

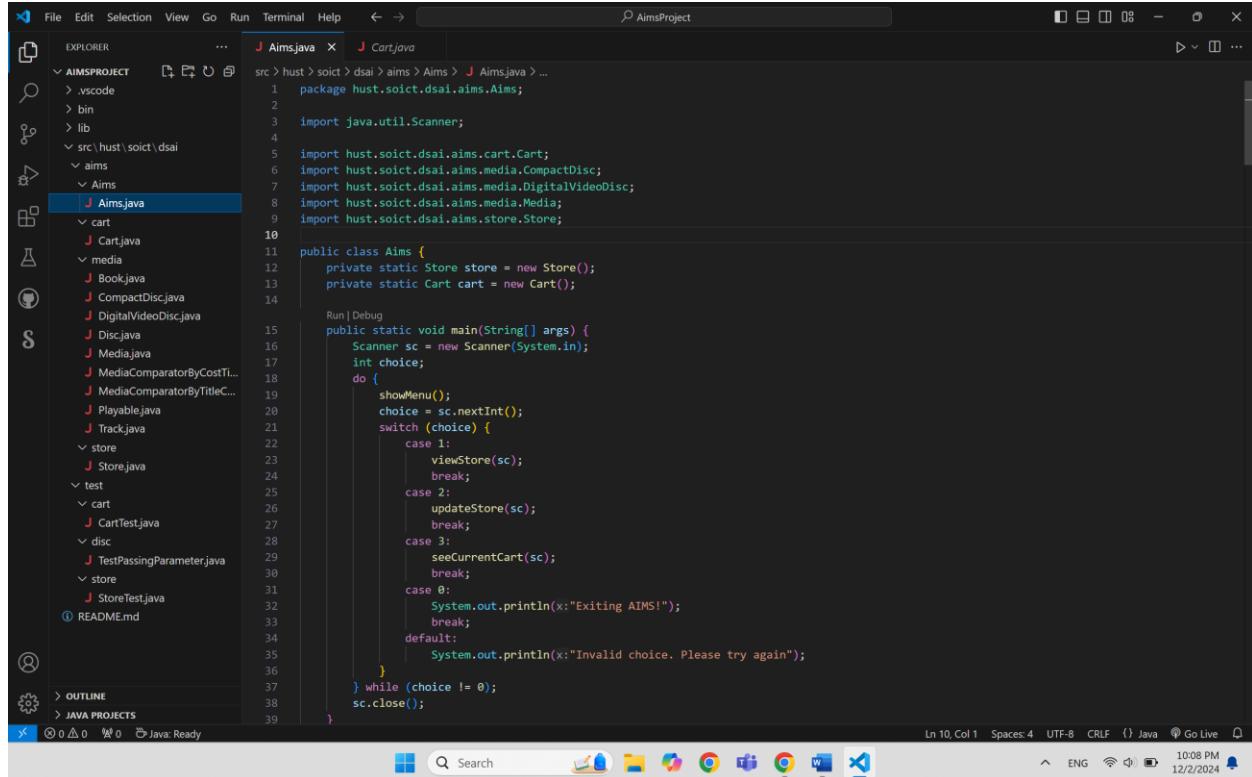
Lab04 Report

Name: Lương Minh Hiếu

ID: 20230083

1. Source Code

1.1. Aims class



```
File Edit Selection View Go Run Terminal Help <- > AimsProject

EXPLORER J Aims.java J Cart.java
AIMSPROJECT .vscode bin lib
src hust soict dsai
  aims
    Aims
      Aims.java
    cart
      Cart.java
    media
      Book.java
      CompactDisc.java
      Disc.java
      Media.java
      MediaComparatorByCost...
      MediaComparatorByTitle...
      Playable.java
      Track.java
    store
      Store.java
    test
      CartTest.java
    disc
      TestPassingParameter.java
    store
      StoreTest.java
 README.md

OUTLINE JAVA PROJECTS

Aims.java

1 package hust.soict.dsai.aims.Aims;
2
3 import java.util.Scanner;
4
5 import hust.soict.dsai.aims.cart.Cart;
6 import hust.soict.dsai.aims.media.CompactDisc;
7 import hust.soict.dsai.aims.media.DigitalVideoDisc;
8 import hust.soict.dsai.aims.media.Media;
9 import hust.soict.dsai.aims.store.Store;
10
11 public class Aims {
12     private static Store store = new Store();
13     private static Cart cart = new Cart();
14
15     Run|Debug
16     public static void main(String[] args) {
17         Scanner sc = new Scanner(System.in);
18         int choice;
19         do {
20             showMenu();
21             choice = sc.nextInt();
22             switch (choice) {
23                 case 1:
24                     viewStore(sc);
25                     break;
26                 case 2:
27                     updateStore(sc);
28                     break;
29                 case 3:
30                     seeCurrentCart(sc);
31                     break;
32                 case 0:
33                     System.out.println("Exiting AIMS!");
34                     break;
35                 default:
36                     System.out.println("Invalid choice. Please try again");
37             }
38         } while (choice != 0);
39         sc.close();
}

Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Java ENG 10:08 PM 12/2/2024 Go Live
```

A screenshot of the Visual Studio Code interface. The title bar says "AimsProject". The left sidebar shows a project structure under "AIMSPROJECT": .vscode, bin, src (containing hust, soict, dsai, aims), and a selected file "Aims.java". The main editor tab is "Aims.java". The code in the editor is:

```
public static void showMenu() {
    System.out.println("AIMS:");
    System.out.println("-----");
    System.out.println("1. View store");
    System.out.println("2. Update store");
    System.out.println("3. See current cart");
    System.out.println("0. Exit");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2-3");
}

public static void storeMenu() {
    System.out.println("Options:");
    System.out.println("-----");
    System.out.println("1. See a media's details");
    System.out.println("2. Add a media to cart");
    System.out.println("3. Play a media");
    System.out.println("4. See current cart");
    System.out.println("0. Back");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2-3-4");
}

public static void mediaDetailsMenu() {
    System.out.println("Options:");
    System.out.println("-----");
    System.out.println("1. Add to cart");
    System.out.println("2. Play");
    System.out.println("0. Back");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2");
}

public static void cartMenu() {
    System.out.println("Options:");
    System.out.println("-----");
    System.out.println("1. Filter medias in cart");
    System.out.println("2. Sort medias in cart");
    System.out.println("3. Remove media from cart");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2-3-4-5");
}
```

The status bar at the bottom right shows "Ln 10, Col 1" and "10:09 PM 12/2/2024".

A screenshot of the Visual Studio Code interface, continuing from the previous one. The title bar says "AimsProject". The left sidebar shows the same project structure. The main editor tab is "Aims.java". The code in the editor continues from the previous snippet:

```
System.out.println("4. Play a media");
System.out.println("5. Place order");
System.out.println("0. Back");
System.out.println("-----");
System.out.println("Please choose a number: 0-1-2-3-4-5");

public static void viewStore(Scanner sc) {
    store.displayStore();
    int choice;
    do {
        storeMenu();
        choice = sc.nextInt();
        sc.nextLine();
        switch (choice) {
            case 1:
                seeMediaDetail(sc);
                break;
            case 2:
                addMediaToCart(sc);
                break;
            case 3:
                playMedia(sc);
                break;
            case 4:
                seeCurrentCart(sc);
                break;
            case 0:
                break;
            default:
                System.out.println("Invalid choice. Please try again");
        }
    } while (choice != 0);
}

public static void seeMediaDetail(Scanner sc) {
    System.out.println("Please enter the title of the media");
    String title = sc.nextLine();
    Media media = store.getMedia(title);
    if (media != null) {
```

The status bar at the bottom right shows "Ln 10, Col 1" and "10:09 PM 12/2/2024".

The screenshot shows a Java IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AimsProject
- Toolbars:** Standard toolbar icons.
- Left Sidebar:** Explorer view showing the project structure:
 - AIMSPROJECT
 - .vscode
 - bin
 - lib
 - src/hust/soict/dsai
 - aims
 - Aims
 - J Aims.java (selected)
 - cart
 - J Cart.java
 - media
 - J Book.java
 - J CompactDisc.java
 - J DigitalVideoDisc.java
 - J Disc.java
 - J Media.java
 - J MediaComparatorByCostTi...
 - J MediaComparatorByTitleC...
 - J Playable.java
 - J Track.java
 - store
 - J Store.java
 - test
 - cart
 - J CartTest.java
 - disc
 - J TestPassingParameter.java
 - store
 - J StoreTest.java
- Central Area:** Code editor showing the content of Aims.java.

```
public static void seeMediaDetail(Scanner sc) {
    System.out.println("Please enter the title of the media");
    String title = sc.nextLine();
    Media media = store.getMedia(title);
    if (media != null) {
        media.toString();
        int choice;
        do {
            mediaDetailsMenu();
            choice = sc.nextInt();
            sc.nextLine();
            switch (choice) {
                case 1:
                    cart.addMedia(media);
                    break;
                case 2:
                    if (media instanceof DigitalVideoDisc) {
                        ((DigitalVideoDisc) media).play();
                    } else if (media instanceof CompactDisc) {
                        ((CompactDisc) media).play();
                    } else {
                        System.out.println("Media cannot be played");
                    }
                    break;
                case 0:
                    break;
                default:
                    System.out.println("Invalid choice. Please try again");
            }
        } while (choice != 0);
    } else {
        System.out.println("Media not found");
    }
}

public static void addMediaToCart(Scanner sc) {
    System.out.println("Please enter the title of the media");
    String title = sc.nextLine();
    Media media = store.getMedia(title);
    if (media != null) {
        cart.addMedia(media);
        cart.displayCart();
    } else {
        System.out.println("Media not found");
    }
}
```
- Bottom Status Bar:** Line 10, Col 1 | Spaces: 4 | UTF-8 | CRLF | Java | Go Live | 10:09 PM | 12/2/2024

