**BÁO CÁO THỰC HÀNH**

**Lab04: Inheritance Polymorphism**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Mục lục**

[1. Import the existing project into the workspace of IntelIJ 2](#_Toc152778503)

[2. Additional requirements of AIMS 2](#_Toc152778504)

[3. Creating the Book class 2](#_Toc152778505)

[4. Creating the abstract Media class 4](#_Toc152778506)

[5. Creating the CompactDisc Class 9](#_Toc152778507)

[5.1 . Create the Disc class extending the Media Class 9](#_Toc152778508)

[5.2 . Create the *Trach class* which models a trach on a compact disc and will store information incuding the *title* and *length* of the track 11](#_Toc152778509)

[5.3 . Open the CompactDisc 12](#_Toc152778510)

[6. Create the Playable interface 14](#_Toc152778511)

[7. Update the Cart class to work with Media 15](#_Toc152778512)

[8. Update the Store class to work with Media 17](#_Toc152778513)

[9. Constructor of whole classes and parent classes 19](#_Toc152778514)

[10. Unique item in a list 20](#_Toc152778515)

[11. Polymorphism with toString() method 21](#_Toc152778516)

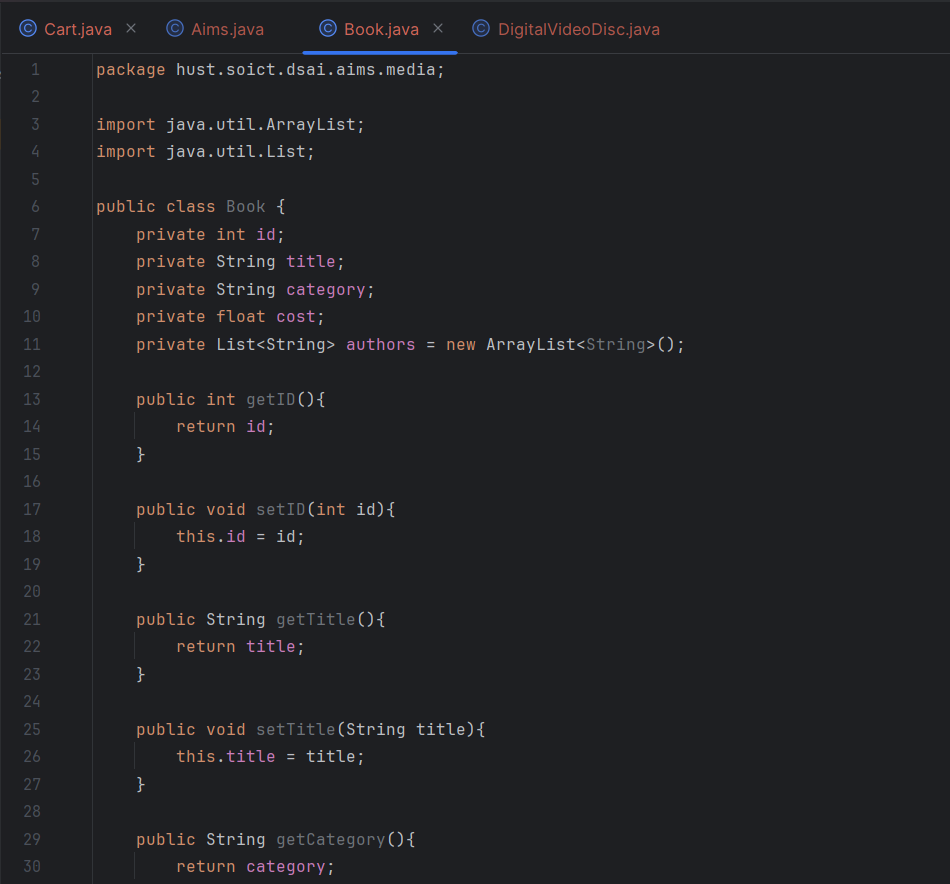
[12. Sort media in the cart 22](#_Toc152778517)

[13. Create a complete console application in the Aims class 23](#_Toc152778518)

