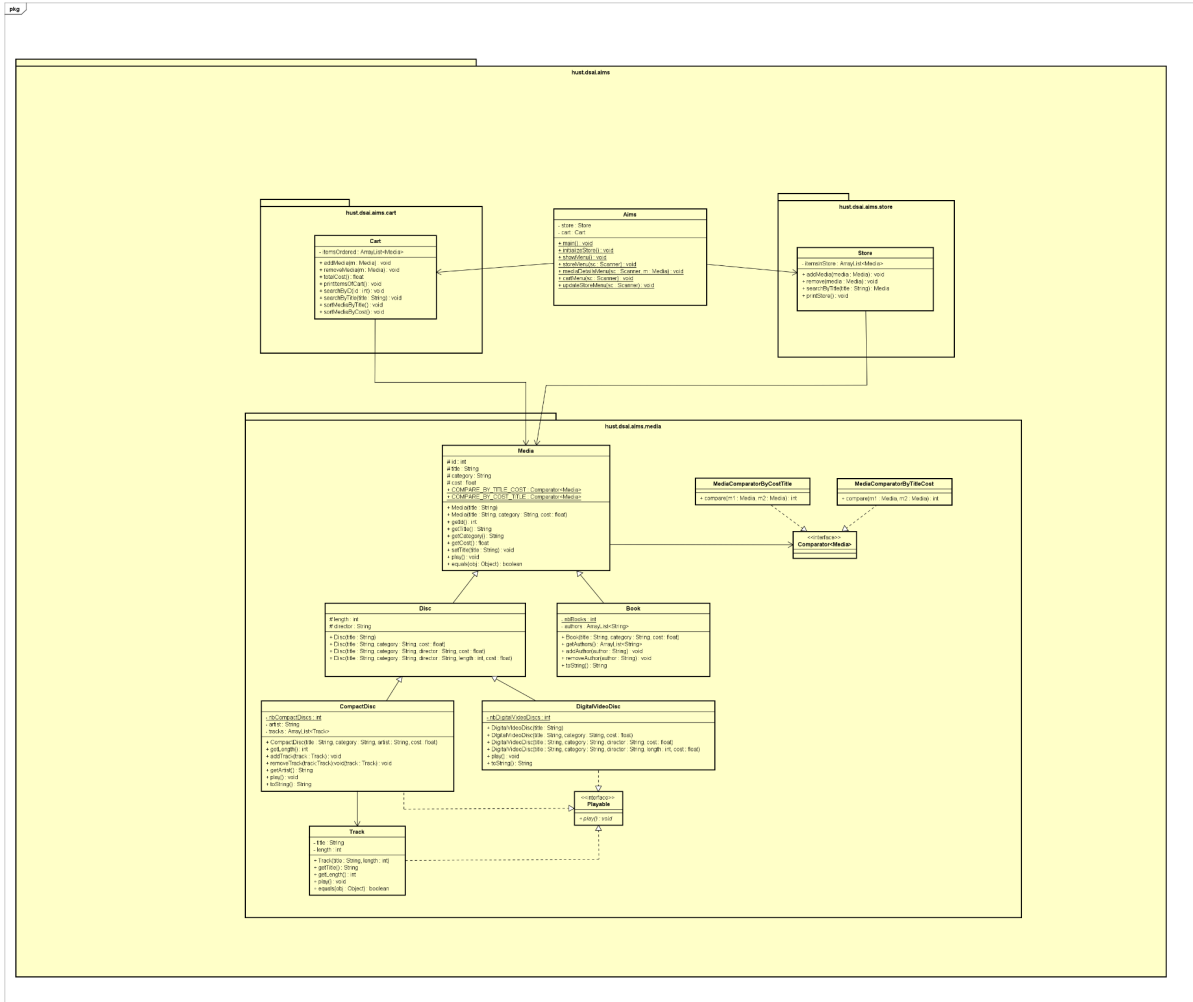


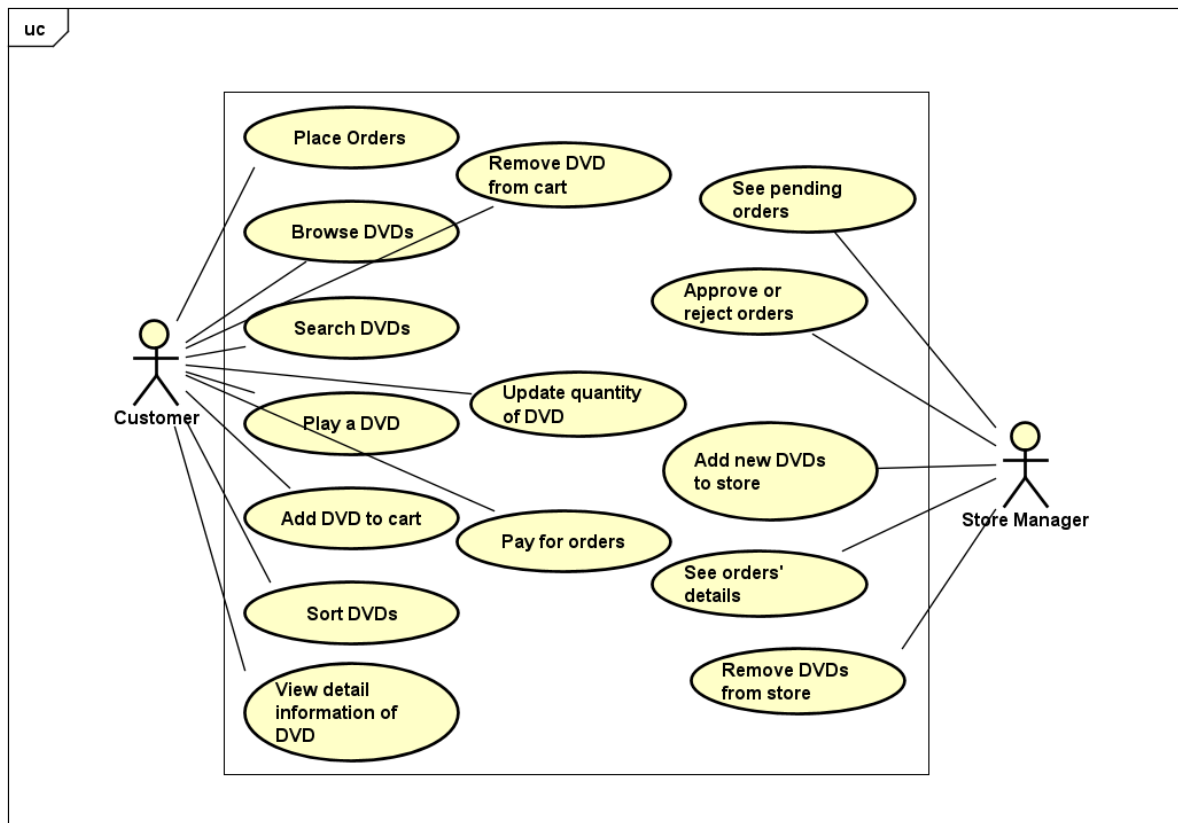
Lab04 Report

1. Updated Diagram.

1.1. Class Diagram.

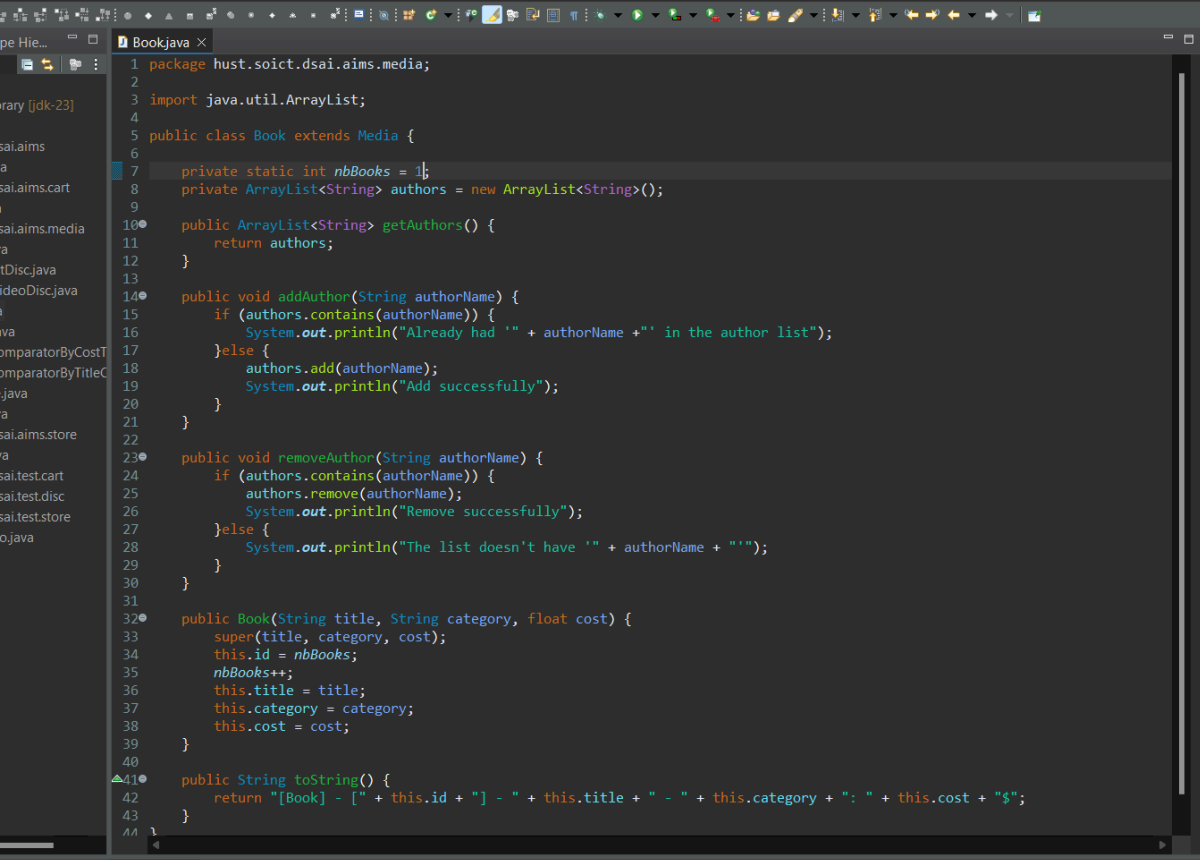


1.2. Use case Diagram.



2. Source code.

2.1. Book class.



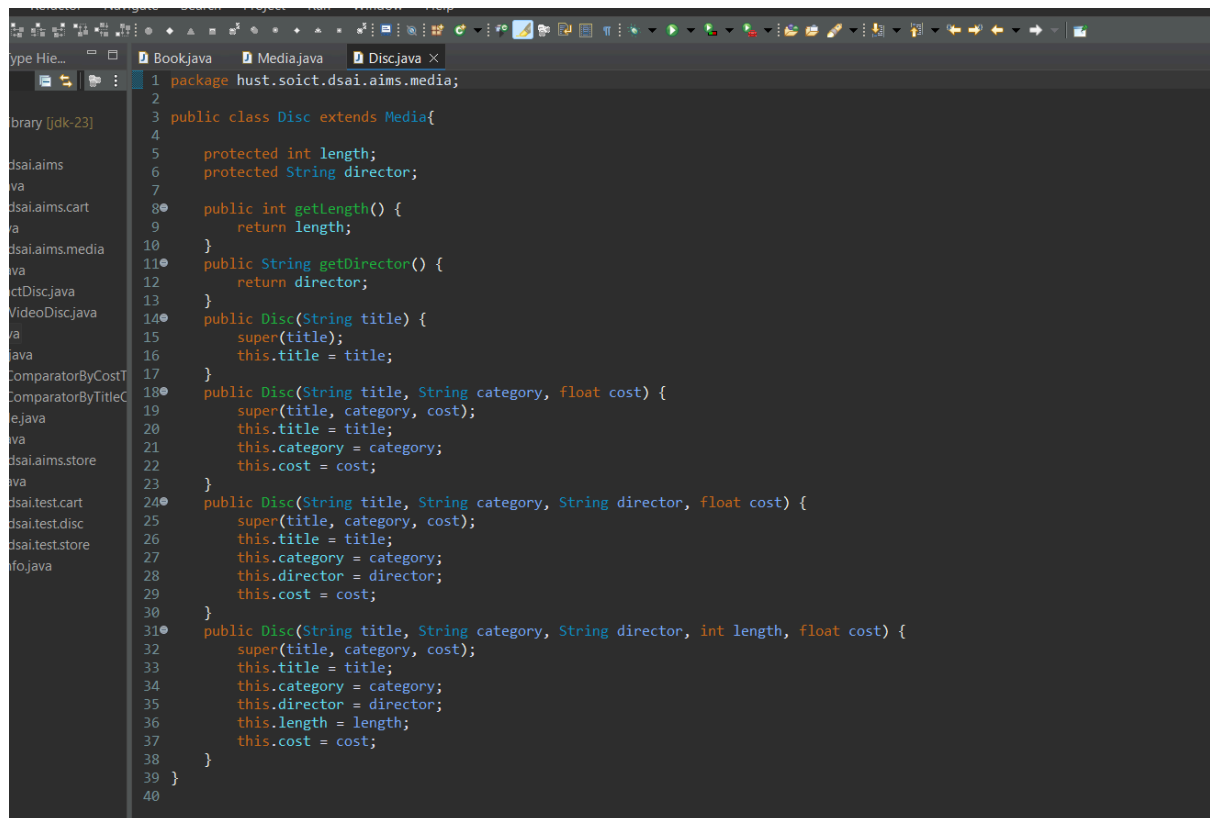
```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4
5 public class Book extends Media {
6
7     private static int nbBooks = 1;
8     private ArrayList<String> authors = new ArrayList<String>();
9
10    public ArrayList<String> getAuthors() {
11        return authors;
12    }
13
14    public void addAuthor(String authorName) {
15        if (authors.contains(authorName)) {
16            System.out.println("Already had '" + authorName + "' in the author list");
17        } else {
18            authors.add(authorName);
19            System.out.println("Add successfully");
20        }
21    }
22
23    public void removeAuthor(String authorName) {
24        if (authors.contains(authorName)) {
25            authors.remove(authorName);
26            System.out.println("Remove successfully");
27        } else {
28            System.out.println("The list doesn't have '" + authorName + "'");
29        }
30    }
31
32    public Book(String title, String category, float cost) {
33        super(title, category, cost);
34        this.id = nbBooks;
35        nbBooks++;
36        this.title = title;
37        this.category = category;
38        this.cost = cost;
39    }
40
41    public String toString() {
42        return "[Book] - [" + this.id + "] - " + this.title + " - " + this.category + ": " + this.cost + "$";
43    }
44 }
```

2.2. Media class.

```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public class Media {
6
7     protected int id;
8     protected String title;
9     protected String category;
10    protected float cost;
11
12    public int getId() {
13        return id;
14    }
15    public String getTitle() {
16        return title;
17    }
18    public String getCategory() {
19        return category;
20    }
21    public float getCost() {
22        return cost;
23    }
24    public void setTitle(String title) {
25        this.title = title;
26    }
27    public Media(String title) {
28        this.title = title;
29    }
30    public Media(String title, String category, float cost) {
31        this.title = title;
32        this.category = category;
33        this.cost = cost;
34    }
35
36    public void play() {
