Họ và tên: Nguyễn Hồng Quân

Mã số sinh viên: 20225908

Lớp: 744521 – Lập trình hướng đối tượng – IT3103

**LAP 05: GUI Programming**

**1. Swing components**

1.1. AWTAccumulator

Code:

package hust.soict.dsai.swing;

import java.awt.Frame;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class AWTAccumulator extends Frame{

    private TextField tfInput;

    private TextField tfOutput;

    private int sum = 0;

    public AWTAccumulator() {

        setLayout(new GridLayout(2, 2));

        add(new Label("Enter an Integer: "));

        tfInput = new TextField(10); //Nhap vao text tu ban phim

        add(tfInput);

        tfInput.addActionListener(new TFInputListener());

        add(new Label("The Accumulated Sum is: "));

        tfOutput = new TextField(10);

        add(tfOutput);

        setTitle("AWT Accumulator");

        setSize(350, 120);

        setVisible(true); // hien thi

        addWindowListener(new WindowAdapter() {

            @Override

            public void windowClosing(WindowEvent e) {

                dispose(); // Đóng frame

            }

        });

    }

    public static void main(String[] args) {

        new AWTAccumulator();

    }

    private class TFInputListener implements ActionListener {

        public void actionPerformed(ActionEvent evt) {

            int numberIn = Integer.parseInt(tfInput.getText());

            sum += numberIn;

            tfInput.setText("");

            tfOutput.setText(sum + "");

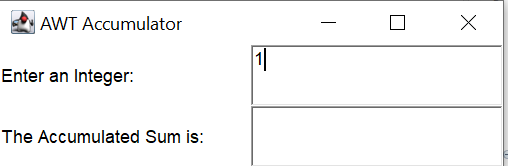
        }

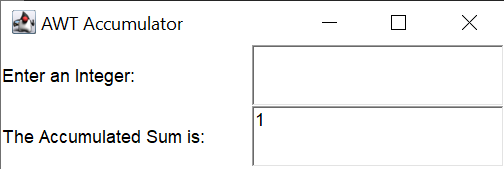
    }

}

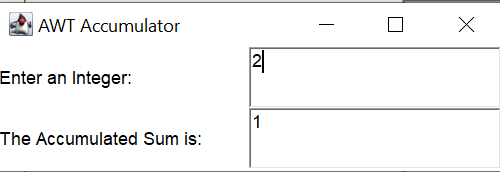
Result:

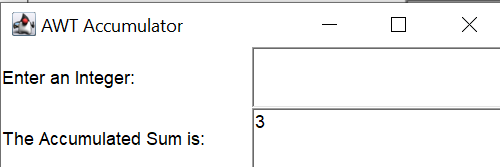
* *Enter an Interger* lần 1 là 1 🡒 *The Accumalated Sum is:* 1.





* *Enter an Interger* lần 2 là 2 🡒 *The Accumalated Sum is:* 3.





1.2. SwingAccumulator

Code:

package hust.soict.dsai.swing;

import java.awt.Container;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

public class SwingAccumulator extends JFrame{

    private JTextField tfInput;

    private JTextField tfOutput;

    private int sum = 0;

    public SwingAccumulator() {

        Container cp = getContentPane();

        cp.setLayout(new GridLayout(2, 2));

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

        tfInput = new JTextField(10);

        cp.add(tfInput);

        tfInput.addActionListener((ActionListener) new TFInputListener());

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

        tfOutput = new JTextField(10);

        tfOutput.setEditable(false);  //Khong cho nhap

        cp.add(tfOutput);

        setTitle("Swing Accumulator");

        setSize(350,120);

        setVisible(true);

    }

    public static void main(String[] args) {

       new SwingAccumulator();

    }

    private class TFInputListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

           int numberIn = Integer.parseInt(tfInput.getText());

           sum += numberIn;

           tfInput.setText("");

            tfOutput.setText(sum + "");

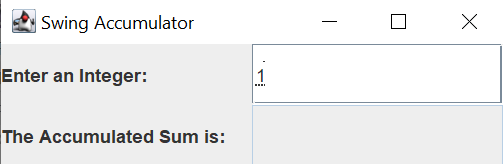
        }

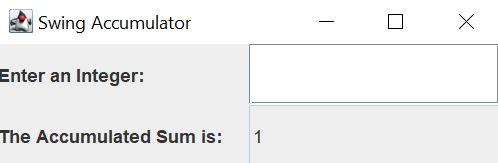
    }

}

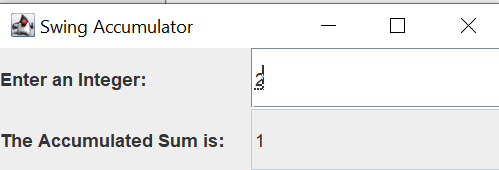
Result:

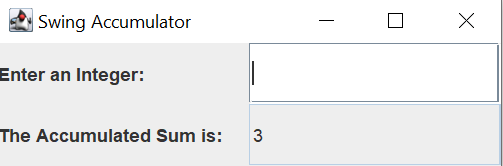
* *Enter an Interger* lần 1 là 1 🡒 *The Accumalated Sum is:* 1.





* *Enter an Interger* lần 2 là 2 🡒 *The Accumalated Sum is:* 3.





**2. Organizing Swing components with Layout Managers**

**2.1. Swing top-level and secondary-level containers**

**2.2. Using JPanel as secondary-level container to organize components**

Code:

package hust.soict.dsai.swing;

import java.awt.BorderLayout;

import java.awt.ComponentOrientation;

import java.awt.Container;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class NumberGrid extends JFrame {

    private JButton[] btnNumbers = new JButton[10];

    private JButton btnDelete, btnResert;

    private JTextField tfDisplay;

    private String str = "";

    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 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);

        btnResert = new JButton("C");

        panelButtons.add(btnResert);

        btnResert.addActionListener(btnListener);

    }

    public String deleteLastCharacter(String str) {

        if (str == null || str.isEmpty()) {

            return str;

        }

        StringBuilder stringBuilder = new StringBuilder(str);

        stringBuilder.deleteCharAt(str.length() - 1);

        return stringBuilder.toString();

    }

    private class ButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            String button = e.getActionCommand();

            if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {

                 str += button;

                tfDisplay.setText(str);

            } else if (button.equals("DEL")) {

                str = deleteLastCharacter(str);

                tfDisplay.setText(str);

            } else {

                str = "";

                tfDisplay.setText(str);

            }

        }

    }

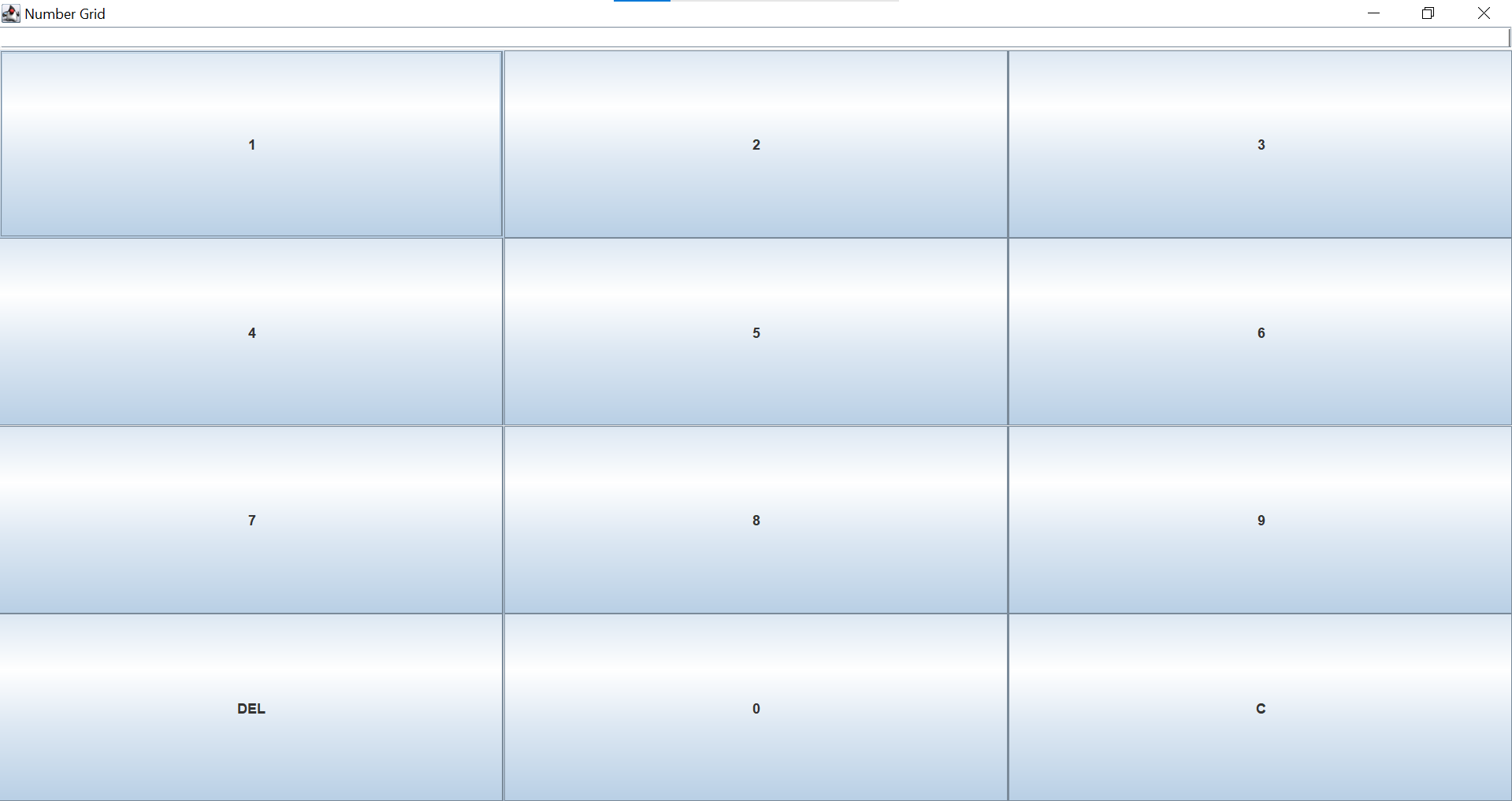
    public static void main(String[] args) {

        new NumberGrid();