# Import the existing project into the workspace of IntelIJ

# Additional requirements of AIMS

# Creating the Book class



A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

A computer screen with colorful text

Description automatically generated

1. Creating the abstract Media class

* **Book class:**

A screen shot of a computer program

Description automatically generated A screen shot of a computer code

Description automatically generated

A computer screen shot of text

Description automatically generated

A computer screen shot of text

Description automatically generated

* **Media class:**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A computer screen shot of a code

Description automatically generated**

* **DigitalVideoDisc Class:**

**A screen shot of a computer program

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

1. **Creating the CompactDisc Class**
2. **. Create the Disc class extending the Media Class**

* Disc Class:

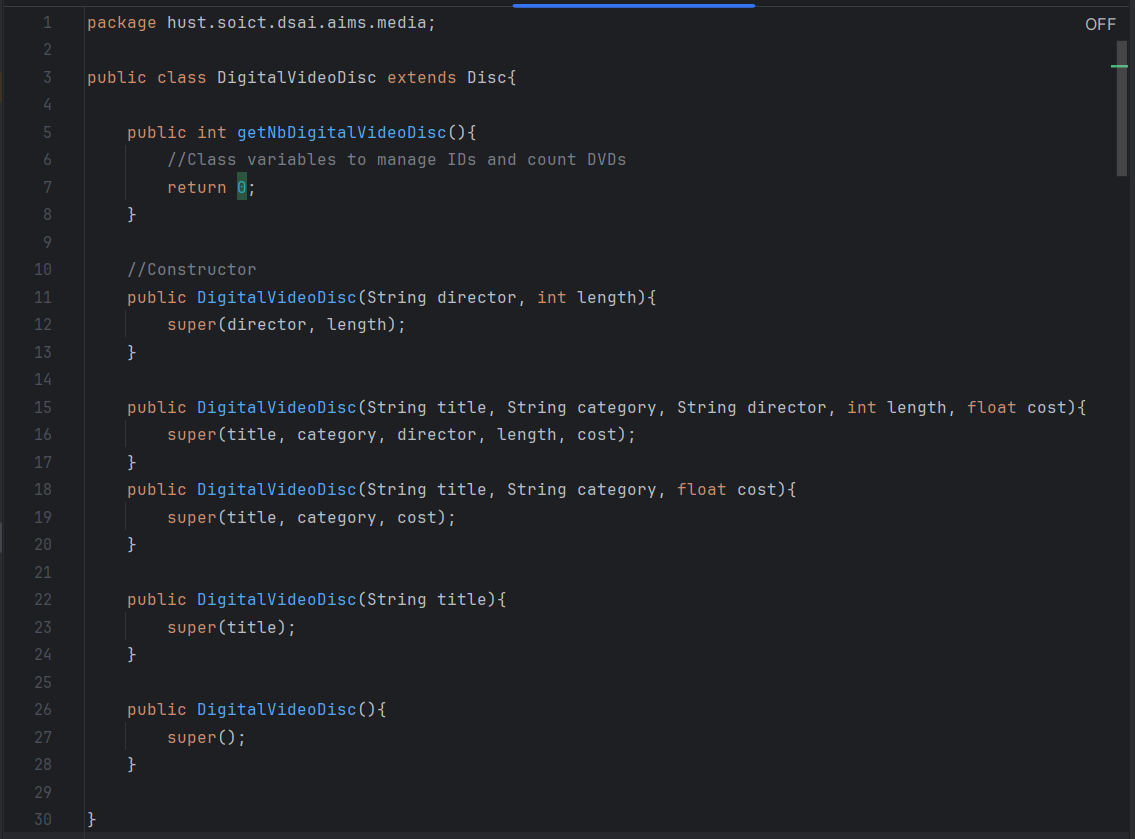
A screenshot of a computer program

Description automatically generated

A computer screen shot of a code

Description automatically generated

* DigitalVideoDisc Class:



* CompactDisc Class:

A screen shot of a computer

Description automatically generated

1. **. Create the *Trach class* which models a trach on a compact disc and will store information incuding the *title* and *length* of the track**

* **Track class:**

**A screen shot of a computer program

Description automatically generated**

1. **. Open the CompactDisc**

* **CompactDisc Class:**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

1. **Create the Playable interface**

* **Playable Interface**

package hust.soict.dsai.aims.media;  
  
public interface Playable {  
 void play();  
}

* **DigitalVideoDisc class:**

package hust.soict.dsai.aims.media;  
  
public class DigitalVideoDisc extends Disc implements Playable{  
 public DigitalVideoDisc (String director, int length){  
 super(director, length);  
 }  
  
 public DigitalVideoDisc (String title, String category, String director, int length, float cost){  
 super(title, category, director, length, cost);  
 }  
  
 public DigitalVideoDisc (String title, String category, float cost){  
 super(title, category, cost);  
 }  
  
 public DigitalVideoDisc (String title){  
 super(title);  
 }  
  
 public DigitalVideoDisc (){  
 super();  
 }  
  
 @Override  
 public void play(){  
 System.*out*.println("Playing DVD: " + this.getTitle());  
 System.*out*.println("DVD length: " + this.getLength());  
 }  
}

* **Track class:**

package hust.soict.dsai.aims.media;  
  
public class Track implements Playable {  
 private final String title;  
 private final int length;  
  
 //Getter methods  
 public String getTitle() {  
 return title;  
 }  
  
 public int getLength() {  
 return length;  
 }  
  
 //Constructor  
 public Track(String title, int length) {  
 this.title = title;  
 this.length = length;  
 }  
 @Override  
 public void play(){  
 System.*out*.println("Playing track: " + this.getTitle());  
 System.*out*.println("Track length: " + this.getLength());  
 }  
}

* Với các thay đổi trên, đã thêm khả năng phát của CompactDisc

DigitalVideoDisc và Track thông qua việc triển khai interface Playable, đồng thời triển khai phương thức play() cho mỗi lớp tương ứng.

1. **Update the Cart class to work with Media**

* **Cart class**

package hust.soict.dsai.aims.cart;  
import hust.soict.dsai.aims.media.DigitalVideoDisc;  
import hust.soict.dsai.aims.media.Media;  
  
import java.io.PrintStream;  
import java.util.ArrayList;  
  
public class Cart {  
  
 // Max Number of order  
 public static final int *MAX\_NUMBERS\_ORDERED* = 20;  
  
 //Arraylíst store orders  
 private ArrayList<Media> itemsOrdered = new ArrayList<>();  
  
 //add a media item to the list  
 public void addMedia(Media media) {  
 //check the current quantity to assure that the cart isn't full  
 if (itemsOrdered.size() < *MAX\_NUMBERS\_ORDERED*) {  
 itemsOrdered.add(media);  
 System.*out*.println("The media has been added!");  
 } else {  
 System.*out*.println("The cart is full!");  
 }  
 }  
  
 //remove the item passed by argument from the list  
 public void removeMedia(Media media) {  
 if (itemsOrdered.remove(media)) {  
 System.*out*.println("The media has been removed!");  
 } else {  
 System.*out*.println("The media not found!");  
 }  
 }  
  
 //method which loops through the values of the array and sums the cost of the individual Media items  
 public float totalCost(){  
 float total = 0;  
 for (Media media : itemsOrdered){  
 total += media.getCost();  
 }  
 return total;  
 }  
  
 public void listCart(){  
 for (Media media : itemsOrdered){  
 System.*out*.println("id: " + media.getId() + " " + media.getTitle());  
 }  
 }  
  
 //Search by ID  
 public boolean search(int id){  
 for (Media media : itemsOrdered){  
 if (media.getId() == id){  
 return true;  
 }  
 }  
 return false;  
 }  
  
