

BÁO CÁO THỰC HÀNH LAB 5
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
LÓP: 143577 (LT) – 732870 (TN)

Table of Contents

1. Swing components.....	3
1.1. AWTAccumulator.....	3
1.2. SwingAccumulator.....	4
1.3. Compare Swing and AWT elements.....	6
The top-level containers in Swing and AWT.....	6
The class name of components in AWT and corresponding class's name in Swing	6
2. Organizing Swing components with Layout Managers.....	6
2.1. Create class NumberGrid.....	7
2.2. Adding buttons.....	7
2.2.3. Complete inner class ButtonListener.....	8
3. Create a graphical user interface for AIMS with Swing.....	9
3.1. View Store Screen.....	9
3.1.1. Create the StoreScreen class.....	9
3.1.2. The NORTH component.....	10
3.1.3. The CENTER component.....	12
3.1.4. The MediaStore class.....	13
3.1.5. Putting it all together.....	14
3.2. Adding more user interaction.....	15
4. JavaFX API.....	16
4.1. Create the FXML file.....	16
4.2. Create the controller class.....	16
4.3. Create the application.....	16
4.4. Practice exercise: Add the Eraser feature.....	16
5. Setting up the View Cart Screen with ScreenBuilder.....	17
6. Integrating JavaFX into Swing application – The JFXPanel class.....	17
7. View the items in cart – JavaFX's data-driven UI.....	19
8. Updating buttons based on selected item in TableView – ChangeListener.....	21
9. Deleting a media.....	21

10. Filter items in cart – FilteredList (optional).....	22
11. Complete the Aims GUI application.....	23
CartScreen.....	23
StoreScreen.....	25
12. Check all the previous source codes to catch/handle/delegate runtime exceptions.....	27
13. Create a class which inherits from Exception.....	27
13.1. Create new class named PlayerException.....	27
13.2. Raise the PlayerException in the play() method.....	27
13.3. Update play() in the Playable interface.....	27
13.4. Update play() in CompactDisc.....	28
14. Update the Aims class.....	29
15. Modify the equals() method of Media class.....	29
16. Reading Document.....	30
17. Update Aims class diagram.....	30

1. Swing components

1.1. AWTAccumulator

```
AWTAccumulator

package hust.soict.hedspi.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;

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 Interger: "));
        tfInput = new TextField(10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());
        add(new Label("The Accumulated Sum is: "));
        tfOutput = new TextField(10);
        tfOutput.setEditable(false);
        add(tfOutput);
        setTitle("AWT Accumulator");
        setSize(350,120);
        setVisible(true);
    }

    public static void main (String arg[]) {
        new AWTAccumulator();
    }

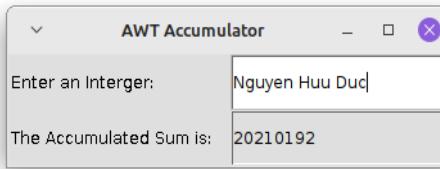
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum+= numberIn;
            tfInput.setText("");
            tfOutput.setText(sum + "");
    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

AWTAccumulator.java

```

1/* Student: Nguyen Huu Duc - 20210192
2 * Class: 732870
3 */
4 package hust.soict.hedspi.swing;
5 import java.awt.Frame;
6 import java.awt.GridLayout;
7 import java.awt.Label;
8 import java.awt.TextField;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11
12 public class AWTAccumulator extends Frame {
13     private TextField tfInput;
14     private TextField tfOutput;
15     private int sum = 0;
16
17     public AWTAccumulator() {
18         setLayout(new GridLayout(2,2));
19
20         add(new Label("Enter an Integer: "));
21
22         tfInput = new TextField(10);
23         add(tfInput);
24         tfInput.addActionListener(new TFInputListener());
25
26         add(new Label("The Accumulated Sum is: "));
27
28         tfOutput = new TextField(10);

```



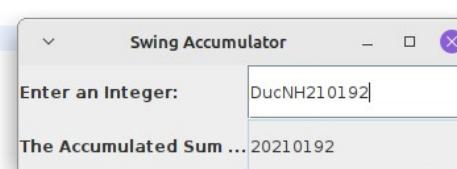
1.2. SwingAccumulator

SwingAccumulator.java

```

1/* Student: Nguyen Huu Duc - 20210192
2 * Class: 732870
3 */
4 package hust.soict.hedspi.swing;
5 import java.awt.Container;
12
13 public class SwingAccumulator extends JFrame {
14     private JTextField tfInput;
15     private JTextField tfOutput;
16     private int sum = 0;
17
18     public SwingAccumulator() {
19         Container cp = getContentPane();
20         cp.setLayout(new GridLayout(2,2));
21
22         cp.add(new JLabel("Enter an Integer: "));
23
24         tfInput = new JTextField(10);
25         cp.add(tfInput);
26         tfInput.addActionListener(new TFInputListener());
27
28         cp.add(new JLabel("The Accumulated Sum is: "));
29
30         tfOutput = new JTextField(10);
31         tfOutput.setEditable(false);
32         cp.add(tfOutput);
33
34         setTitle("Swing Accumulator");

```



The screenshot shows a Java Swing application window titled "SwingAccumulator". The code within the window is as follows:

```
package hust.soict.hedspi.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(new TFInputListener());
        cp.add(new JLabel("The Accumulated Sum is: "));
        tfOutput = new JTextField(10);
        tfOutput.setEditable(false);
        cp.add(tfOutput);
        setTitle("Swing Accumulator");
        setSize(350, 120);
        setVisible(true);
    }
    public static void main(String arg[]) {
        new SwingAccumulator();
    }

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

1.3. Compare Swing and AWT elements

Programming with AWT and Swing is quite similar (similar elements including container/components, event-handling).

However, there are some differences that need to note:

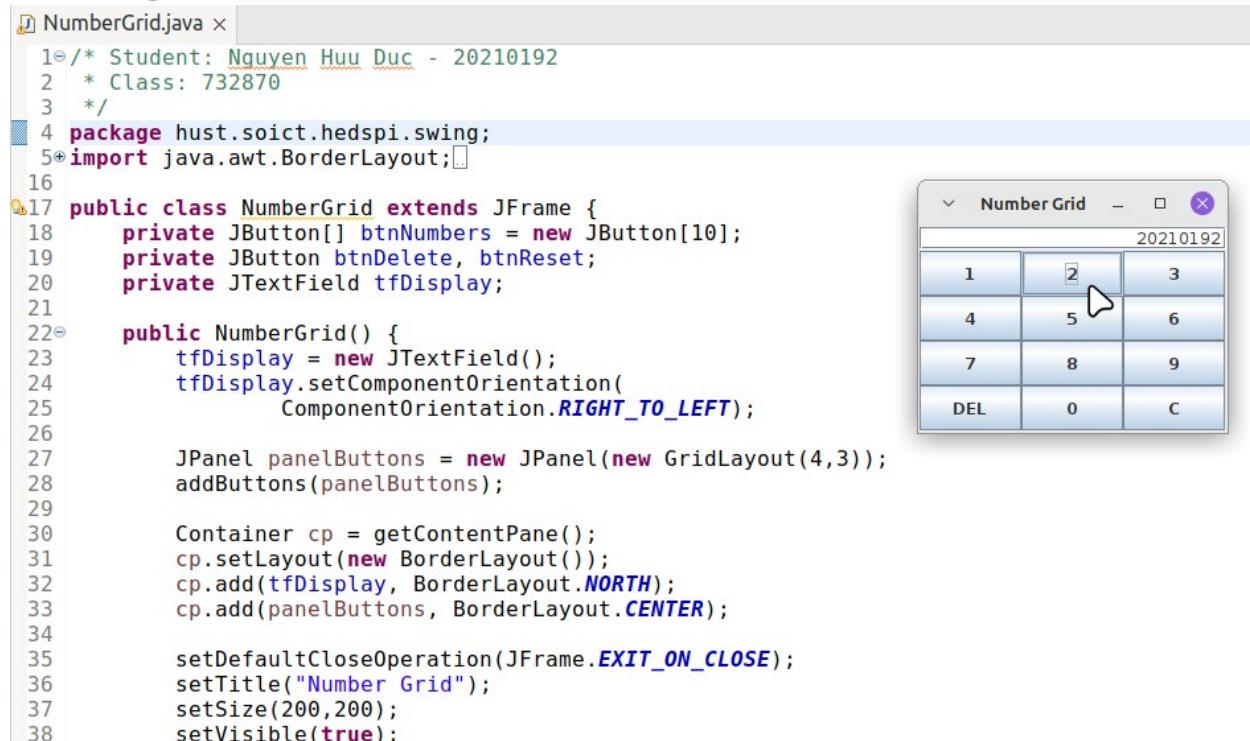
The top-level containers in Swing and AWT

- AWT: The top-level container in AWT is called "Frame".
- Swing: The top-level container in Swing is called "JFrame". It is an extension of Frame and provides additional features and functionality.