37        System.out.println("Playing media");
38    }
39
40    public boolean equals(Object obj) {
41        if (obj == this) {
42            return true;
43        }
```

```
10    protected float cost;
11
12    public int getId() {
13        return id;
14    }
15    public String getTitle() {
16        return title;
17    }
18    public String getCategory() {
19        return category;
20    }
21    public float getCost() {
22        return cost;
23    }
24    public void setTitle(String title) {
25        this.title = title;
26    }
27    public Media(String title) {
28        this.title = title;
29    }
30    public Media(String title, String category, float cost) {
31        this.title = title;
32        this.category = category;
33        this.cost = cost;
34    }
35
36    public void play() {
37        System.out.println("Playing media");
38    }
39
40    public boolean equals(Object obj) {
41        if (obj == this) {
42            return true;
43        }
44        if (!(obj instanceof Media)) {
45            return false;
46        }
47        return ((Media)obj).getTitle() == this.getTitle();
48    }
49
50    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
51    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
52 }
53
```

2.3. Disc class.



```
1 package hust.soict.dsai.aims.media;
2
3 public class Disc extends Media{
4     protected int length;
5     protected String director;
6
7     public int getLength() {
8         return length;
9     }
10    public String getDirector() {
11        return director;
12    }
13    public Disc(String title) {
14        super(title);
15        this.title = title;
16    }
17
18    public Disc(String title, String category, float cost) {
19        super(title, category, cost);
20        this.title = title;
21        this.category = category;
22        this.cost = cost;
23    }
24    public Disc(String title, String category, String director, float cost) {
25        super(title, category, cost);
26        this.title = title;
27        this.category = category;
28        this.director = director;
29        this.cost = cost;
30    }
31    public Disc(String title, String category, String director, int length, float cost) {
32        super(title, category, cost);
33        this.title = title;
34        this.category = category;
35        this.director = director;
36        this.length = length;
37        this.cost = cost;
38    }
39 }
40
```

