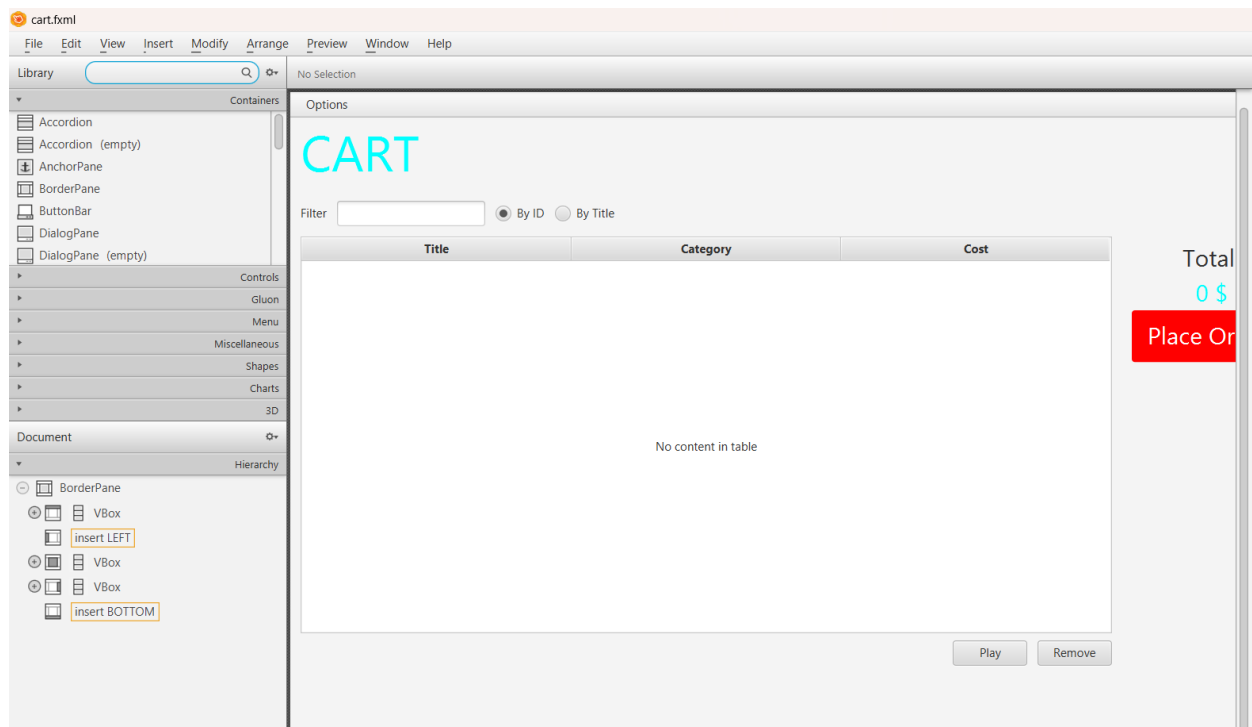
    @FXML
    void initialize() {
        assert drawingAreaPane != null : "fx:id=\"drawingAreaPane\" was not injected: check your FXML file 'Painter.fxml'.";
    }
}
```

```

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
11     @Override
12     public void start(Stage stage) throws Exception {
13         Parent root = FXMLLoader.load(getClass()
14             .getResource(name: "/hust/soict/cybersec/javafx/Painter.fxml"));
15
16         Scene scene = new Scene(root);
17         stage.setTitle("Painter");
18         stage.setScene(scene);
19         stage.show();
20     }
21
22     Run | Debug
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }

```

4. Setting up the View Cart Screen with ScreenBuilder



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

```
public class CartScreen extends JFrame {

    private static Cart cart;

    Run | Debug
    public static void main(String[] args) {
        new CartScreen(cart);
    }

    public CartScreen(Cart cart) {

        super();

        CartScreen.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle(title:"Cart");
        this.setSize(width:1024, height:768);
        this.setVisible(b:true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"cart.fxml"));

                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

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

```

public class CartScreenController {

    private Cart cart;
    private ControllerScreen controllerScreen;

    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private TableColumn<Media, String> colMediaCategory;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private Label totalCost;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private Label playingMedia;

    @FXML
    private Button btnStop;

    @FXML
    private Button btnOrder;
}

```

```

public CartScreenController(Cart cart, ControllerScreen controllerScreen) {
    this.cart = cart;
    this.controllerScreen = controllerScreen;
}

private void updateButtonBar(Media media) {
    btnRemove.setVisible(true);
    btnPlay.setVisible(media instanceof Playable);
}

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

```

```

@FXML
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<>("title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<>("cost"));

    tblMedia.setItems(this.cart.getItemsOrdered());
    totalCost.setText(this.cart.totalCost() + "$");
    btnPlay.setVisible(false);
    btnRemove.setVisible(false);
    playingMedia.setVisible(false);
    btnStop.setVisible(false);

    tffilter.textProperty().addListener((observable, oldValue, newValue) -> showFilterMedia(newValue));
    tblMedia.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newValue) -> {
        if (newValue != null) {
            updateButtonBar(newValue);
            totalCost.setText(cart.totalCost() + "$");
        }
    });
}

```

```

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

@FXML
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    if (media != null) {
        playingMedia.setText("Playing " + media.getTitle() + "....");
        playingMedia.setVisible(true);
        btnStop.setVisible(true);
    }
}

@FXML
void btnStopPressed(ActionEvent event) {
    playingMedia.setVisible(false);
    btnStop.setVisible(false);
}

```



```

@FXML
void btnOrderPressed(ActionEvent event) {
    cart.getItemsOrdered().clear();
    totalCost.setText("0.0$");
    btnOrder.setText("Success!!!");
    btnOrder.setDisable(true);

    PauseTransition pt = new PauseTransition(Duration.seconds(1));
    pt.setOnFinished(e -> {
        btnOrder.setDisable(false);
        btnOrder.setText("Order");
    });
    pt.playFromStart();
}

@FXML
void changeToStoreScreen(ActionEvent event) {
    controllerScreen.showStoreScreen();
}

@FXML
void changeToAddBookScreen(ActionEvent event) {
    controllerScreen.showAddBookScreen();
}

@FXML
void changeToAddCDScreen(ActionEvent event) {
    controllerScreen.showAddCDCreen();
}

@FXML
void changeToAddDVDScreen(ActionEvent event) {
    controllerScreen.showAddDVDScreen();
}

@FXML
void changeToCartScreen(ActionEvent event) {
    controllerScreen.showCartScreen();
}

@FXML
void updateFilter(InputMethodEvent event) {
    System.out.println(event.toString());
}
}

```

12. Create a class which inherits from **Exception**

```
package hust.soict.dsai.aims.exception;

public class PlayerException extends Exception {
    public PlayerException(String message) {
        super(message);
    }
}
```

```
package hust.soict.dsai.aims.media;

import hust.soict.dsai.aims.exception.PlayerException;

public interface Playable {
    public void play() throws PlayerException;
}
```

```
@Override
public void play() throws PlayerException{
    if (this.getLength() > 0) {
        System.out.println("Compactdisc: " + getTitle());
        System.out.println("Artist: " + getArtist());
        System.out.println("Length: " + 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!");
    }
}
```