The class name of components in AWT and corresponding class's name in Swing

- AWT: The top-level container in AWT is called "Frame".
- Swing: The top-level container in Swing is called "JFrame". It is an extension of Frame and provides additional features and functionality.

2. Organizing Swing components with Layout Managers



The image shows a Java code editor and a running application window. The code editor displays `NumberGrid.java` with the following content:

```

1  /* Student: Nguyen Huu Duc - 20210192
2  * Class: 732870
3  */
4 package hust.soict.hedspi.swing;
5 import java.awt.BorderLayout;
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        tfDisplay = new JTextField();
14        tfDisplay.setComponentOrientation(
15            ComponentOrientation.RIGHT_TO_LEFT);
16
17        JPanel panelButtons = new JPanel(new GridLayout(4,3));
18        addButtons(panelButtons);
19
20        Container cp = getContentPane();
21        cp.setLayout(new BorderLayout());
22        cp.add(tfDisplay, BorderLayout.NORTH);
23        cp.add(panelButtons, BorderLayout.CENTER);
24
25        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26        setTitle("Number Grid");
27        setSize(200,200);
28        setVisible(true);
29
30    }
31
32    void addButtons(JPanel panel) {
33        panel.setLayout(new GridLayout(4,3));
34        panel.add(btnNumbers[0], 0, 0);
35        panel.add(btnNumbers[1], 0, 1);
36        panel.add(btnNumbers[2], 0, 2);
37        panel.add(btnNumbers[3], 1, 0);
38        panel.add(btnNumbers[4], 1, 1);
39        panel.add(btnNumbers[5], 1, 2);
40        panel.add(btnNumbers[6], 2, 0);
41        panel.add(btnNumbers[7], 2, 1);
42        panel.add(btnNumbers[8], 2, 2);
43        panel.add(btnNumbers[9], 3, 0);
44        panel.add(btnDelete, 3, 1);
45        panel.add(btnReset, 3, 2);
46    }
47
48    void setNumber(int value) {
49        tfDisplay.setText(Integer.toString(value));
50    }
51
52    void clear() {
53        tfDisplay.setText("");
54    }
55
56    void delete() {
57        tfDisplay.setText(tfDisplay.getText().substring(0,
58            tfDisplay.getText().length() - 1));
59    }
60
61    void reset() {
62        tfDisplay.setText("0");
63    }
64
65    void addNumber(int value) {
66        tfDisplay.setText(tfDisplay.getText() + Integer.toString(value));
67    }
68
69    void setDelete() {
70        tfDisplay.setEditable(false);
71    }
72
73    void setReset() {
74        tfDisplay.setEditable(true);
75    }
76
77    void setDisplay() {
78        tfDisplay.setEditable(false);
79    }
80
81    void setDisplay() {
82        tfDisplay.setEditable(true);
83    }
84
85    void setDisplay() {
86        tfDisplay.setEditable(true);
87    }
88
89    void setDisplay() {
90        tfDisplay.setEditable(true);
91    }
92
93    void setDisplay() {
94        tfDisplay.setEditable(true);
95    }
96
97    void setDisplay() {
98        tfDisplay.setEditable(true);
99    }
100   void setDisplay() {
101      tfDisplay.setEditable(true);
102  }
103
104  void setDisplay() {
105      tfDisplay.setEditable(true);
106  }
107
108  void setDisplay() {
109      tfDisplay.setEditable(true);
110  }
111
112  void setDisplay() {
113      tfDisplay.setEditable(true);
114  }
115
116  void setDisplay() {
117      tfDisplay.setEditable(true);
118  }
119
120  void setDisplay() {
121      tfDisplay.setEditable(true);
122  }
123
124  void setDisplay() {
125      tfDisplay.setEditable(true);
126  }
127
128  void setDisplay() {
129      tfDisplay.setEditable(true);
130  }
131
132  void setDisplay() {
133      tfDisplay.setEditable(true);
134  }
135
136  void setDisplay() {
137      tfDisplay.setEditable(true);
138  }
139
140  void setDisplay() {
141      tfDisplay.setEditable(true);
142  }
143
144  void setDisplay() {
145      tfDisplay.setEditable(true);
146  }
147
148  void setDisplay() {
149      tfDisplay.setEditable(true);
150  }
151
152  void setDisplay() {
153      tfDisplay.setEditable(true);
154  }
155
156  void setDisplay() {
157      tfDisplay.setEditable(true);
158  }
159
160  void setDisplay() {
161      tfDisplay.setEditable(true);
162  }
163
164  void setDisplay() {
165      tfDisplay.setEditable(true);
166  }
167
168  void setDisplay() {
169      tfDisplay.setEditable(true);
170  }
171
172  void setDisplay() {
173      tfDisplay.setEditable(true);
174  }
175
176  void setDisplay() {
177      tfDisplay.setEditable(true);
178  }
179
180  void setDisplay() {
181      tfDisplay.setEditable(true);
182  }
183
184  void setDisplay() {
185      tfDisplay.setEditable(true);
186  }
187
188  void setDisplay() {
189      tfDisplay.setEditable(true);
190  }
191
192  void setDisplay() {
193      tfDisplay.setEditable(true);
194  }
195
196  void setDisplay() {
197      tfDisplay.setEditable(true);
198  }
199
200  void setDisplay() {
201      tfDisplay.setEditable(true);
202  }
203
204  void setDisplay() {
205      tfDisplay.setEditable(true);
206  }
207
208  void setDisplay() {
209      tfDisplay.setEditable(true);
210  }
211
212  void setDisplay() {
213      tfDisplay.setEditable(true);
214  }
215
216  void setDisplay() {
217      tfDisplay.setEditable(true);
218  }
219
220  void setDisplay() {
221      tfDisplay.setEditable(true);
222  }
223
224  void setDisplay() {
225      tfDisplay.setEditable(true);
226  }
227
228  void setDisplay() {
229      tfDisplay.setEditable(true);
230  }
231
232  void setDisplay() {
233      tfDisplay.setEditable(true);
234  }
235
236  void setDisplay() {
237      tfDisplay.setEditable(true);
238  }
239
240  void setDisplay() {
241      tfDisplay.setEditable(true);
242  }
243
244  void setDisplay() {
245      tfDisplay.setEditable(true);
246  }
247
248  void setDisplay() {
249      tfDisplay.setEditable(true);
250  }
251
252  void setDisplay() {
253      tfDisplay.setEditable(true);
254  }
255
256  void setDisplay() {
257      tfDisplay.setEditable(true);
258  }
259
260  void setDisplay() {
261      tfDisplay.setEditable(true);
262  }
263
264  void setDisplay() {
265      tfDisplay.setEditable(true);
266  }
267
268  void setDisplay() {
269      tfDisplay.setEditable(true);
270  }
271
272  void setDisplay() {
273      tfDisplay.setEditable(true);
274  }
275
276  void setDisplay() {
277      tfDisplay.setEditable(true);
278  }
279
280  void setDisplay() {
281      tfDisplay.setEditable(true);
282  }
283
284  void setDisplay() {
285      tfDisplay.setEditable(true);
286  }
287
288  void setDisplay() {
289      tfDisplay.setEditable(true);
290  }
291
292  void setDisplay() {
293      tfDisplay.setEditable(true);
294  }
295
296  void setDisplay() {
297      tfDisplay.setEditable(true);
298  }
299
300  void setDisplay() {
301      tfDisplay.setEditable(true);
302  }
303
304  void setDisplay() {
305      tfDisplay.setEditable(true);
306  }
307
308  void setDisplay() {
309      tfDisplay.setEditable(true);
310  }
311
312  void setDisplay() {
313      tfDisplay.setEditable(true);
314  }
315
316  void setDisplay() {
317      tfDisplay.setEditable(true);
318  }
319
320  void setDisplay() {
321      tfDisplay.setEditable(true);
322  }
323
324  void setDisplay() {
325      tfDisplay.setEditable(true);
326  }
327
328  void setDisplay() {
329      tfDisplay.setEditable(true);
330  }
331
332  void setDisplay() {
333      tfDisplay.setEditable(true);
334  }
335
336  void setDisplay() {
337      tfDisplay.setEditable(true);
338  }
339
340  void setDisplay() {
341      tfDisplay.setEditable(true);
342  }
343
344  void setDisplay() {
345      tfDisplay.setEditable(true);
346  }
347
348  void setDisplay() {
349      tfDisplay.setEditable(true);
350  }
351
352  void setDisplay() {
353      tfDisplay.setEditable(true);
354  }
355
356  void setDisplay() {
357      tfDisplay.setEditable(true);
358  }
359
360  void setDisplay() {
361      tfDisplay.setEditable(true);
362  }
363
364  void setDisplay() {
365      tfDisplay.setEditable(true);
366  }
367
368  void setDisplay() {
369      tfDisplay.setEditable(true);
370  }
371
372  void setDisplay() {
373      tfDisplay.setEditable(true);
374  }
375
376  void setDisplay() {
377      tfDisplay.setEditable(true);
378  }
379
380  void setDisplay() {
381      tfDisplay.setEditable(true);
382  }
383
384  void setDisplay() {
385      tfDisplay.setEditable(true);
386  }
387
388  void setDisplay() {
389      tfDisplay.setEditable(true);
390  }
391
392  void setDisplay() {
393      tfDisplay.setEditable(true);
394  }
395
396  void setDisplay() {
397      tfDisplay.setEditable(true);
398  }
399
400  void setDisplay() {
401      tfDisplay.setEditable(true);
402  }
403
404  void setDisplay() {
405      tfDisplay.setEditable(true);
406  }
407
408  void setDisplay() {
409      tfDisplay.setEditable(true);
410  }
411
412  void setDisplay() {
413      tfDisplay.setEditable(true);
414  }
415
416  void setDisplay() {
417      tfDisplay.setEditable(true);
418  }
419
420  void setDisplay() {
421      tfDisplay.setEditable(true);
422  }
423
424  void setDisplay() {
425      tfDisplay.setEditable(true);
426  }
427
428  void setDisplay() {
429      tfDisplay.setEditable(true);
430  }
431
432  void setDisplay() {
433      tfDisplay.setEditable(true);
434  }
435
436  void setDisplay() {
437      tfDisplay.setEditable(true);
438  }
439
440  void setDisplay() {
441      tfDisplay.setEditable(true);
442  }
443
444  void setDisplay() {
445      tfDisplay.setEditable(true);
446  }
447
448  void setDisplay() {
449      tfDisplay.setEditable(true);
450  }
451
452  void setDisplay() {
453      tfDisplay.setEditable(true);
454  }
455
456  void setDisplay() {
457      tfDisplay.setEditable(true);
458  }
459
460  void setDisplay() {
461      tfDisplay.setEditable(true);
462  }
463
464  void setDisplay() {
465      tfDisplay.setEditable(true);
466  }
467
468  void setDisplay() {
469      tfDisplay.setEditable(true);
470  }
471
472  void setDisplay() {
473      tfDisplay.setEditable(true);
474  }
475
476  void setDisplay() {
477      tfDisplay.setEditable(true);
478  }
479
480  void setDisplay() {
481      tfDisplay.setEditable(true);
482  }
483
484  void setDisplay() {
485      tfDisplay.setEditable(true);
486  }
487
488  void setDisplay() {
489      tfDisplay.setEditable(true);
490  }
491
492  void setDisplay() {
493      tfDisplay.setEditable(true);
494  }
495
496  void setDisplay() {
497      tfDisplay.setEditable(true);
498  }
499
500  void setDisplay() {
501      tfDisplay.setEditable(true);
502  }
503
504  void setDisplay() {
505      tfDisplay.setEditable(true);
506  }
507
508  void setDisplay() {
509      tfDisplay.setEditable(true);
510  }
511
512  void setDisplay() {
513      tfDisplay.setEditable(true);
514  }
515
516  void setDisplay() {
517      tfDisplay.setEditable(true);
518  }
519
520  void setDisplay() {
521      tfDisplay.setEditable(true);
522  }
523
524  void setDisplay() {
525      tfDisplay.setEditable(true);
526  }
527
528  void setDisplay() {
529      tfDisplay.setEditable(true);
530  }
531
532  void setDisplay() {
533      tfDisplay.setEditable(true);
534  }
535
536  void setDisplay() {
537      tfDisplay.setEditable(true);
538  }
539
540  void setDisplay() {
541      tfDisplay.setEditable(true);
542  }
543
544  void setDisplay() {
545      tfDisplay.setEditable(true);
546  }
547
548  void setDisplay() {
549      tfDisplay.setEditable(true);
550  }
551
552  void setDisplay() {
553      tfDisplay.setEditable(true);
554  }
555
556  void setDisplay() {
557      tfDisplay.setEditable(true);
558  }
559
560  void setDisplay() {
561      tfDisplay.setEditable(true);
562  }
563
564  void setDisplay() {
565      tfDisplay.setEditable(true);
566  }
567
568  void setDisplay() {
569      tfDisplay.setEditable(true);
570  }
571
572  void setDisplay() {
573      tfDisplay.setEditable(true);
574  }
575
576  void setDisplay() {
577      tfDisplay.setEditable(true);
578  }
579
580  void setDisplay() {
581      tfDisplay.setEditable(true);
582  }
583
584  void setDisplay() {
585      tfDisplay.setEditable(true);
586  }
587
588  void setDisplay() {
589      tfDisplay.setEditable(true);
590  }
591
592  void setDisplay() {
593      tfDisplay.setEditable(true);
594  }
595
596  void setDisplay() {
597      tfDisplay.setEditable(true);
598  }
599
599 
```