The screenshot shows a Java IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AimsProject
- Toolbars:** Standard toolbar icons.
- Left Sidebar:** Explorer view showing the project structure (same as the first screenshot).
- Central Area:** Code editor showing the content of Aims.java with additional methods added.

```
public static void addMediaToCart(Scanner sc) {
    System.out.println("Please enter the title of the media");
    String title = sc.nextLine();
    Media media = store.getMedia(title);
    if (media != null) {
        cart.addMedia(media);
        cart.displayCart();
    } else {
        System.out.println("Media not found");
    }
}

public static void playMedia(Scanner sc) {
    System.out.println("Please enter the title of the media");
    String title = sc.nextLine();
    Media media = store.getMedia(title);
    if (media != null) {
        if (media instanceof DigitalVideoDisc) {
            ((DigitalVideoDisc) media).play();
        } else if (media instanceof CompactDisc) {
            ((CompactDisc) media).play();
        }
    } else {
        System.out.println("Media not found or cannot be played");
    }
}

public static void updateStore(Scanner sc) {
    System.out.println("1. Add a media");
    System.out.println("2. Remove a media");
    int choice = sc.nextInt();
    sc.nextLine();
    if (choice == 1) {
        store.addMedia(sc);
    } else if (choice == 2) {
        System.out.println("Please enter the id of the media");
        int id = sc.nextInt();
        sc.nextLine();
        Media media = store.getMedia(id);
        if (media != null) {
            store.removeMedia(id);
        } else {
            System.out.println("Media not found");
        }
    }
}
```
- Bottom Status Bar:** Line 10, Col 1 | Spaces: 4 | UTF-8 | CRLF | Java | Go Live | 10:10 PM | 12/2/2024

```
src > hust > soict > dsai > aims > Aims > J Aims.java > ...
187     sc.nextLine();
188     Media media = store.getMedia(id);
189     if (media != null) {
190         store.removeMedia(media);
191     } else {
192         System.out.println(x:"Media not found");
193     }
194 } else {
195     System.out.println(x:"Invalid choice");
196 }
197
public static void seeCurrentCart(Scanner sc) {
    cart.displayCart();
    int choice;
    do {
        cartMenu();
        choice = sc.nextInt();
        sc.nextLine();
        switch (choice) {
            case 1:
                System.out.println(x:"1. Filter by id");
                System.out.println(x:"2. Filter by title");
                int option = sc.nextInt();
                if (option == 1) {
                    int id = sc.nextInt();
                    sc.nextLine();
                    cart.searchMedia(id);
                } else if (option == 2) {
                    String title = sc.nextLine();
                    cart.filterMedia(title);
                } else {
                    System.out.println(x:"Invalid choice");
                }
                break;
            case 2:
                System.out.println(x:"1. Sort by tilte");
                System.out.println(x:"2. Sort by cost");
                int option2 = sc.nextInt();
                if (option2 == 1) {
                    cart.sortByTitleCost();
                } else if (option2 == 2) {
                    cart.sortByCostTitle();
                } else {
                    System.out.println(x:"Invalid choice");
                }
                break;
            case 3:
                System.out.println(x:"Please enter the title of the media");
                String title = sc.nextLine();
                Media media = cart.getMedia(title);
                if (media != null) {
                    cart.removeMedia(media);
                } else {
                    System.out.println(x:"Media not found");
                }
                break;
            case 4:
                playMedia(sc);
                break;
            case 5:
                System.out.println(x:"Order created!");
                cart.emptyCart();
                break;
            case 0:
                break;
            default:
                System.out.println(x:"Invalid choice. Please try again");
        }
    } while (choice != 0);
}

```

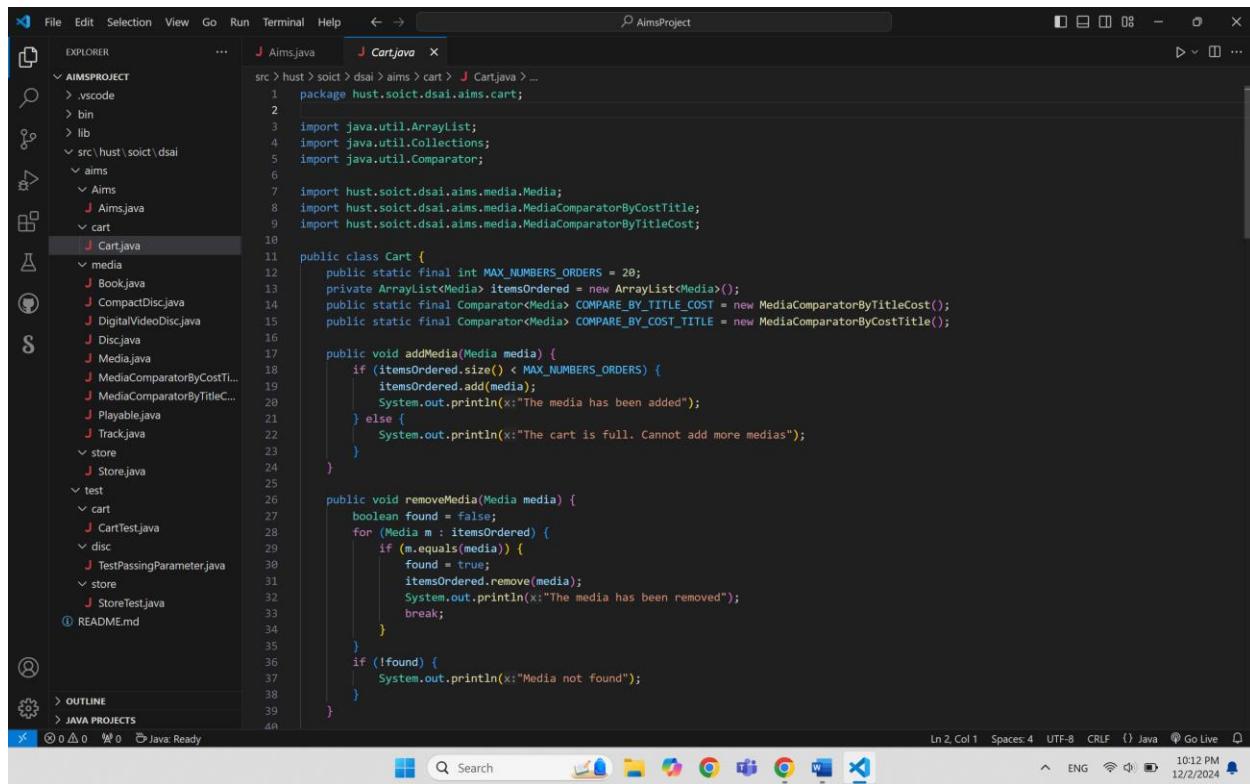
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:10 PM 12/2/2024

```
src > hust > soict > dsai > aims > Aims > J Aims.java > ...
219     System.out.println(x:"Invalid choice");
220 }
221 break;
222 case 2:
223     System.out.println(x:"1. Sort by tilte");
224     System.out.println(x:"2. Sort by cost");
225     int option2 = sc.nextInt();
226     if (option2 == 1) {
227         cart.sortByTitleCost();
228     } else if (option2 == 2) {
229         cart.sortByCostTitle();
230     } else {
231         System.out.println(x:"Invalid choice");
232     }
233 break;
234 case 3:
235     System.out.println(x:"Please enter the title of the media");
236     String title = sc.nextLine();
237     Media media = cart.getMedia(title);
238     if (media != null) {
239         cart.removeMedia(media);
240     } else {
241         System.out.println(x:"Media not found");
242     }
243 break;
244 case 4:
245     playMedia(sc);
246     break;
247 case 5:
248     System.out.println(x:"Order created!");
249     cart.emptyCart();
250     break;
251 case 0:
252     break;
253 default:
254     System.out.println(x:"Invalid choice. Please try again");
255 }
256 }
257 } while (choice != 0);
}

```

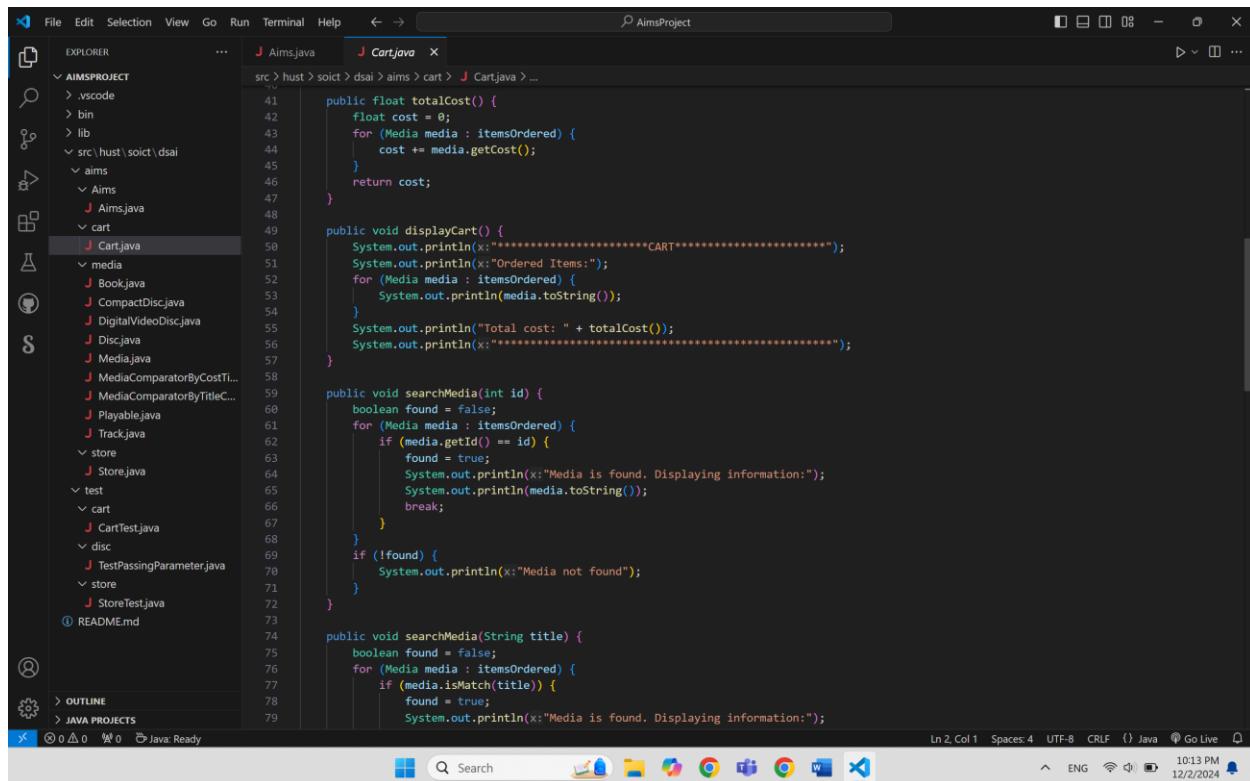
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:10 PM 12/2/2024

