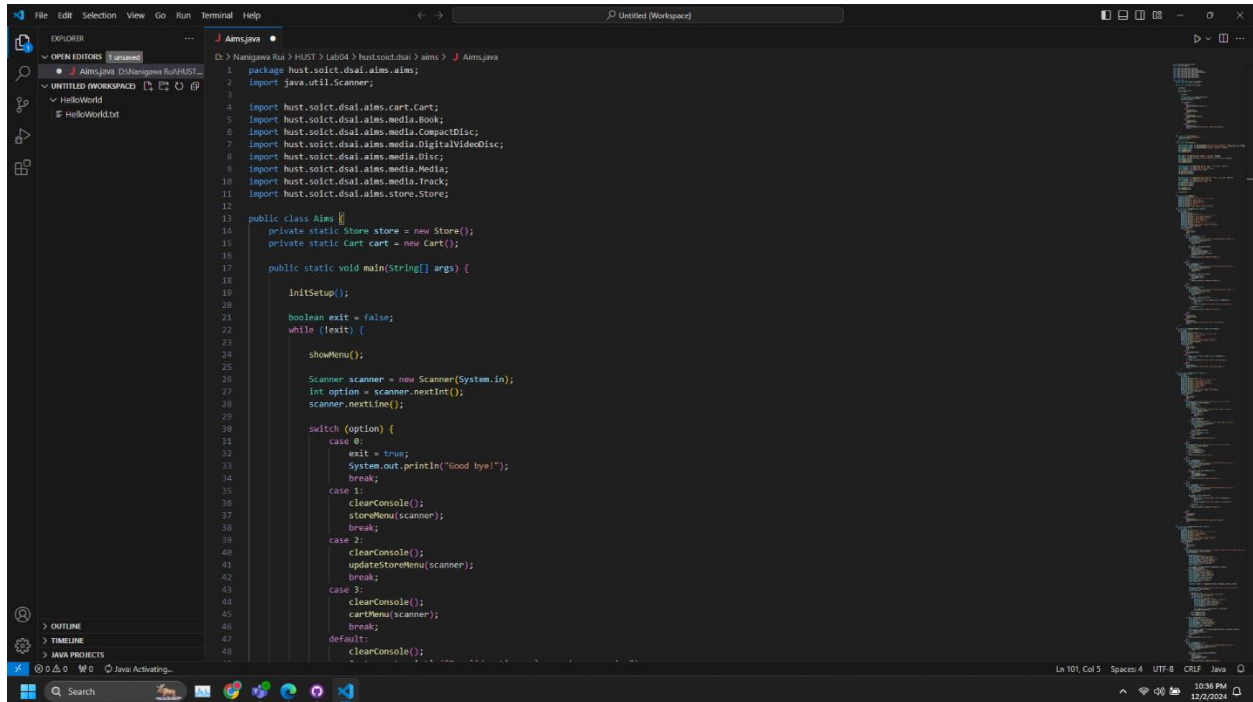


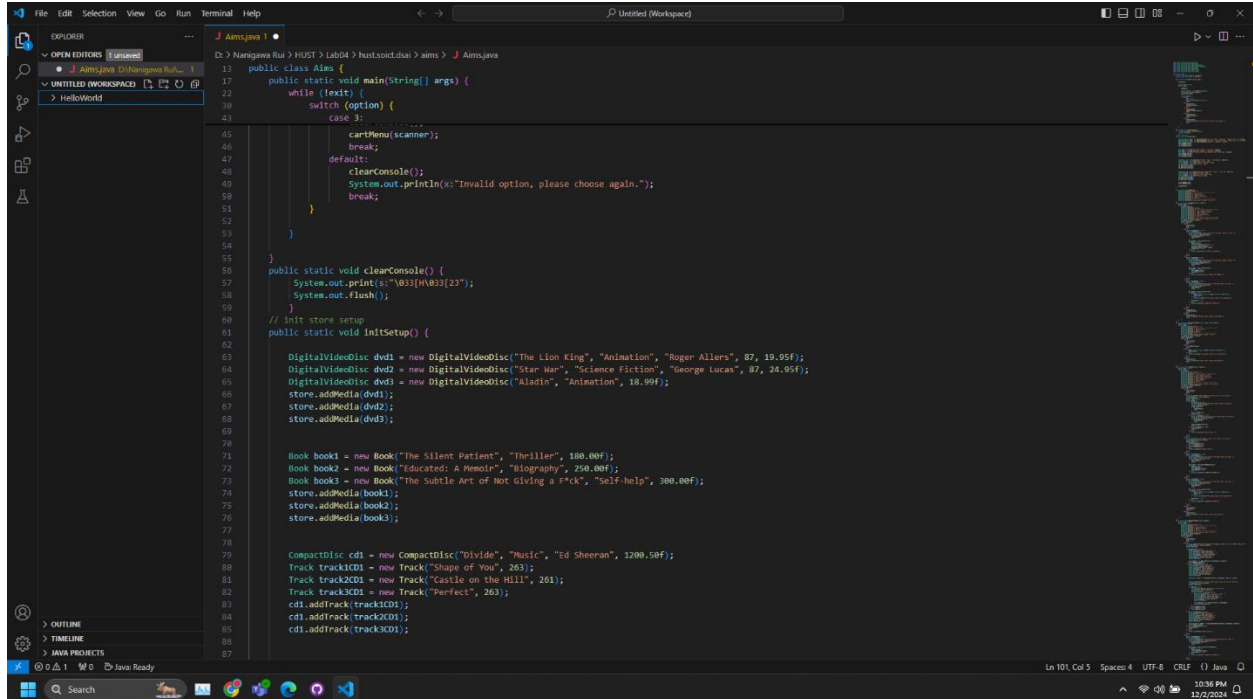
Nguyen Duc Anh – 20235585

2. Additional requirements of AIMS



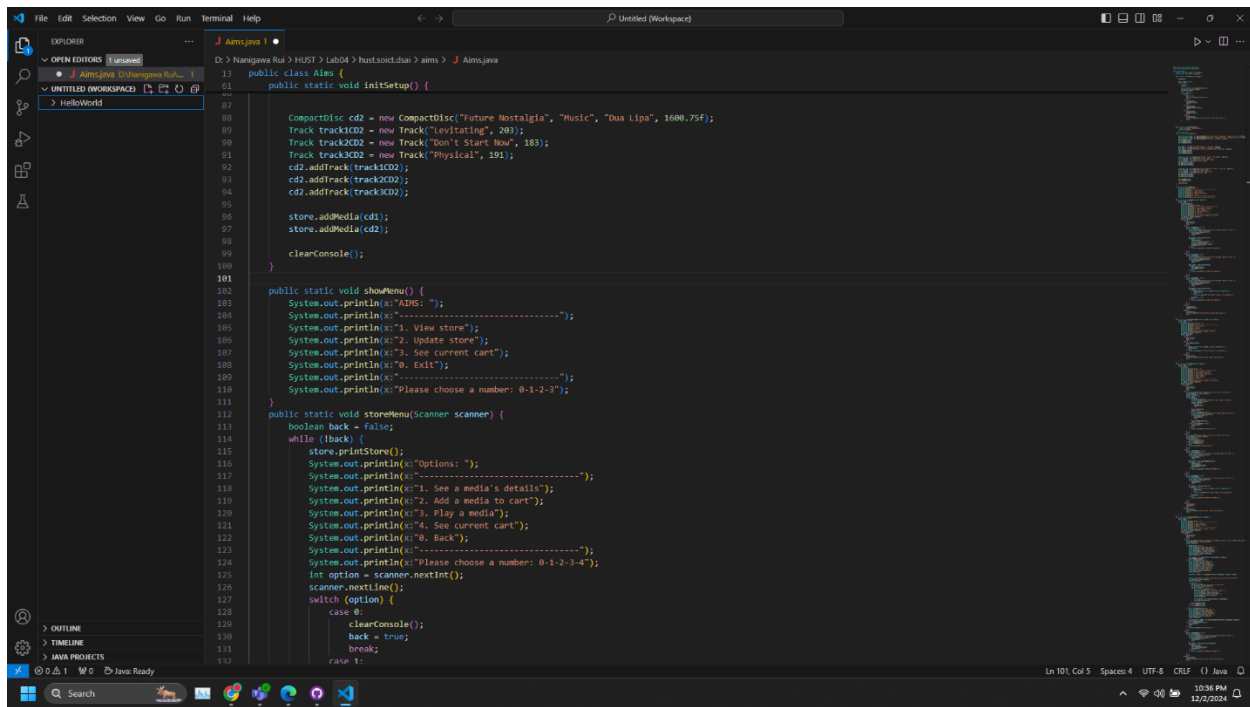
This screenshot shows the initial state of the Aims.java file in an IDE. The code defines a package, imports necessary classes, and sets up a main method with a menu loop. The menu options are limited to exiting the program, clearing the console, and updating the store menu.

```
1 package hust.soict.dsai.aims.aims;
2 import java.util.Scanner;
3
4 import hust.soict.dsai.aims.cart.Cart;
5 import hust.soict.dsai.aims.media.Book;
6 import hust.soict.dsai.aims.media.CompactDisc;
7 import hust.soict.dsai.aims.media.DigitalVideoDisc;
8 import hust.soict.dsai.aims.media.Disc;
9 import hust.soict.dsai.aims.media.Media;
10 import hust.soict.dsai.aims.media.Track;
11 import hust.soict.dsai.aims.store.Store;
12
13 public class Aims {
14     private static Store store = new Store();
15     private static Cart cart = new Cart();
16
17     public static void main(String[] args) {
18         initSetup();
19
20         boolean exit = false;
21         while (!exit) {
22             showMenu();
23
24             Scanner scanner = new Scanner(System.in);
25             int option = scanner.nextInt();
26             scanner.nextLine();
27
28             switch (option) {
29                 case 0:
30                     exit = true;
31                     System.out.println("Good bye!");
32                     break;
33                 case 1:
34                     clearConsole();
35                     storeMenu(scanner);
36                     break;
37                 case 2:
38                     clearConsole();
39                     updateStoreMenu(scanner);
40                     break;
41                 case 3:
42                     clearConsole();
43                     cartMenu(scanner);
44                     break;
45                 default:
46                     clearConsole();
47             }
48         }
49     }
50 }
```

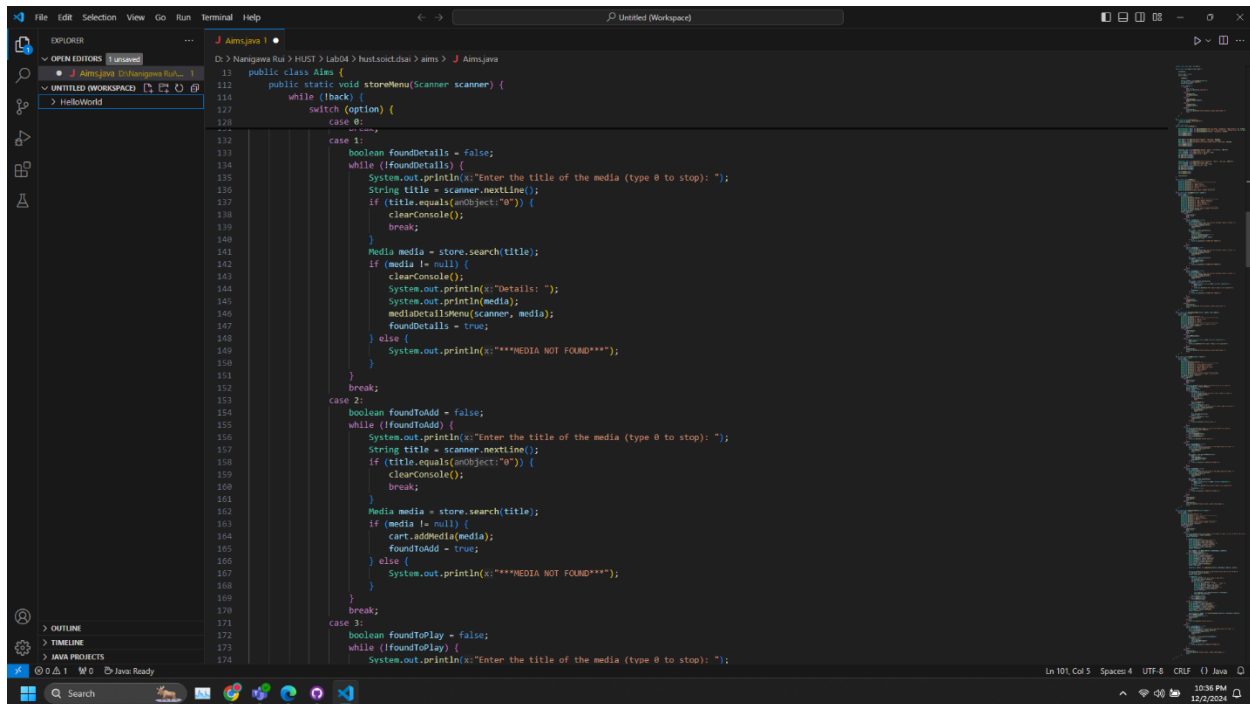


This screenshot shows the completed Aims.java file. It includes the initialization of the store with various media items (DVDs, books, and compact discs) and tracks, and the implementation of the menu options to interact with these items.