2.4. CompactDisc class.

```

1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc implements Playable{
6     private static int nbCompactDiscs = 1;
7     private String artist;
8     private ArrayList<Track> tracks = new ArrayList<Track>();
9     public String getArtist() {
10         return artist;
11     }
12
13     public CompactDisc(String title, String category, String artist, float cost) {
14         super(title, category, cost);
15         this.id = nbCompactDiscs;
16         nbCompactDiscs++;
17         this.title = title;
18         this.category = category;
19         this.cost = cost;
20         this.artist = artist;
21         this.length = getLength();
22     }
23
24     public int getLength() {
25         int length = 0;
26         for (int i = 0; i < tracks.size(); i++) {
27             length += tracks.get(i).getLength();
28         }
29         return length;
30     }
31
32     public void addTrack(Track track) {
33         if (tracks.contains(track)) {
34             System.out.println("Already have '" + track.getTitle() + "' in the track list");
35         } else {
36             tracks.add(track);
37             System.out.println("Add successfully");
38         }
39     }
40
41     public void removeTrack(Track track) {
42         if (tracks.contains(track)) {
43             tracks.remove(track);
44             System.out.println("Remove successfully");
45         }
46     }
47 }

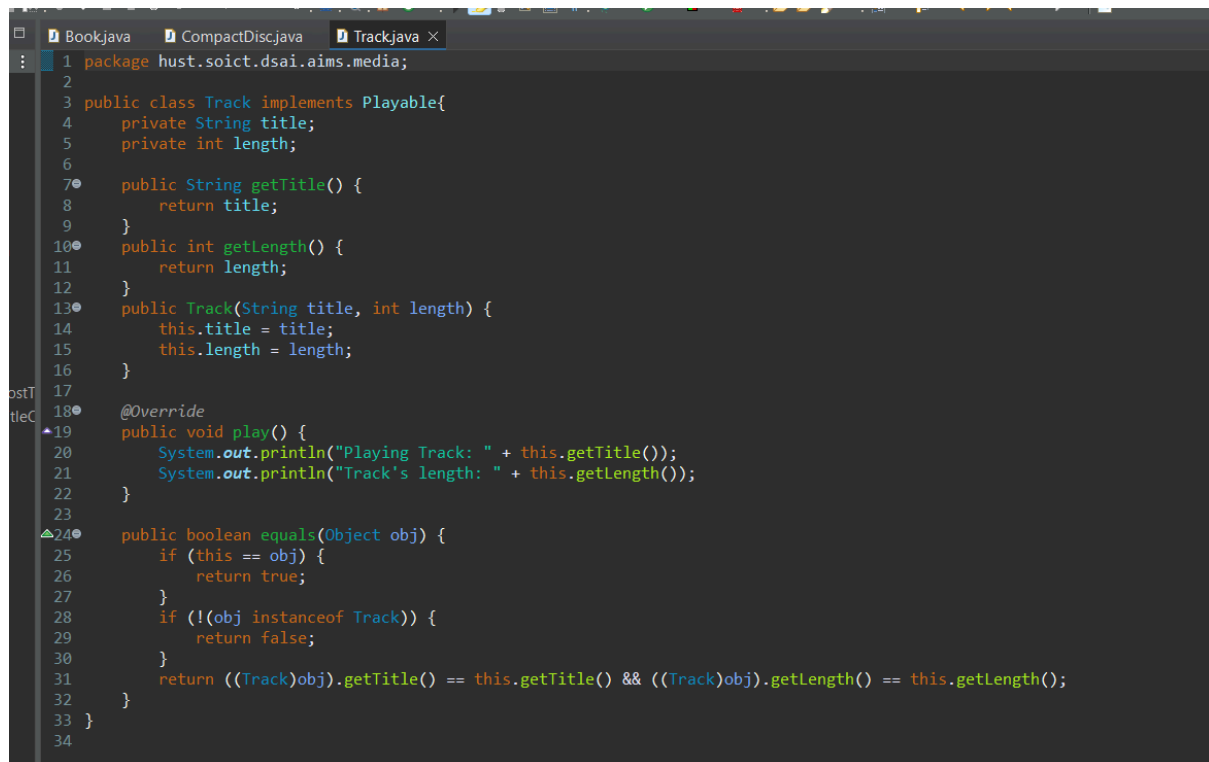
```

```
Bookjava CompactDisc.java
22 }
23
24 public int getLength() {
25     int length = 0;
26     for (int i = 0; i < tracks.size(); i++) {
27         length += tracks.get(i).getLength();
28     }
29     return length;
30 }
31
32 public void addTrack(Track track) {
33     if (tracks.contains(track)) {
34         System.out.println("Already have " + track.getTitle() + " in the track list");
35     } else {
36         tracks.add(track);
37         System.out.println("Add successfully");
38     }
39 }
40
41 public void removetrack(Track track) {
42     if (tracks.contains(track)) {
43         tracks.remove(track);
44         System.out.println("Remove successfully");
45     } else {
46         System.out.println("The track list doesn't have " + track.getTitle() + "");
47     }
48 }
49
50 @Override
51 public void play() {
52     System.out.println("Playing CD: " + this.getTitle());
53     System.out.println("CD length: " + this.getLength());
54     System.out.println("*****TRACK*****");
55     for (int i = 0; i < tracks.size(); i++) {
56         tracks.get(i).play();
57         System.out.println("*****");
58     }
59 }
60
61 public String toString() {
62     return "[CD] - [" + this.id + "] - " + this.title + " - " + this.category + " - " + this.artist + " - " + getLength() +
63 }
64 }
65
```

2.5. DigitalVideoDisc class.

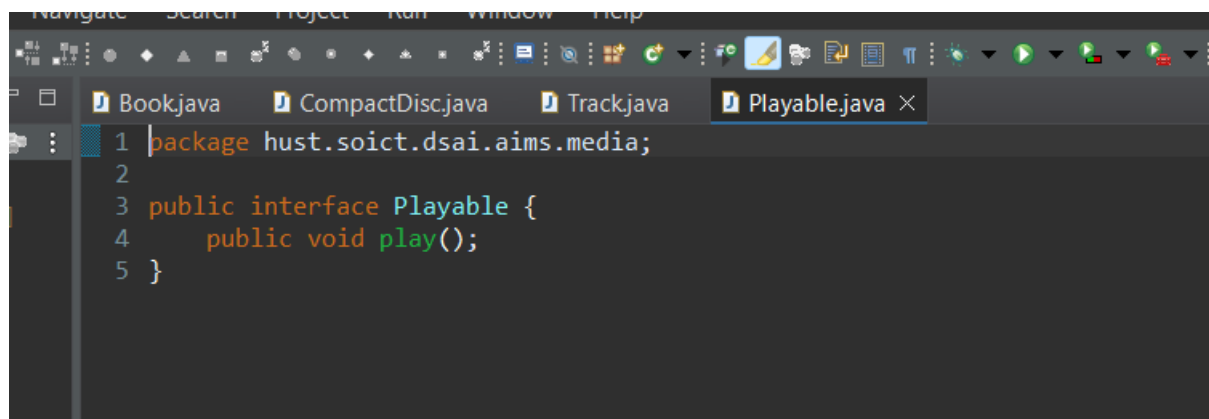
```
1 package hust.soict.dsai.aims.media;
2
3 public class DigitalVideoDisc extends Disc implements Playable{
4     private static int nbDigitalVideoDiscs = 1;
5
6     //Question 6
7     public DigitalVideoDisc(String title) {
8         super(title);
9         this.id = nbDigitalVideoDiscs;
10        nbDigitalVideoDiscs++;
11        this.title = title;
12    }
13    public DigitalVideoDisc(String title, String category, float cost) {
14        super(title, category, cost);
15        this.id = nbDigitalVideoDiscs;
16        nbDigitalVideoDiscs++;
17        this.title = title;
18        this.category = category;
19        this.cost = cost;
20    }
21    public DigitalVideoDisc(String title, String category, String director, float cost) {
22        super(title, category, director, cost);
23        this.id = nbDigitalVideoDiscs;
24        nbDigitalVideoDiscs++;
25        this.title = title;
26        this.category = category;
27        this.director = director;
28        this.cost = cost;
29    }
30    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
31        super(title, category, director, length, cost);
32        this.id = nbDigitalVideoDiscs;
33        nbDigitalVideoDiscs++;
34        this.title = title;
35        this.category = category;
36        this.director = director;
37        this.length = length;
38        this.cost = cost;
39    }
40    @Override
41    public void play() {
42        System.out.println("Playing DVD: " + this.getTitle());
43        System.out.println("DVD length: " + this.getLength());
44    }
45
46    public String toString() {
47        return "[DVD] - [" + this.id + "] - " + this.title + " - " + this.category + " - " + this.director
48            + " - " + this.length + " : " + this.cost + "$";
49    }
50 }
51
```