1.2. Cart class



```
File Edit Selection View Go Run Terminal Help <- > AimsProject
EXPLORER J Aims.java J Cart.java ...
src > hust > soict > dsai > aims > cart > J Cart.java > ...
1 package hust.soict.dsai.aims.cart;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.Comparator;
6
7 import hust.soict.dsai.aims.media.Media;
8 import hust.soict.dsai.aims.media.MediaComparatorByCostTitle;
9 import hust.soict.dsai.aims.media.MediaComparatorByTitleCost;
10
11 public class Cart {
12     public static final int MAX_NUMBERS_ORDERS = 20;
13     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
14     public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
15     public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
16
17     public void addMedia(Media media) {
18         if (itemsOrdered.size() < MAX_NUMBERS_ORDERS) {
19             itemsOrdered.add(media);
20             System.out.println("The media has been added");
21         } else {
22             System.out.println("The cart is full. Cannot add more medias");
23         }
24     }
25
26     public void removeMedia(Media media) {
27         boolean found = false;
28         for (Media m : itemsOrdered) {
29             if (m.equals(media)) {
30                 found = true;
31                 itemsOrdered.remove(media);
32                 System.out.println("The media has been removed");
33                 break;
34             }
35         }
36         if (!found) {
37             System.out.println("Media not found");
38         }
39     }
40 }
```

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:12 PM 12/2/2024



```
File Edit Selection View Go Run Terminal Help <- > AimsProject
EXPLORER J Aims.java J Cart.java ...
src > hust > soict > dsai > aims > cart > J Cart.java > ...
41     public float totalCost() {
42         float cost = 0;
43         for (Media media : itemsOrdered) {
44             cost += media.getCost();
45         }
46         return cost;
47     }
48
49     public void displayCart() {
50         System.out.println("*****CART*****");
51         System.out.println("Ordered Items:");
52         for (Media media : itemsOrdered) {
53             System.out.println(media.toString());
54         }
55         System.out.println("Total cost: " + totalCost());
56         System.out.println("*****");
57     }
58
59     public void searchMedia(int id) {
60         boolean found = false;
61         for (Media media : itemsOrdered) {
62             if (media.getId() == id) {
63                 found = true;
64                 System.out.println("Media is found. Displaying information:");
65                 System.out.println(media.toString());
66                 break;
67             }
68         }
69         if (!found) {
70             System.out.println("Media not found");
71         }
72     }
73
74     public void searchMedia(String title) {
75         boolean found = false;
76         for (Media media : itemsOrdered) {
77             if (media.isMatch(title)) {
78                 found = true;
79                 System.out.println("Media is found. Displaying information:");
80             }
81         }
82     }
83 }
```

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:13 PM 12/2/2024

A screenshot of the Visual Studio Code interface. The title bar says "AimsProject". The left sidebar shows a project structure under "AIMSPROJECT" with files like .vscode, bin, lib, src, aims, Aims.java, cart, media, Book.java, CompactDisc.java, DigitalVideoDisc.java, Disc.java, Media.java, MediaComparatorByCostTitle..., MediaComparatorByTitleC..., Playable.java, Track.java, store, CartTest.java, TestPassingParameter.java, StoreTest.java, and README.md. The "Cart.java" file is selected in the Explorer view. The main editor area displays the Java code for "Cart.java". The status bar at the bottom right shows "Ln 2, Col 1" and "10:13 PM 12/2/2024".

```
src > hust > soict > dsai > aims > cart > Cart.java > ...
80     System.out.println(media.toString());
81 }
82 }
83 }
84 if (!found) {
85     System.out.println("Media not found");
86 }
87 }

public Media getMedia(String title) {
    Media result = null;
    for (Media media : itemsOrdered) {
        if (media.isMatch(title)) {
            result = media;
            break;
        }
    }
    return result;
}

public void filterMedia(String title) {
    boolean found = false;
    Cart cart = new Cart();
    for (Media media : itemsOrdered) {
        if (media.isMatch(title)) {
            found = true;
            cart.addMedia(media);
        }
    }
    if (!found) {
        System.out.println("Media not found");
    } else {
        cart.displayCart();
    }
}

public void emptyCart() {
    itemsOrdered.clear();
}

public void sortByTitleCost() {
    Collections.sort(itemsOrdered, COMPARISON_BY_TITLE_COST);
}

public void sortByCostTitle() {
    Collections.sort(itemsOrdered, COMPARISON_BY_COST_TITLE);
}
```

A screenshot of the Visual Studio Code interface. The title bar says "AimsProject". The left sidebar shows a project structure under "AIMSPROJECT" with files like .vscode, bin, lib, src, aims, Aims.java, cart, media, Book.java, CompactDisc.java, DigitalVideoDisc.java, Disc.java, Media.java, MediaComparatorByCostTitle..., MediaComparatorByTitleC..., Playable.java, Track.java, store, CartTest.java, TestPassingParameter.java, StoreTest.java, and README.md. The "Cart.java" file is selected in the Explorer view. The main editor area displays the Java code for "Cart.java". The status bar at the bottom right shows "Ln 2, Col 1" and "10:13 PM 12/2/2024".

```
src > hust > soict > dsai > aims > cart > Cart.java > ...
96 }
97 return result;
98 }

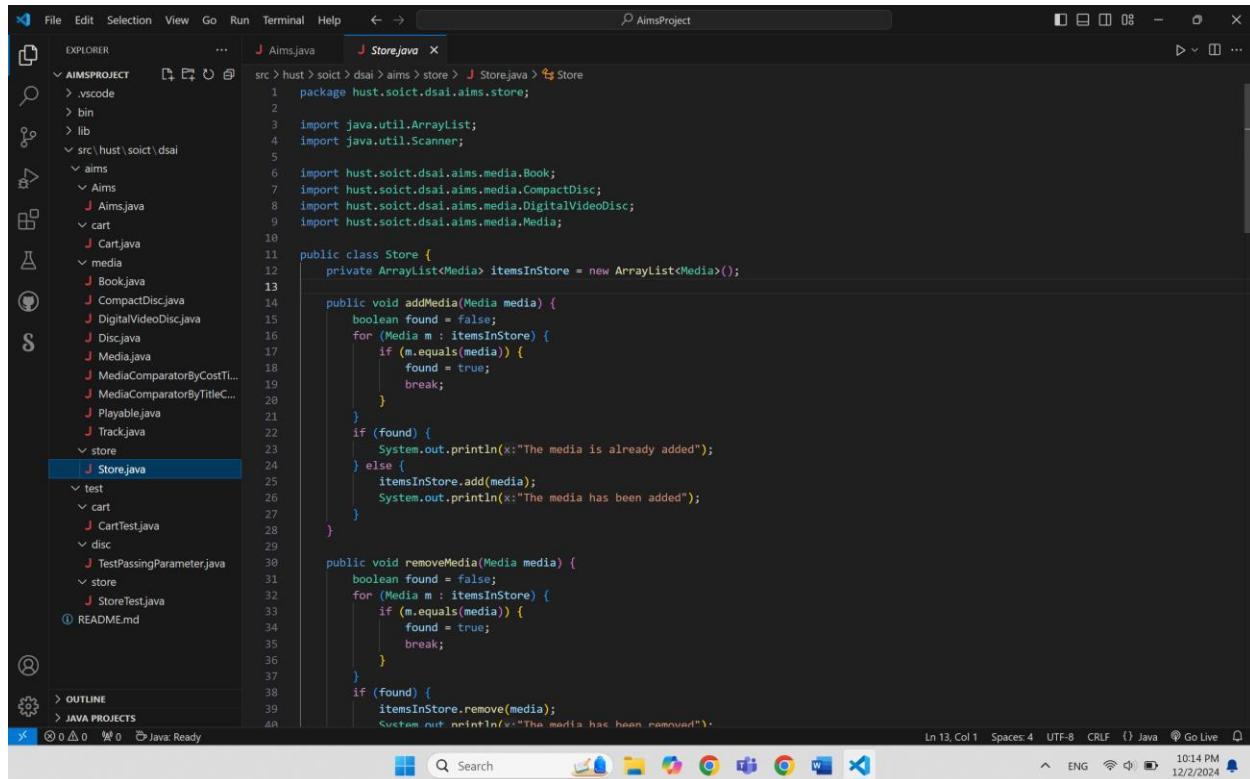
public void filterMedia(String title) {
    boolean found = false;
    Cart cart = new Cart();
    for (Media media : itemsOrdered) {
        if (media.isMatch(title)) {
            found = true;
            cart.addMedia(media);
        }
    }
    if (!found) {
        System.out.println("Media not found");
    } else {
        cart.displayCart();
    }
}

public void emptyCart() {
    itemsOrdered.clear();
}

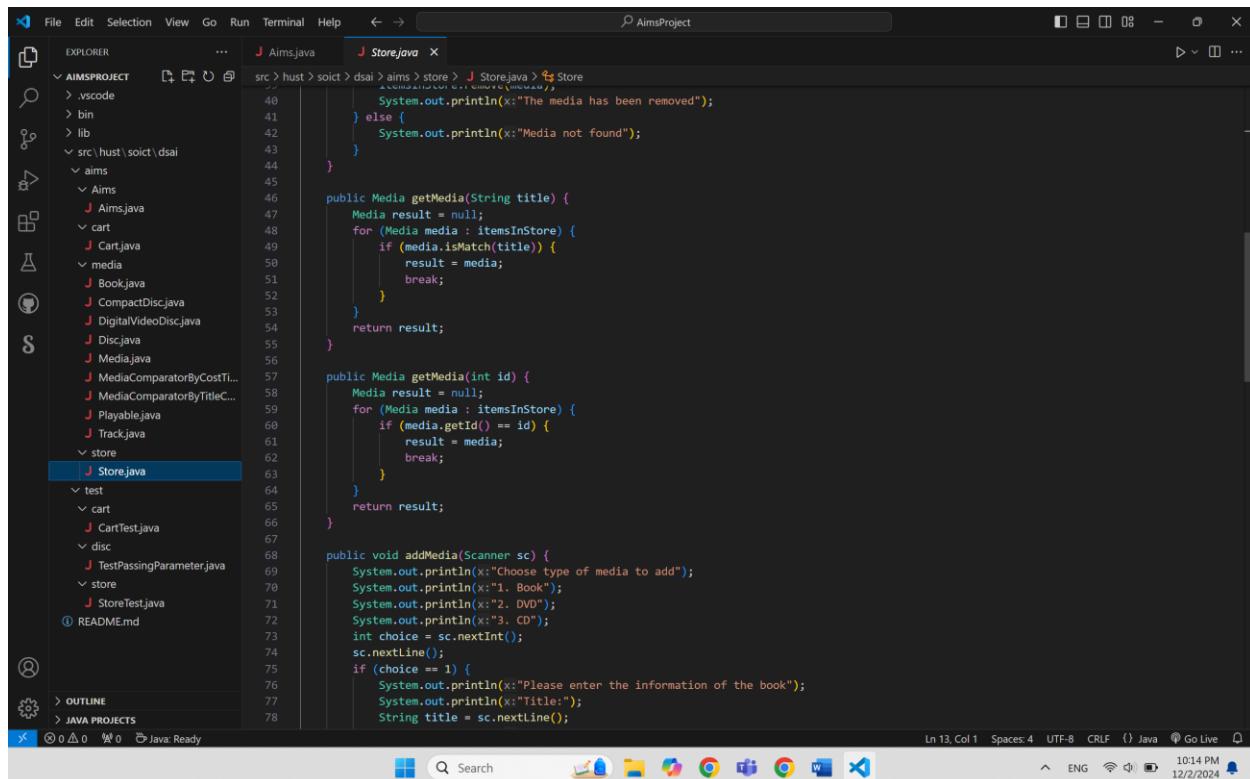
public void sortByTitleCost() {
    Collections.sort(itemsOrdered, COMPARISON_BY_TITLE_COST);
}

public void sortByCostTitle() {
    Collections.sort(itemsOrdered, COMPARISON_BY_COST_TITLE);
}
```