2.1. Create class NumberGrid

```

1/* Student: Nguyen Huu Duc - 20210192
2 * Class: 732870
3 */
4 package hust.soict.hedspi.swing;
5import java.awt.BorderLayout;
16
17 public class NumberGrid extends JFrame {
18     private JButton[] btnNumbers = new JButton[10];
19     private JButton btnDelete, btnReset;
20     private JTextField tfDisplay;
21
22     public NumberGrid() {
23         tfDisplay = new JTextField();
24         tfDisplay.setComponentOrientation(
25             ComponentOrientation.RIGHT_TO_LEFT);
26
27         JPanel panelButtons = new JPanel(new GridLayout(4,3));
28         addButtons(panelButtons);
29
30         Container cp = getContentPane();
31         cp.setLayout(new BorderLayout());
32         cp.add(tfDisplay, BorderLayout.NORTH);
33         cp.add(panelButtons, BorderLayout.CENTER);
34
35         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36         setTitle("Number Grid");
37         setSize(200,200);
38         setVisible(true);

```

2.2. Adding buttons

```

41     void addButtons(JPanel panelButtons) {
42         ButtonListener btnListener = new ButtonListener();
43         for(int i = 1; i<=9;i++) {
44             btnNumbers[i]= new JButton(""+i);
45             panelButtons.add(btnNumbers[i]);
46             btnNumbers[i].addActionListener(btnListener);
47         }
48         btnDelete = new JButton("DEL");
49         panelButtons.add(btnDelete);
50         btnDelete.addActionListener(btnListener);
51
52         btnNumbers[0] = new JButton("0");
53         panelButtons.add(btnNumbers[0]);
54         btnNumbers[0].addActionListener(btnListener);
55
56         btnReset = new JButton("C");
57         panelButtons.add(btnReset);
58         btnReset.addActionListener(btnListener);
59     }
60

```

2.2.3. Complete inner class ButtonListener

```
61@ private class ButtonListener implements ActionListener {  
62@     @Override  
63@     public void actionPerformed(ActionEvent e) {  
64@         String button = e.getActionCommand();  
65@         if(button.charAt(0)>= '0' && button.charAt(0)<= '9') {  
66@             tfDisplay.setText(tfDisplay.getText()+button);  
67@         }  
68@         else if (button.equals("DEL")) {  
69@             //handles the "DEL" case  
70@             String text = tfDisplay.getText();  
71@             if (text.length() > 0) {  
72@                 tfDisplay.setText(text.substring(0, text.length() - 1));  
73@             }  
74@         }  
75@         else {  
76@             //handles the "C" case  
77@             tfDisplay.setText("");  
78@         }  
79@     }  
80@ }  
81@  
82@ public static void main(String arg[]) {  
83@     new NumberGrid();  
84@ }  
85@ }
```