```
13 public class Aims {
14     public static void main(String[] args) {
15         while (!exit) {
16             switch (option) {
17                 case 0:
18                     exit = true;
19                     System.out.println("Good bye!");
20                     break;
21                 case 1:
22                     clearConsole();
23                     storeMenu(scanner);
24                     break;
25                 case 2:
26                     clearConsole();
27                     updateStoreMenu(scanner);
28                     break;
29                 case 3:
30                     clearConsole();
31                     cartMenu(scanner);
32                     break;
33                 default:
34                     clearConsole();
35                     System.out.println("Invalid option, please choose again.");
36                     break;
37             }
38         }
39     }
40
41     public static void clearConsole() {
42         System.out.print("\033[H\033[2J");
43         System.out.flush();
44     }
45
46     // Init store setup
47     public static void initSetup() {
48         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
49         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
50         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
51         store.addMedia(dvd1);
52         store.addMedia(dvd2);
53         store.addMedia(dvd3);
54
55         Book book1 = new Book("The Silent Patient", "Thriller", 180.00f);
56         Book book2 = new Book("Educated: A Memoir", "Biography", 250.00f);
57         Book book3 = new Book("The Subtle Art of Not Giving a F*ck", "Self-help", 180.00f);
58         store.addMedia(book1);
59         store.addMedia(book2);
60         store.addMedia(book3);
61
62         CompactDisc cd1 = new CompactDisc("Divide", "Music", "Ed Sheeran", 1200.50f);
63         Track track1CD1 = new Track("Shape of You", 263);
64         Track track2CD1 = new Track("Castle on the Hill", 261);
65         Track track3CD1 = new Track("Perfect", 263);
66         cd1.addTrack(track1CD1);
67         cd1.addTrack(track2CD1);
68         cd1.addTrack(track3CD1);
69     }
70 }
```