1.3. Store class



```
src > hust > soict > dsai > aims > store > J Store.java > Store
1 package hust.soict.dsai.aims.store;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 import hust.soict.dsai.aims.media.Book;
7 import hust.soict.dsai.aims.media.CompactDisc;
8 import hust.soict.dsai.aims.media.DigitalVideoDisc;
9 import hust.soict.dsai.aims.media.Media;
10
11 public class Store {
12     private ArrayList<Media> itemsInStore = new ArrayList<Media>();
13
14     public void addMedia(Media media) {
15         boolean found = false;
16         for (Media m : itemsInStore) {
17             if (m.equals(media)) {
18                 found = true;
19                 break;
20             }
21         }
22         if (found) {
23             System.out.println("The media is already added");
24         } else {
25             itemsInStore.add(media);
26             System.out.println("The media has been added");
27         }
28     }
29
30     public void removeMedia(Media media) {
31         boolean found = false;
32         for (Media m : itemsInStore) {
33             if (m.equals(media)) {
34                 found = true;
35                 break;
36             }
37         }
38         if (found) {
39             itemsInStore.remove(media);
40             System.out.println("The media has been removed");
41         } else {
42             System.out.println("Media not found");
43         }
44     }
45
46     public Media getMedia(String title) {
47         Media result = null;
48         for (Media media : itemsInStore) {
49             if (media.isMatch(title)) {
50                 result = media;
51                 break;
52             }
53         }
54         return result;
55     }
56
57     public Media getMedia(int id) {
58         Media result = null;
59         for (Media media : itemsInStore) {
60             if (media.getId() == id) {
61                 result = media;
62                 break;
63             }
64         }
65         return result;
66     }
67
68     public void addMedia(Scanner sc) {
69         System.out.println("Choose type of media to add");
70         System.out.println("1. Book");
71         System.out.println("2. DVD");
72         System.out.println("3. CD");
73         int choice = sc.nextInt();
74         sc.nextLine();
75         if (choice == 1) {
76             System.out.println("Please enter the information of the book");
77             System.out.println("Title:");
78             String title = sc.nextLine();
79         }
80     }
81 }
```



```
src > hust > soict > dsai > aims > store > J Store.java > Store
82     String title = sc.nextLine();
83     String author = sc.nextLine();
84     String publisher = sc.nextLine();
85     String year = sc.nextLine();
86     String price = sc.nextLine();
87     String genre = sc.nextLine();
88     String condition = sc.nextLine();
89     String barcode = sc.nextLine();
90
91     Book book = new Book(title, author, publisher, year, price, genre, condition, barcode);
92     itemsInStore.add(book);
93     System.out.println("Book added successfully");
94 }
95
96 public void removeMedia(Scanner sc) {
97     System.out.println("Enter media ID to remove");
98     int id = sc.nextInt();
99     sc.nextLine();
100    for (Media media : itemsInStore) {
101        if (media.getId() == id) {
102            itemsInStore.remove(media);
103            System.out.println("Media removed successfully");
104            break;
105        }
106    }
107 }
108
109 public void printInventory() {
110     System.out.println("Inventory Report");
111     System.out.println("-----");
112     for (Media media : itemsInStore) {
113         System.out.println(media);
114     }
115 }
```

```
src > hust > soict > dsai > aims > store > J Store.java > $ Store
    System.out.println("*****STORE*****");
    String title = sc.nextLine();
    String category = sc.nextLine();
    String cost = sc.nextLine();
    float cost = sc.nextFloat();
    sc.nextLine();
    Book book = new Book(title, category, cost);
    itemsInStore.add(book);
} else if (choice == 2) {
    System.out.println("Please enter the information of the DVD");
    System.out.println("Title:");
    String title = sc.nextLine();
    System.out.println("Category:");
    String category = sc.nextLine();
    System.out.println("Director:");
    String director = sc.nextLine();
    System.out.println("Length:");
    int length = sc.nextInt();
    System.out.println("Cost:");
    float cost = sc.nextFloat();
    sc.nextLine();
    DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
    itemsInStore.add(dvd);
} else if (choice == 3) {
    System.out.println("Please enter the information of the CD");
    System.out.println("Title:");
    String title = sc.nextLine();
    System.out.println("Category:");
    String category = sc.nextLine();
    System.out.println("Artist:");
    String artist = sc.nextLine();
    System.out.println("Director:");
    String director = sc.nextLine();
    System.out.println("Length:");
    int length = sc.nextInt();
    System.out.println("Cost:");
    float cost = sc.nextFloat();
    sc.nextLine();
    CompactDisc cd = new CompactDisc(title, category, artist, director, length, cost);
    itemsInStore.add(cd);
} else {
    System.out.println("Invalid choice");
}
}

public void displayStore() {
    System.out.println("*****STORE*****");
    for (Media media : itemsInStore) {
        System.out.println(media.toString());
    }
    System.out.println("*****STORE*****");
}
```

File Edit Selection View Go Run Terminal Help ↶ → AimsProject

EXPLORER Aims.java J Store.java

AIMSPROJECT .vscode bin src hust soict dsai aims Aims.java cart Book.java CompactDisc.java DigitalVideoDisc.java Disc.java Media.java MediaComparatorByCostTitle.java MediaComparatorByTitleCost.java Playable.java Track.java store J Store.java

test CartTest.java

disc TestPassingParameter.java

store StoreTest.java

README.md

OUTLINE JAVA PROJECTS

0 0 0 0 Java: Ready

Search

113 Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:15 PM 12/2/2024

```
src > hust > soict > dsai > aims > store > J Store.java > $ Store
    DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
    itemsInStore.add(dvd);
} else if (choice == 3) {
    System.out.println("Please enter the information of the CD");
    System.out.println("Title:");
    String title = sc.nextLine();
    System.out.println("Category:");
    String category = sc.nextLine();
    System.out.println("Artist:");
    String artist = sc.nextLine();
    System.out.println("Director:");
    String director = sc.nextLine();
    System.out.println("Length:");
    int length = sc.nextInt();
    System.out.println("Cost:");
    float cost = sc.nextFloat();
    sc.nextLine();
    CompactDisc cd = new CompactDisc(title, category, artist, director, length, cost);
    itemsInStore.add(cd);
} else {
    System.out.println("Invalid choice");
}
}

public void displayStore() {
    System.out.println("*****STORE*****");
    for (Media media : itemsInStore) {
        System.out.println(media.toString());
    }
    System.out.println("*****STORE*****");
}
```