2.6. Track class.



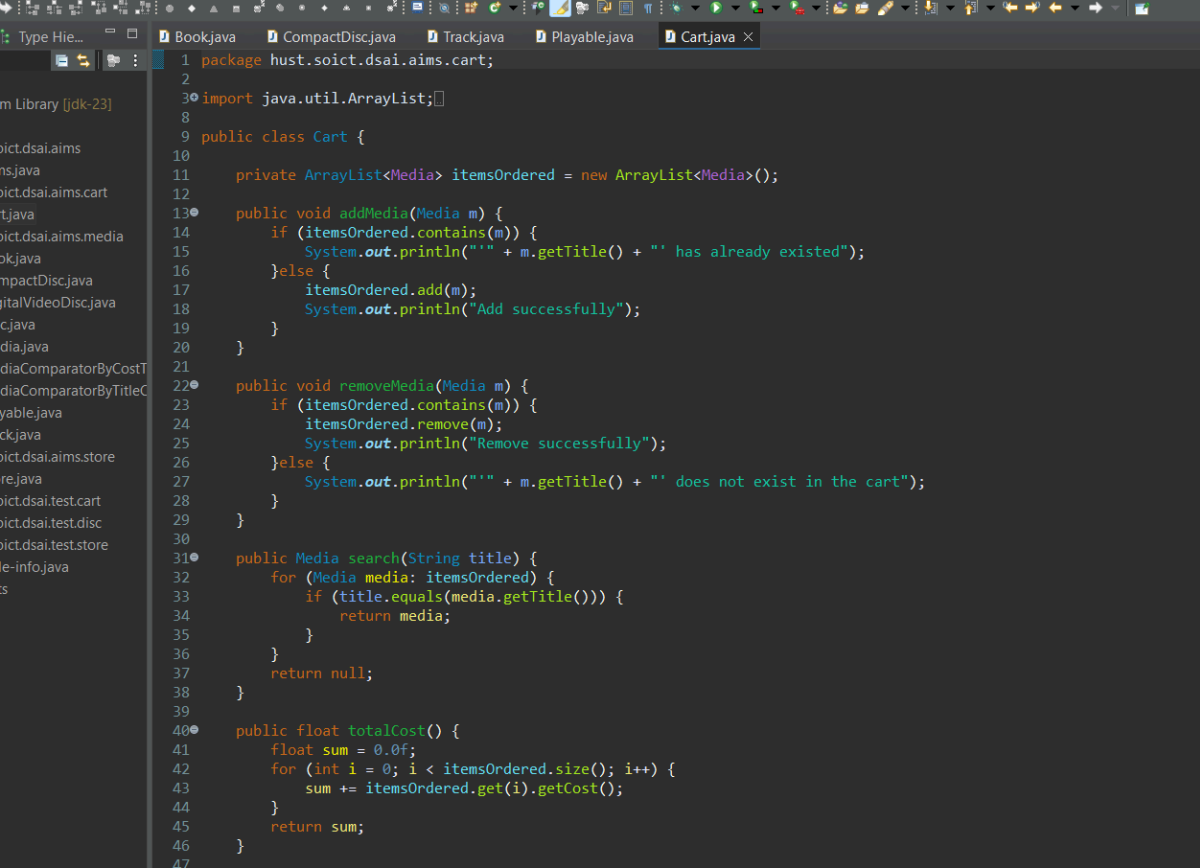
```
1 package hust.soict.dsai.aims.media;
2
3 public class Track implements Playable{
4     private String title;
5     private int length;
6
7     public String getTitle() {
8         return title;
9     }
10    public int getLength() {
11        return length;
12    }
13    public Track(String title, int length) {
14        this.title = title;
15        this.length = length;
16    }
17
18    @Override
19    public void play() {
20        System.out.println("Playing Track: " + this.getTitle());
21        System.out.println("Track's length: " + this.getLength());
22    }
23
24    public boolean equals(Object obj) {
25        if (this == obj) {
26            return true;
27        }
28        if (!(obj instanceof Track)) {
29            return false;
30        }
31        return ((Track)obj).getTitle() == this.getTitle() && ((Track)obj).getLength() == this.getLength();
32    }
33 }
34
```

2.7. Playable interface.

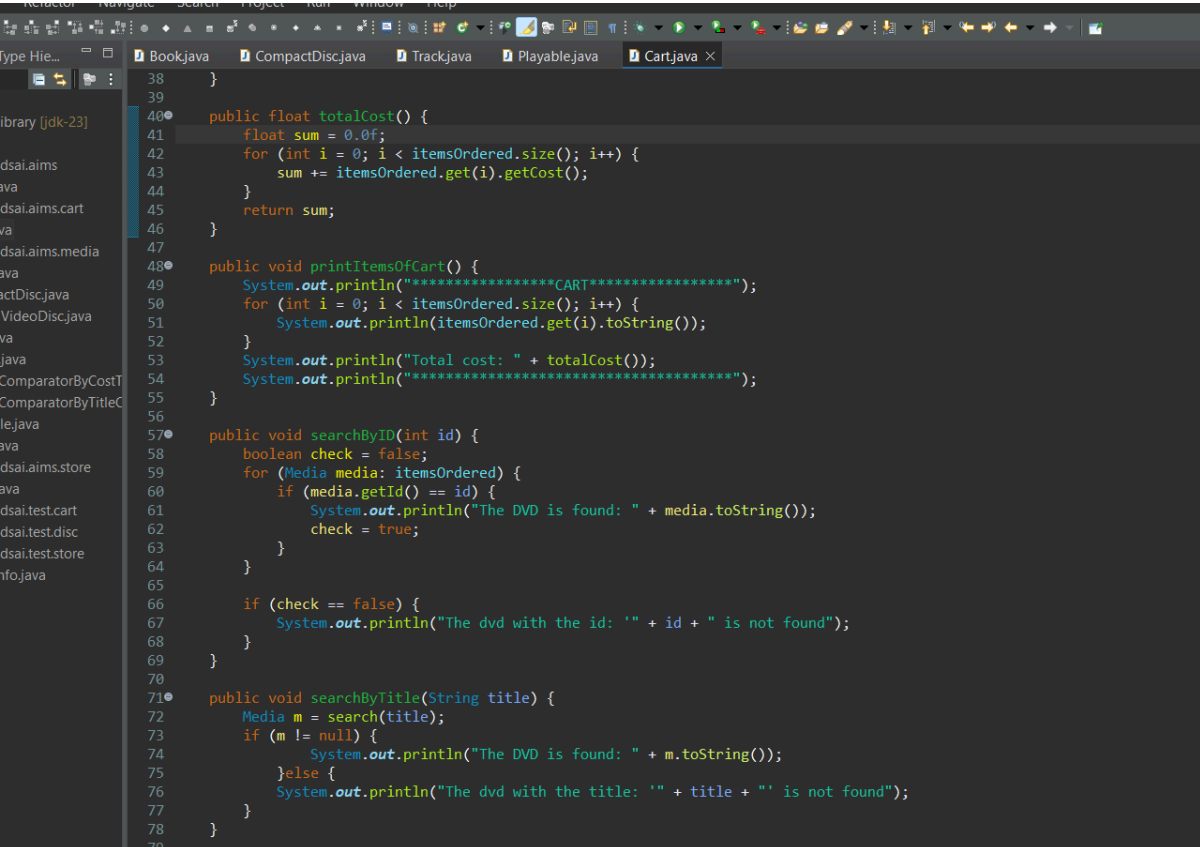


```
1 package hust.soict.dsai.aims.media;
2
3 public interface Playable {
4     public void play();
5 }
```