    }

}

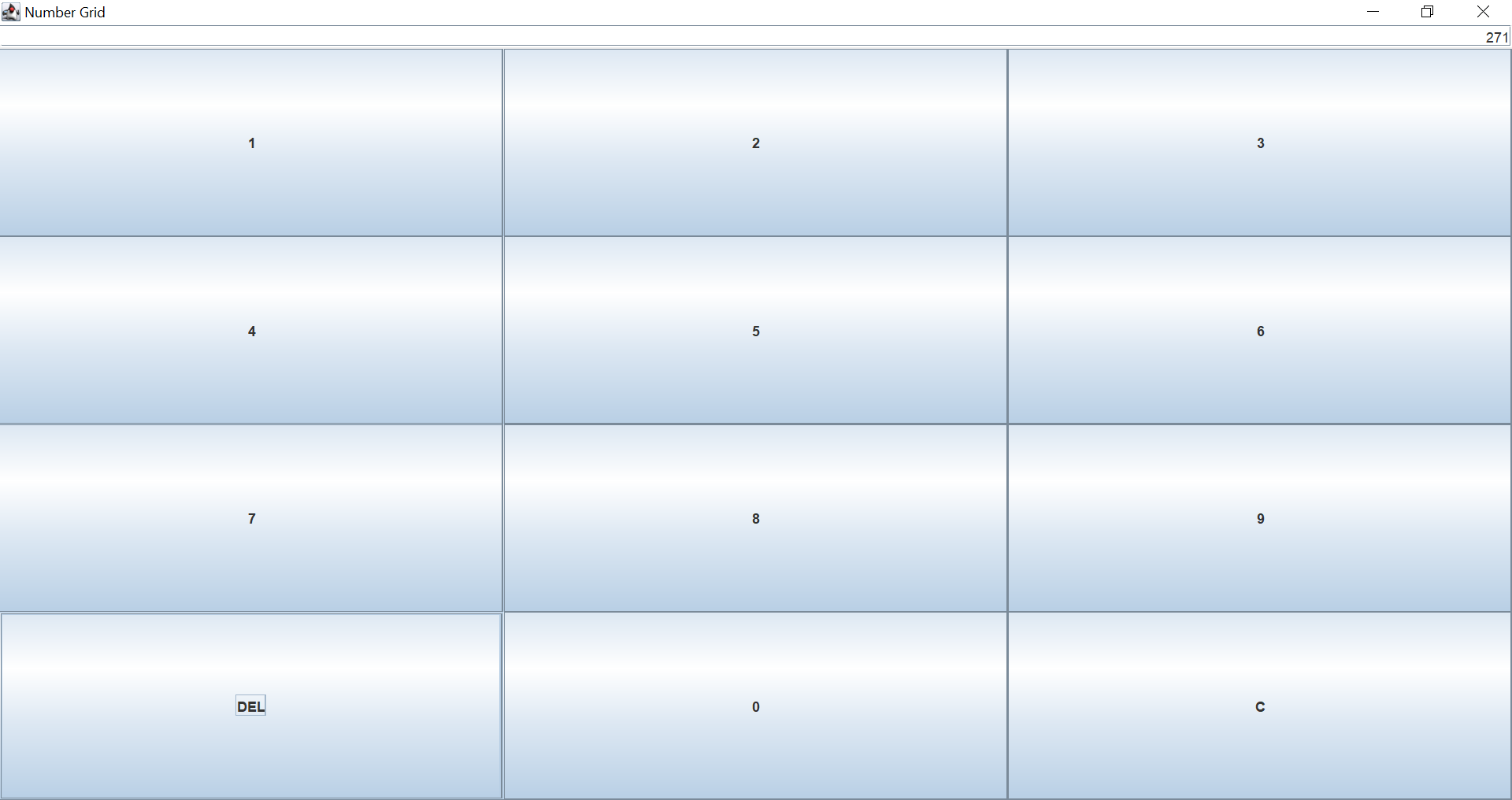
Result:



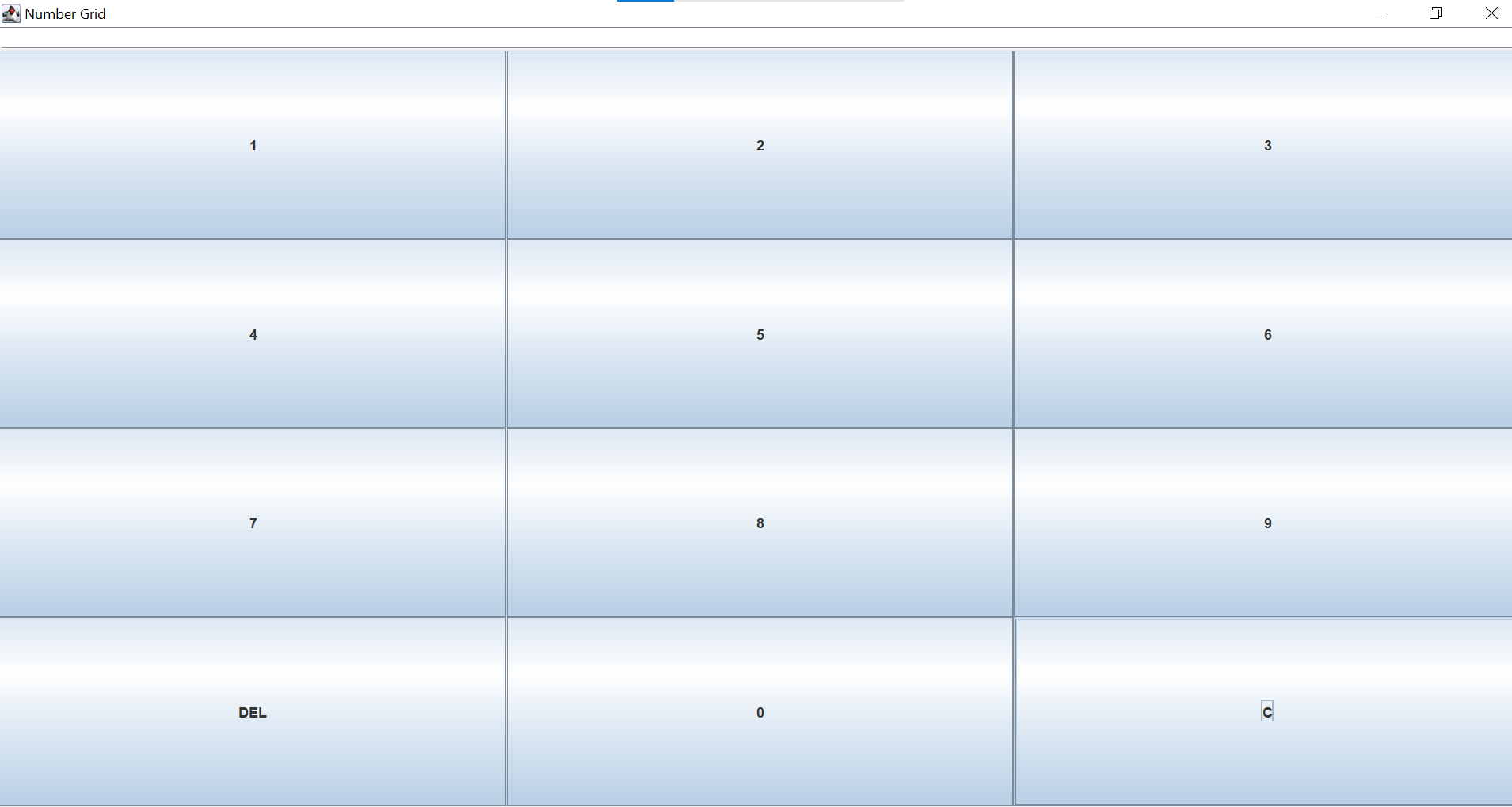
* Nhấn 2710:



* Nhấn **DEL:**



* Nhấn **C:**



**3. Create a graphical user interface for AIMS with Swing**

**3.1. View Store Screen**

**3.1.1. Create the StoreScreen class**

public class StoreScreen extends JFrame {

public static AddDVDscreen sc;

public static AddCDscreen sc1;

public static AddBookscreen sc2;

public static CartScreen scCart;

private Store store;

**3.1.2. The NORTH component**

Create the method createNorth(), which will create our NORTH component:

JPanel createNorth()

{

    JPanel north = new JPanel();

    north.setLayout( new BoxLayout(north, BoxLayout.Y\_AXIS));

    north.add(createMenuBar());

    north.add(createHeader());

    return north;

}

Create the method createMenuBar():

JMenuBar createMenuBar()

{

    JMenu menu =new JMenu("Options");

    JMenu smUpdateStore = new JMenu("Update Store");

    JMenuItem addDVD =new JMenuItem("Add DVD");

    addDVD.addActionListener(new MenuItemListener());

    smUpdateStore.add(addDVD);

    JMenuItem addCD=new JMenuItem("Add CD");

    addCD.addActionListener(new MenuItemListener());

    smUpdateStore.add(addCD);

    JMenuItem addBook=new JMenuItem("Add Book");

    addBook.addActionListener(new MenuItemListener());

    smUpdateStore.add(addBook);

    menu.add(smUpdateStore);

    menu.add(new JMenuItem("View store"));

    JMenuItem viewcart=new JMenuItem("View cart");

    viewcart.addActionListener(new MenuItemListener());

    menu.add(viewcart);

    JMenuBar menuBar = new JMenuBar();

    menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));

    menuBar.add(menu);

    return menuBar;

}

Create the method createHeader():

JPanel createHeader()

 {

     JPanel header = new JPanel();

     header.setLayout(new BoxLayout(header, BoxLayout.X\_AXIS));

     JLabel title = new JLabel("AIMS");

     title.setFont(new Font(title.getFont().getName(),Font.PLAIN, 50));

     title.setForeground(Color.CYAN);

     JButton cart = new JButton("View cart");

     cart.addActionListener(new ButtonListener());

     cart.setPreferredSize(new Dimension(100,50));

     cart.setMaximumSize(new Dimension(100,50));

     header.add(Box.createRigidArea(new Dimension(10,10)));

     header.add(title);

     header.add(Box.createHorizontalGlue());

     header.add(cart);

     header.add(Box.createRigidArea(new Dimension(10,10)));

     return header;

 }

**3.1.3. The CENTER component**

JPanel  createCenter()

 {

     JPanel center = new JPanel();

     center.setLayout(new GridLayout(3,3,2,2));

     ArrayList<Media> mediaInStore = store.getItemsInStore();

     for(int i=0; i<mediaInStore.size(); i++)

     {

         MediaStore cell = new MediaStore(mediaInStore.get(i));

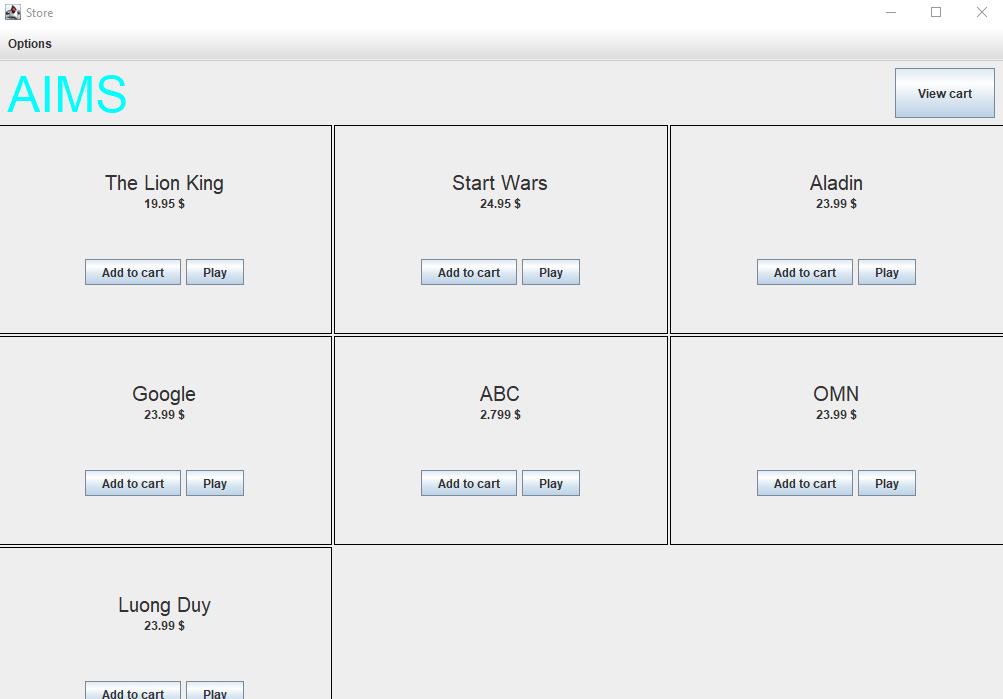
         center.add(cell);