File Edit Selection View Go Run Terminal Help ↶ → AimsProject

EXPLORER Aims.java J Store.java

AIMSPROJECT .vscode bin src hust soict dsai aims Aims.java cart Book.java CompactDisc.java DigitalVideoDisc.java Disc.java Media.java MediaComparatorByCostTitle.java MediaComparatorByTitleCost.java Playable.java Track.java store J Store.java

test CartTest.java

disc TestPassingParameter.java

store StoreTest.java

README.md

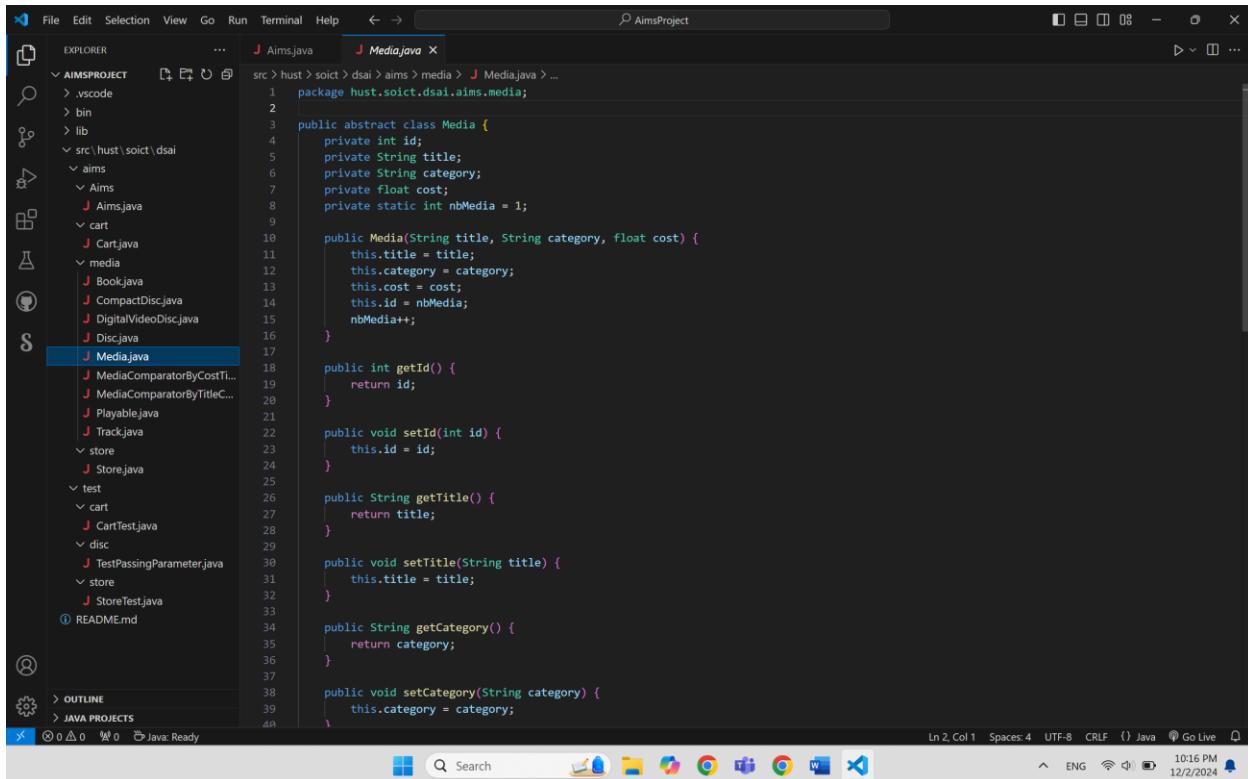
OUTLINE JAVA PROJECTS

0 0 0 0 Java: Ready

Search

113 Col 1 Spaces: 4 UTF-8 CRLF {} Java Go Live 10:15 PM 12/2/2024

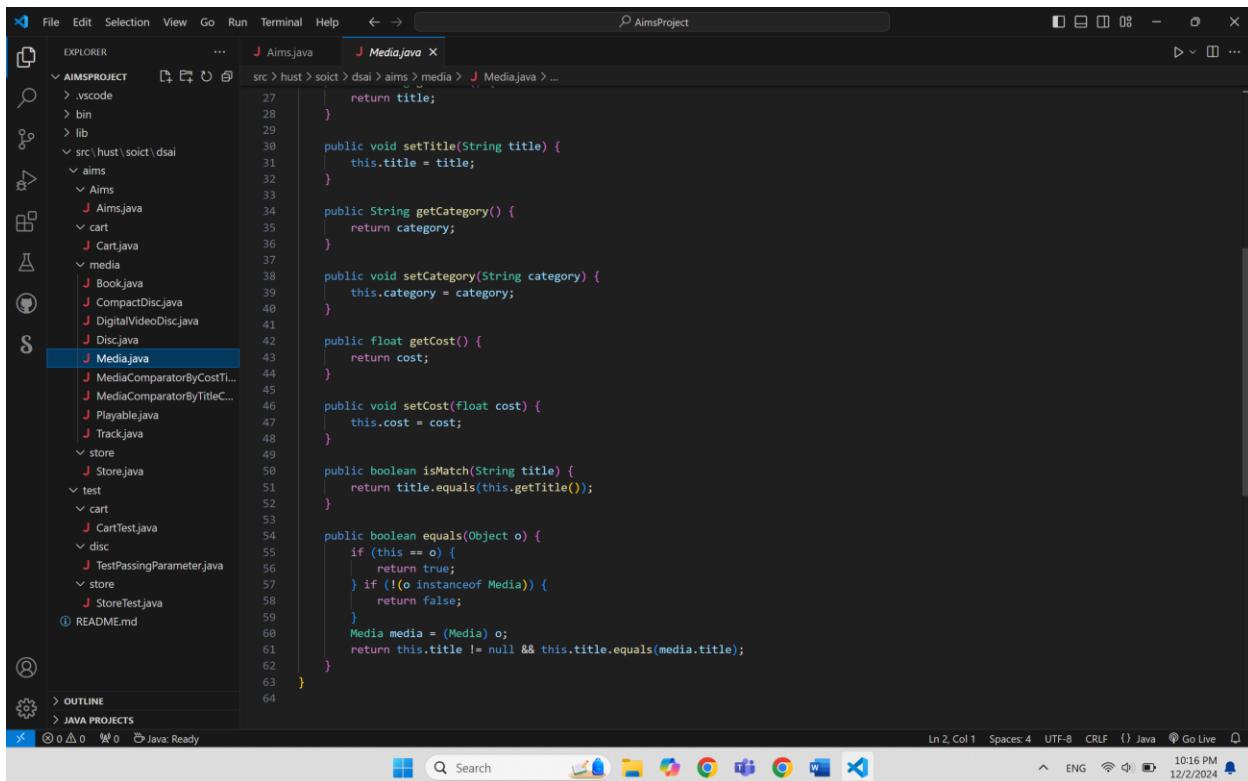
1.4. Media class



The screenshot shows the Visual Studio Code interface with the 'Media.java' file open in the editor. The code defines an abstract class 'Media' with various methods and fields. The code is as follows:

```
1 package hust.soict.dsai.aims.media;
2
3 public abstract class Media {
4     private int id;
5     private String title;
6     private String category;
7     private float cost;
8     private static int nbMedia = 1;
9
10    public Media(String title, String category, float cost) {
11        this.title = title;
12        this.category = category;
13        this.cost = cost;
14        this.id = nbMedia;
15        nbMedia++;
16    }
17
18    public int getId() {
19        return id;
20    }
21
22    public void setId(int id) {
23        this.id = id;
24    }
25
26    public String getTitle() {
27        return title;
28    }
29
30    public void setTitle(String title) {
31        this.title = title;
32    }
33
34    public String getCategory() {
35        return category;
36    }
37
38    public void setCategory(String category) {
39        this.category = category;
40    }
41 }
```

The 'EXPLORER' sidebar on the left shows the project structure, including 'Aims.java', 'Cart.java', 'media', 'Book.java', 'CompactDisc.java', 'DigitalVideoDisc.java', 'Disc.java', and 'Media.java'. The 'OUTLINE' and 'JAVA PROJECTS' sections are also visible.

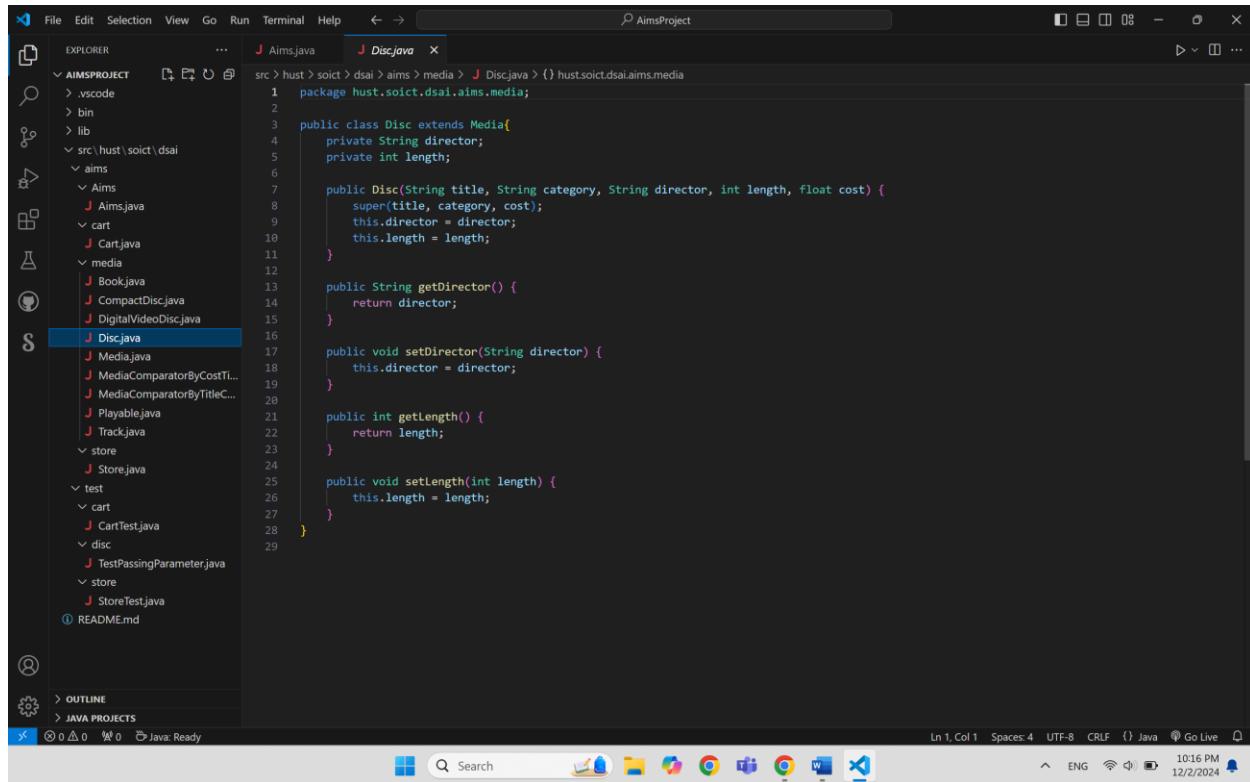


The screenshot shows the Visual Studio Code interface with the 'Media.java' file open in the editor. The code has been modified to include additional methods and logic. The code is as follows:

```
1 package hust.soict.dsai.aims.media;
2
3 public abstract class Media {
4     private int id;
5     private String title;
6     private String category;
7     private float cost;
8     private static int nbMedia = 1;
9
10    public int getId() {
11        return id;
12    }
13
14    public void setTitle(String title) {
15        this.title = title;
16    }
17
18    public String getCategory() {
19        return category;
20    }
21
22    public void setCategory(String category) {
23        this.category = category;
24    }
25
26    public float getCost() {
27        return cost;
28    }
29
30    public void setCost(float cost) {
31        this.cost = cost;
32    }
33
34    public boolean isMatch(String title) {
35        return title.equals(this.getTitle());
36    }
37
38    public boolean equals(Object o) {
39        if (this == o) {
40            return true;
41        } if (!(o instanceof Media)) {
42            return false;
43        }
44        Media media = (Media) o;
45        return this.title != null && this.title.equals(media.title);
46    }
47 }
```

The 'EXPLORER' sidebar on the left shows the project structure, including 'Aims.java', 'Cart.java', 'media', 'Book.java', 'CompactDisc.java', 'DigitalVideoDisc.java', 'Disc.java', and 'Media.java'. The 'OUTLINE' and 'JAVA PROJECTS' sections are also visible.