2.8. Cart class.



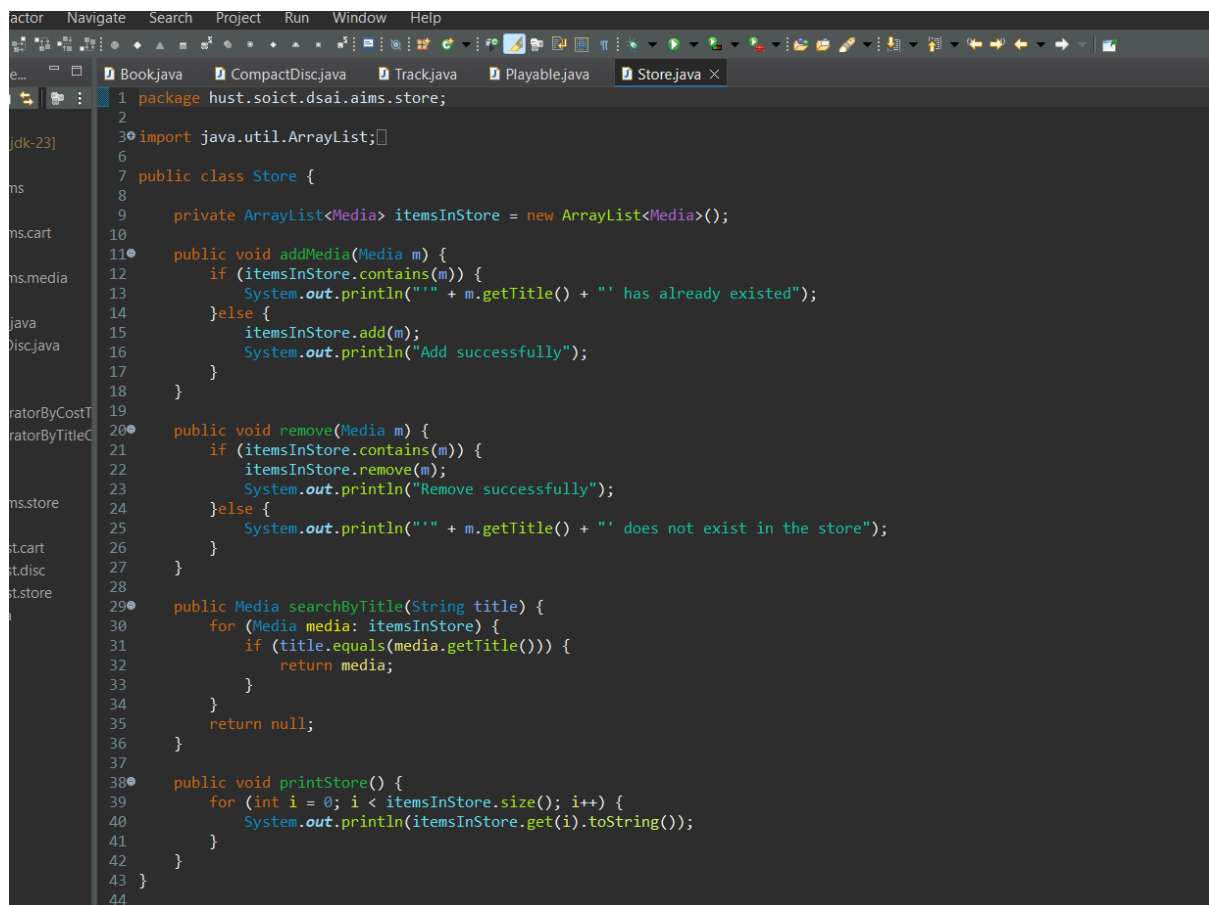
```
1 package hust.soict.dsai.aims.cart;
2
3 import java.util.ArrayList;
4
5 public class Cart {
6
7     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
8
9     public void addMedia(Media m) {
10         if (itemsOrdered.contains(m)) {
11             System.out.println("'" + m.getTitle() + "' has already existed");
12         } else {
13             itemsOrdered.add(m);
14             System.out.println("Add successfully");
15         }
16     }
17
18     public void removeMedia(Media m) {
19         if (itemsOrdered.contains(m)) {
20             itemsOrdered.remove(m);
21             System.out.println("Remove successfully");
22         } else {
23             System.out.println("'" + m.getTitle() + "' does not exist in the cart");
24         }
25     }
26
27     public Media search(String title) {
28         for (Media media: itemsOrdered) {
29             if (title.equals(media.getTitle())) {
30                 return media;
31             }
32         }
33         return null;
34     }
35
36     public float totalCost() {
37         float sum = 0.0f;
38         for (int i = 0; i < itemsOrdered.size(); i++) {
39             sum += itemsOrdered.get(i).getCost();
40         }
41         return sum;
42     }
43
44 }
45
46
47
```



```
38 }
39
40 public float totalCost() {
41     float sum = 0.0f;
42     for (int i = 0; i < itemsOrdered.size(); i++) {
43         sum += itemsOrdered.get(i).getCost();
44     }
45     return sum;
46 }
47
48 public void printItemsOfCart() {
49     System.out.println("*****CART*****");
50     for (int i = 0; i < itemsOrdered.size(); i++) {
51         System.out.println(itemsOrdered.get(i).toString());
52     }
53     System.out.println("Total cost: " + totalCost());
54     System.out.println("*****");
55 }
56
57 public void searchByID(int id) {
58     boolean check = false;
59     for (Media media: itemsOrdered) {
60         if (media.getId() == id) {
61             System.out.println("The DVD is found: " + media.toString());
62             check = true;
63         }
64     }
65
66     if (check == false) {
67         System.out.println("The dvd with the id: '" + id + "' is not found");
68     }
69 }
70
71 public void searchByTitle(String title) {
72     Media m = search(title);
73     if (m != null) {
74         System.out.println("The DVD is found: " + m.toString());
75     } else {
76         System.out.println("The dvd with the title: '" + title + "' is not found");
77     }
78 }
79
```

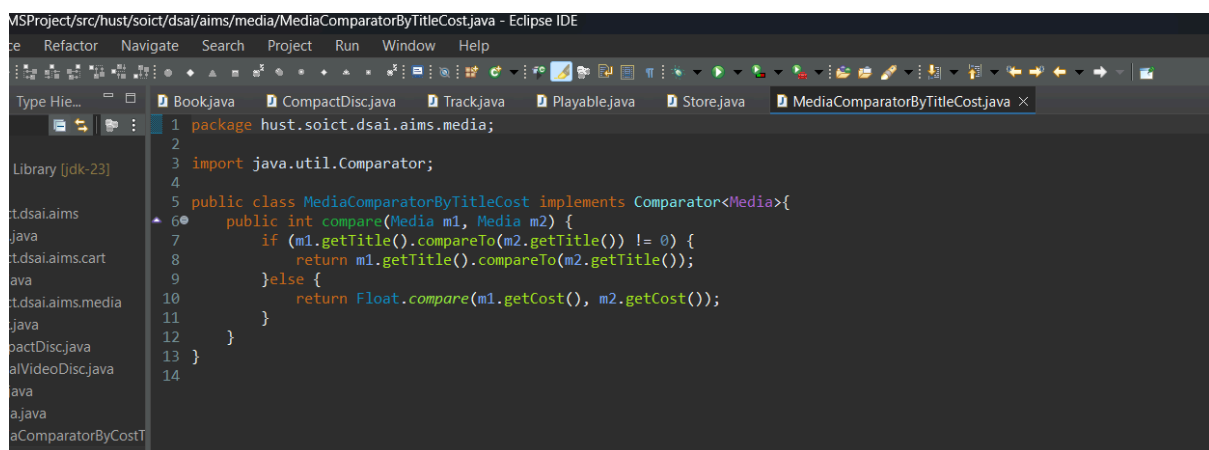
```
Book.java CompactDisc.java Track.java Playable.java Cart.java X
64 }
65
66 if (check == false) {
67     System.out.println("The dvd with the id: " + id + " is not found");
68 }
69 }
70
71 public void searchByTitle(String title) {
72     Media m = search(title);
73     if (m != null) {
74         System.out.println("The DVD is found: " + m.toString());
75     } else {
76         System.out.println("The dvd with the title: '" + title + "' is not found");
77     }
78 }
79
80 /*public boolean isMatching(String title) {
81     for (int i = 0; i < itemsOrdered.size(); i++) {
82         if (title == itemsOrdered.get(i).getTitle()) {
83             return true;
84         }
85     }
86     return false;
87 }*/
88
89 public void sortMediaByTitle() {
90     Collections.sort(itemsOrdered, Media.COMPARE_BY_TITLE_COST);
91     Iterator<Media> iterator = itemsOrdered.iterator();
92
93     while (iterator.hasNext()) {
94         System.out.println(((Media)iterator.next()).toString());
95     }
96 }
97
98 public void sortMediaByCost() {
99     Collections.sort(itemsOrdered, Media.COMPARE_BY_COST_TITLE);
100     Iterator<Media> iterator = itemsOrdered.iterator();
101
102     while (iterator.hasNext()) {
103         System.out.println(((Media)iterator.next()).toString());
104     }
105 }
106 }
107
```

2.9. Store class.



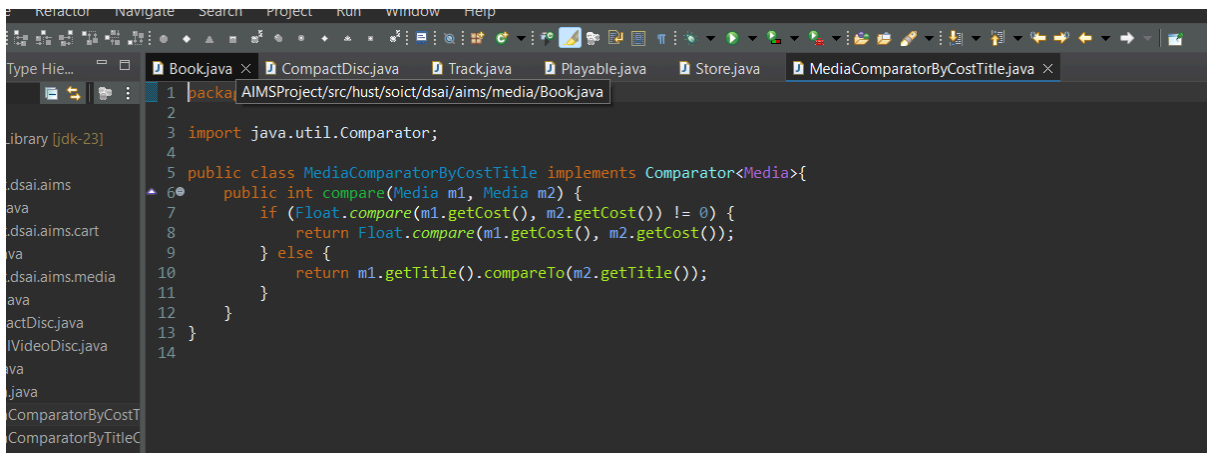
```
1 package hust.soict.dsai.aims.store;
2
3 import java.util.ArrayList;
4
5
6
7 public class Store {
8
9     private ArrayList<Media> itemsInStore = new ArrayList<Media>();
10
11     public void addMedia(Media m) {
12         if (itemsInStore.contains(m)) {
13             System.out.println("'" + m.getTitle() + "' has already existed");
14         } else {
15             itemsInStore.add(m);
16             System.out.println("Add successfully");
17         }
18     }
19
20     public void remove(Media m) {
21         if (itemsInStore.contains(m)) {
22             itemsInStore.remove(m);
23             System.out.println("Remove successfully");
24         } else {
25             System.out.println("'" + m.getTitle() + "' does not exist in the store");
26         }
27     }
28
29     public Media searchByTitle(String title) {
30         for (Media media: itemsInStore) {
31             if (title.equals(media.getTitle())) {
32                 return media;
33             }
34         }
35         return null;
36     }
37
38     public void printStore() {
39         for (int i = 0; i < itemsInStore.size(); i++) {
40             System.out.println(itemsInStore.get(i).toString());
41         }
42     }
43 }
44
```