     }

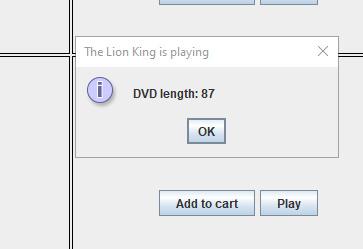
     return center;

 }

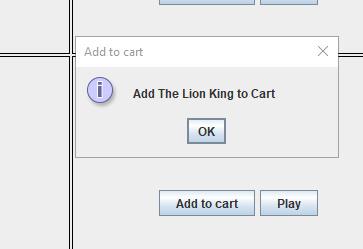
Chạy chương trình:

****

Ấn nút play của “The Lion King”, chương trình sẽ hiển thị ra 1 dialog như sau:

****

Ấn nút Add to cart thì chương trình sẽ hiển thị thông báo:

****

**3.1.4. The MediaStore class**

**3.1.5. Putting it all together**

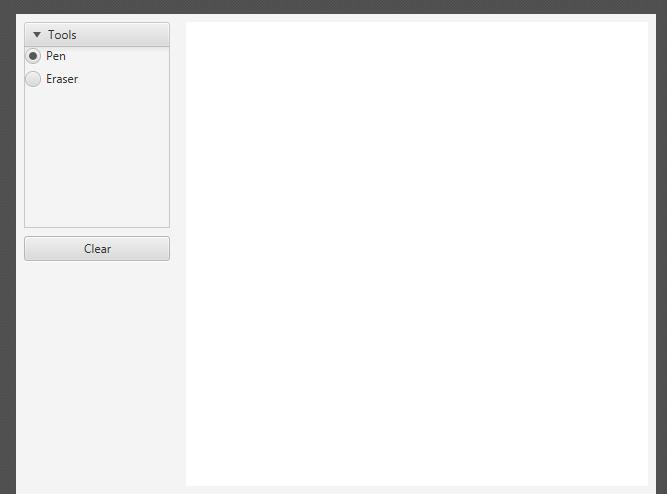
**3.2. Adding more user interaction**

**4. JavaFX API**

**4.1. Create the FXML file**

**4.1.1. Create and open the FXML file in Scene Builder from Eclipse**

**4.1.2. Building the GUI**



**4.2. Create the controller class**

package hust.soict.dsai.javafx;

import javafx.scene.paint.Color;

import javafx.event.ActionEvent;

import javafx.scene.control.RadioButton;

import javafx.scene.input.MouseEvent;

import javafx.fxml.FXML;

import javafx.scene.layout.Pane;

import javafx.scene.shape.Circle;

public class PainterController {

        @FXML

        private Pane drawingAreaPane;

        @FXML

        private RadioButton rbtnPen;

        @FXML

        private RadioButton rbtnEraser;

        @FXML

        void drawingAreaMouseDragged(MouseEvent event) {

            Circle newCircle;

            if(rbtnPen.isSelected())

                 newCircle= new Circle(event.getX(),event.getY(),4,Color.BLACK);

            else newCircle= new Circle(event.getX(),event.getY(),4,Color.WHITE);

            drawingAreaPane.getChildren().add(newCircle);

        }

        @FXML

        void clearButtonPressed(ActionEvent event) {

            drawingAreaPane.getChildren().clear();

        }

}

**4.3. Create the application**

package hust.soict.dsai.javafx;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

public class Painter extends Application{

    @Override

    public void start(Stage stage) throws Exception {

        // TODO Auto-generated method stub

        Parent root= FXMLLoader.load(getClass().getResource("Painter.fxml"));

        Scene scene= new Scene(root);

        stage.setTitle("Painter");

        stage.setScene(scene);

        stage.show();

        }

    public static void main(String[] args)

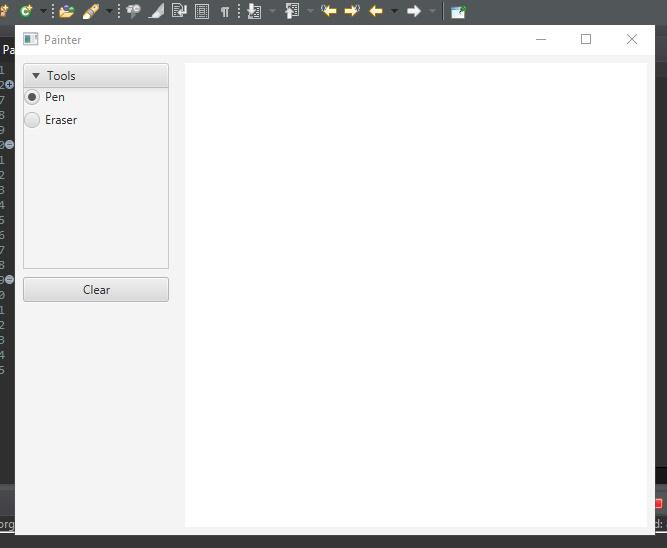
    {

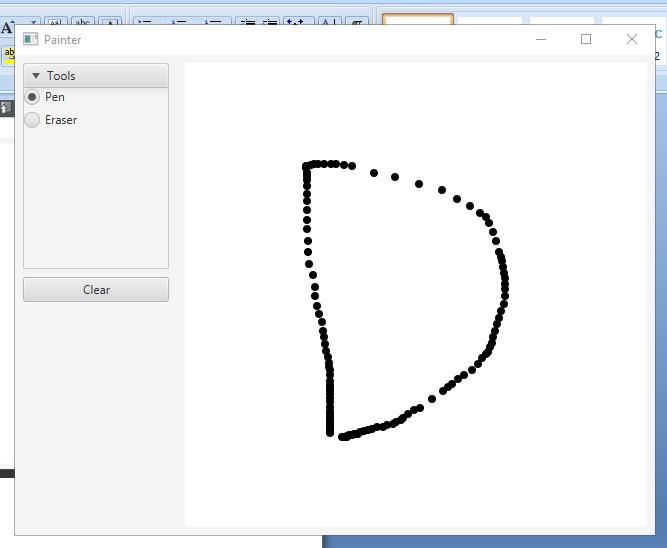
        launch(args);

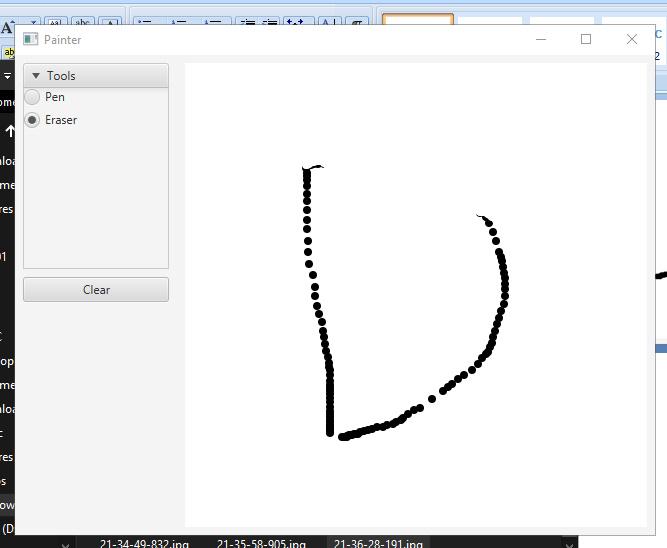
    }

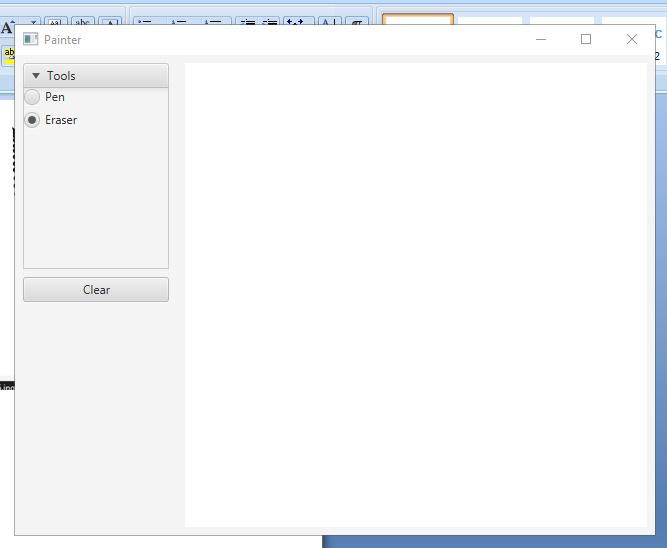
}

Chạy chương trình:

****

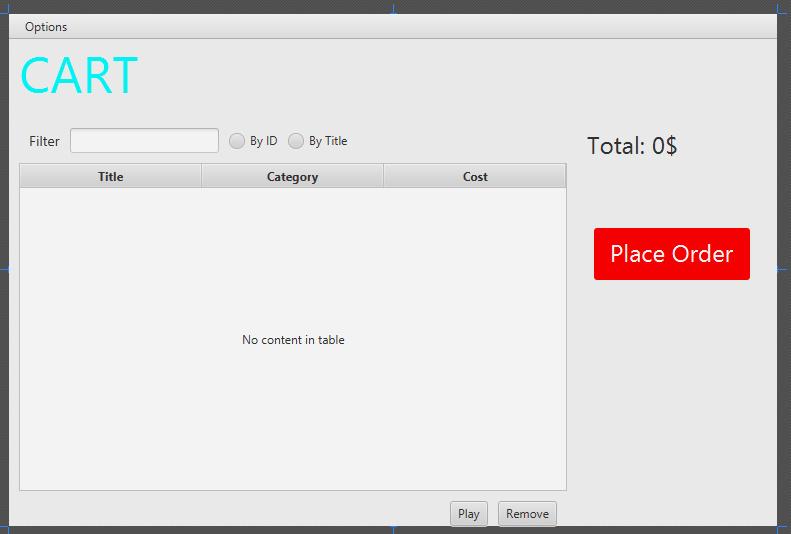
Vẽ chữ D  
****

Xóa một phần chữ D  
****

Xóa toàn bộ chữ D với clear  
****

**4.4. Practice exercise**

**5. Setting up the View Cart Screen with ScreenBuilder**

****

**6. Integrating JavaFX into Swing application**

package hust.soict.dsai.aims.screen;

import java.io.IOException;

import javax.swing.\*;

import hust.soict.dsai.aims.Aims;

import hust.soict.dsai.aims.cart.Cart;

import javafx.application.Platform;

import javafx.embed.swing.JFXPanel;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class CartScreen extends JFrame{

    private Cart cart;

    public CartScreen(Cart cart)

    {   super();

        this.cart=cart;