 //Search by Title  
 public boolean search(String title){  
 for (Media media : itemsOrdered){  
 if (media.getTitle().equals(title){  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public void printCart() {  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 System.*out*.println(" Ordered Iteams:");  
 for (Media media : itemsOrdered) {  
 System.*out*.printf("id: %d. %s\n" + media.getId(), media.getTitle());  
 }  
 float totalCost = totalCost();  
 System.*out*.printf("Total cost: %.2f $\n", totalCost);  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
   
 }  
}

* Sử dụng ArrayList<Media> thay vì mảng để lưu trữ các đối tượng Media
* Sử dụng phương thức addMedia() thay vì addDigitalVideoDisc()
* Sử dụng phương thức removeMedia() thay vì removeDigitalVideoDisc()
* Sử dụng size() để kiểm tra số lượng phần tử trong giỏ hàng
* Sử dụng vòng lặp for-each để duyệt qua danh sách các đối tượng Media
* Sửa lại phương thức printCart() để in thông tin đối tượng Media

1. **Update the Store class to work with Media**

* **Store class:**
* package hust.soict.dsai.aims.media;  
    
  import java.util.ArrayList;  
    
  public class Store {  
     
   //Max Number of items in store  
   public static final int *MAX\_NUMBERS\_STORED* = 100;  
     
   //ArrayList store items  
   private ArrayList<Media> itemsInStore = new ArrayList<>();  
     
   //add a media item to the store  
   public void addMedia(Media media) {  
     
   //check the current quantity to assure that the store is not full  
   if (itemsInStore.size() < *MAX\_NUMBERS\_STORED*){  
   itemsInStore.add(media);  
   System.*out*.println("The media has been added to the store!");  
   }  
   else {  
   System.*out*.println("The store is full!");  
   }  
   }  
     
   //remove items  
   public void removeMedia(Media media) {  
   if (itemsInStore.remove(media)){  
   System.*out*.println("The media has been removed!");  
   }  
   else{  
   System.*out*.println("The media not found!");  
   }  
   }  
     
   //list all items  
   public void listItems() {  
   for (Media media : itemsInStore) {  
   System.*out*.println("id: " + media.getId() + " " + media.getTitle());  
   }  
   }  
     
   //search by ID  
   public boolean search(int id){  
   for (Media media : itemsInStore){  
   if (media.getId() == id){  
   return true;  
   }  
   }  
   return false;  
   }  
     
   //search by title  
   public boolean search(String title){  
   for (Media media : itemsInStore){  
   if (media.getTitle().equals(title)){  
   return true;  
   }  
   }  
   return false;  
   }  
  }
* itemsInStore từ mảng DigitalVideoDisc[] sang ArrayList<Media> trong lớp Store.
* Sau đó, cần thay thế các phương thức addDVD() và removeDVD() bằng addMedia() và removeMedia().

1. **Constructor of whole classes and parent classes**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

1. **Unique item in a list**

* **Track class:**

@Override  
public boolean equals(Object o) {  
 Track tmp = (Track) o;  
 if (this.getTitle() == tmp.getTitle() && this.getLength() == tmp.getLength()) {  
 return true;  
 } else return false;  
}

* **Media class:**

@Override  
public boolean equals(Object o){  
 Media tmp = (Media) o;  
 if (this.getTitle() == tmp.getTitle()){  
 return true;  
 }  
 else return false;  
}

1. **Polymorphism with toString() method**

* **MediaTest class:**

package hust.soict.dsai.aims.media;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MediaTest {  
 public static void main(String args[]) {  
 List<Media> media = new ArrayList<Media>();  
  
 DigitalVideoDisc dvd = new DigitalVideoDisc("Conan", "Comic", "aidji", 87, 19.95f);  
  
 Book book = new Book();  
  
 CompactDisc cd = new CompactDisc();  
  
 media.add(cd);  
 media.add(dvd);  
 media.add(book);  
  
 for(Media m: media) {  
 System.*out*.println(m.toString());  
 }  
 }  
}

* **Kết quả:**

**A screenshot of a computer program

Description automatically generated**

# **Sort media in the cart**

* **MediaComparatorByCostTitle Class:**

package hust.soict.dsai.aims.media;  
  
import java.util.Comparator;  
  
public class MediaComparatorByCostTitle implements Comparator<Media>{  
  
 @Override  
 public int compare(Media o1, Media o2) {  
 // *TODO Auto-generated method stub* return o1.getTitle().compareTo(o2.getTitle());  
 }  
  