1.5. Disc class



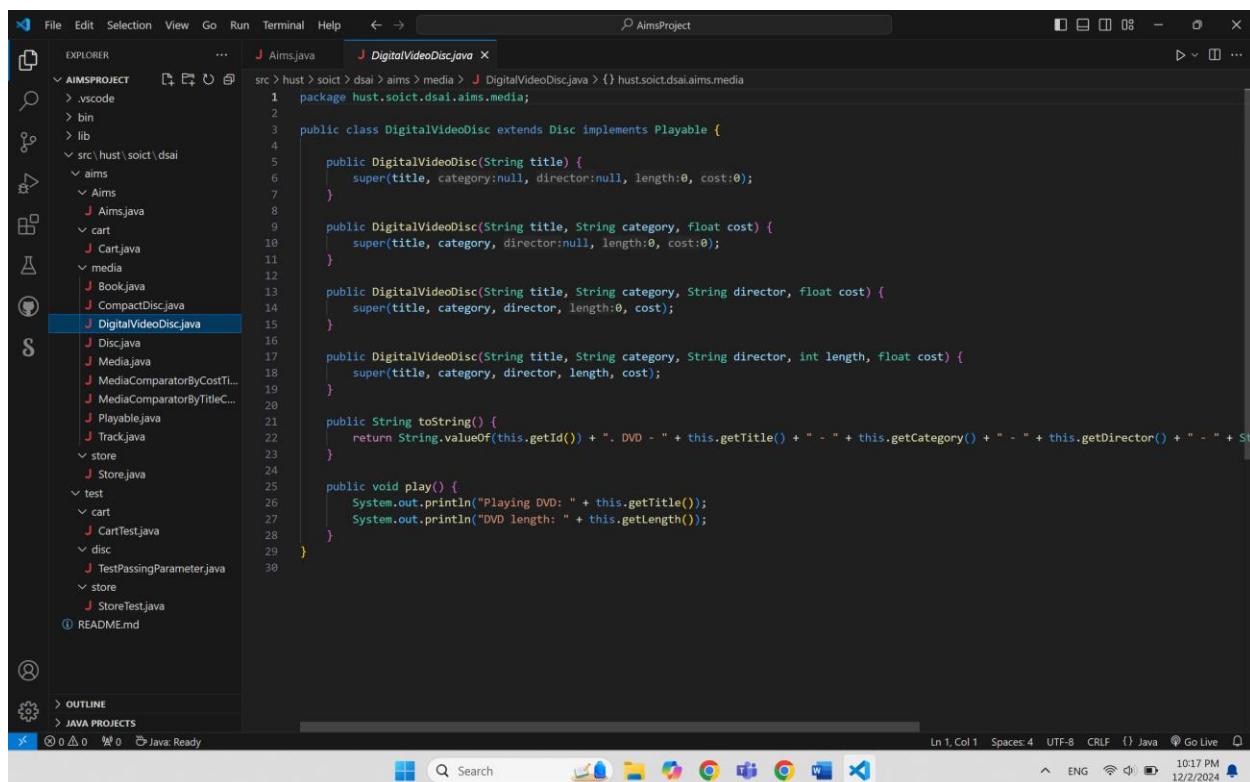
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AimsProject, Disc.java.
- Explorer:** Shows the project structure under AIMS_PROJECT, including .vscode, bin, lib, src, aims, cart, media, test, store, and various Java files like Aims.java, Cart.java, Book.java, CompactDisc.java, DigitalVideoDisc.java, Disc.java, Media.java, MediaComparatorByCostTitle..., MediaComparatorByTitleC..., Playable.java, Track.java, and TestPassingParameter.java.
- Editor:** Displays the code for Disc.java:

```
1 package hust.soict.dsai.aims.media;
2
3 public class Disc extends Media{
4     private String director;
5     private int length;
6
7     public Disc(String title, String category, String director, int length, float cost) {
8         super(title, category, cost);
9         this.director = director;
10        this.length = length;
11    }
12
13    public String getDirector() {
14        return director;
15    }
16
17    public void setDirector(String director) {
18        this.director = director;
19    }
20
21    public int getLength() {
22        return length;
23    }
24
25    public void setLength(int length) {
26        this.length = length;
27    }
28 }
```

- Bottom Status Bar:** Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Java, Go Live, ENG, 10:16 PM, 12/2/2024.

1.6. DVD class



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AimsProject, DigitalVideoDisc.java.
- Explorer:** Shows the project structure under AIMS_PROJECT, including .vscode, bin, lib, src, aims, cart, media, test, store, and various Java files like Aims.java, Cart.java, Book.java, CompactDisc.java, DigitalVideoDisc.java, Disc.java, Media.java, MediaComparatorByCostTitle..., MediaComparatorByTitleC..., Playable.java, Track.java, and TestPassingParameter.java.
- Editor:** Displays the code for DigitalVideoDisc.java:

```
1 package hust.soict.dsai.aims.media;
2
3 public class DigitalVideoDisc extends Disc implements Playable {
4
5     public DigitalVideoDisc(String title) {
6         super(title, category:null, director:null, length:0, cost:0);
7     }
8
9     public DigitalVideoDisc(String title, String category, float cost) {
10        super(title, category, director:null, length:0, cost:0);
11    }
12
13    public DigitalVideoDisc(String title, String category, String director, float cost) {
14        super(title, category, director, length:0, cost);
15    }
16
17    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
18        super(title, category, director, length, cost);
19    }
20
21    public String toString() {
22        return String.valueOf(this.getId()) + ". DVD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " + String.valueOf(this.getLength());
23    }
24
25    public void play() {
26        System.out.println("Playing DVD: " + this.getTitle());
27        System.out.println("DVD length: " + this.getLength());
28    }
29 }
```

- Bottom Status Bar:** Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Java, Go Live, ENG, 10:17 PM, 12/2/2024.

1.7. CD class

The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left displays the project structure of 'AIMSPROJECT'. The current file being edited is 'CompactDisc.java' in the 'src' folder. The code implements the `Playable` interface and contains methods for setting and getting the artist name, as well as adding tracks to the disc.

```
src > hust > soict > dsai > aims > media > J CompactDisc.java > CompactDisc.java
1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc implements Playable {
6     private String artist;
7     private ArrayList<Track> tracks = new ArrayList<Track>();
8
9     public CompactDisc(String title, String category, String artist, String director, int length, float cost) {
10        super(title, category, director, length, cost);
11        this.artist = artist;
12    }
13
14    public String getArtist() {
15        return artist;
16    }
17
18    public void setArtist(String artist) {
19        this.artist = artist;
20    }
21
22    public ArrayList<Track> getTracks() {
23        return tracks;
24    }
25
26    public void setTracks(ArrayList<Track> tracks) {
27        this.tracks = tracks;
28    }
29
30    public void addTrack(Track track) {
31        boolean found = false;
32        for (Track t : tracks) {
33            if (t.equals(track)) {
34                found = true;
35                break;
36            }
37        }
38        if (found) {
39            System.out.println("The track already in the list");
40        }
41    }
42}
```

The Status Bar at the bottom right shows the following information: Line 29, Column 1, Spaces: 4, UTF-8, CRLF, Java, Go Live, ENG, and a battery icon indicating 12/2/2024.

The screenshot shows a Java code editor with the following details:

- File Explorer:** Shows the project structure under "AIMSPROJECT". The "CompactDiscJava" file is selected.
- Code Editor:** Displays the "CompactDiscJava" class with its methods and logic for managing tracks on a CD.
- Status Bar:** Shows "Java: Ready" and other system information.
- Bottom Bar:** Includes icons for file operations like Open, Save, and Close, along with tabs for "Search", "Replace", "Find", and "Outline".

```
src > hust > soict > aims > media > J CompactDiscJava > CompactDisc.java
39     System.out.println(x:"The track already in the list");
40   } else {
41     tracks.add(track);
42   }
43 }

44 public void removeTrack(Track track) {
45   boolean found = false;
46   for (Track t : tracks) {
47     if (t.equals(track)) {
48       found = true;
49       break;
50     }
51   }
52   if (!found) {
53     System.out.println(x:"The track is not in the list");
54   } else {
55     tracks.add(track);
56   }
57 }

58 public int getLength() {
59   int length = 0;
60   for (Track track : tracks) {
61     length += track.getLength();
62   }
63   return length;
64 }

65 public String toString() {
66   return String.valueOf(this.getId()) + ". CD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getArtist() + " - " + this.getYear();
67 }

68 public void play() {
69   System.out.println("Playing CD: " + this.getTitle() + " - " + this.getArtist());
70   for (Track track : this.tracks) {
71     track.play();
72   }
73 }

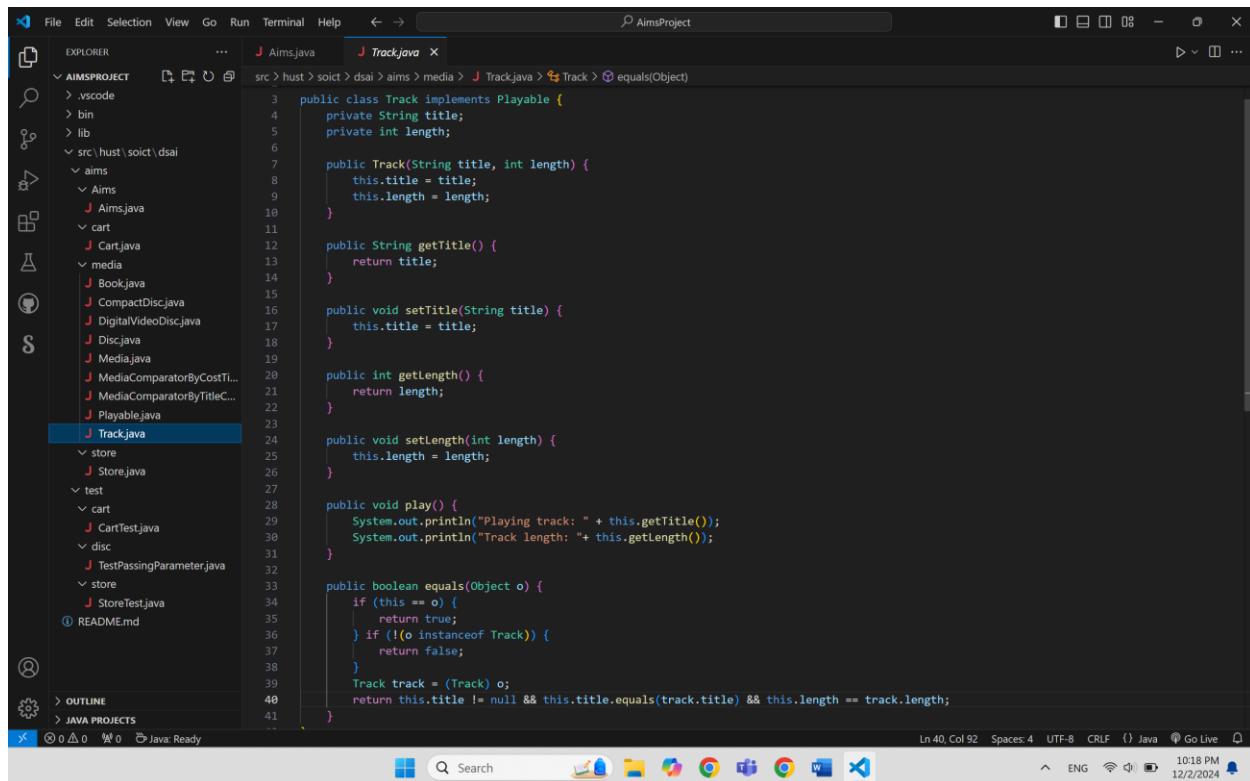
74 }

75 }

76 }

77 }
```

1.8. Track class



The screenshot shows the Visual Studio Code interface with the 'AimsProject' workspace open. The 'EXPLORER' view on the left shows the project structure with 'Track.java' selected. The 'TRACKS' tab in the top navigation bar is active, displaying the code for the 'Track' class. The code implements the 'Playable' interface and defines methods for getting and setting the title and length, as well as playing the track and performing equality checks.

```
public class Track implements Playable {
    private String title;
    private int length;

    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

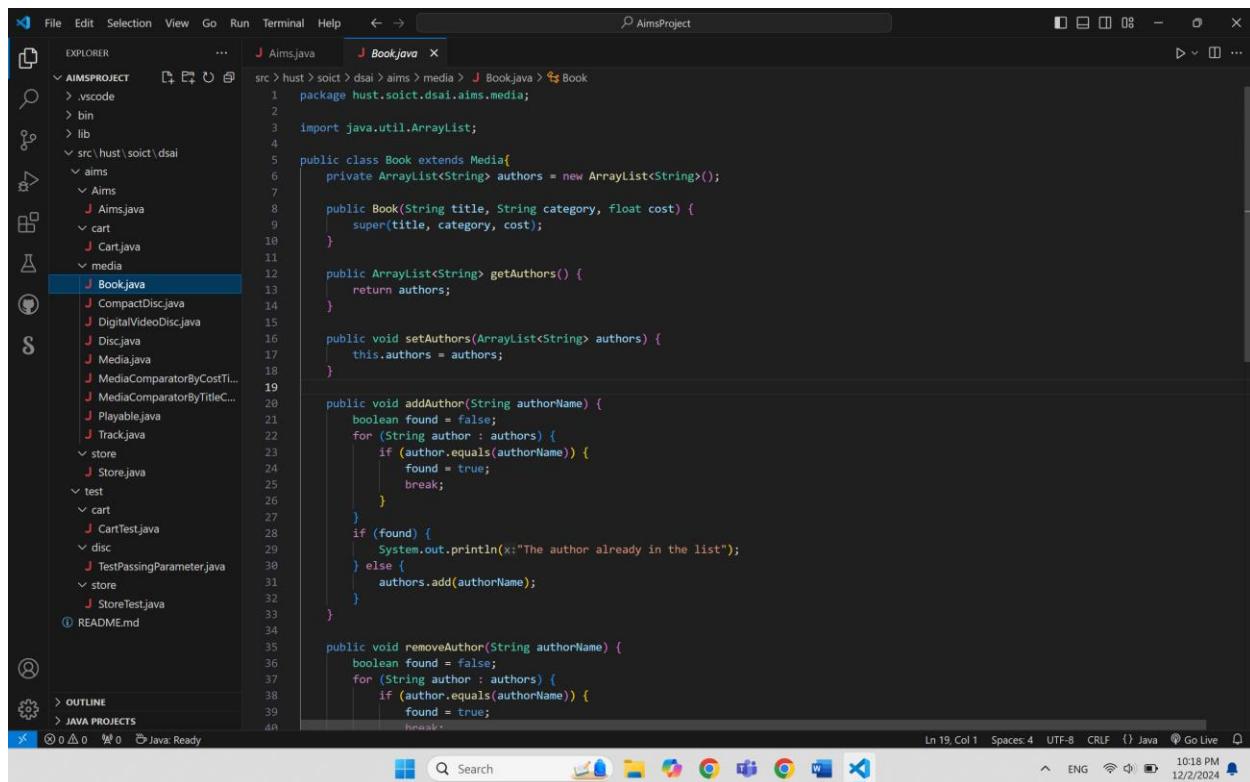
    public int getLength() {
        return length;
    }

    public void setLength(int length) {
        this.length = length;
    }

    public void play() {
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }

    public boolean equals(Object o) {
        if (this == o) {
            return true;
        } if (!(o instanceof Track)) {
            return false;
        }
        Track track = (Track) o;
        return this.title != null && this.title.equals(track.title) && this.length == track.length;
    }
}
```

1.9. Book class



The screenshot shows the Visual Studio Code interface with the 'AimsProject' workspace open. The 'EXPLORER' view on the left shows the project structure with 'Book.java' selected. The 'TRACKS' tab in the top navigation bar is active, displaying the code for the 'Book' class, which extends the 'Media' class and implements the 'Comparable' interface. The code includes methods for adding and removing authors from an ArrayList.

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

import java.util.ArrayList;

public class Book extends Media{
    private ArrayList<String> authors = new ArrayList<String>();

    public Book(String title, String category, float cost) {
        super(title, category, cost);
    }

    public ArrayList<String> getAuthors() {
        return authors;
    }

    public void setAuthors(ArrayList<String> authors) {
        this.authors = authors;
    }

    public void addAuthor(String authorName) {
        boolean found = false;
        for (String author : authors) {
            if (author.equals(authorName)) {
                found = true;
                break;
            }
        }
        if (found) {
            System.out.println("The author already in the list");
        } else {
            authors.add(authorName);
        }
    }

    public void removeAuthor(String authorName) {
        boolean found = false;
        for (String author : authors) {
            if (author.equals(authorName)) {
                found = true;
                break;
            }
        }
    }
}
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under 'AIMSPROJECT'. The 'src' folder contains 'hust', 'soict', 'dsai', 'aims', 'cart', 'media', and 'test' subfolders. Inside 'media', there are files like 'Book.java', 'CompactDisc.java', 'DigitalVideoDisc.java', 'Disc.java', 'Media.java', 'MediaComparatorByCostTi...', 'MediaComparatorByTitleC...', 'Playable.java', and 'Track.java'. Other files include 'Aims.java', 'Cart.java', 'Store.java', 'CartTest.java', 'TestPassingParameter.java', 'StoreTest.java', and 'README.md'.
- Code Editor (Center):** Displays the 'Book.java' file. The code implements the 'Book' class with methods for adding and removing authors, and a toString() method. The code editor has syntax highlighting for Java and includes line numbers.
- Bottom Status Bar:** Shows 'Ln 34, Col 1' and other system information like battery level, signal strength, and time ('10:19 PM 12/2/2024').

1.10. Interface Playable

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under 'AIMSPROJECT'. The 'src' folder contains 'hust', 'soict', 'dsai', 'aims', 'cart', 'media', and 'test' subfolders. Inside 'media', there are files like 'Book.java', 'CompactDisc.java', 'DigitalVideoDisc.java', 'Disc.java', 'Media.java', 'MediaComparatorByCostTi...', 'MediaComparatorByTitleC...', 'Playable.java', and 'Track.java'. Other files include 'Aims.java', 'Cart.java', 'Store.java', 'CartTest.java', 'TestPassingParameter.java', 'StoreTest.java', and 'README.md'.
- Code Editor (Center):** Displays the 'Playable.java' file. It defines a public interface named 'Playable' with a single abstract method 'play()'.
- Bottom Status Bar:** Shows 'Ln 1, Col 1' and other system information like battery level, signal strength, and time ('10:20 PM 12/2/2024').