        JFXPanel fxPanel = new JFXPanel();

        this.add(fxPanel);

        this.setTitle("Cart");

        this.setVisible(true);

        Platform.runLater(new Runnable() {

            public void run()

            {

                try {

                    FXMLLoader loader = new FXMLLoader(getClass().getResource("Cart.fxml"));

                    CartScreenController controller =

                              new CartScreenController(cart);

                    loader.setController(controller);

                    Parent root= loader.load();

                    fxPanel.setScene(new Scene(root));

                }catch(IOException e)

                {

                    e.printStackTrace();

                }

            }

        });

}

}

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

package hust.soict.dsai.aims.screen;

import java.text.DecimalFormat;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import hust.soict.dsai.aims.Aims;

import hust.soict.dsai.aims.cart.Cart;

import hust.soict.dsai.aims.media.CompactDisc;

import hust.soict.dsai.aims.media.DigitalVideoDisc;

import hust.soict.dsai.aims.media.Media;

import hust.soict.dsai.aims.media.Playable;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.beans.value.ChangeListener;

import javafx.beans.value.ObservableValue;

import javafx.fxml.FXML;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {

    private ObservableList<Media> x= FXCollections.observableArrayList();

    private ObservableList<Media> y= FXCollections.observableArrayList();

    private Float total;

    @FXML

    private Label lblTotal;

    @FXML

    private TextField tfFilter;

    private Cart cart;

    @FXML

    private Button btnPlay;

    @FXML

    private RadioButton rbtnFilterID;

    @FXML

    private RadioButton rbtnFilterTitle;

    @FXML

    private Button btnRemove;

    @FXML

    private ToggleGroup filter;

    @FXML

    private TableColumn<Media, String> colMediacategory;

    @FXML

    private TableView<Media> tblMedia;

    @FXML

    private TableColumn<Media, Float> colMediaCost;

    @FXML

    private TableColumn<Media, String> colMediaTitle;

    public CartScreenController(Cart cart)

    {

        super();

        this.cart= cart;

    }

    @FXML

    private void initialize()

    {  //Khoi tao

        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media,String>("title"));

        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media,String>("category"));

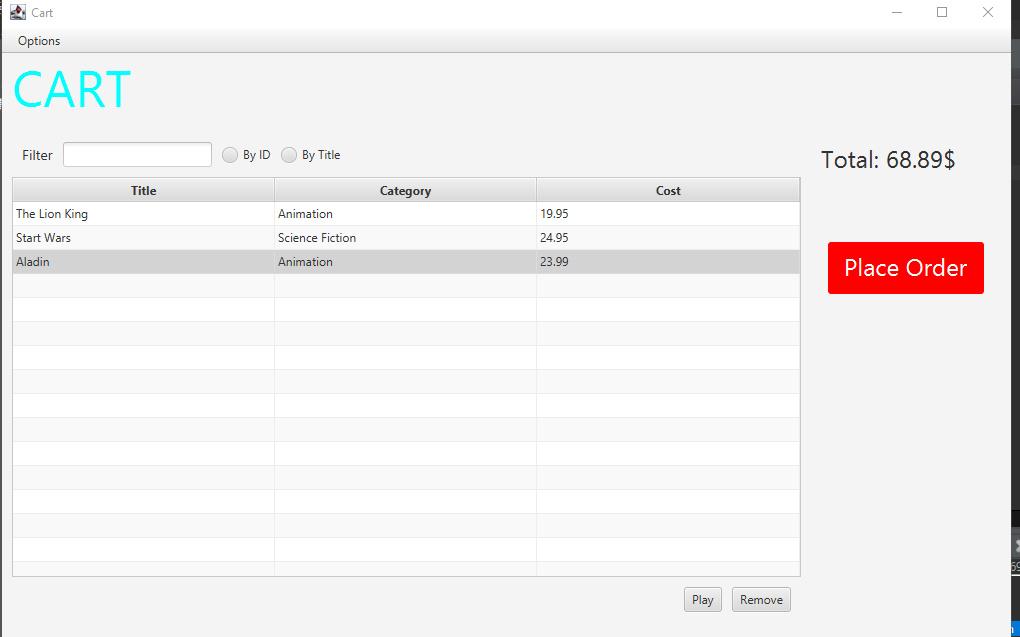
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media,Float>("cost"));

        tblMedia.setItems(this.cart.getItemsOrdered());

        setTotal();

    }

Chạy chương trình

****

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

@FXML

    private void initialize()

    {  //Khoi tao

        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media,String>("title"));

        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media,String>("category"));

        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media,Float>("cost"));

        tblMedia.setItems(this.cart.getItemsOrdered());

        setTotal();

        btnPlay.setVisible(false);

        btnRemove.setVisible(false);

        tblMedia.getSelectionModel().selectedItemProperty().addListener(

                new ChangeListener<Media>()

                {

                    @Override

                    public void changed(ObservableValue<? extends Media> arg0, Media arg1, Media arg2) {

                        // TODO Auto-generated method stub

                        if( arg2 != null)

                            updateButtonBar(arg2);

                                            }

                    private void updateButtonBar(Media arg2) {

                        btnRemove.setVisible( true);

                        if( arg2 instanceof Playable)

                            btnPlay.setVisible( true);

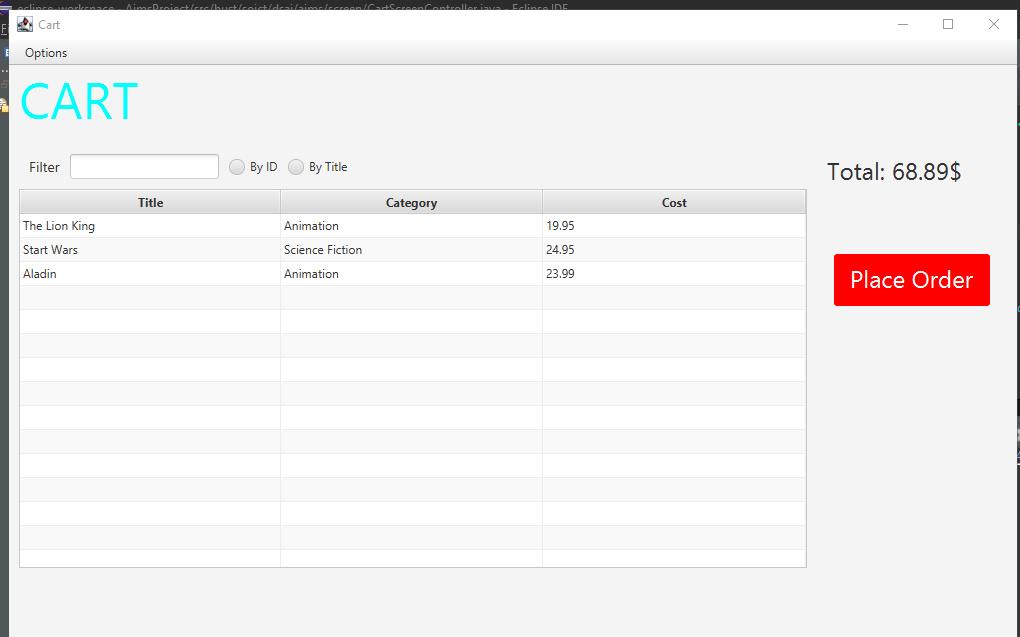
                         else

                            btnPlay.setVisible( false);

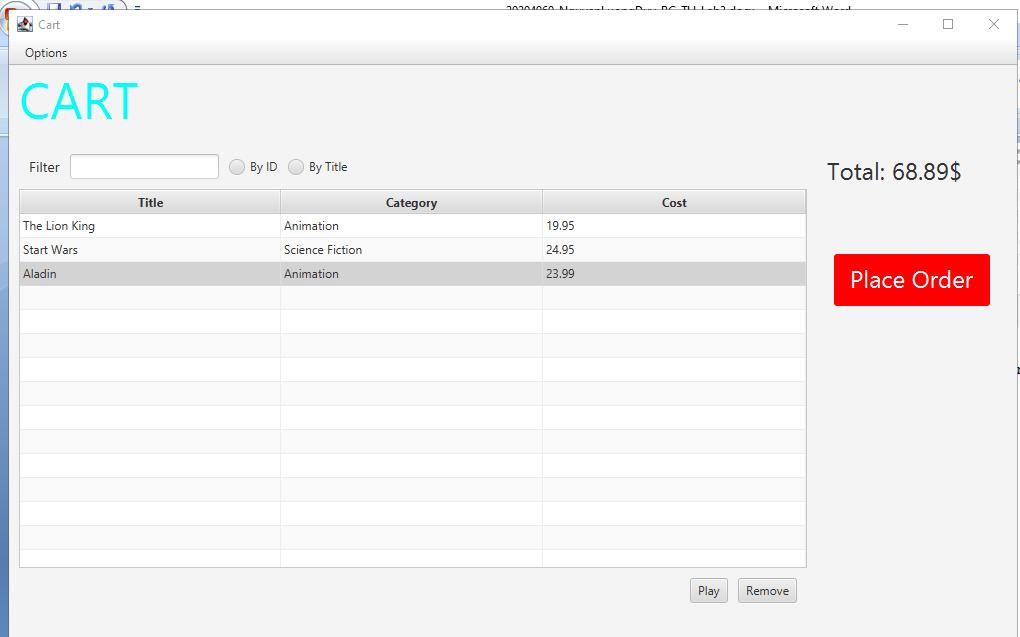
                        };

**Chạy chương trình**

Khi chưa chọn gì thì chương trình sẽ không hiển thị nút Play và Remove

****

Khi chọn 1 phần tử thì chương trình hiển thị nút play và remove



**9. Deleting a media**

@FXML

    void btnRemovePressed(ActionEvent event)

// remove button

    {

        Media media = tblMedia.getSelectionModel().getSelectedItem();

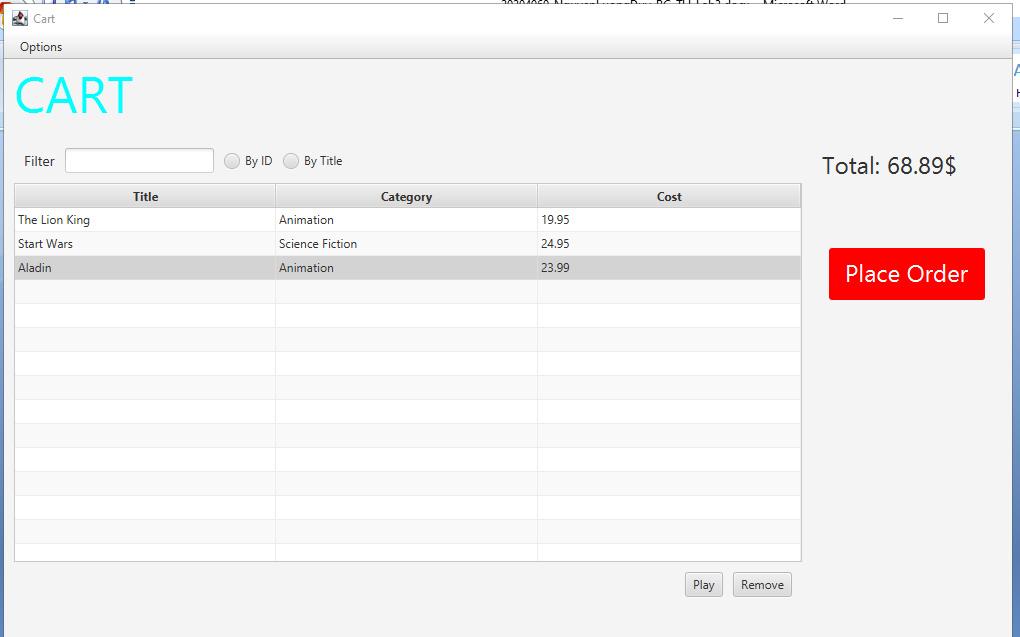
        cart.removeMedia(media);

        setTotal();