}

* **MediaComparatorByTitleCost class:**

package hust.soict.dsai.aims.media;  
  
import java.util.Comparator;  
  
public class MediaComparatorByTitleCost implements Comparator<Media>{  
 @Override  
 public int compare(Media o1, Media o2) {  
 // *TODO Auto-generated method stub* return o1.getTitle().compareTo(o2.getTitle());  
 }  
}

* **Media class:**

public static final Comparator<Media> *COMPARE\_BY\_TITLE\_COST* = new MediaComparatorByTitleCost();  
public static final Comparator<Media> *COMPARE\_BY\_COST\_TITLE* = new MediaComparatorByCostTitle();

1. **Create a complete console application in the Aims class**

* **Menu chính:**

**A screenshot of a computer

Description automatically generated**

* **View Store**

**A screenshot of a computer program

Description automatically generated**

* **Media’s Details:**

**A screenshot of a computer program

Description automatically generated**

* **Play Media:**

**A screenshot of a computer program

Description automatically generated**

* **Thêm vào Cart:**

**A screenshot of a computer program

Description automatically generated**

* **Xem Cart:**

**A screen shot of a computer

Description automatically generated**

* **Filter theo title:**

**A screenshot of a computer program

Description automatically generated**

* **Sort theo Cost/Tite:**

**A screenshot of a computer program

Description automatically generated**

* **Xóa Media khỏi Order:**

**A screenshot of a computer program

Description automatically generated**

* **Place order**

**A screenshot of a computer program

Description automatically generated**

* **Update Store:**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

* **Remove Media khỏi Store:**

**A screenshot of a computer program

Description automatically generated**

* **Code:**

package hust.soict.dsai.aims.aims;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.util.Scanner;  
  
import hust.soict.dsai.aims.cart.Cart;  
import hust.soict.dsai.aims.media.\*;  
import hust.soict.dsai.aims.store.Store;  
  
public class Aims {  
  
 private static Scanner *sc* = new Scanner(System.*in*);  
 private static Cart *order* = new Cart();  
 private static Store *store* = new Store();  
 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("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");  
 }  
  
 //======================================View store=========================================//  
 public static void test(){}  
 public static void showDetailsOfEachMediaType(Media media) {  
  
 int maxLen = 20;  
 if (media instanceof DigitalVideoDisc) {  
 System.*out*.println("╔══════════════════════════════════════════════╗");  
 System.*out*.printf("║ Title: %-"+ maxLen +"s \n", media.getTitle());  
 System.*out*.printf("║ Director: %-"+ maxLen +"s \n", ((DigitalVideoDisc) media).getDirector());  
 System.*out*.printf("║ Length: %-3d minutes \n", ((DigitalVideoDisc) media).getLength());  
 System.*out*.printf("║ Cost: $%-5.2f \n", media.getCost());  
 System.*out*.println("╚══════════════════════════════════════════════╝");  
  
 } else if (media instanceof CompactDisc) {  
  
 System.*out*.println("╔══════════════════════════════════════════════╗");  
 System.*out*.printf("║ Title: %-"+ maxLen +"s \n", media.getTitle());  
 System.*out*.printf("║ Artist: %-"+ maxLen +"s \n", ((CompactDisc) media).getArtist());  
 for(Track t: ((CompactDisc) media).getTrack()){  
 System.*out*.printf("║ Track: %-"+ maxLen +"s \n", t.getTitle());  
 System.*out*.printf("║ Length: %-3d minutes \n", t.getLength());  
 }  
 System.*out*.printf("║ Cost: $%-5.2f \n", media.getCost());  
 System.*out*.println("╚══════════════════════════════════════════════╝");  
 } else if (media instanceof Book) {  
  