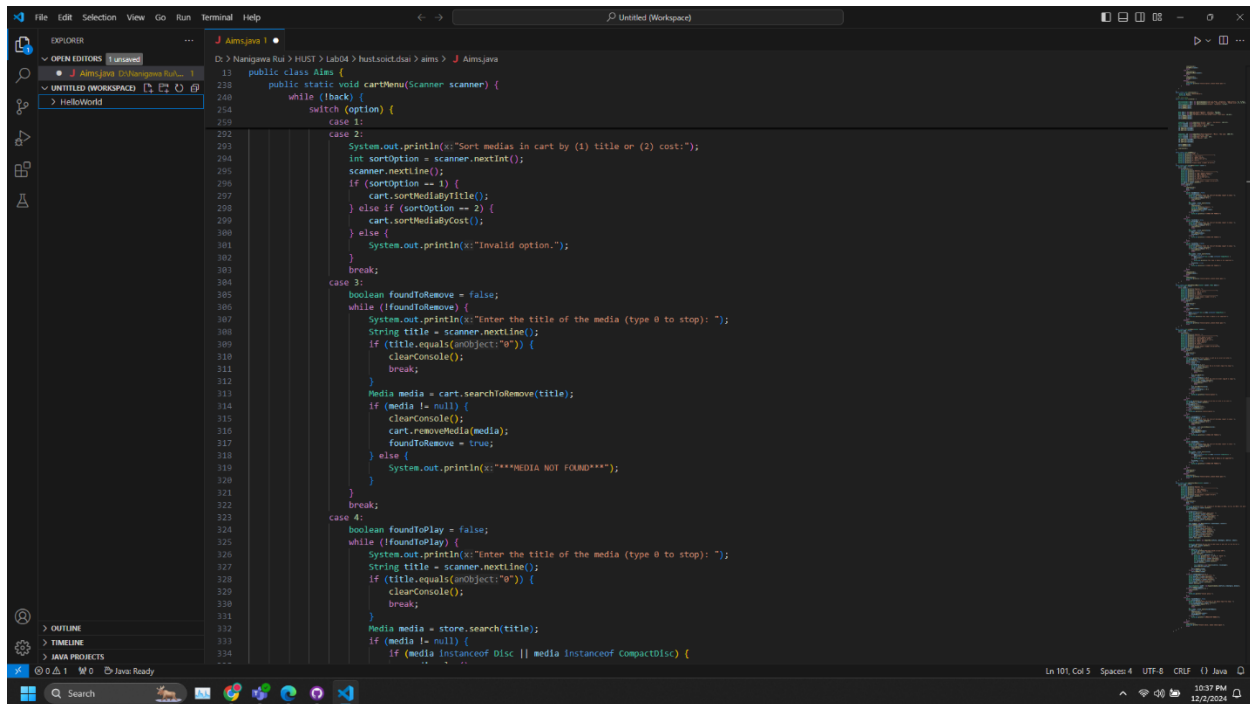
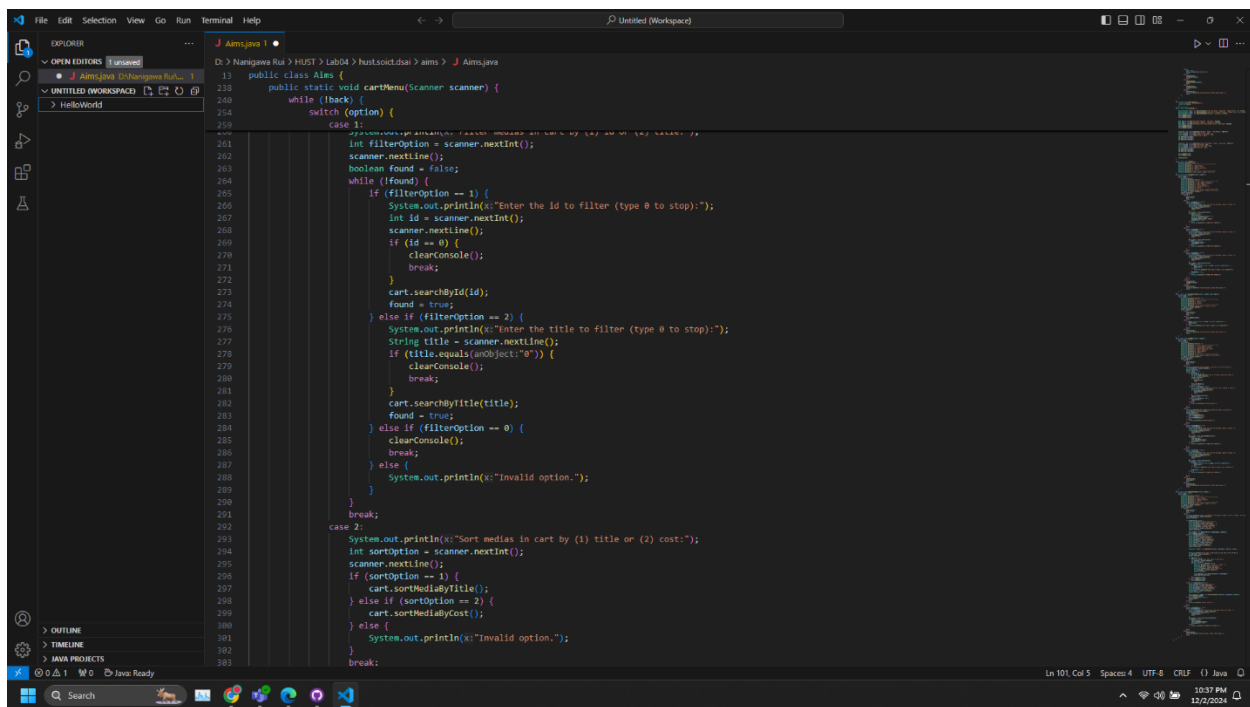
```
13 public class Aims {
61     public static void initSetup() {
87
88         CompactDisc cd2 = new CompactDisc("Future Nostalgia", "Music", "Dua Lipa", 1600, 75f);
89         Track track1CD2 = new Track("Levitating", 209);
90         Track track2CD2 = new Track("Don't Start Now", 183);
91         Track track3CD2 = new Track("Physical", 191);
92         cd2.addTrack(track1CD2);
93         cd2.addTrack(track2CD2);
94         cd2.addTrack(track3CD2);
95
96         store.addMedia(cd1);
97         store.addMedia(cd2);
98
99         clearConsole();
100     }
101
102     public static void showMenu() {
103         System.out.println("AIMS: ");
104         System.out.println("-----");
105         System.out.println("1. View store");
106         System.out.println("2. Update store");
107         System.out.println("3. See current cart");
108         System.out.println("0. Exit");
109         System.out.println("-----");
110         System.out.println("Please choose a number: 0-1-2-3");
111     }
112     public static void storeMenu(Scanner scanner) {
113         boolean back = false;
114         while (!back) {
115             store.printStore();
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             int option = scanner.nextInt();
126             scanner.nextLine();
127             switch (option) {
128                 case 0:
129                     clearConsole();
130                     back = true;
131                     break;
132                 case 1:
```



```
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 = scanner.nextLine();
138                     if (title.equals("0")) {
139                         clearConsole();
140                         break;
141                     }
142                     Media media = store.search(title);
143                     if (media != null) {
144                         clearConsole();
145                         System.out.println("Details: ");
146                         System.out.println(media);
147                         mediaDetailsMenu(scanner, media);
148                         foundDetails = true;
149                     } else {
150                         System.out.println("****MEDIA NOT FOUND****");
151                     }
152                 }
153             case 2:
154                 boolean foundToAdd = false;
155                 while (!foundToAdd) {
156                     System.out.println("Enter the title of the media (type 0 to stop): ");
157                     String title = scanner.nextLine();
158                     if (title.equals("0")) {
159                         clearConsole();
160                         break;
161                     }
162                     Media media = store.search(title);
163                     if (media != null) {
164                         cart.addMedia(media);
165                         foundToAdd = true;
166                     } else {
167                         System.out.println("****MEDIA NOT FOUND****");
168                     }
169                 }
170             case 3:
171                 boolean foundToPlay = false;
172                 while (!foundToPlay) {
173                     System.out.println("Enter the title of the media (type 0 to stop): ");
174                     String title = scanner.nextLine();
175                     if (title.equals("0")) {
176                         clearConsole();
177                         break;
178                     }
179                     Media media = store.search(title);
180                     if (media != null) {
181                         media.play();
182                         foundToPlay = true;
183                     } else {
184                         System.out.println("****MEDIA NOT FOUND****");
185                     }
186                 }
187             case 4:
188                 boolean foundToClear = false;
189                 while (!foundToClear) {
190                     System.out.println("Enter the title of the media (type 0 to stop): ");
191                     String title = scanner.nextLine();
192                     if (title.equals("0")) {
193                         clearConsole();
194                         break;
195                     }
196                     Media media = store.search(title);