1.11. Comparator

The screenshot shows the Visual Studio Code interface with the title bar "AimsProject". The Explorer sidebar on the left shows the project structure under "AIMSPROJECT". The main editor area displays the code for `MediaComparatorByCostTitle.java`:

```
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 o1, Media o2) {
7         if (o1.getCost() > o2.getCost()) return -1;
8         if (o1.getCost() < o2.getCost()) return 1;
9         return o1.getTitle().compareToIgnoreCase(o2.getTitle());
10    }
11 }
```

The status bar at the bottom indicates "Ln 1, Col 1" and "Java".

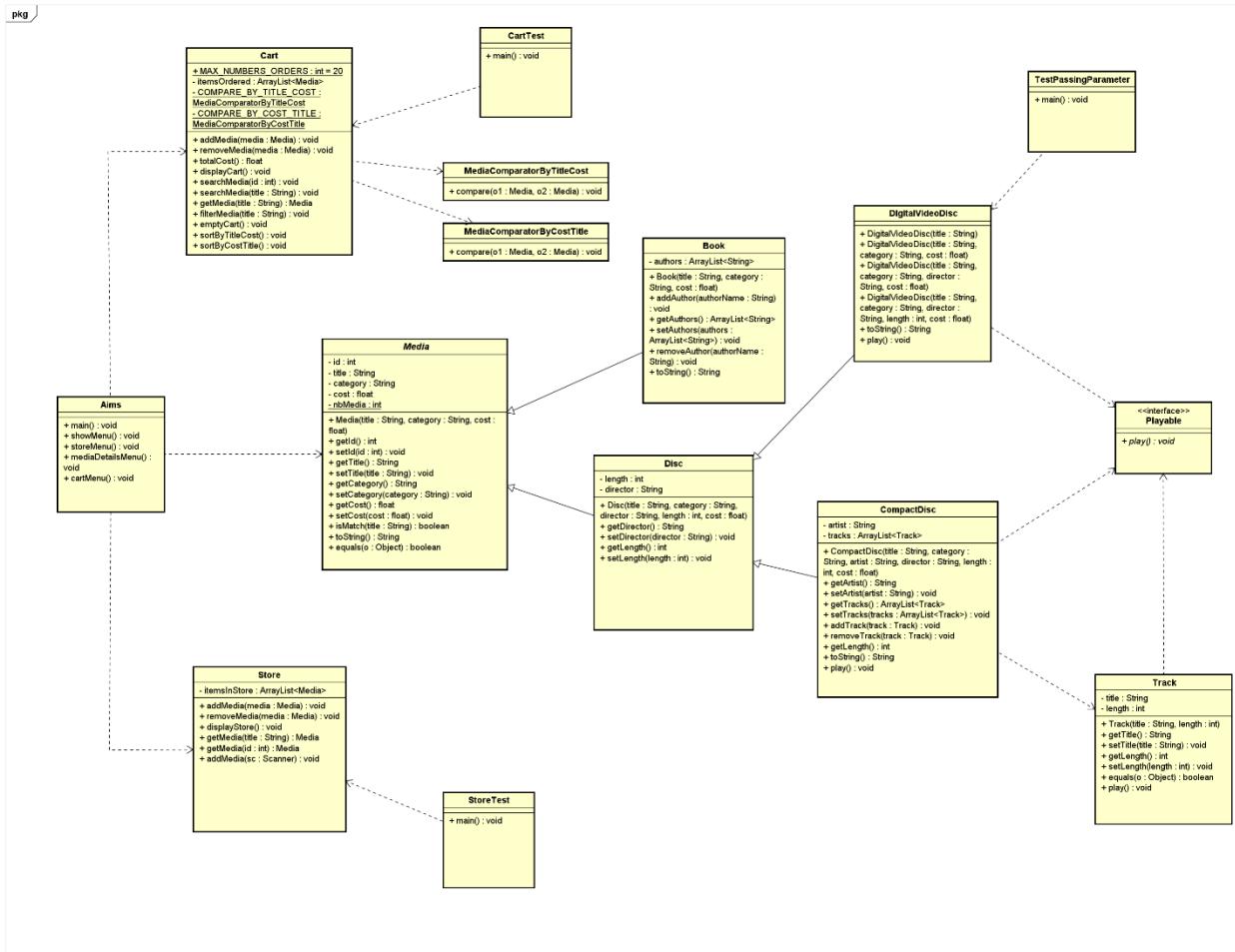
The screenshot shows the Visual Studio Code interface with the title bar "AimsProject". The Explorer sidebar on the left shows the project structure under "AIMSPROJECT". The main editor area displays the code for `MediaComparatorByTitleCost.java`:

```
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 o1, Media o2) {
7         if (!o1.getTitle().equals(o2.getTitle())) {
8             return o1.getTitle().compareToIgnoreCase(o2.getTitle());
9         }
10        return o1.getCost() > o2.getCost() ? -1 : o1.getCost() < o2.getCost() ? 1 : 0;
11    }
12 }
```

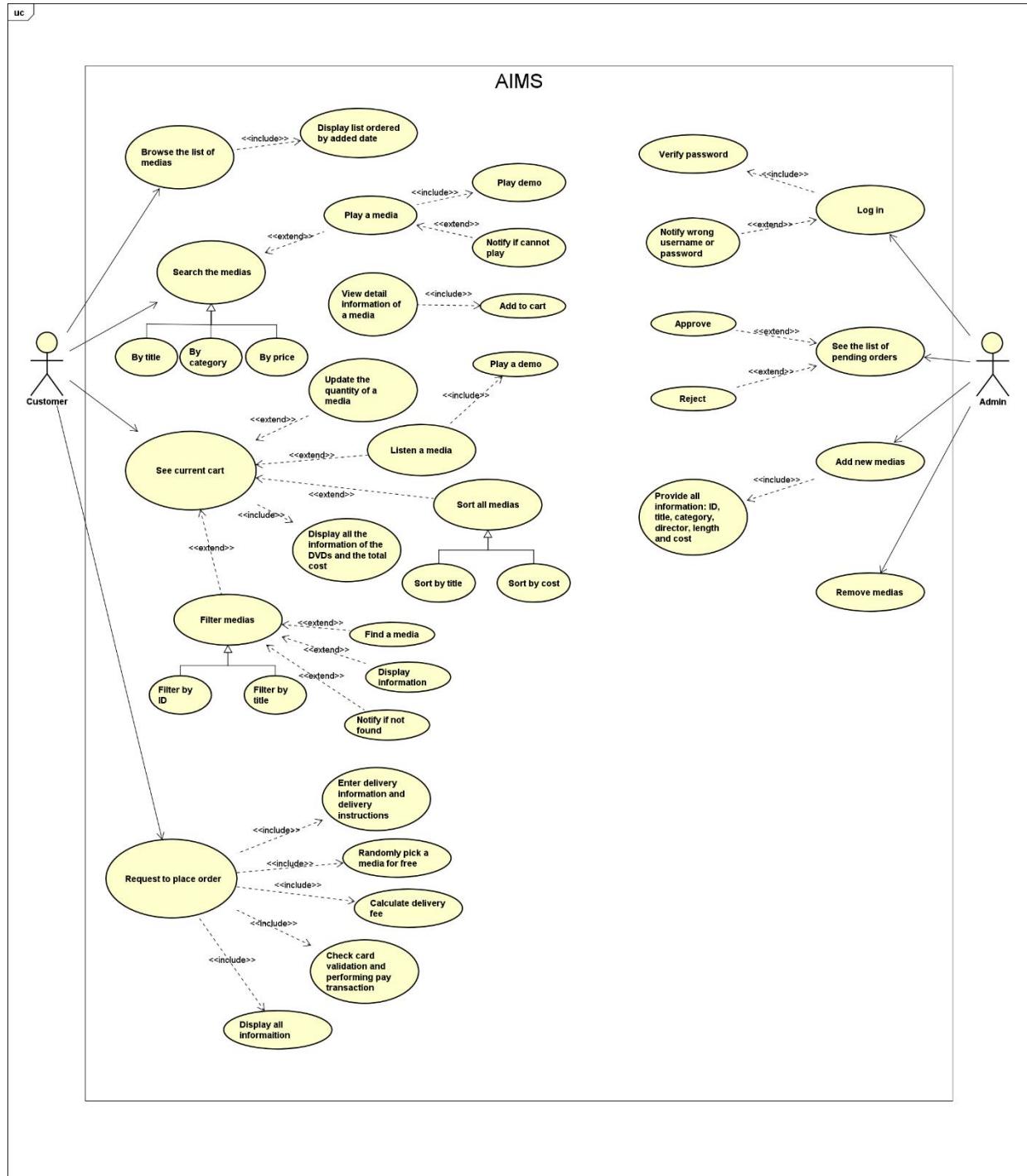
The status bar at the bottom indicates "Ln 1, Col 1" and "Java".

2. Diagram

2.1. Class diagram



2.2. Usecase diagram



3. Answer questions

3.1 Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts?

- Cart aggregates Media objects
- Store aggregates Media objects
- CompactDisc aggregates Track objects

3.2 When overriding the equals() method of the Object class, you will have to cast the Object parameter obj to the type of Object that you are dealing with. For example, in the Media class, you must cast the Object obj to a Media, and then check the equality of the two objects' attributes as the above requirements (i.e. title for Media; title and length for Track). If the passing object is not an instance of Media, what happens?

- If the passing object is not an instance of Media, it will return false.

3.3 **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?
 - In those classes, how should you implement the compareTo()method be to reflect the ordering that we want?
 - 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?
 - 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?
- Class media should implement the Comparable interface and subclass of media class such as Book, Disc, DVD, CD should also override that method.

- Each class should override the `compareTo()` method to define the natural ordering, for example by title, by cost, ...
- Using Comparable interface, a class can define only one ordering using `compareTo()` method.
- We should override this method for DVD class

```
public int compareTo(Media media) {  
    if (media instanceof DigitalVideoDisc) {  
        DigitalVideoDisc dvd = (DigitalVideoDisc) media;  
        int titleCompare = this.getTitle().compareTo(dvd.getTitle());  
        if (titleCompare != 0) {  
            return titleCompare;  
        }  
        int lengthCompare = Integer.compare(dvd.getLength(), this.getLength());  
        if (lengthCompare != 0) {  
            return lengthCompare;  
        }  
        return Float.compare(this.getCost(), dvd.getCost());  
    }  
    return super.compareTo(media);  
}
```