2.10. MediaComparatorByTitleCost.



```
MSPProject/src/hust/soict/dsai/aims/media/MediaComparatorByTitleCost.java - Eclipse IDE
1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByTitleCost implements Comparator<Media>{
6     public int compare(Media m1, Media m2) {
7         if (m1.getTitle().compareTo(m2.getTitle()) != 0) {
8             return m1.getTitle().compareTo(m2.getTitle());
9         } else {
10             return Float.compare(m1.getCost(), m2.getCost());
11         }
12     }
13 }
14
```

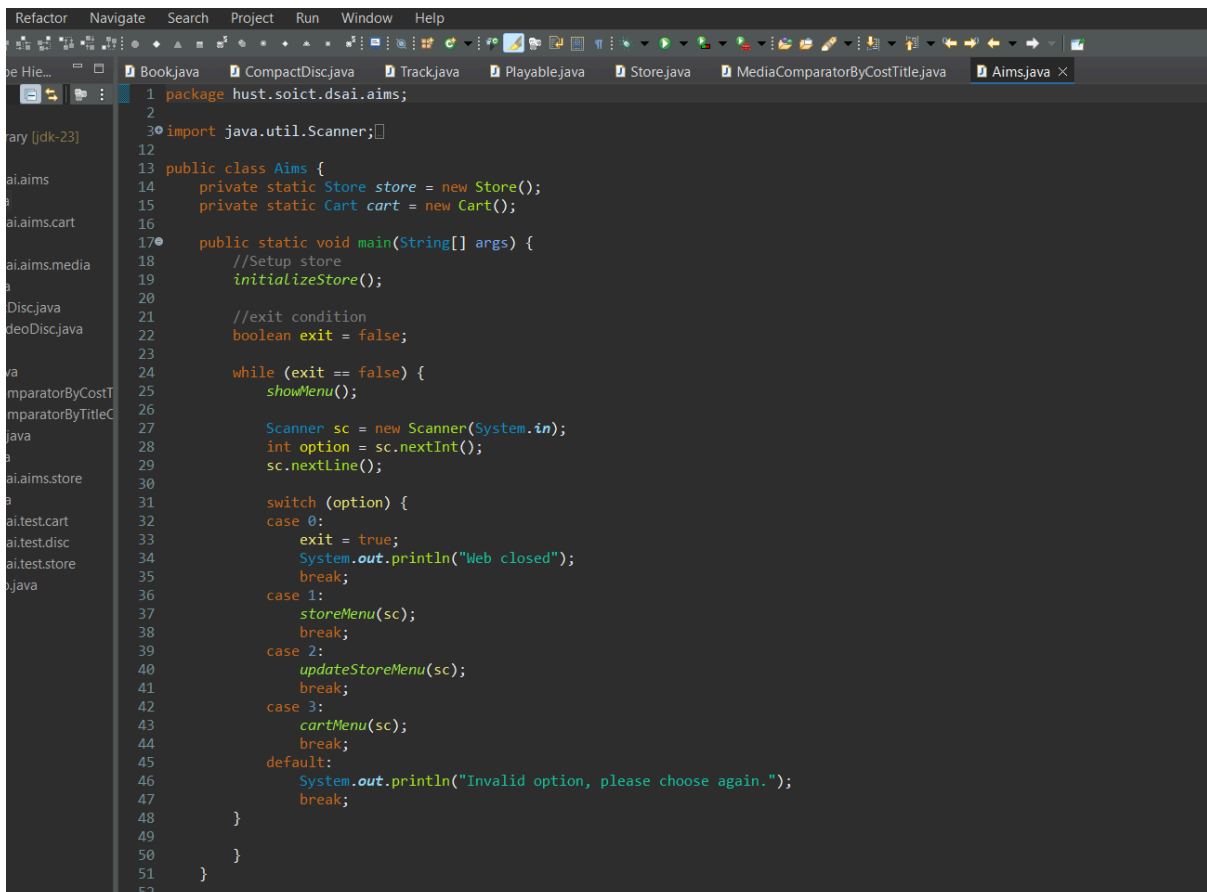
2.11. MediaComparatorByCostTitle.



The screenshot shows an IDE window with the file `MediaComparatorByCostTitle.java` open. The code defines a class `MediaComparatorByCostTitle` that implements the `Comparator<Media>` interface. It has a `compare` method that compares the cost of two media objects. If the costs are equal, it compares the titles.

```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByCostTitle implements Comparator<Media>{
6     public int compare(Media m1, Media m2) {
7         if (Float.compare(m1.getCost(), m2.getCost()) != 0) {
8             return Float.compare(m1.getCost(), m2.getCost());
9         } else {
10            return m1.getTitle().compareTo(m2.getTitle());
11        }
12    }
13 }
14
```