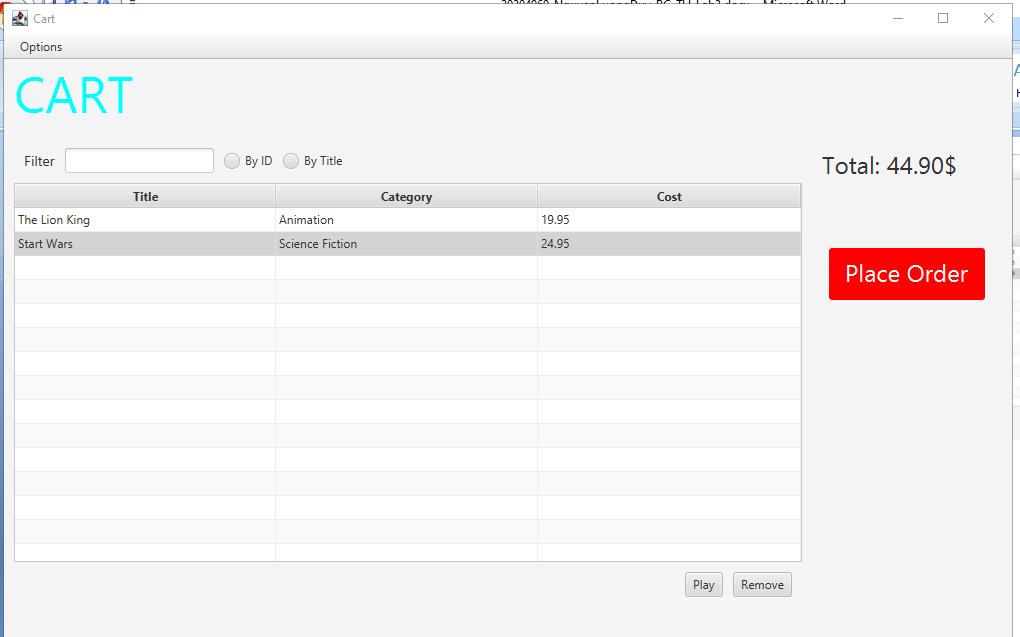
    }

Chạy chương trình

Lúc đầu chương trình đang có 3 items:

****

Chúng ta sẽ xoá đi item có title: “Aladin” bằng cách nhấn button Remove, thì khi đó cart chỉ còn 2 items



**10.Filter items in cart – FilteredList**

tfFilter.textProperty().addListener(new ChangeListener<String>() {

            @Override

            public void changed(ObservableValue<? extends String> arg0, String arg1, String arg2) {

                // TODO Auto-generated method stub

                showFilteredMedia(arg2);

            }

            private void showFilteredMedia(String arg2) {

                if (arg2.length() == 0) {

                    tblMedia.setItems(cart.getItemsOrdered());

                    setTotal();

                } else {

                    if (rbtnFilterID.isSelected()) {

                        tblMedia.setItems(cart.searchID(Integer.parseInt(arg2)));

                        setTotal();

                    }

                    if (rbtnFilterTitle.isSelected()) {

                        tblMedia.setItems(cart.searchTitle(arg2));

                        setTotal();

                    }

                }

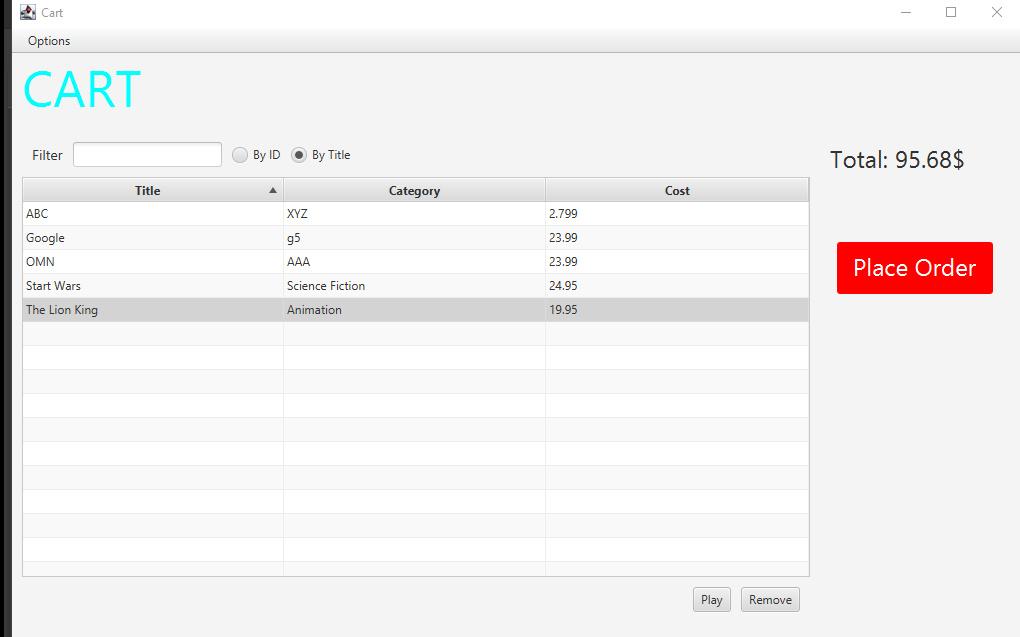
            }

        });

Chạy chương trình

Khi chúng ta thêm hoặc xoá 1 item trong Cart thì phần danh sách trong Table sẽ tự động được sắp xếp dựa trên title hoặc Id:

Trong trường hợp ví dụ này thì chúng ta đang chọn “title” nên chương trình sẽ tự động sắp xếp theo title khi chúng ta thêm vài item vào cart:

****

**11.Complete the Aims GUI application**

Complete the remaining UI of Aims to make a functioning GUI application

• Cart Screen:

- “Place order” Button

- “Play” Button

- The total cost Label

- should updated along with changes in the current cart (add/remove).

- MenuBar

“Place order” Button:

@FXML

    void PlaceOrderPressed(ActionEvent event)

//PlaceOrder button

    {

        cart.itemsOrdered.clear();

        setTotal();

        JFrame frame = new JFrame("JOptionPane showMessageDialog example");

    }

        JOptionPane.showMessageDialog(frame, "Cart was Ordered", "Ordered Information",

                JOptionPane.INFORMATION\_MESSAGE);

    }

“Play” Button:

@FXML

    void btnPlayPressed(ActionEvent event)

    // play button

    {

        Media media = tblMedia.getSelectionModel().getSelectedItem();

        if (media instanceof DigitalVideoDisc) {

            DigitalVideoDisc dvd = (DigitalVideoDisc) media;

            JFrame frame = new JFrame("JOptionPane showMessageDialog example");

            if (dvd.getLength() > 0)

                JOptionPane.showMessageDialog(frame, "DVD length: " + dvd.getLength(), dvd.title + " is playing",

                        JOptionPane.INFORMATION\_MESSAGE);

            else

                JOptionPane.showMessageDialog(frame, "ERROR: DVD length is non-positive!", dvd.title + " ERROR",

                        JOptionPane.ERROR\_MESSAGE);

        } else {

            CompactDisc cd = (CompactDisc) media;

            JFrame frame = new JFrame("JOptionPane showMessageDialog example");

            if (cd.getLength() > 0)

                JOptionPane.showMessageDialog(frame, "CD length: " + cd.getLength(), cd.title + " is playing",

                        JOptionPane.INFORMATION\_MESSAGE);

            else

                JOptionPane.showMessageDialog(frame, "ERROR: CD length is non-positive!", cd.title + " ERROR",

                        JOptionPane.ERROR\_MESSAGE);

        }

    }

The total cost Label

B1: Tạo hàm setTotal() để reset total:

private void setTotal() { // tinh tong

    x = tblMedia.getItems();

    total = (float) 0;

    for (Media m : x)

        total = total + m.getCost();

    DecimalFormat df = new DecimalFormat("#.00");

    lblTotal.setText(df.format(total) + "$");

}

B2: Mỗi khi remove/add thì chúng ta sẽ thêm setTotal() vào để reset Total:

@FXML

    void btnRemovePressed(ActionEvent event)

// remove button

    {

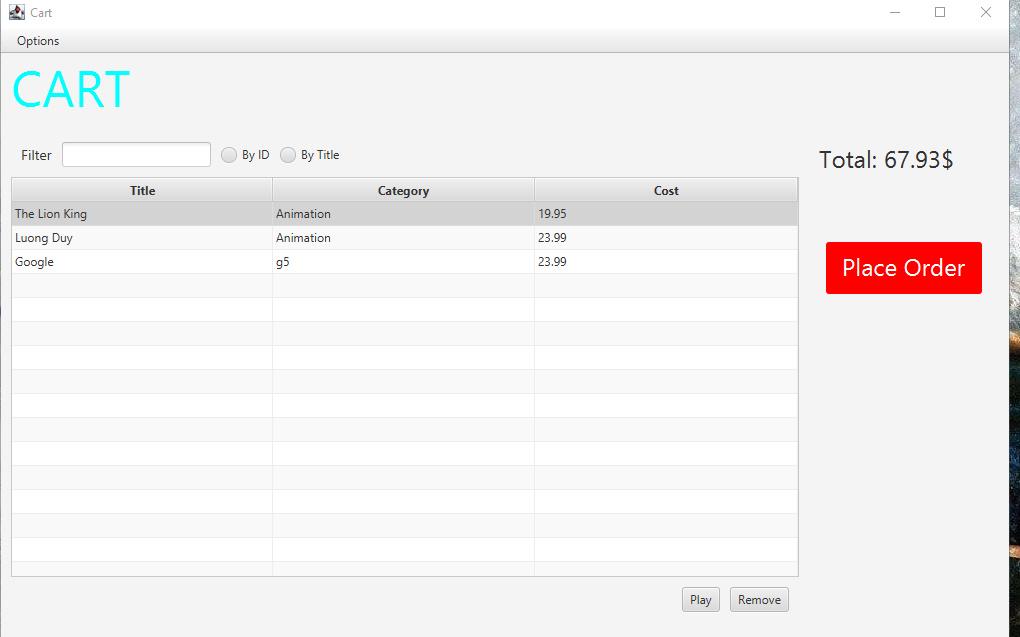
        Media media = tblMedia.getSelectionModel().getSelectedItem();

        cart.removeMedia(media);

        setTotal();