3. Create a graphical user interface for AIMS with Swing

3.1. View Store Screen



3.1.1. Create the StoreScreen class

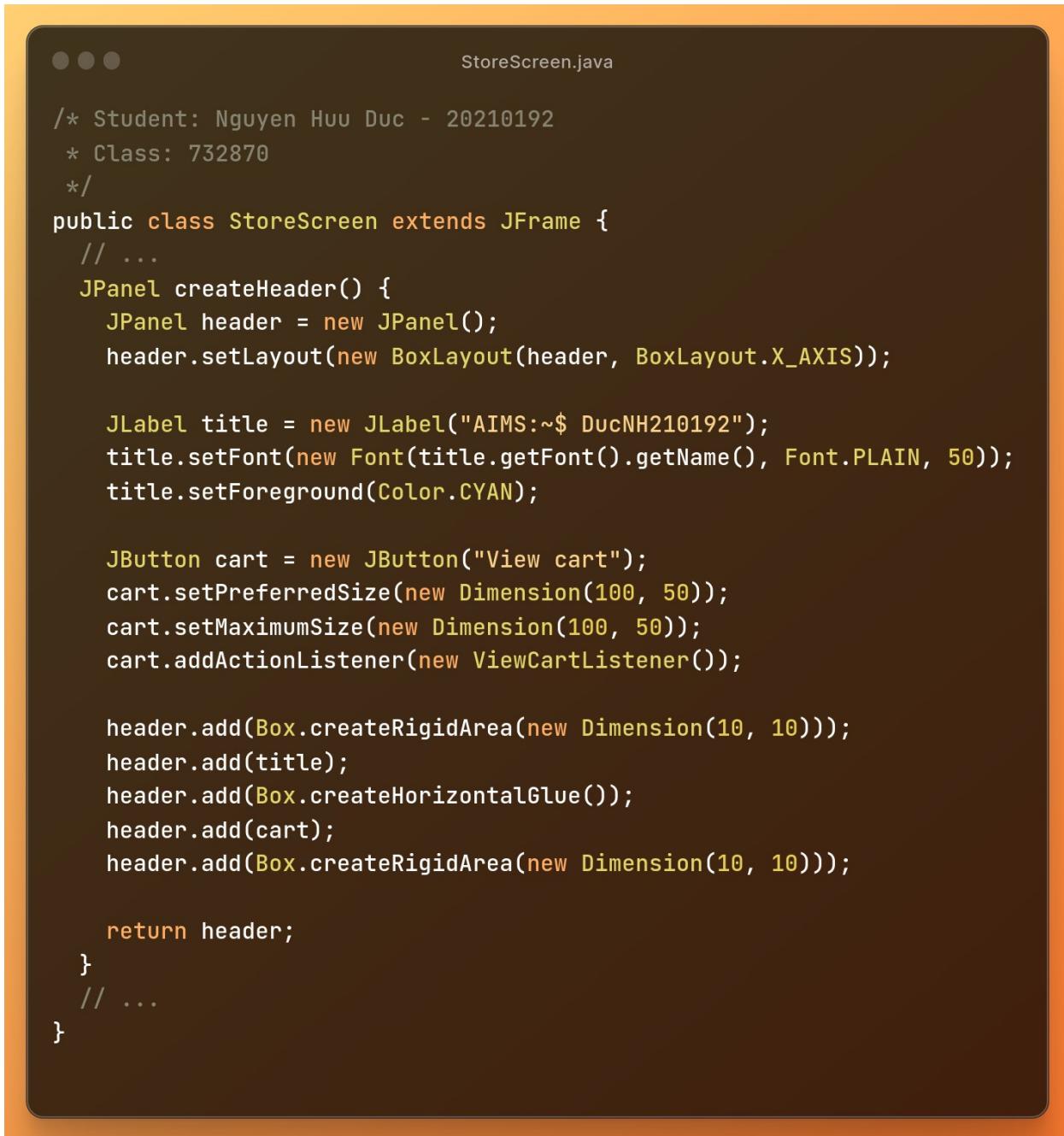
```
StoreScreen.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class StoreScreen extends JFrame {
    private Store store;
    private Cart cart;
```

3.1.2. The NORTH component

```
StoreScreen.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class StoreScreen extends JFrame {
    // ...
    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("Options");
        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBook = new JMenuItem("Add Book");
        addBook.addActionListener(new AddBookListener());
        smUpdateStore.add(addBook);
        JMenuItem addCD = new JMenuItem("Add CD");
        addCD.addActionListener(new AddCDListener());
        smUpdateStore.add(addCD);
        JMenuItem addDVD = new JMenuItem("Add DVD");
        addDVD.addActionListener(new AddDVDListener());
        smUpdateStore.add(addDVD);
        menu.add(smUpdateStore);
        menu.add(new JMenuItem("View store"));
        JMenuItem cart = new JMenuItem("View cart");
        cart.addActionListener(new ViewCartListener());
        menu.add(cart);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);
        return menuBar;
    }
    // ...
}
```



The screenshot shows a Java code editor window with a dark theme. The title bar reads "StoreScreen.java". The code itself is as follows:

```
/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class StoreScreen extends JFrame {
    // ...
    JPanel createHeader() {
        JPanel header = new JPanel();
        header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

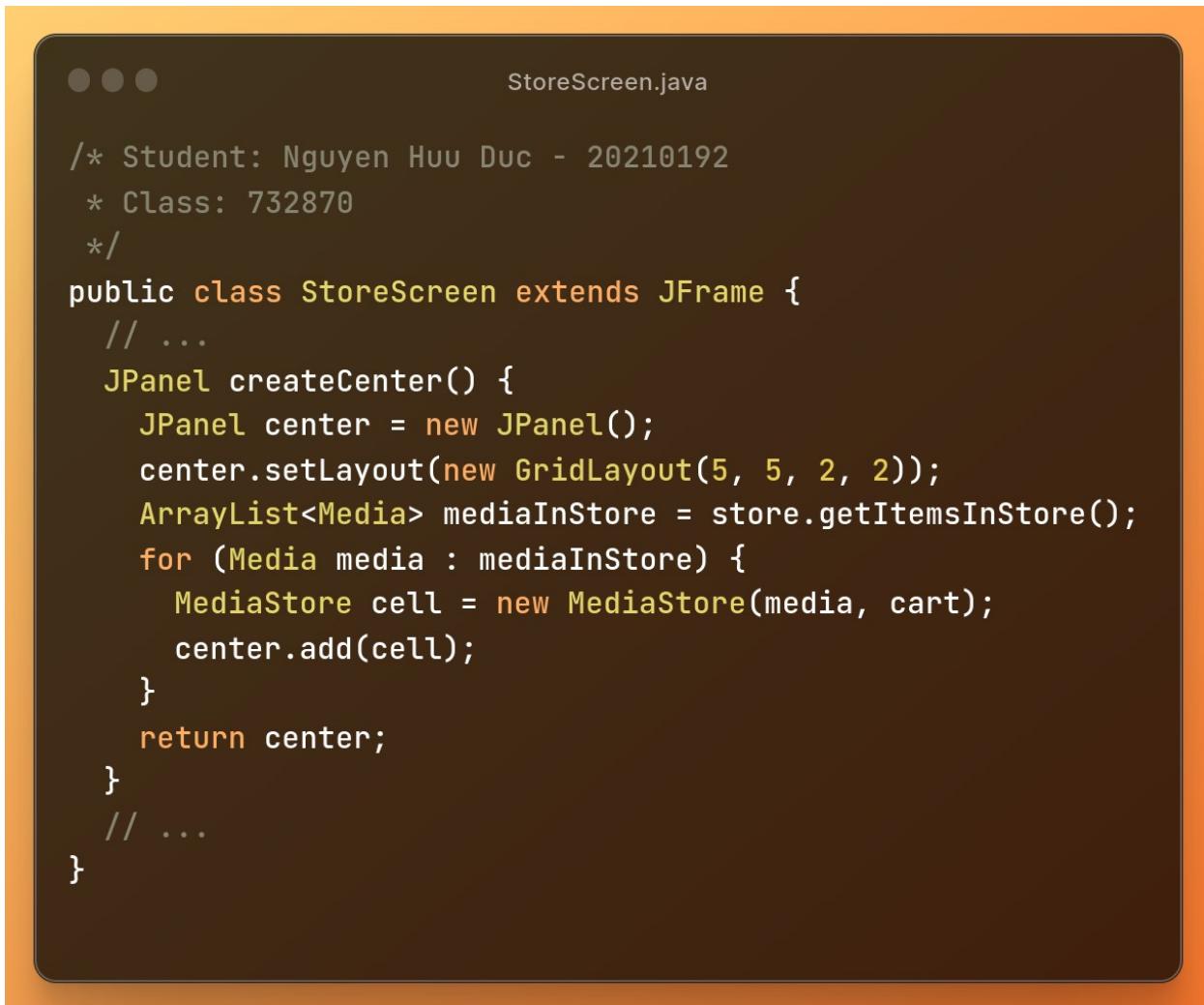
        JLabel title = new JLabel("AIMS:~$ DucNH210192");
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
        title.setForeground(Color.CYAN);

        JButton cart = new JButton("View cart");
        cart.setPreferredSize(new Dimension(100, 50));
        cart.setMaximumSize(new Dimension(100, 50));
        cart.addActionListener(new ViewCartListener());

        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



The screenshot shows a Java code editor window with a dark theme. The title bar says "StoreScreen.java". The code is as follows:

```
/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class StoreScreen extends JFrame {
    // ...
    JPanel createCenter() {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(5, 5, 2, 2));
        ArrayList<Media> mediaInStore = store.getItemsInStore();
        for (Media media : mediaInStore) {
            MediaStore cell = new MediaStore(media, cart);
            center.add(cell);
        }
        return center;
    }
    // ...
}
```

3.1.4. The MediaStore class

```
MediaStore.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class MediaStore extends JPanel {
    private Media media;
    private Cart cart;
    public MediaStore(Media media, Cart cart) {
        this.media = media;
        this.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, 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("Add to cart");
        addToCartButton.addActionListener(new AddToCartListener());
        container.add(addToCartButton);
        JButton detailsButton = new JButton("View details");
        detailsButton.addActionListener(new DetailsListener());
        container.add(detailsButton);
        if (media instanceof Playable) {
            JButton playButton = new JButton("Play");
            playButton.addActionListener(new PlayButtonListener());
            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));
    }
}
```

3.1.5. Putting it all together



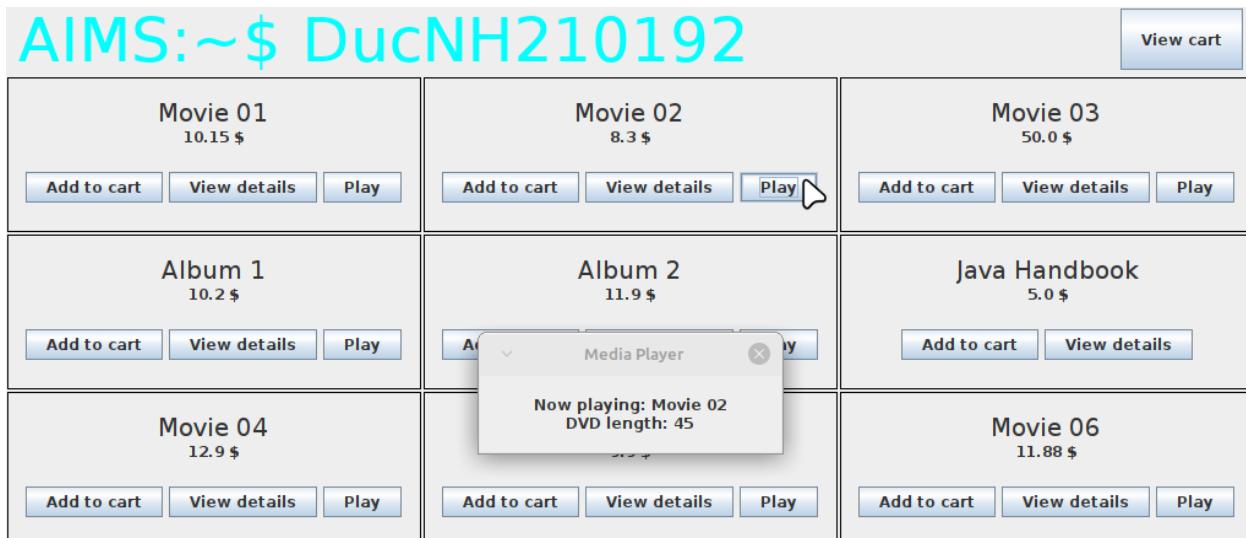
The screenshot shows a Java code editor window with a dark theme. The title bar says "StoreScreen.java". The code is as follows:

```
/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class StoreScreen extends JFrame {
    private Store store;
    private Cart cart;

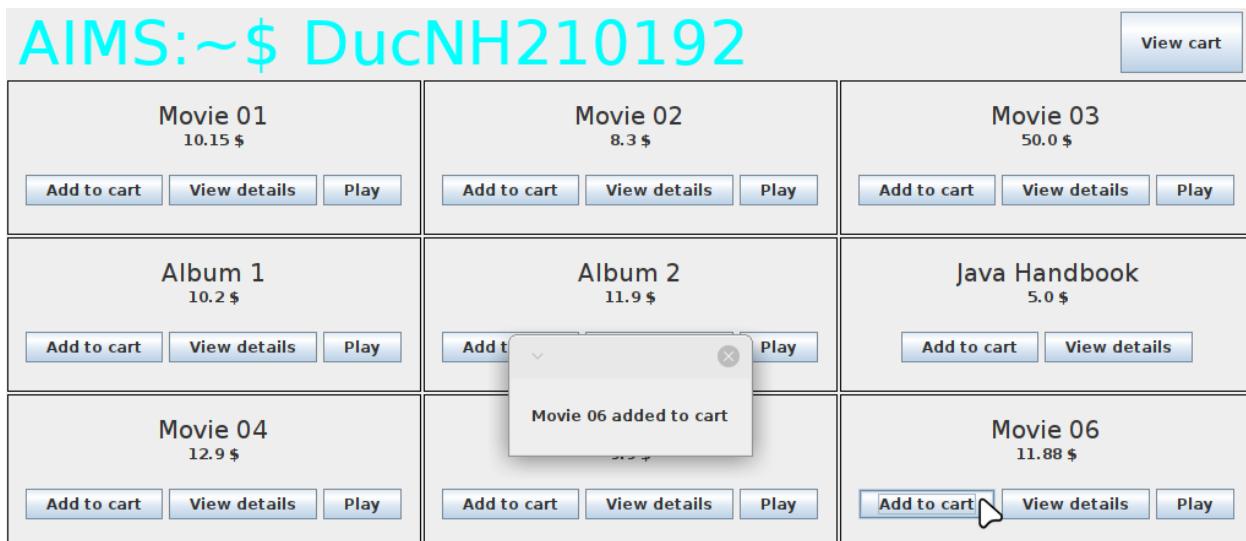
    public StoreScreen(Store store, Cart cart) {
        this.store = store;
        this.cart = cart;
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(), BorderLayout.CENTER);
        setVisible(true);
        setTitle("Store");
        setSize(1024, 768);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int w = getSize().width;
        int h = getSize().height;
        int x = (dim.width - w) / 2;
        int y = (dim.height - h) / 2;
        setLocation(x, y);
    }
    // ...
}
```

3.2. Adding more user interaction

When user clicks on the “Play” button, the Media should be played in a dialog window.



When user clicks on the “Add to cart” button, the Media should be added to the cart.



4. JavaFX API

4.1. Create the FXML file

4.2. Create the controller class

```
PainterController.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class PainterController {
    private Pane drawingAreaPane;
    @FXML
    void clearButtonPressed(ActionEvent event) {
        drawingAreaPane.getChildren().clear();
    }
    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
        drawingAreaPane.getChildren().add(newCircle);
    }
}
```

4.3. Create the application