197                     if (media != null) {
198                         cart.removeMedia(media);
199                         foundToClear = true;
200                     } else {
201                         System.out.println("****MEDIA NOT FOUND****");
202                     }
203                 }
204             case 0:
205                 back = true;
206                 break;
207             default:
208                 System.out.println("Invalid option. Please choose a number from 0-4.");
209                 scanner.nextLine();
210             }
211         }
212     }
213 }
```

```
13 public class Aims {
14     public static void storeMenu(Scanner scanner) {
15         while (!back) {
16             switch (option) {
17                 case 1:
18                     boolean foundToPlay = false;
19                     while (!foundToPlay) {
20                         System.out.println("Enter the title of the media (type 0 to stop): ");
21                         String title = scanner.nextLine();
22                         if (title.equals("0")) {
23                             clearConsole();
24                             break;
25                         }
26                         Media media = store.search(title);
27                         if (media != null) {
28                             if (media instanceof Disc || media instanceof CompactDisc) {
29                                 media.play();
30                             } else {
31                                 System.out.println("This type of media is not supported!");
32                             }
33                             foundToPlay = true;
34                         } else {
35                             System.out.println("****MEDIA NOT FOUND****");
36                         }
37                     }
38                 case 2:
39                     clearConsole();
40                     cartMenu(scanner);
41                     break;
42                 default:
43                     clearConsole();
44                     System.out.println("Invalid option, please choose again.");
45                     break;
46             }
47         }
48     }
49     public static void mediaDetailMenu(Scanner scanner, Media media) {
50         boolean back = false;
51         while (!back) {
52             System.out.println("Options: ");
53             System.out.println("-----");
54             System.out.println("1. Add to cart");
55             System.out.println("2. Play");
56             System.out.println("0. Back");
57             System.out.println("-----");
58             System.out.println("Please choose a number: 0-1-2");
59         }
60     }
61 }
```

```
217     case 0:
218         clearConsole();
219         back = true;
220         break;
221     case 1:
222         cart.addMedia(media);
223         break;
224     case 2:
225         if (media instanceof Disc || media instanceof CompactDisc) {
226             media.play();
227         } else {
228             System.out.println("This type of media is not supported!");
229         }
230         break;
231     default:
232         clearConsole();
233         System.out.println("Invalid option, please choose again.");
234         break;
235     }
236 }
237
238 public static void cartMenu(Scanner scanner) {
239     boolean back = false;
240     while (!back) {
241         cart.print();
242         System.out.println("Options: ");
243         System.out.println("-----");
244         System.out.println("1. Filter medias in cart");
245         System.out.println("2. Sort medias in cart");
246         System.out.println("3. Remove media from cart");
247         System.out.println("4. Play a media");
248         System.out.println("5. Place order");
249         System.out.println("0. Back");
250         System.out.println("-----");
251         System.out.println("Please choose a number: 0-1-2-3-4-5");
252         int option = scanner.nextInt();
253         scanner.nextLine();
254         switch (option) {
255             case 0:
256                 clearConsole();
257                 back = true;
258                 break;
259             case 1:
260                 System.out.println("Filter medias in cart by (1) id or (2) title");
261                 int filterOption = scanner.nextInt();
262             }
263     }
264 }
```