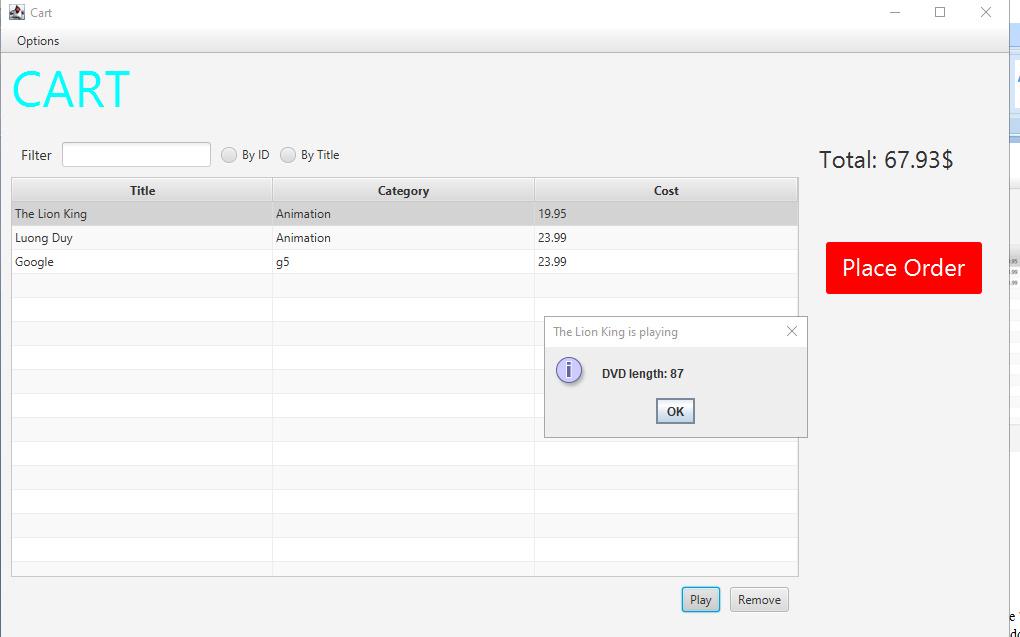
    }

Chạy chương trình

****

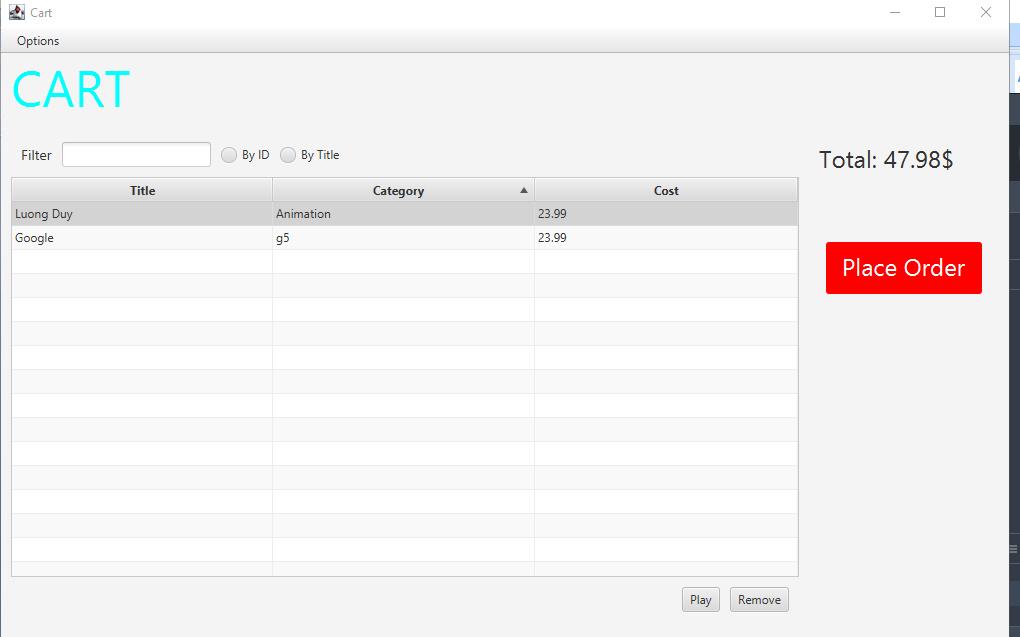
**Test chức năng “Play”:**

Chúng ta sẽ chọn Play của The Lion King thì chương trình sẽ show ra 1 dialog như thế này:

****

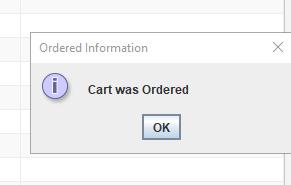
**Test chức năng “total cost”:**

Khi 1 cd là The Lion King thì chương trình sẽ tự động tính lại total cho chúng ta:

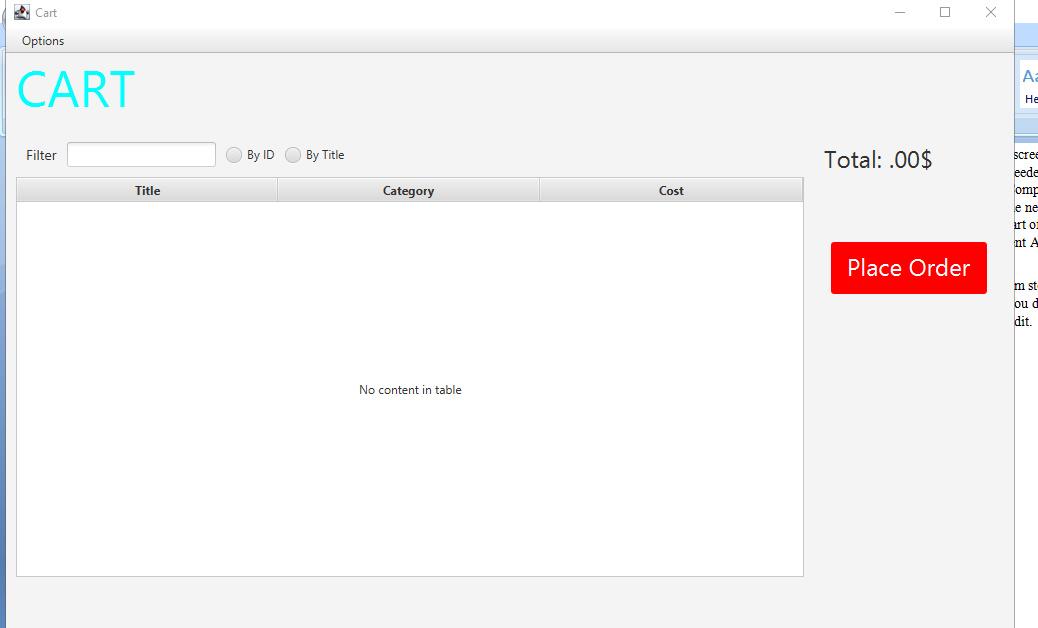
****

**Test chức năng “Place Order”:**

Khi chọn chức năng Place Order thì chương trình sẽ hiển thị thông báo Cart was Ordered

****

Và cart được xoá hết:

****

• **Store Screen:**

- “Add to cart” Button

JButton btncart= new JButton("Add to cart");

       btncart.addActionListener(new ButtonListener());

       container.add(btncart);

private class ButtonListener implements ActionListener

   {

               Aims.Mediacart.addMedia(media);

               JFrame frame = new JFrame("JOptionPane showMessageDialog example");

                  JOptionPane.showMessageDialog(frame,

                            "Add "+media.getTitle()+ " to Cart",

                            "Add to cart",

                            JOptionPane.INFORMATION\_MESSAGE);

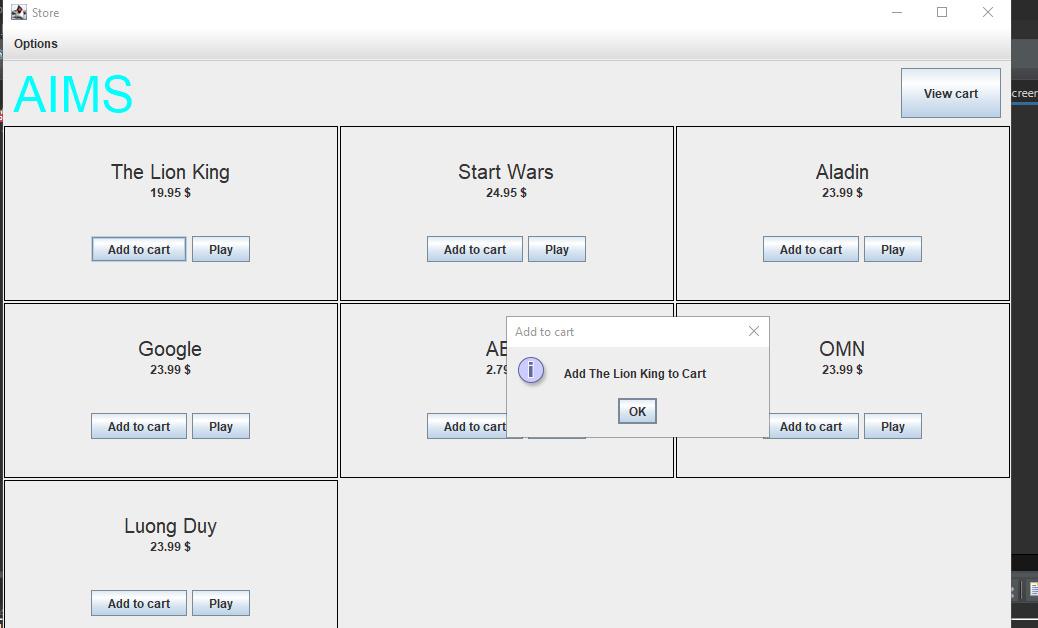
           }

    }

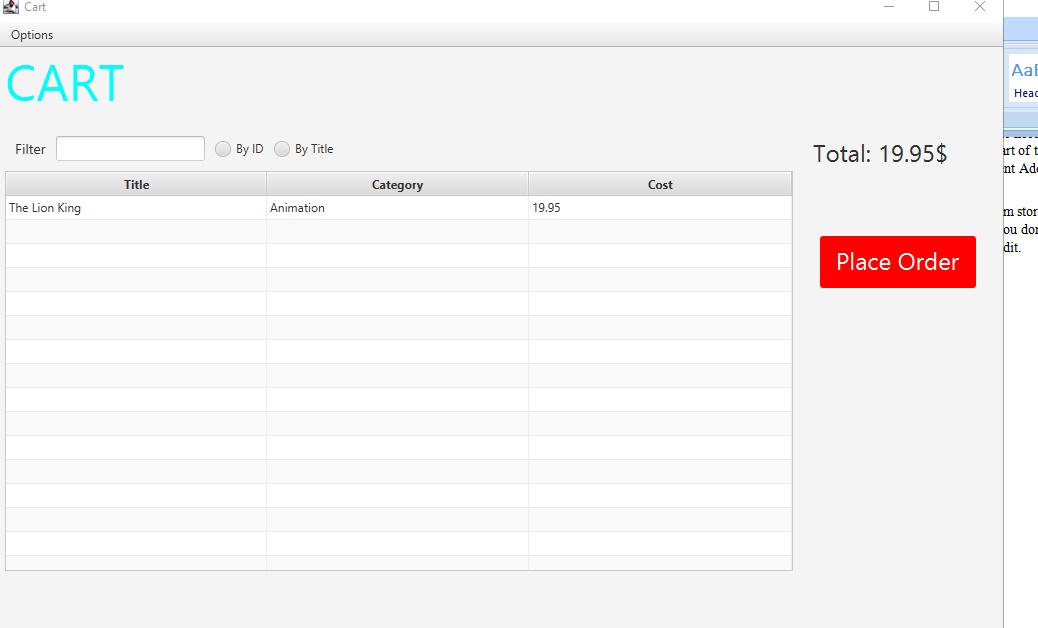
   }

Chạy chương trình

Add to cart của The Lion King thì khi đó chương trình sẽ hiển thị 1 dialog như thế này:

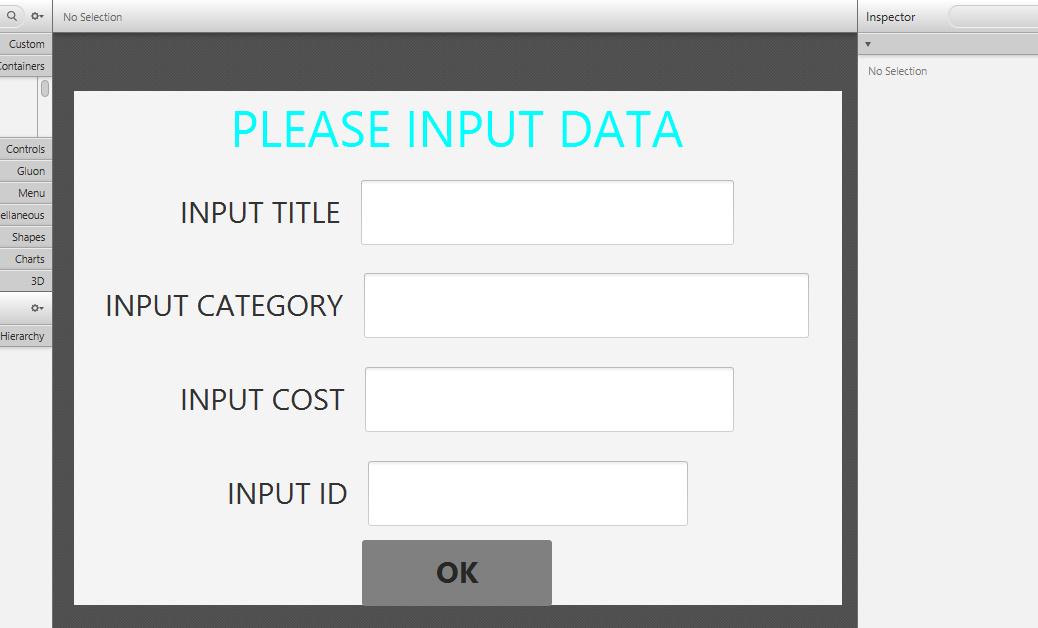


Và Cart được cập nhật



• Update Store Screen:

B1: Tạo 1 file AddItemToStore.fxml trong Screen:

****B2: Tạo file AddBookScreen.java, AddCDScreen.java, AddDVDScreen.java trong Screen

**AddBookScreen.java:**

package hust.soict.dsai.aims.screen;

import java.io.IOException;

import javax.swing.JFrame;

import javafx.application.Platform;

import javafx.embed.swing.JFXPanel;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class AddBookscreen extends JFrame {

    public AddBookscreen()

    {

        super();

        JFXPanel fxPanel = new JFXPanel();

        this.add(fxPanel);

        this.setTitle("Add Book");

        this.setVisible(true);

        Platform.runLater(new Runnable() {

            public void run()

            {

                try {

                    FXMLLoader loader = new FXMLLoader(getClass().getResource("AddItemtoStore.fxml"));

                    AddBookController controller =

                              new AddBookController();

                    loader.setController(controller);

                    Parent root= loader.load();

                    fxPanel.setScene(new Scene(root));

                }catch(IOException e)

                {

                    e.printStackTrace();

                }

            }

        });

    }

}

**AddCDScreen.java**

package hust.soict.dsai.aims.screen;

import java.io.IOException;

import javax.swing.JFrame;

import javafx.application.Platform;

import javafx.embed.swing.JFXPanel;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class AddCDscreen extends JFrame {

    public AddCDscreen()

    {

        super();

        JFXPanel fxPanel = new JFXPanel();

        this.add(fxPanel);

        this.setTitle("Add CD");

        this.setVisible(true);

        Platform.runLater(new Runnable() {

            public void run()

            {

                try {

                    FXMLLoader loader = new FXMLLoader(getClass().getResource("AddItemtoStore.fxml"));

                    AddCDController controller =

                              new AddCDController();

                    loader.setController(controller);

                    Parent root= loader.load();

                    fxPanel.setScene(new Scene(root));

                }catch(IOException e)

                {

                    e.printStackTrace();

                }

            }

        });

    }

}

**AddDVDScreen.java**

package hust.soict.dsai.aims.screen;

import java.io.IOException;

import javax.swing.JFrame;

import javafx.application.Platform;

import javafx.embed.swing.JFXPanel;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class AddDVDscreen extends JFrame {

    public AddDVDscreen()

    {

        super();

        JFXPanel fxPanel = new JFXPanel();

        this.add(fxPanel);

        this.setTitle("Add DVD");

        this.setVisible(true);

        Platform.runLater(new Runnable() {

            public void run()

            {

                try {

                    FXMLLoader loader = new FXMLLoader(getClass().getResource("AddItemtoStore.fxml"));

                    AddDVDController controller =

                              new AddDVDController();

                    loader.setController(controller);

                    Parent root= loader.load();

                    fxPanel.setScene(new Scene(root));

                }catch(IOException e)

                {

                    e.printStackTrace();

                }

            }

        });

    }

}

B3: Tạo file AddBookController.java, AddCDControler.java, AddDVDController.java trong Screen:

**AddBookController.java**

package hust.soict.dsai.aims.screen;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import hust.soict.dsai.aims.Aims;

import hust.soict.dsai.aims.media.Book;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.TextField;

public class AddBookController {

    @FXML

    private TextField tfCost;

    @FXML

    private TextField tfTitle;

    @FXML

    private TextField tfCategory;

    @FXML

    private TextField tfID;

    @FXML

    void OkPressed(ActionEvent event) {

      Book book =new Book();

      book.setTitle(tfTitle.getText());

      book.setCategory(tfCategory.getText());

      book.setId(Integer.parseInt(tfID.getText()));

      book.setCost(Float.parseFloat(tfCost.getText()));

      Aims.Mediastore.addMedia(book);

      JFrame frame = new JFrame("JOptionPane showMessageDialog example");

      JOptionPane.showMessageDialog(frame,

                "Update Store",

                book.getTitle()+ " Added to Store",

                JOptionPane.INFORMATION\_MESSAGE);

      new StoreScreen(Aims.Mediastore);

      StoreScreen.sc2.setVisible(false);

    }

    @FXML

    void e8e408(ActionEvent event) {

    }

}

**AddCDControler.java**

package hust.soict.dsai.aims.screen;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import hust.soict.dsai.aims.Aims;

import hust.soict.dsai.aims.media.CompactDisc;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.TextField;

public class AddCDController {

    @FXML

    private TextField tfCost;

    @FXML

    private TextField tfTitle;

    @FXML

    private TextField tfCategory;

    @FXML

    private TextField tfID;

    @FXML

    void OkPressed(ActionEvent event) {

      CompactDisc cd =new CompactDisc();

      cd.setTitle(tfTitle.getText());

      cd.setCategory(tfCategory.getText());

      cd.setId(Integer.parseInt(tfID.getText()));

      cd.setCost(Float.parseFloat(tfCost.getText()));

      Aims.Mediastore.addMedia(cd);

      JFrame frame = new JFrame("JOptionPane showMessageDialog example");

      JOptionPane.showMessageDialog(frame,

                "Update Store",

                cd.getTitle()+ " Added to Store",

                JOptionPane.INFORMATION\_MESSAGE);

      new StoreScreen(Aims.Mediastore);

      StoreScreen.sc1.setVisible(false);

    }

    @FXML

    void e8e408(ActionEvent event) {

    }

}

**AddDVDController.java**

package hust.soict.dsai.aims.screen;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import hust.soict.dsai.aims.Aims;

import hust.soict.dsai.aims.media.DigitalVideoDisc;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.TextField;

public class AddDVDController {

    @FXML

    private TextField tfCost;

    @FXML

    private TextField tfTitle;

    @FXML

    private TextField tfCategory;

    @FXML

    private TextField tfID;

    @FXML

    void OkPressed(ActionEvent event) {

      DigitalVideoDisc dvd =new DigitalVideoDisc();

      dvd.setTitle(tfTitle.getText());

      dvd.setCategory(tfCategory.getText());

      dvd.setId(Integer.parseInt(tfID.getText()));

      dvd.setCost(Float.parseFloat(tfCost.getText()));

      Aims.Mediastore.addMedia(dvd);

      JFrame frame = new JFrame("JOptionPane showMessageDialog example");

      JOptionPane.showMessageDialog(frame,

                "Update Store",

                dvd.getTitle()+ " Added to Store",

                JOptionPane.INFORMATION\_MESSAGE);

      new StoreScreen(Aims.Mediastore);

      StoreScreen.sc.setVisible(false);

    }

    @FXML

    void e8e408(ActionEvent event) {

    }

}

B4: Sửa JmenuItem trong hàm createMenuBar() trong StoreScreen.java

JMenuBar createMenuBar()

 {

     JMenu menu =new JMenu("Options");

     JMenu smUpdateStore = new JMenu("Update Store");

     JMenuItem addDVD =new JMenuItem("Add DVD");

     addDVD.addActionListener(new MenuItemListener());

     smUpdateStore.add(addDVD);

     JMenuItem addCD=new JMenuItem("Add CD");

     addCD.addActionListener(new MenuItemListener());

     smUpdateStore.add(addCD);

     JMenuItem addBook=new JMenuItem("Add Book");

     addBook.addActionListener(new MenuItemListener());

     smUpdateStore.add(addBook);

     menu.add(smUpdateStore);

     menu.add(new JMenuItem("View store"));

     JMenuItem viewcart=new JMenuItem("View cart");

     viewcart.addActionListener(new MenuItemListener());

     menu.add(viewcart);

     JMenuBar menuBar = new JMenuBar();

     menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));

     menuBar.add(menu);

     return menuBar;

 }

B5: Thêm hàm MenuItemListener():

public class MenuItemListener implements ActionListener {

         @Override

         public void actionPerformed(ActionEvent e) {

                String menuitem = e.getActionCommand();

                if(menuitem.equals("Add DVD"))

                      {  setVisible(false);

                         sc= new AddDVDscreen();

                         sc.setSize(934,595);

                      }

                else if(menuitem.equals("Add CD"))

                {

                    setVisible(false);

                    sc1=new AddCDscreen();

                    sc1.setSize(934,595);

                }

                else if(menuitem.equals("Add Book"))

                {

                    setVisible(false);

                    sc2=new AddBookscreen();

                    sc2.setSize(934,595);

                }

                else if(menuitem.equals("View cart"))

                {    setVisible(false);

                     scCart =new CartScreen(Aims.Mediacart);

                     scCart.setSize(1024,768);