```
Painter.java

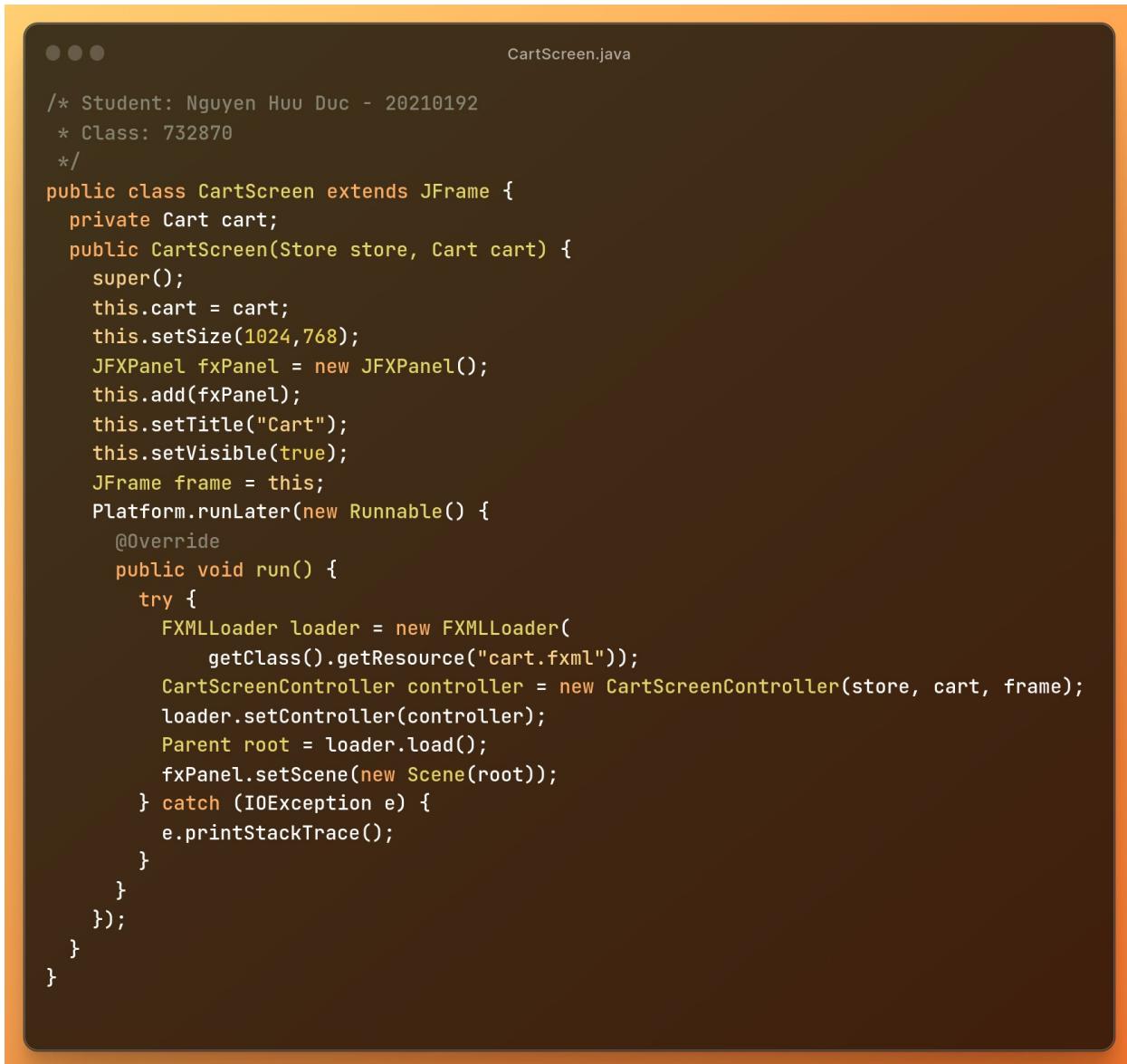
/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class Painter extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().
            getResource("/hust/soict/hedspi/javafx/Painter.fxml"));
        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

4.4. Practice exercise: Add the Eraser feature



5. Setting up the View Cart Screen with ScreenBuilder

6. Integrating JavaFX into Swing application - The JFXPanel class



```
CartScreen.java

/*
 * Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class CartScreen extends JFrame {
    private Cart cart;
    public CartScreen(Store store, Cart cart) {
        super();
        this.cart = cart;
        this.setSize(1024,768);
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        this.setVisible(true);
        JFrame frame = this;
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(
                        getClass().getResource("cart.fxml"));
                    CartScreenController controller = new CartScreenController(store, cart, frame);
                    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

```
CartScreen.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class CartScreenController {
    private Cart cart;
    @FXML
    private TableView<Media> tblMedia;
    @FXML
    private TableColumn<Media, String> colMediaTitle;
    @FXML
    private TableColumn<Media, String> colMediaCategory;
    @FXML
    private TableColumn<Media, String> colMediaCost;

    public CartScreenController(Store store, Cart cart, JFrame stage) {
        super();
        this.cart = cart;
    }
    @FXML
    public void initialize() {
        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());
    }
}
```

```
Cart.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class Cart {
    private static final int MAX_NUMBERS_ORDERED = 20;
    private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();
    // ...
}
```

The screenshot shows a software application window titled "Cart". At the top, there is a toolbar with a dropdown menu, a "Cart" button, and standard window control buttons (minimize, maximize, close). Below the toolbar is a section labeled "Options" and a "Filter:" input field with two radio buttons: "By ID" (selected) and "By Title". The main area displays a table of items:

Title	Category	Cost	Total
The Goblin King	Animation	19.95	
Star Wars	Sci-fi	24.95	
Aladin	Animation	18.99	
ROTK OST (Part 1)	Drama	30.95	
ROTK OST (Part 2)	Drama	25.99	

To the right of the table, a red button labeled "Place Order" is visible. At the bottom of the window, there is a "Sort" button and three radio buttons for sorting: "By Title" (selected), "By Cost", and "By Date".

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

```
CartScreenController.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
private void updateButtonBar(Media media) {
    if (media == null) {
        btnRemove.setVisible(false);
        btnDetails.setVisible(false);
        btnPlay.setVisible(false);
    } else {
        btnRemove.setVisible(true);
        btnDetails.setVisible(true);
        if (media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
    }
}
```

9. Deleting a media

```
CartScreenController.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
private void removeButtonPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
}
```

10. Filter items in cart - FilteredList (optional)

```
CartScreenController.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
@FXML
public void initialize() {
    filteredCart = new FilteredList<Media>(this.cart.getItemsOrdered(), s -> true);
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, String>("cost"));
    tblMedia.setItems(filteredCart);
    btnPlay.setVisible(false);
    btnRemove.setVisible(false);
    btnDetails.setVisible(false);
    costLabel.setText(String.valueOf(this.cart.totalCost()));
    tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>()
    () {
        @Override
        public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
            updateButtonBar(newValue);
        }
    });
    tfFilter.textProperty().addListener(new ChangeListener<String> () {
        @Override
        public void changed(ObservableValue<? extends String> observable, String oldValue,
String newValue) {
            showFilteredMedia(newValue);
        }
    });
}
```