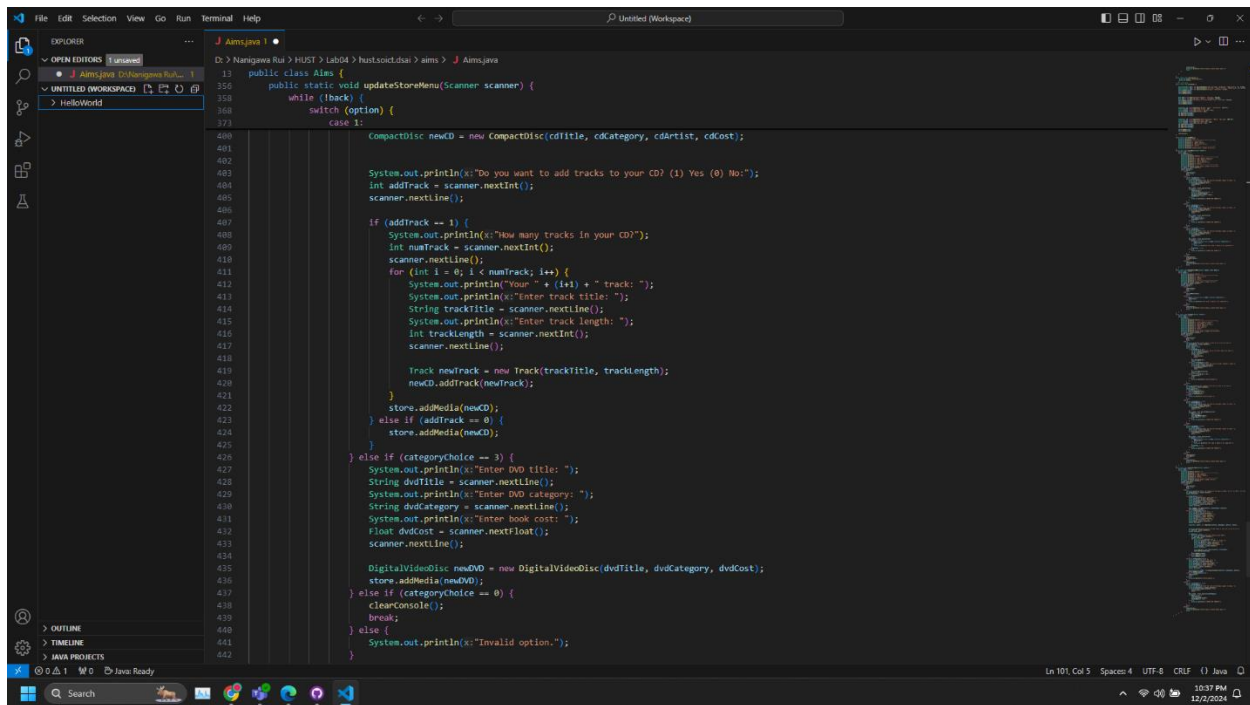


```
File Edit Selection View Go Run Terminal Help
D:\Narigawa Rai > HUST > Lab04 > hustscid.dai > ams > J Ams.java

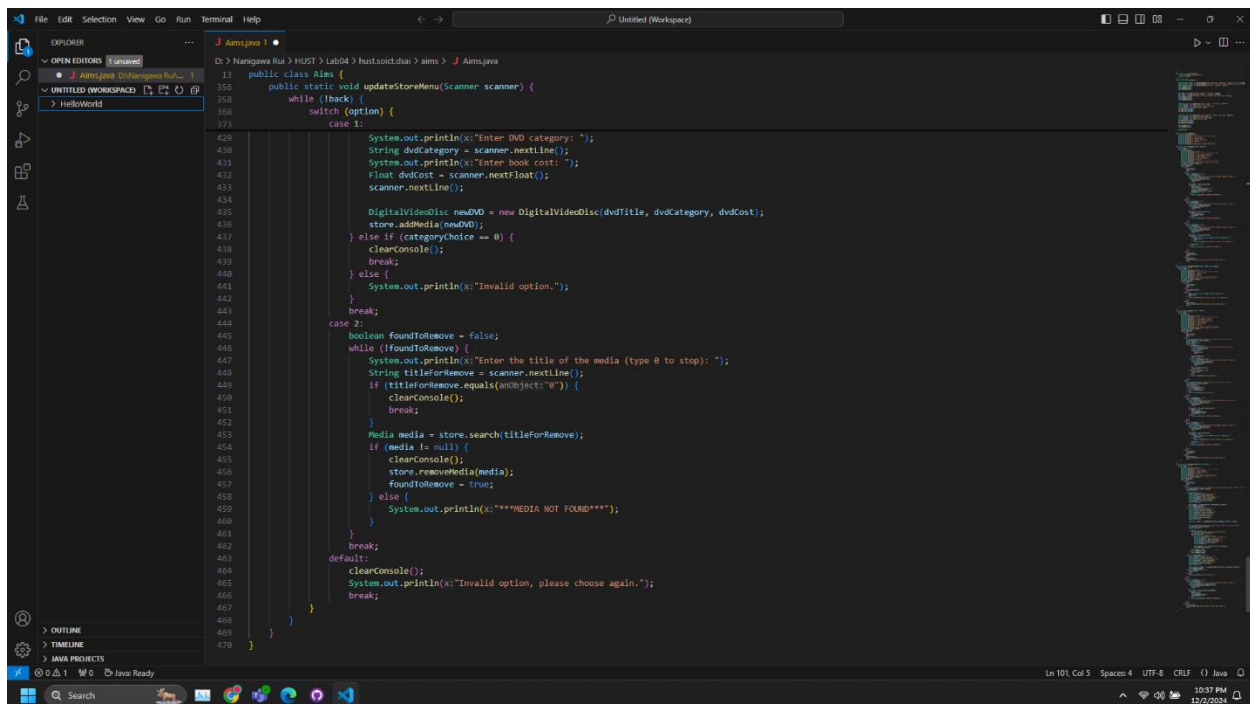
13 public class Ams {
238 public static void carMenu(Scanner scanner) {
240 while (!back) {
254 switch (option) {
321 case 3:
322 }
323 break;
324 case 4:
325 boolean foundToPlay = false;
326 while (!foundToPlay) {
327 System.out.println("Enter the title of the media (type 0 to stop): ");
328 String title = scanner.nextLine();
329 if (title.equals("0")) {
330 clearConsole();
331 break;
332 }
333 Media media = store.search(title);
334 if (media != null) {
335 if (media instanceof Disc || media instanceof CompactDisc) {
336 media.play();
337 } else {
338 System.out.println("This type of media is not supported!");
339 }
340 foundToPlay = true;
341 } else {
342 System.out.println("****MEDIA NOT FOUND****");
343 }
344 break;
345 case 5:
346 clearConsole();
347 cart.empty();
348 break;
349 default:
350 clearConsole();
351 System.out.println("Invalid option, please choose again.");
352 break;
353 }
354 }
355 }
356 public static void updateStoreMenu(Scanner scanner) {
357 boolean back = false;
358 while (!back) {
359 System.out.println("Options: ");
360 System.out.println("-----");
361 System.out.println("1. Add a media");
362 System.out.println("2. Remove a media");
363 System.out.println("0. Back");
364 System.out.println("-----");
365 System.out.println("Please choose a number: 0-2");
366 int option = scanner.nextInt();
367 scanner.nextLine();
368 switch (option) {
369 case 0:
370 clearConsole();
371 back = true;
372 break;
373 case 1:
374 System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
375 int categoryChoice = scanner.nextInt();
376 scanner.nextLine();
377 if (categoryChoice == 1) {
378 System.out.println("Enter book title: ");
379 String bookTitle = scanner.nextLine();
380 System.out.println("Enter book category: ");
381 String bookCategory = scanner.nextLine();
382 System.out.println("Enter book cost: ");
383 Float bookCost = scanner.nextFloat();
384 scanner.nextLine();
385 Book newBook = new Book(bookTitle, bookCategory, bookCost);
386 store.addMedia(newBook);
387 } else if (categoryChoice == 2) {
388 System.out.println("Enter title: ");
389 String cdTitle = scanner.nextLine();
390 System.out.println("Enter category: ");
391 String cdCategory = scanner.nextLine();
392 System.out.println("Enter artist: ");
393 String cdArtist = scanner.nextLine();
394 System.out.println("Enter cost: ");
395 Float cdCost = scanner.nextFloat();
396 scanner.nextLine();
397 CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);
400 }
```

```
File Edit Selection View Go Run Terminal Help
D:\Narigawa Rai > HUST > Lab04 > hustscid.dai > ams > J Ams.java

13 public class Ams {
238 public static void carMenu(Scanner scanner) {
355 }
356 public static void updateStoreMenu(Scanner scanner) {
357 boolean back = false;
358 while (!back) {
359 System.out.println("Options: ");
360 System.out.println("-----");
361 System.out.println("1. Add a media");
362 System.out.println("2. Remove a media");
363 System.out.println("0. Back");
364 System.out.println("-----");
365 System.out.println("Please choose a number: 0-2");
366 int option = scanner.nextInt();
367 scanner.nextLine();
368 switch (option) {
369 case 0:
370 clearConsole();
371 back = true;
372 break;
373 case 1:
374 System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
375 int categoryChoice = scanner.nextInt();
376 scanner.nextLine();
377 if (categoryChoice == 1) {
378 System.out.println("Enter book title: ");
379 String bookTitle = scanner.nextLine();
380 System.out.println("Enter book category: ");
381 String bookCategory = scanner.nextLine();
382 System.out.println("Enter book cost: ");
383 Float bookCost = scanner.nextFloat();
384 scanner.nextLine();
385 Book newBook = new Book(bookTitle, bookCategory, bookCost);
386 store.addMedia(newBook);
387 } else if (categoryChoice == 2) {
388 System.out.println("Enter title: ");
389 String cdTitle = scanner.nextLine();
390 System.out.println("Enter category: ");
391 String cdCategory = scanner.nextLine();
392 System.out.println("Enter artist: ");
393 String cdArtist = scanner.nextLine();
394 System.out.println("Enter cost: ");
395 Float cdCost = scanner.nextFloat();
396 scanner.nextLine();
397 CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);
400 }
```