2.12. Aims class.



The screenshot shows an IDE window with the file `Aims.java` open. The code defines a class `Aims` that manages a store and a cart. It has a `main` method that runs a loop where the user can interact with the store and cart. The code uses `Scanner` for input and `System.out` for output.

```
1 package hust.soict.dsai.aims;
2
3 import java.util.Scanner;
4
5 public class Aims {
6     private static Store store = new Store();
7     private static Cart cart = new Cart();
8
9     public static void main(String[] args) {
10        //Setup store
11        initializeStore();
12
13        //exit condition
14        boolean exit = false;
15
16        while (exit == false) {
17            showMenu();
18
19            Scanner sc = new Scanner(System.in);
20            int option = sc.nextInt();
21            sc.nextLine();
22
23            switch (option) {
24                case 0:
25                    exit = true;
26                    System.out.println("Web closed");
27                    break;
28                case 1:
29                    storeMenu(sc);
30                    break;
31                case 2:
32                    updateStoreMenu(sc);
33                    break;
34                case 3:
35                    cartMenu(sc);
36                    break;
37                default:
38                    System.out.println("Invalid option, please choose again.");
39                    break;
40            }
41        }
42    }
43 }
44
```

```
Type Hie...  Bookjava  CompactDisc.java  Trackjava  Playable.java  Store.java  MediaComparatorByCostTitle.java  Aimsjava x
48
49
50
51
52
53 public static void initializeStore() {}
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 public static void showMenu() {
101     System.out.println("AIMS: ");
102     System.out.println("-----");
103     System.out.println("1. View store");
104     System.out.println("2. Update store");
105     System.out.println("3. See current cart");
106     System.out.println("0. Exit");
107     System.out.println("-----");
108     System.out.println("Please choose a number: 0-1-2-3");
109 }
110 public static void storeMenu(Scanner sc) {
111     boolean back = false;
112
113     while (back == false) {
114         store.printStore();
115         System.out.println("-----");
116         System.out.println("Options: ");
117         System.out.println("-----");
118         System.out.println("1. See a media's details");
119         System.out.println("2. Add a media to cart");
120         System.out.println("3. Play a media");
121         System.out.println("4. See current cart");
122         System.out.println("0. Back");
123         System.out.println("-----");
124         System.out.println("Please choose a number: 0-1-2-3-4");
125
126         int option = sc.nextInt();
127         sc.nextLine();
128
129         switch (option) {
130             case 0:
131                 back = true;
132                 break;
133             case 1:
134                 boolean foundDetails = false;
135                 while (!foundDetails) {
```

```
123         System.out.println("-----");
124         System.out.println("Please choose a number: 0-1-2-3-4");
125
126         int option = sc.nextInt();
127         sc.nextLine();
128
129         switch (option) {
130             case 0:
131                 back = true;
132                 break;
133             case 1:
134                 boolean foundDetails = false;
135                 while (!foundDetails) {
136                     System.out.println("Enter the title of the media (type 0 to stop): ");
137                     String title = sc.nextLine();
138                     if (title.equals("0")) {
139                         break;
140                     }
141                     Media media = store.searchByTitle(title);
142                     if (media != null) {
143                         System.out.println("Details: ");
144                         System.out.println(media);
145                         mediaDetailsMenu(sc, media);
146                         foundDetails = true;
147                     } else {
148                         System.out.println("----MEDIA NOT FOUND----");
149                     }
150                 }
151                 break;
152             case 2:
153                 boolean foundToAdd = false;
154                 while (!foundToAdd) {
155                     System.out.println("Enter the title of the media (type 0 to stop): ");
156                     String title = sc.nextLine();
157                     if (title.equals("0")) {
158                         break;
159                     }
160                     Media media = store.searchByTitle(title);
161                     if (media != null) {
162                         cart.addMedia(media);
163                     } else {
164                         System.out.println("----MEDIA NOT FOUND----");
165                     }
166                 }
```

```
Refactor  Navigate  Search  Project  Run  Window  Help
Bookjava  CompactDiscjava  Trackjava  Playablejava  Storejava  MediaComparatorByCostTitlejava  Aimsjava x
159
160     Media media = store.searchByTitle(title);
161     if (media != null) {
162         cart.addMedia(media);
163     } else {
164         System.out.println("----MEDIA NOT FOUND----");
165     }
166 }
167 break;
168 case 3:
169     boolean foundToPlay = false;
170     while (!foundToPlay) {
171         System.out.println("Enter the title of the media (type 0 to stop): ");
172         String title = sc.nextLine();
173         if (title.equals("0")) {
174             break;
175         }
176         Media media = store.searchByTitle(title);
177         if (media != null) {
178             if (media instanceof Disc || media instanceof CompactDisc) {
179                 media.play();
180             } else {
181                 System.out.println("This is a book!");
182             }
183             foundToPlay = true;
184         } else {
185             System.out.println("----MEDIA NOT FOUND----");
186         }
187     }
188     break;
189 case 4:
190     cartMenu(sc);
191     break;
192 default:
193     System.out.println("Invalid option, please choose again.");
194     break;
195 }
196 }
197 }
198 public static void mediaDetailsMenu(Scanner sc, Media media) {
199     boolean back = false;
200     while (back == false) {
201         System.out.println("Options: ");
202     }
203 }
```

```
Navigate  Search  Project  Run  Window  Help
Bookjava  CompactDiscjava  Trackjava  Playablejava  Storejava  MediaComparatorByCostTitlejava  Aimsjava x
195
196 }
197 }
198 public static void mediaDetailsMenu(Scanner sc, Media media) {
199     boolean back = false;
200     while (back == false) {
201         System.out.println("Options: ");
202         System.out.println("-----");
203         System.out.println("1. Add to cart");
204         System.out.println("2. Play");
205         System.out.println("0. Back");
206         System.out.println("-----");
207         System.out.println("Please choose a number: 0-1-2");
208
209         int option = sc.nextInt();
210         sc.nextLine();
211
212         switch (option) {
213             case 0:
214                 back = true;
215                 break;
216             case 1:
217                 cart.addMedia(media);
218                 break;
219             case 2:
220                 if (media instanceof Disc || media instanceof CompactDisc) {
221                     media.play();
222                 } else {
223                     System.out.println("This is a book");
224                 }
225                 break;
226             default:
227                 System.out.println("Invalid option, please choose again.");
228                 break;
229         }
230     }
231 }
232 }
233 public static void cartMenu(Scanner sc) {
234     boolean back = false;
235     while (back == false) {
236         cart.printItemsOfCart();
237         System.out.println("Options: ");
238         System.out.println("-----");
239     }
240 }
```

```
efactor  Navigate  Search  Project  Run  Window  Help
Bookjava  CompactDisc.java  Track.java  Playable.java  Store.java  MediaComparatorByCostTitle.java  Aims.java
225     }
226     break;
227     default:
228         System.out.println("Invalid option, please choose again.");
229         break;
230     }
231 }
232 }
233 public static void cartMenu(Scanner sc) {
234     boolean back = false;
235     while (back == false) {
236         cart.printItemsOfCart();
237         System.out.println("Options: ");
238         System.out.println("-----");
239         System.out.println("1. Filter medias in cart");
240         System.out.println("2. Sort medias in cart");
241         System.out.println("3. Remove media from cart");
242         System.out.println("4. Play a media");
243         System.out.println("5. Place order");
244         System.out.println("0. Back");
245         System.out.println("-----");
246         System.out.println("Please choose a number: 0-1-2-3-4-5");
247
248         int option = sc.nextInt();
249         sc.nextLine();
250
251         switch (option) {
252             case 0:
253                 back = true;
254                 break;
255             case 1:
256                 System.out.println("Filter medias in cart by (1) id or (2) title:");
257                 int filterOption = sc.nextInt();
258                 sc.nextLine();
259                 boolean found = false;
260                 while (!found) {
261                     if (filterOption == 1) {
262                         System.out.println("Enter the id to filter (type 0 to stop):");
263                         int id = sc.nextInt();
264                         sc.nextLine();
265                         if (id == 0) {
266                             break;
267                         }
268                         cart.searchById(id);
```

```
Type Hierarchy  Bookjava  CompactDisc.java  Track.java  Playable.java  Store.java  MediaComparatorByCostTitle.java  Aims.java
Library [jdk-23]
264         sc.nextLine();
265         if (id == 0) {
266             break;
267         }
268         cart.searchById(id);
269         found = true;
270     } else if (filterOption == 2) {
271         System.out.println("Enter the title to filter (type 0 to stop):");
272         String title = sc.nextLine();
273         if (title.equals("0")) {
274             break;
275         }
276         cart.searchByTitle(title);
277         found = true;
278     } else if (filterOption == 0) {
279         break;
280     } else {
281         System.out.println("Invalid option.");
282     }
283     }
284     break;
285     case 2:
286         System.out.println("Sort medias in cart by (1) title or (2) cost:");
287         int sortOption = sc.nextInt();
288         sc.nextLine();
289         if (sortOption == 1) {
290             cart.sortMediaByTitle();
291         } else if (sortOption == 2) {
292             cart.sortMediaByCost();
293         } else {
294             System.out.println("Invalid option.");
295         }
296         break;
297     case 3:
298         boolean foundToRemove = false;
299         while (!foundToRemove) {
300             System.out.println("Enter the title of the media (type 0 to stop): ");
301             String title = sc.nextLine();
302             if (title.equals("0")) {
303                 break;
304             }
305             Media media = cart.search(title);
306             if (media != null) {
307                 cart.removeMedia(media);
```



```
Book.java CompactDisc.java Track.java Playable.java Store.java MediaComparatorByCostTitle.java Aims.java x
303         break;
304     }
305     Media media = cart.search(title);
306     if (media != null) {
307         cart.removeMedia(media);
308         foundToRemove = true;
309     } else {
310         System.out.println("----MEDIA NOT FOUND----");
311     }
312 }
313 break;
314 case 4:
315     boolean foundToPlay = false;
316     while (!foundToPlay) {
317         System.out.println("Enter the title of the media (type 0 to stop): ");
318         String title = sc.nextLine();
319         if (title.equals("0")) {
320             break;
321         }
322         Media media = store.searchByTitle(title);
323         if (media != null) {
324             if (media instanceof Disc || media instanceof CompactDisc) {
325                 media.play();
326             } else {
327                 System.out.println("This is a book");
328             }
329             foundToPlay = true;
330         } else {
331             System.out.println("----MEDIA NOT FOUND----");
332         }
333     }
334     break;
335 default:
336     System.out.println("Invalid option, please choose again.");
337     break;
338 }
339 }
340 }
341
342 public static void updateStoreMenu(Scanner sc) {
343     boolean back = false;
344     while (back == false) {
345         store.printStore();
346         System.out.println("-----");
347         System.out.println("Options: ");