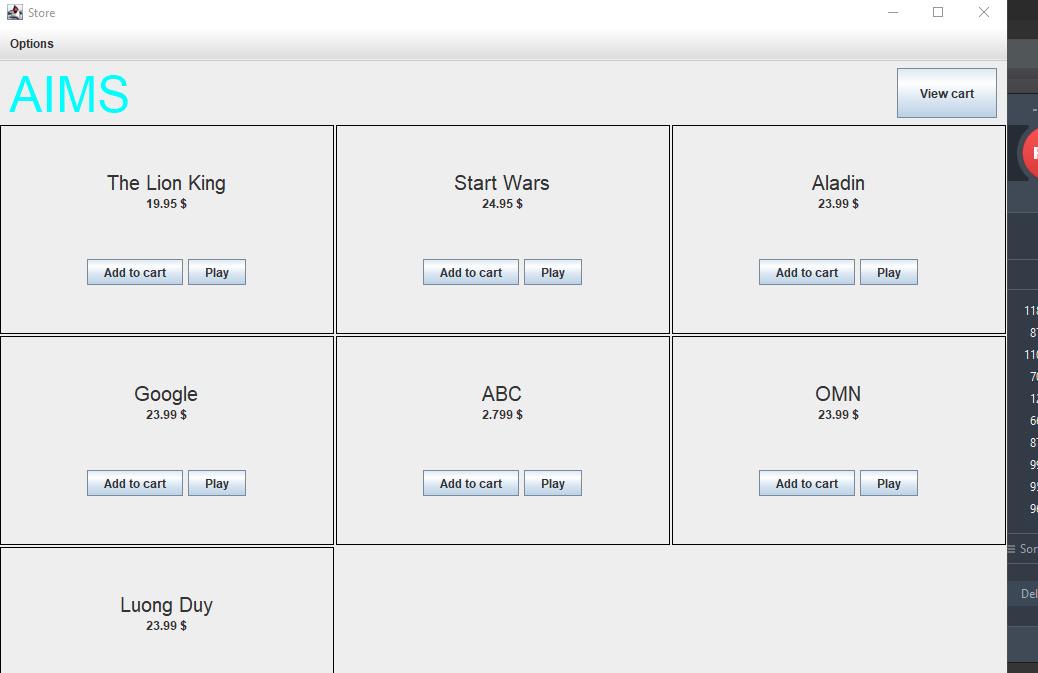
                }

            }

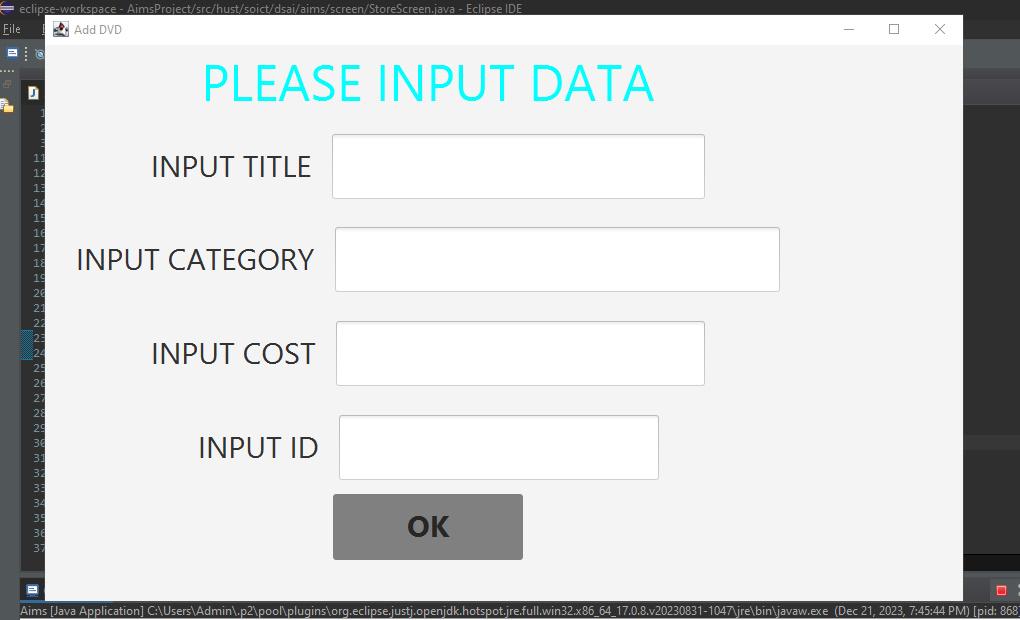
            }

}

Chạy chương trình:

****

Chọn Option -> UpdateStore -> AddDVD, khi đó chương trình sẽ hiển thị ra

****

Chức năng ViewStore, ViewCart:

ViewCart:

JMenuItem viewcart=new JMenuItem("View cart");

     viewcart.addActionListener(new MenuItemListener());

     menu.add(viewcart);

public class MenuItemListener implements ActionListener {

         @Override

         public void actionPerformed(ActionEvent e) {

                String menuitem = e.getActionCommand();

                …

                …

                else if(menuitem.equals("View cart"))

                {    setVisible(false);

                     scCart =new CartScreen(Aims.Mediacart);

                     scCart.setSize(1024,768);

                }

View Store: Thêm actionEvent vào button viewStore:

@FXML

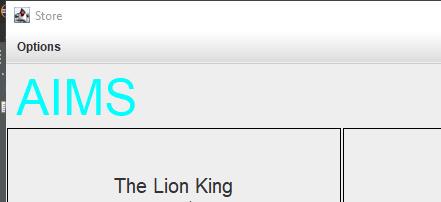
    void ViewStorePressed(ActionEvent event) {

        new StoreScreen(Aims.Mediastore);

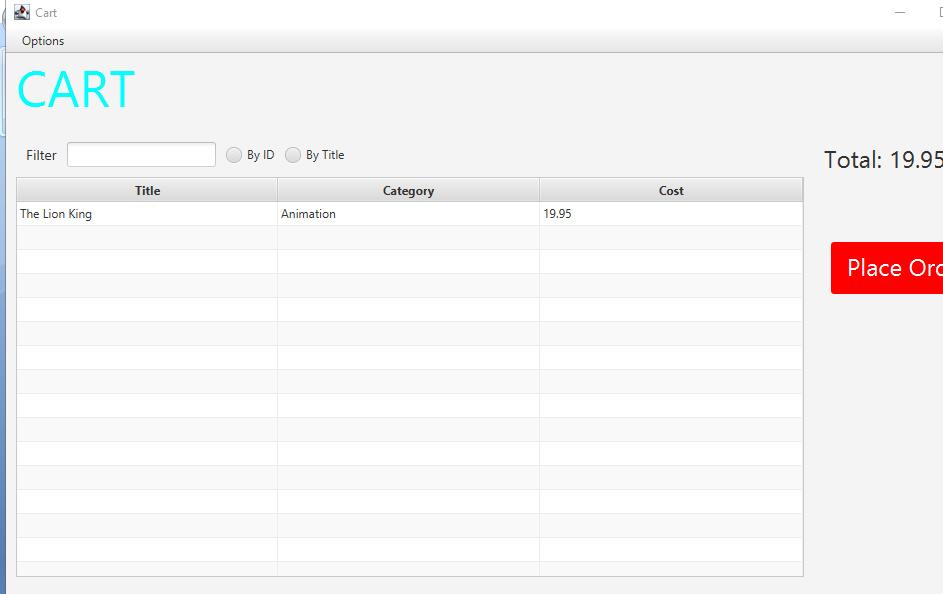
        StoreScreen.scCart.setVisible(false);

    }

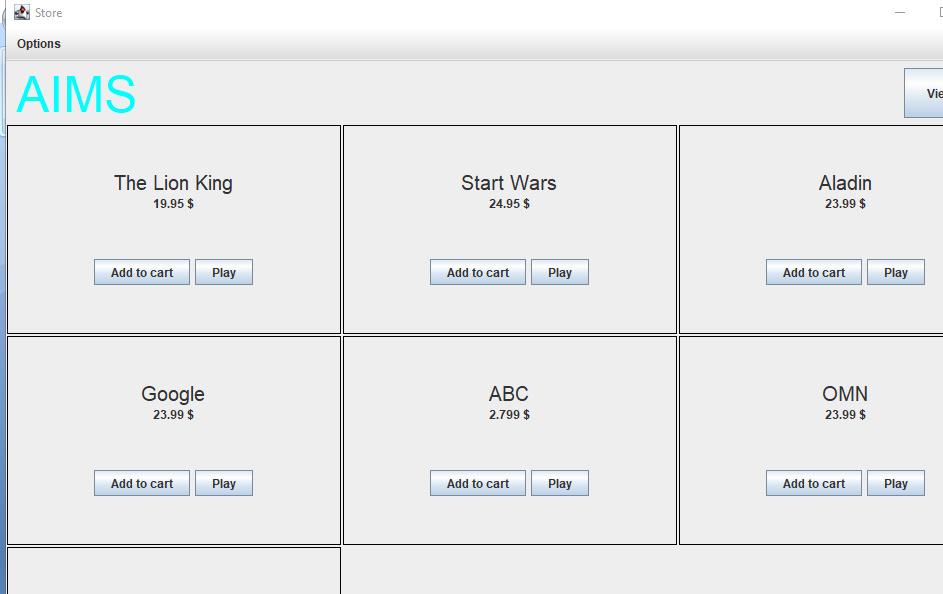
Chạy chương trình



Chọn options -> View Cart:



Chọn Option->View Store: chương trình sẽ hiển thị ra Store Menu



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

Sửa hàm ID của addBookController, addDVDController,addCDController để ID không thể nhận giá trị âm

try {

          int id = Integer.parseInt(tfID.getText());

          if (id >= 0) {

              book.setId(id);

          } else {

              throw new IllegalArgumentException("ID không thể là giá trị âm");

          }

      } catch (NumberFormatException e) {

          System.err.println("Lỗi: ID phải là số nguyên");

          e.printStackTrace();

      }

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

**13.1.Create new class named PlayerException**

package hust.soict.dsai.aims.exception;

public class PlayerException extends Exception {

    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

    }

}

**13.2. Raise the PlayerException in the play() method**

Cập nhật phương thức Play() trong DigitalVideoDisc:

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("ERROR: DVD length is non-positive!");

        }

    }

Cập nhật phương thức Play() trong Track:

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("ERROR: Track length is non-positive!");

            }

    }

**13.3.Update play() in the Playable interface**

Cập nhập Interface Playable():

public interface Playable {

   public void play() throws PlayerException;

**13.4.Update play() in CompactDisc**

Cập nhập Play trong CD:

public void play() throws PlayerException

    {

        if(this.getLength()>0)

        {

            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: CD length is non-postive!");

        }

    }

**14.Update the Aims class**

- The Aims class must be updated to handle any exceptions generated when the play() methods are called. What happens when you don’t update for them to catch?

- Try to use try-catch block when you call the play() method of Media's objects. With all these steps, you have practiced with User-defined Exception (PlayerException), try-catch block and also throw. The try-catch block is used in the main method of class Aims.java and in the play() method of the CompactDisc.java. Print all information of the exception object, e.g.

getMessage(), toString(), printStackTrace(), display a dialog box to the user with the content of the exception.

The example of codes and results for the play() of DigitalVideoDisc in Swing are illustrated in the following figure

**15.Modify the equals() method of Media class**

**16.Reading Document**

**17. Update Aims class diagram**