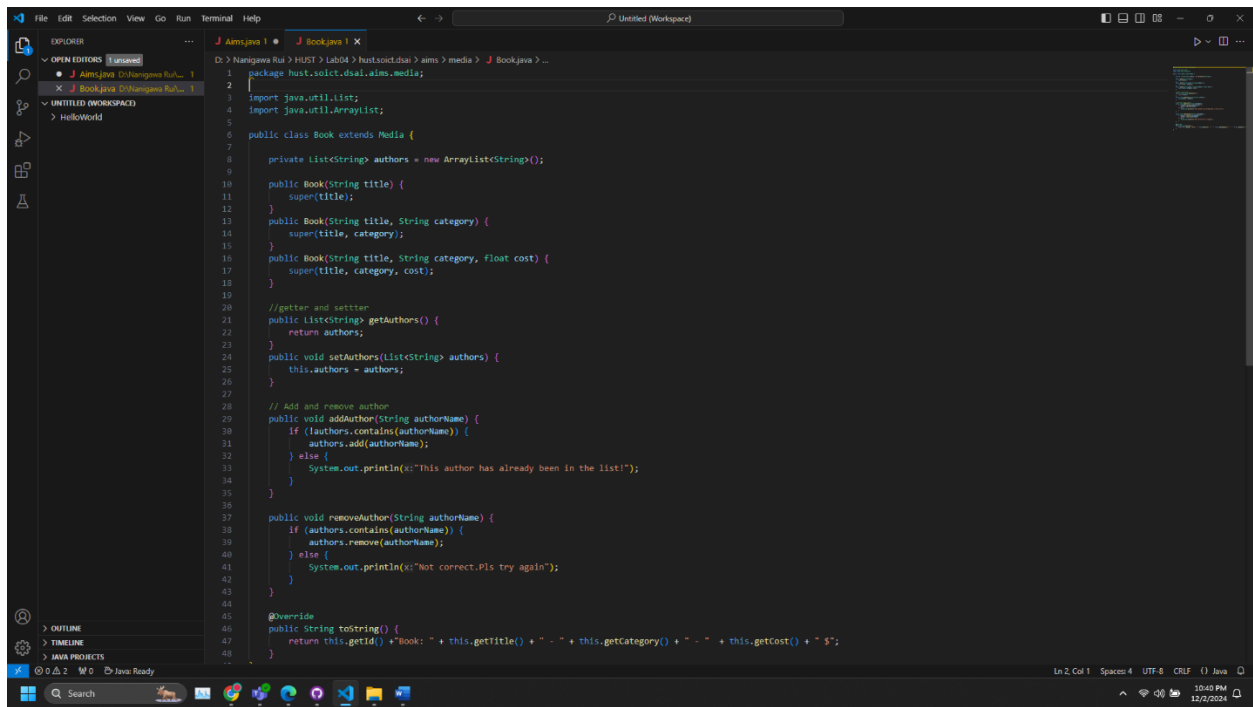


```
13 public class Aims {
14     public static void updateStoreMenu(Scanner scanner) {
15         while (!back) {
16             switch (option) {
17                 case 1:
18                     CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);
19
20                     System.out.println("Do you want to add tracks to your CD? (1) Yes (0) No:");
21                     int numTrack = scanner.nextInt();
22                     scanner.nextLine();
23
24                     if (addTrack == 1) {
25                         System.out.println("How many tracks in your CD?");
26                         int numTrack = scanner.nextInt();
27                         scanner.nextLine();
28                         for (int i = 0; i < numTrack; i++) {
29                             System.out.println("Your " + (i+1) + " track: ");
30                             String trackTitle = scanner.nextLine();
31                             System.out.println("Enter track length: ");
32                             int trackLength = scanner.nextInt();
33                             scanner.nextLine();
34
35                             Track newTrack = new Track(trackTitle, trackLength);
36                             newCD.addTrack(newTrack);
37                         }
38                         store.addMedia(newCD);
39                     } else if (addTrack == 0) {
40                         store.addMedia(newCD);
41                     }
42
43                     } else if (categoryChoice == 1) {
44                         System.out.println("Enter DVD title: ");
45                         String dvdTitle = scanner.nextLine();
46                         System.out.println("Enter DVD category: ");
47                         String dvdCategory = scanner.nextLine();
48                         System.out.println("Enter book cost: ");
49                         Float dvdCost = scanner.nextFloat();
50                         scanner.nextLine();
51
52                         DigitalVideoDisc newDVD = new DigitalVideoDisc(dvdTitle, dvdCategory, dvdCost);
53                         store.addMedia(newDVD);
54                     } else if (categoryChoice == 0) {
55                         clearConsole();
56                         break;
57                     } else {
58                         System.out.println("Invalid option.");
59                     }
60             }
61         }
62     }
63 }
```



```
13 public class Aims {
14     public static void updateStoreMenu(Scanner scanner) {
15         while (!back) {
16             switch (option) {
17                 case 1:
18                     System.out.println("Enter DVD category: ");
19                     String dvdCategory = scanner.nextLine();
20                     System.out.println("Enter book cost: ");
21                     Float dvdCost = scanner.nextFloat();
22                     scanner.nextLine();
23
24                     DigitalVideoDisc newDVD = new DigitalVideoDisc(dvdTitle, dvdCategory, dvdCost);
25                     store.addMedia(newDVD);
26                 } else if (categoryChoice == 0) {
27                     clearConsole();
28                     break;
29                 } else {
30                     System.out.println("Invalid option.");
31                 }
32             }
33         }
34     }
35
36     case 2:
37         boolean foundToRemove = false;
38         while (!foundToRemove) {
39             System.out.println("Enter the title of the media (type 0 to stop): ");
40             String titleToRemove = scanner.nextLine();
41             if (titleToRemove.equals("0")) {
42                 clearConsole();
43                 break;
44             }
45             Media media = store.search(titleToRemove);
46             if (media != null) {
47                 clearConsole();
48                 store.removeMedia(media);
49                 foundToRemove = true;
50             } else {
51                 System.out.println("****MEDIA NOT FOUND****");
52             }
53         }
54         break;
55     default:
56         clearConsole();
57         System.out.println("Invalid option, please choose again.");
58         break;
59     }
60 }
61 }
```