348         System.out.println("-----");
349         System.out.println("1. Add a media to store");
350         System.out.println("2. Remove a media from store");
351         System.out.println("0. Back");
352         System.out.println("-----");
353         System.out.println("Please choose a number: 0-1-2");
354
355         int option = sc.nextInt();
356         sc.nextLine();
357
358         switch (option) {
359             case 0:
360                 back = true;
361                 break;
362             case 1:
363                 System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
364                 int category = sc.nextInt();
365                 sc.nextLine();
366
367                 if (category == 1) {
368                     System.out.println("Enter book title: ");
369                     String bookTitle = sc.nextLine();
370                     System.out.println("Enter book category: ");
371                     String bookCategory = sc.nextLine();
372                     System.out.println("Enter book cost: ");
373                     Float bookCost = sc.nextFloat();
374                     sc.nextLine();
375
376                     Book newBook = new Book(bookTitle, bookCategory, bookCost);
377                     store.addMedia(newBook);
378                 } else if (category == 2) {
379                     System.out.println("Enter CD title: ");
380                     String cdTitle = sc.nextLine();
381                     System.out.println("Enter CD category: ");
382                     String cdCategory = sc.nextLine();
383                 }
384             case 2:
385                 System.out.println("Enter the title of the media (type 0 to stop): ");
386                 String title = sc.nextLine();
387                 if (title.equals("0")) {
388                     break;
389                 }
390                 Media media = store.searchByTitle(title);
391                 if (media != null) {
392                     if (media instanceof Disc || media instanceof CompactDisc) {
393                         media.play();
394                     } else {
395                         System.out.println("This is a book");
396                     }
397                 } else {
398                     System.out.println("----MEDIA NOT FOUND----");
399                 }
400             case 3:
401                 System.out.println("Enter the title of the media (type 0 to stop): ");
402                 String title = sc.nextLine();
403                 if (title.equals("0")) {
404                     break;
405                 }
406                 Media media = cart.search(title);
407                 if (media != null) {
408                     cart.removeMedia(media);
409                     foundToRemove = true;
410                 } else {
411                     System.out.println("----MEDIA NOT FOUND----");
412                 }
413             case 4:
414                 boolean foundToPlay = false;
415                 while (!foundToPlay) {
416                     System.out.println("Enter the title of the media (type 0 to stop): ");
417                     String title = sc.nextLine();
418                     if (title.equals("0")) {
419                         break;
420                     }
421                     Media media = store.searchByTitle(title);
422                     if (media != null) {
423                         if (media instanceof Disc || media instanceof CompactDisc) {
424                             media.play();
425                         } else {
426                             System.out.println("This is a book");
427                         }
428                         foundToPlay = true;
429                     } else {
430                         System.out.println("----MEDIA NOT FOUND----");
431                     }
432                 }
433                 break;
434             default:
435                 System.out.println("Invalid option, please choose again.");
436                 break;
437         }
438     }
439 }
440 }
```

```
Book.java CompactDisc.java Track.java Playable.java Store.java MediaComparatorByCostTitle.java Aims.java x
339 }
340 }
341
342 public static void updateStoreMenu(Scanner sc) {
343     boolean back = false;
344     while (back == false) {
345         store.printStore();
346         System.out.println("-----");
347         System.out.println("Options: ");
348         System.out.println("-----");
349         System.out.println("1. Add a media to store");
350         System.out.println("2. Remove a media from store");
351         System.out.println("0. Back");
352         System.out.println("-----");
353         System.out.println("Please choose a number: 0-1-2");
354
355         int option = sc.nextInt();
356         sc.nextLine();
357
358         switch (option) {
359             case 0:
360                 back = true;
361                 break;
362             case 1:
363                 System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
364                 int category = sc.nextInt();
365                 sc.nextLine();
366
367                 if (category == 1) {
368                     System.out.println("Enter book title: ");
369                     String bookTitle = sc.nextLine();
370                     System.out.println("Enter book category: ");
371                     String bookCategory = sc.nextLine();
372                     System.out.println("Enter book cost: ");
373                     Float bookCost = sc.nextFloat();
374                     sc.nextLine();
375
376                     Book newBook = new Book(bookTitle, bookCategory, bookCost);
377                     store.addMedia(newBook);
378                 } else if (category == 2) {
379                     System.out.println("Enter CD title: ");
380                     String cdTitle = sc.nextLine();
381                     System.out.println("Enter CD category: ");
382                     String cdCategory = sc.nextLine();
383                 }
384             case 2:
385                 System.out.println("Enter the title of the media (type 0 to stop): ");
386                 String title = sc.nextLine();
387                 if (title.equals("0")) {
388                     break;
389                 }
390                 Media media = store.searchByTitle(title);
391                 if (media != null) {
392                     if (media instanceof Disc || media instanceof CompactDisc) {
393                         media.play();
394                     } else {
395                         System.out.println("This is a book");
396                     }
397                 } else {
398                     System.out.println("----MEDIA NOT FOUND----");
399                 }
400             case 3:
401                 System.out.println("Enter the title of the media (type 0 to stop): ");
402                 String title = sc.nextLine();
403                 if (title.equals("0")) {
404                     break;
405                 }
406                 Media media = cart.search(title);
407                 if (media != null) {
408                     cart.removeMedia(media);
409                     foundToRemove = true;
410                 } else {
411                     System.out.println("----MEDIA NOT FOUND----");
412                 }
413             case 4:
414                 boolean foundToPlay = false;
415                 while (!foundToPlay) {
416                     System.out.println("Enter the title of the media (type 0 to stop): ");
417                     String title = sc.nextLine();
418                     if (title.equals("0")) {
419                         break;
420                     }
421                     Media media = store.searchByTitle(title);
422                     if (media != null) {
423                         if (media instanceof Disc || media instanceof CompactDisc) {
424                             media.play();
425                         } else {
426                             System.out.println("This is a book");
427                         }
428                         foundToPlay = true;
429                     } else {
430                         System.out.println("----MEDIA NOT FOUND----");
431                     }
432                 }
433                 break;
434             default:
435                 System.out.println("Invalid option, please choose again.");
436                 break;
437         }
438     }
439 }
440 }
```

```
Bookjava CompactDiscjava Trackjava Playablejava Storejava MediaComparatorByCostTitlejava Aimsjava X
378     } else if (category == 2) {
379         System.out.println("Enter CD title: ");
380         String cdTitle = sc.nextLine();
381         System.out.println("Enter CD category: ");
382         String cdCategory = sc.nextLine();
383         System.out.println("Enter CD artist: ");
384         String cdArtist = sc.nextLine();
385         System.out.println("Enter CD cost: ");
386         Float cdCost = sc.nextFloat();
387         sc.nextLine();
388
389         CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);
390
391         System.out.println("Do you want to add tracks to your CD? (1) Yes (0) No:");
392         int trackChoice = sc.nextInt();
393         sc.nextLine();
394
395         if (trackChoice == 1) {
396             System.out.println("How many tracks in your CD?");
397             int numTrack = sc.nextInt();
398             sc.nextLine();
399             for (int i = 0; i < numTrack; i++) {
400                 System.out.println("Your " + (i+1) + " track: ");
401                 System.out.println("Enter track title: ");
402                 String trackTitle = sc.nextLine();
403                 System.out.println("Enter track length: ");
404                 int trackLength = sc.nextInt();
405                 sc.nextLine();
406
407                 Track newTrack = new Track(trackTitle, trackLength);
408                 newCD.addTrack(newTrack);
409             }
410             store.addMedia(newCD);
411         } else if (trackChoice == 0) {
412             store.addMedia(newCD);
413         }
414     } else if (category == 3) {
415         System.out.println("Enter DVD title: ");
416         String dvdTitle = sc.nextLine();
417         System.out.println("Enter DVD category: ");
418         String dvdCategory = sc.nextLine();
419         System.out.println("Enter book cost: ");
420         Float dvdCost = sc.nextFloat();
```

```
Bookjava CompactDiscjava Trackjava Playablejava Storejava MediaComparatorByCostTitlejava Aimsjava X
414     }
415     } else if (category == 3) {
416         System.out.println("Enter DVD title: ");
417         String dvdTitle = sc.nextLine();
418         System.out.println("Enter DVD category: ");
419         String dvdCategory = sc.nextLine();
420         System.out.println("Enter book cost: ");
421         Float dvdCost = sc.nextFloat();
422         sc.nextLine();
423
424         DigitalVideoDisc newDVD = new DigitalVideoDisc(dvdTitle, dvdCategory, dvdCost);
425         store.addMedia(newDVD);
426     } else if (category == 0) {
427         break;
428     } else {
429         System.out.println("Invalid option.");
430     }
431     break;
432
433     case 2:
434         boolean found = false;
435         while (!found) {
436             System.out.println("Enter the title of the media (type 0 to stop): ");
437             String titleForRemove = sc.nextLine();
438             if (titleForRemove.equals("0")) {
439                 break;
440             }
441             Media media = store.searchByTitle(titleForRemove);
442             if (media != null) {
443                 store.remove(media);
444                 found = true;
445             } else {
446                 System.out.println("----MEDIA NOT FOUND----");
447             }
448         }
449         break;
450     default:
451         System.out.println("Invalid option, please choose again.");
452         break;
453     }
454 }
455 }
456 }
457 }
```

3. Answer questions.

Question: Alternatively, to compare items in the cart, instead of using `Comparator`, we can use the `Comparable` interface and override the `compareTo()` method. You can refer to the Java docs to see the information of this interface.

Suppose we are taking this `Comparable` interface approach.

- What class should implement the `Comparable` interface?
Media class should implement the `Comparable` interface.
- In those classes, how should you implement the `compareTo()` method be to reflect the ordering that we want?
The answer can be found in 'Media.java' file.
- Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this `Comparable` interface approach?
No we cannot. The `Comparable` interface assumes that there is only one natural ordering for the objects being compared.
- Suppose the DVDs has a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?
We can override the `compareTo` method in `Disc` class to reflect the new ordering rule.
The modified code can be found in 'Disc.java' file