```
CartScreenController.java

/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
private void showFilteredMedia(String filter) {
    if (filter == null || filter.length() == 0) filteredCart.setPredicate(s -> true);
    else if (filterByID) filteredCart.setPredicate(s -> s.getID() == Integer.parseInt(filter));
    else filteredCart.setPredicate(s -> s.getTitle().toLowerCase().contains(filter));
}
```

11. Complete the Aims GUI application

Complete the remaining UIs of Aims: CartScreen, StoreScreen.

CartScreen

The screenshot displays two windows of the Aims GUI application, specifically the CartScreen. Both windows show a table of items in a shopping cart:

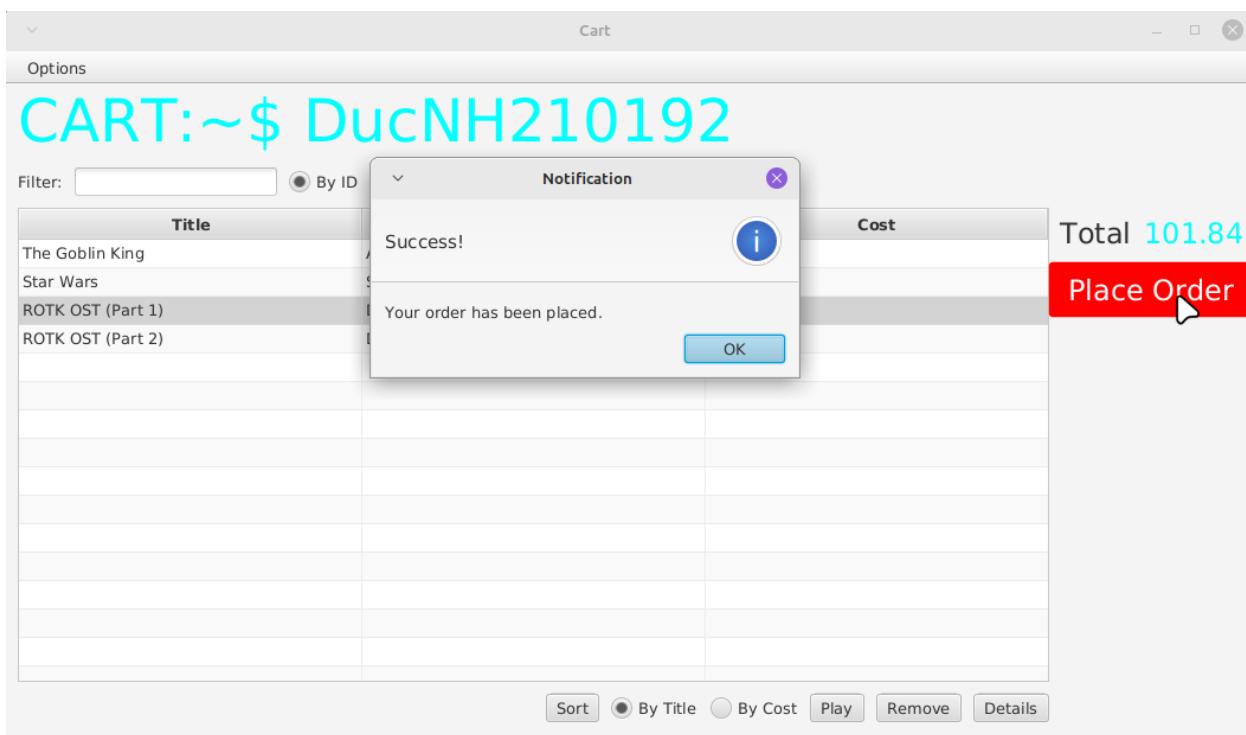
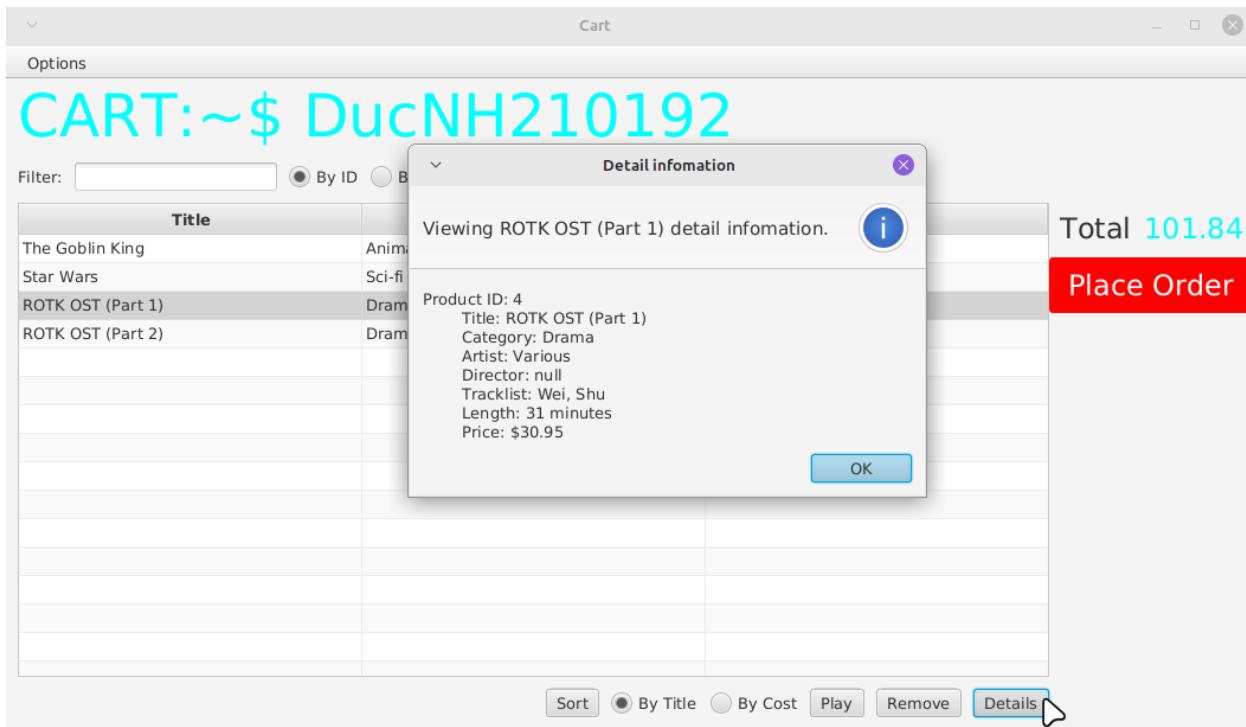
Title	Category	Cost	Total
The Goblin King	Animation	19.95	Total 120.83
Star Wars	Sci-fi	24.95	
Aladdin	Animation	18.99	
ROTK OST (Part 1)	Drama	30.95	
ROTK OST (Part 2)	Drama	25.99	

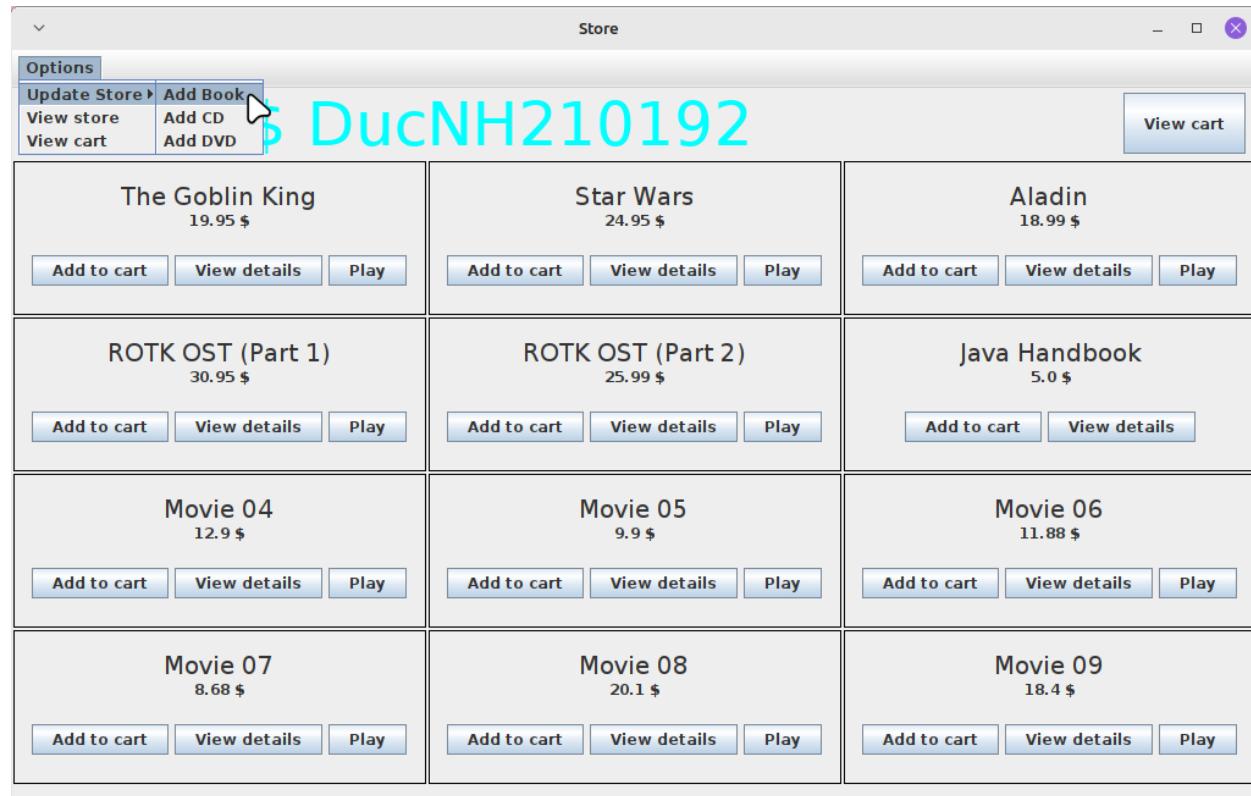
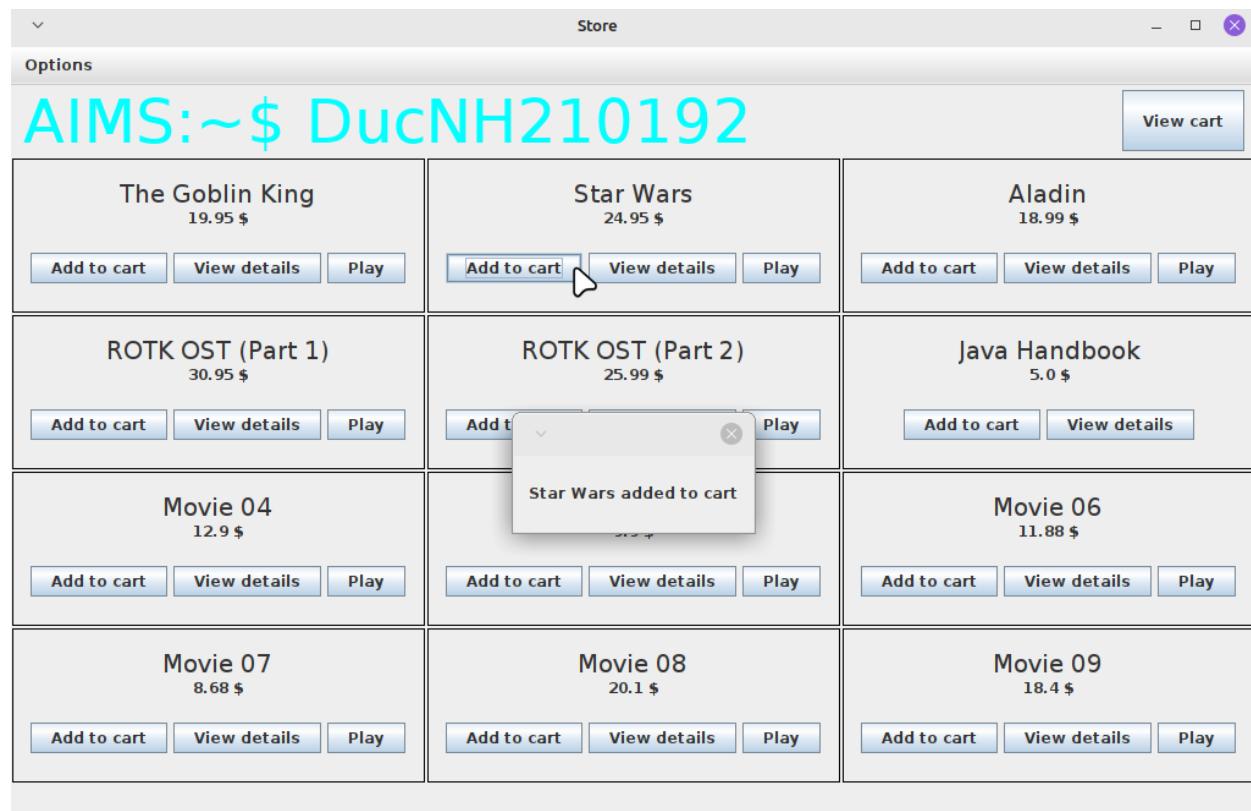
Top Window (Active):

- Contains a "Media Player" dialog box with the message: "Now playing: Shu Track length: 21".
- Buttons at the bottom: Sort, By Title (selected), By Cost, Play (highlighted), Remove, Details.

Bottom Window (Inactive):

- Buttons at the bottom: Sort, By Title (selected), By Cost, Play, Remove (cursor hovering), Details.



StoreScreen

Add book to store

Options

ADD BOOK TO STORE

Title: Think Like A Champion

Category: Business

Authors: Donald Trump

Length (pa...: 220

Cost (\$): 6.95

Notification

Success

Think Like A Champion has been added to the store

OK

Add

Options

AIMS:~\$ DucNH210192

View cart

The Goblin King 19.95 \$ Add to cart View details Play	Star Wars 24.95 \$ Add to cart View details Play	Aladin 18.99 \$ Add to cart View details Play
ROTK OST (Part 1) 30.95 \$ Add to cart View details Play	ROTK OST (Part 2) 25.99 \$ Add to cart View details Play	Java Handbook 5.0 \$ Add to cart View details
Movie 04 12.9 \$ Add to cart View details Play	Movie 05 9.9 \$ Add to cart View details Play	Movie 06 11.88 \$ Add to cart View details Play
Movie 07 8.68 \$ Add to cart View details Play	Movie 08 20.1 \$ Add to cart View details Play	Movie 09 18.4 \$ Add to cart View details Play
Think Like A Champion 6.95 \$ Add to cart View details		

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

Review all methods, classes in AimsProject, catch/handle or delegate all exceptions if necessary. The exception delegation mechanism is especially helpful for constructors so that no object is created if there is any violation of the requirement/constraints.

13. Create a class which inherits from Exception

13.1. Create new class named PlayerException

13.2. Raise the PlayerException in the play() method

Update play() method in DigitalVideoDisc and Track.