3. Creating book class



The screenshot shows an IDE with a dark theme. The Explorer panel on the left shows a project named 'HelloWorld' with a file 'Book.java'. The main editor displays the following Java code:

```
1 package hust.sict.dsai.aims.media;
2
3 import java.util.List;
4 import java.util.ArrayList;
5
6 public class Book extends Media {
7
8     private List<String> authors = new ArrayList<String>();
9
10    public Book(String title) {
11        super(title);
12    }
13    public Book(String title, String category) {
14        super(title, category);
15    }
16    public Book(String title, String category, float cost) {
17        super(title, category, cost);
18    }
19
20    //getter and setter
21    public List<String> getAuthors() {
22        return authors;
23    }
24    public void setAuthors(List<String> authors) {
25        this.authors = authors;
26    }
27
28    // Add and remove author
29    public void addAuthor(String authorName) {
30        if (!authors.contains(authorName)) {
31            authors.add(authorName);
32        } else {
33            System.out.println("This author has already been in the list!");
34        }
35    }
36
37    public void removeAuthor(String authorName) {
38        if (authors.contains(authorName)) {
39            authors.remove(authorName);
40        } else {
41            System.out.println("Not correct.Pls try again");
42        }
43    }
44
45    @Override
46    public String toString() {
47        return this.getId() + "Book: " + this.getTitle() + " - " + this.getCategory() + " - " + this.getCost() + " $";
48    }
49 }
```

The status bar at the bottom indicates 'Ln:2, Col:1, Spaces:4, UTF-8, CRLF, () Java'.

4. Creating the abstract media class

This screenshot shows the initial implementation of the `Media` class in an IDE. The Explorer panel on the left shows a project named 'hust.socit.dsa1.aims.media' with files 'Aims.java', 'Book.java', and 'Media.java'. The 'Media.java' file is open in the editor, showing the following code:

```
1 package hust.socit.dsa1.aims.media;
2
3 import java.util.Comparator;
4
5 public abstract class Media implements Comparable<Media> {
6
7     public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
8     public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
9
10
11     private static int nbMedia = 0;
12     private int id;
13     private String title;
14     private String category;
15     private float cost;
16
17     // Constructor
18     public Media(String title) {
19         this.title = title;
20         this.id = ++nbMedia;
21     }
22     public Media(String title, String category) {
23         this.title = title;
24         this.category = category;
25         this.id = ++nbMedia;
26     }
27     public Media(String title, String category, float cost) {
28         this.title = title;
29         this.category = category;
30         this.cost = cost;
31         this.id = ++nbMedia;
32     }
33
34     // Getter method
35     public int getId() {
36         return id;
37     }
38     public String getTitle() {
39         return title;
40     }
41     public String getCategory() {
42         return category;
43     }
44     public float getCost() {
45         return cost;
46     }
47
48     // Setter method
```

This screenshot shows the completed implementation of the `Media` class in the same IDE. The code continues from the previous screenshot, adding the following methods:

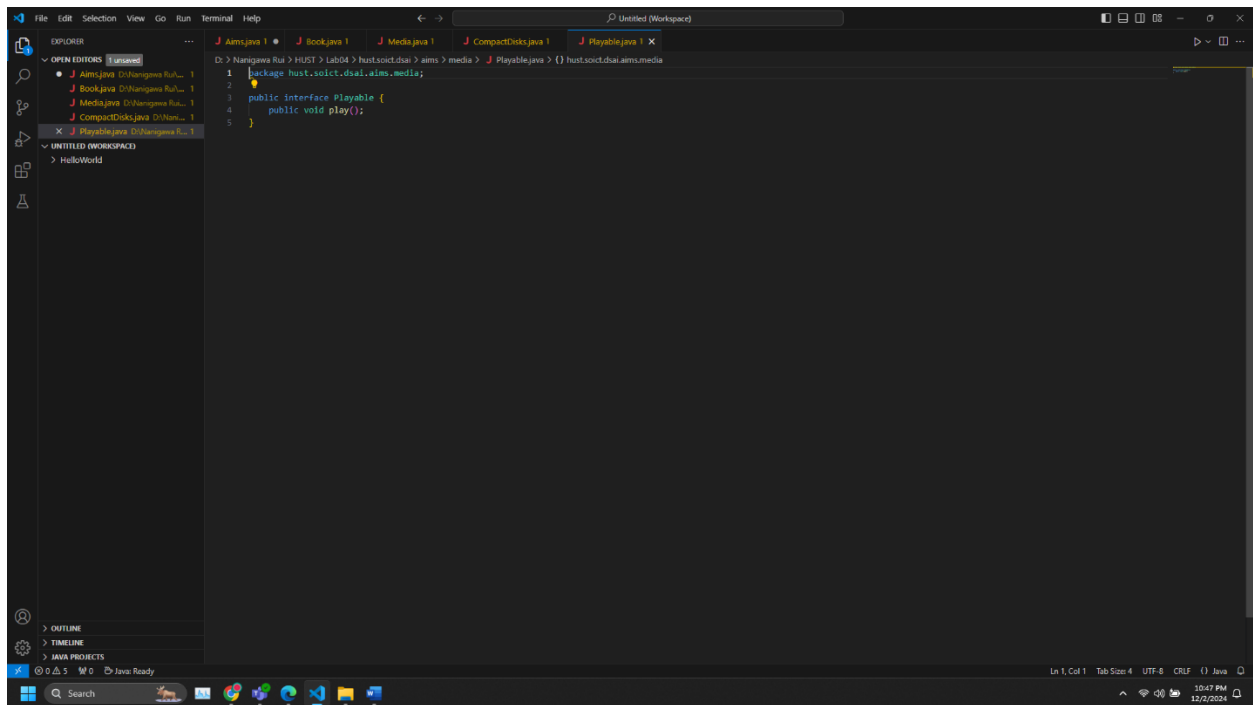
```
49     public String getCategory() {
50         return category;
51     }
52     public float getCost() {
53         return cost;
54     }
55
56     // Setter method
57     public void setTitle(String title) {
58         this.title = title;
59     }
60
61     // Check is title match
62     public boolean isMatch(String title) {
63         return this.getTitle().toLowerCase().contains(title.toLowerCase());
64     }
65
66     public void play() {
67         System.out.println("Playing media");
68     }
69
70     @Override
71     public boolean equals(Object obj) {
72         if (obj == this) {
73             return true;
74         }
75         if (!(obj instanceof Media)) {
76             return false;
77         }
78         return ((Media)obj).getTitle() == this.getTitle();
79     }
80
81     @Override
82     public String toString() {
83         return "Media: " + this.getTitle() +
84             " - Category: " + this.getCategory() +
85             " - Cost: " + this.getCost() + "$";
86     }
87 }
```