```

DigitalVideoDisc.java x Track.java

31  public void play() throws PlayerException {
32      if (this.getLength() <= 0) {
33          throw new PlayerException("ERROR: DVD length is non-positive!");
34      } else {
35          System.out.println("Playing DVD: " + this.getTitle());
36          System.out.println("DVD length: " + this.getLength());
37
38      Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

```



```

DigitalVideoDisc.java x Track.java x

32  public void play() throws PlayerException {
33      if (this.getLength() <= 0) {
34          throw new PlayerException("ERROR: Track length is non-positive!");
35      } else {
36          System.out.println("Playing track: " + this.getTitle());
37          System.out.println("Track length: " + this.getLength());
38

```

13.3. Update play() in the Playable interface

```

Playable.java x CompactDisc.java

1 /* Student: Nguyen Huu Duc - 20210192
2 * Class: 732870
3 */
4 package hust.soict.hedspi.aims.media;
5
6 import hust.soict.hedspi.aims.exception.*;
7 public interface Playable {
8     public void play() throws PlayerException;;
9 }
10

```

13.4. Update play() in CompactDisc

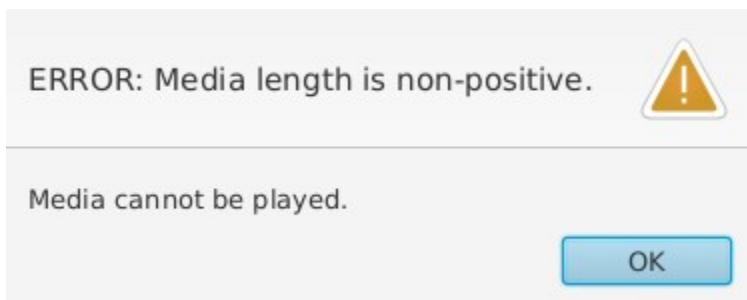
```

  J Playable.java  J CompactDisc.java x
△ 49⊕  public void play() throws PlayerException {
  50      if (this.getLength() <= 0) {
  51          throw new PlayerException("ERROR: CD length is non-positive!");
  52      } else {
  53          System.out.println("Playing CD: " + this.getTitle());
  54          Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
  55          JPanel p = new JPanel();
  56          JDialog d = new JDialog();
  57          JLabel l1 = new JLabel("Now playing: " + this.getTitle());
  58          p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
  59          l1.setAlignmentX(Component.CENTER_ALIGNMENT);
  60          d.setTitle("Media Player");
  61          p.add(Box.createVerticalGlue());
  62          p.add(l1);
  63          p.add(Box.createVerticalGlue());
  64          d.add(p);
  65          d.setSize(250,100);
  66          int w = d.getSize().width;
  67          int h = d.getSize().height;
  68          int x = (dim.width - w) / 2;
  69          int y = (dim.height - h) / 2;
  70          d.setLocation(x, y);
  71          d.setVisible(true);
  72          for (Track track: this.tracks) {
  73              try {
  74                  track.play();
  75              } catch (PlayerException e) {
  76                  throw e;
  
```

14. Update the Aims class

- The Aims class must be updated to handle any exceptions generated when the play() methods are called. When we don't update for them to catch, the program becomes more susceptible to crashes or unpredictable behavior (Classes using play() won't work).
- Try to use try-catch block when you call the play() method of Media's objects.

The example result for the play() of Media is illustrated in the following figure.



15. Modify the equals() method of Media class

```

  Media.java

  /* Student: Nguyen Huu Duc - 20210192
   * Class: 732870
   */
  public boolean equals(Object media) {
    if (media instanceof Media) {
      try {
        Media that = (Media) media;
        return this.title.toLowerCase().equals(that.getTitle().toLowerCase());
      } catch (NullPointerException e1) {
        return false;
      } catch (ClassCastException e2) {
        return false;
      }
    } else return false;
  }

```

16. Reading Document

17. Update Aims class diagram

Make an exception hierarchical tree for all self-defined exceptions in Aims Project.