5. Creating compact disc class

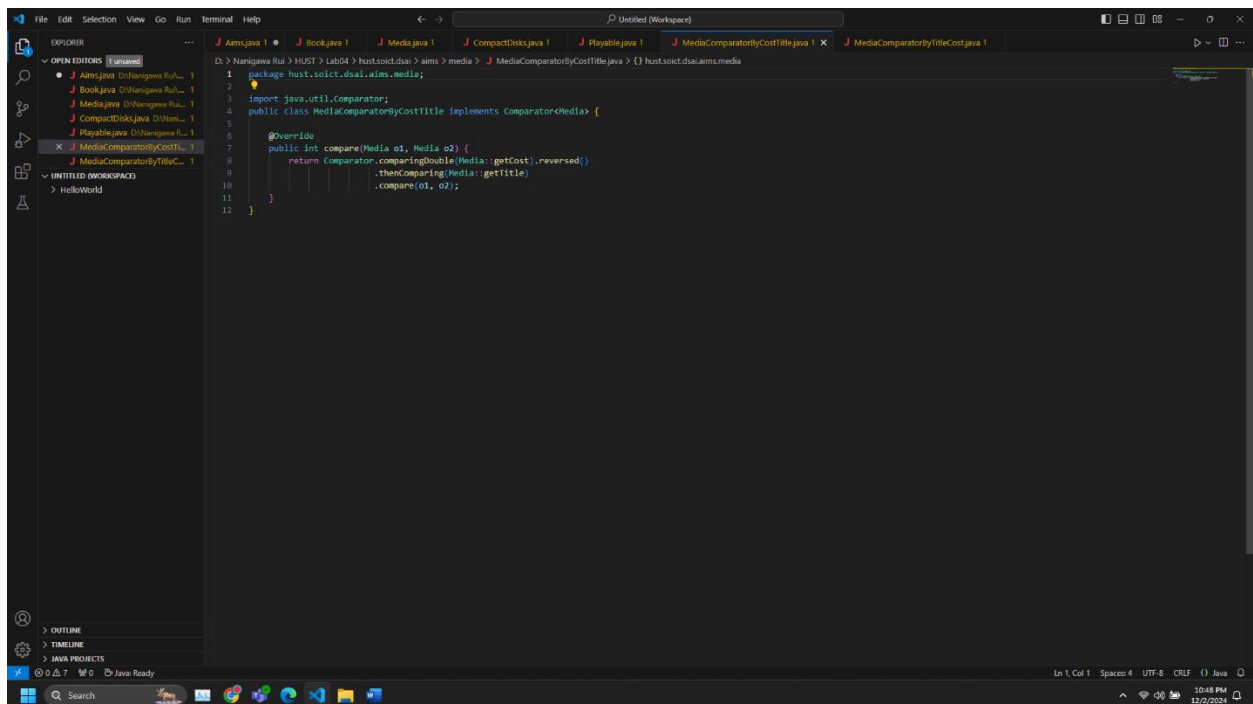

```
1 package hunsoid.dsa1.aims.media;
2 import java.util.ArrayList;
3 import java.util.List;
4
5 public class CompactDisc extends Disc implements Playable {
6
7     private String artist;
8     private List<Track> tracks = new ArrayList<Track>();
9
10    public String getArtist() {
11        return artist;
12    }
13
14    public CompactDisc(String title) {
15        super(title);
16    }
17
18    public CompactDisc(String title, String category, String artist, float cost) {
19        super(title, category, cost);
20        this.artist = artist;
21    }
22
23    // Add and remove track method
24    public void addTrack(Track track) {
25        if (!tracks.contains(track)) {
26            tracks.add(track);
27            System.out.println("Track: " + track.getTitle() + " has been added to CD!");
28        } else {
29            System.out.println("Track already exists in CD.");
30        }
31    }
32
33    public void removeTrack(Track track) {
34        if (tracks.contains(track)) {
35            tracks.remove(track);
36            System.out.println("Track: " + track.getTitle() + " has been removed from CD!");
37        } else {
38            System.out.println("Track does not exist in CD.");
39        }
40    }
41
42    // Get length method
43    public int getLength() {
44        int totalLength = 0;
45        for (Track track : tracks) {
46            totalLength += track.getLength();
47        }
48        return totalLength;
49    }
50 }
```

```
51
52 // Add and remove track method
53 public void addTrack(Track track) {
54     if (!tracks.contains(track)) {
55         tracks.add(track);
56         System.out.println("Track: " + track.getTitle() + " has been added to CD!");
57     } else {
58         System.out.println("Track already exists in CD.");
59     }
60 }
61
62 public void removeTrack(Track track) {
63     if (tracks.contains(track)) {
64         tracks.remove(track);
65         System.out.println("Track: " + track.getTitle() + " has been removed from CD!");
66     } else {
67         System.out.println("Track does not exist in CD.");
68     }
69 }
70
71 // Get length method
72 public int getLength() {
73     int totalLength = 0;
74     for (Track track : tracks) {
75         totalLength += track.getLength();
76     }
77     return totalLength;
78 }
79
80 @Override
81 public void play() {
82     System.out.println("Playing CD: " + this.getTitle());
83     System.out.println("CD length: " + this.getLength());
84     for (Track track : tracks) {
85         track.play();
86     }
87 }
88
89 @Override
90 public String toString() {
91     return this.getId() + " - CD: " + this.getTitle() + " - " + this.getCategory() + " - " +
92         this.getArtist() + " - " + this.getLength() + " - " +
93         this.getCost() + "$";
94 }
95 }
```

6. create playable interface



7. sort media in cart



1. What class should implement the Comparable interface?

- The media class

Reason: It's the base

2. In those classes, how should you implement the compareTo() method to reflect the ordering that we want?

@Override

```
public int compareTo(Media other) {  
    int titleComparison = this.getTitle().compareTo(other.getTitle());  
    if (titleComparison != 0) {  
        return titleComparison;  
    }  
    return Double.compare(this.getCost(), other.getCost());  
}
```

3. 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
- The Comparable interface allows for only one natural ordering.

4. 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?

```
@Override
public int compareTo(Media other) {
    if (other instanceof Disc) {
        Disc otherDVD = (Disc) other;

        int titleComparison =
this.getTitle().compareTo(otherDVD.getTitle());

        if (titleComparison != 0) {
            return titleComparison;
        } else {
            // Compare by decreasing length

            int lengthComparison =
Integer.compare(otherDVD.getLength(), this.getLength());

            if (lengthComparison != 0) {
                return lengthComparison;
            } else {
                return Double.compare(this.getCost(),
otherDVD.getCost());
            }
        }
    } else {
        return super.compareTo(other);
    }
}
```