 List<String> authors = ((Book) media).getAuthors();  
 String authorsString = String.*join*(", ", authors);  
  
 System.*out*.println("╔══════════════════════════════════════════════╗");  
 System.*out*.printf("║ Title: %-"+ maxLen +"s \n", media.getTitle());  
 System.*out*.printf("║ Authors: %-"+ maxLen +"s \n", authorsString);  
 System.*out*.printf("║ Cost: $%-5.2f \n", media.getCost());  
 System.*out*.println("╚══════════════════════════════════════════════╝");  
 } else {  
 System.*out*.println("==============================");  
 System.*out*.println("===>>>Unknown media type<<<===");  
 System.*out*.println("==============================");  
 }  
 }  
  
 public static void addToCart(Media m){  
 System.*out*.println("\n>>>>>>>>>>>>>=======<<<<<<<<<<<<");  
 *order*.addMedia(m);  
 System.*out*.println(">>>>>>>>>>>>>=======<<<<<<<<<<<<\n");  
 }  
  
 public static void playMedia(Media m){  
 if (m instanceof DigitalVideoDisc) {  
 System.*out*.println("\n%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%");  
 ((DigitalVideoDisc) m).play();  
 System.*out*.println("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\n");  
 } else if (m instanceof CompactDisc) {  
 System.*out*.println("\n%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%");  
 ((CompactDisc) m).play();  
 System.*out*.println("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\n");  
 } else {  
 System.*out*.println("======================================================");  
 System.*out*.println("===>>>This type of media does not have play mode<<<===");  
 System.*out*.println("======================================================");  
 }  
 }  
  
 public static boolean solveOptionOfMediaDetailsSelected(Media m){  
 switch (*readOption*()){  
  
 // 1. Add to cart  
 case 1:  
 *addToCart*(m);  
 break;  
  
 // 2. Play  
 case 2:  
 *playMedia*(m);  
 break;  
  
 // 0. Back  
 case 0:  
 return true;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 return false;  
 }  
 return false;  
 }  
  
 public static void showMediaDetails(){  
  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media: ");  
 System.*out*.print(">>>>>");  
 Media tmp = *store*.searchMedia(*sc*.nextLine());  
 if(tmp!=null){  
 *showDetailsOfEachMediaType*(tmp);  
  
 // Menu Media Details  
 boolean backToMenuStore = false;  
  
 while (!backToMenuStore) {  
 *mediaDetailsMenu*();  
 backToMenuStore = *solveOptionOfMediaDetailsSelected*(tmp);  
 }  
 }  
 else {  
 System.*out*.println("=================================================");  
 System.*out*.println("===>>>The Media does not exist in the store<<<===");  
 System.*out*.println("=================================================");  
 }  
  
 }  
  
 public static void addAMediaToCart(){  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media you want to add to the cart: ");  
 System.*out*.print(">>>>>");  
 Media tmp = *store*.searchMedia(*sc*.nextLine());  
 if(tmp!=null){  
 System.*out*.println("\n>>>>>>>>>>>>>=======<<<<<<<<<<<<");  
 *order*.addMedia(tmp);  
 System.*out*.println(">>>>>>>>>>>>>=======<<<<<<<<<<<<\n");  
 }  
 else {  
 System.*out*.println("=================================================");  
 System.*out*.println("===>>>The Media does not exist in the store<<<===");  
 System.*out*.println("=================================================");  
 }  
 int count = 0;  
 if(tmp instanceof DigitalVideoDisc){  
 for(Media m : *order*.getOrderedList())  
 if(m instanceof DigitalVideoDisc) count++;  
  
 System.*out*.println(">>>>>The number of DVDs in the current cart: " + count +"<<<<<\n");  
 }  
 }  
  
 public static void playAMedia(){  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media you want to play: ");  
 System.*out*.print(">>>>>");  
 Media tmp = *store*.searchMedia(*sc*.nextLine());  
 *playMedia*(tmp);  
 }  
 public static boolean solveOptionOfStoreMenuSelected(){  
 switch (*readOption*()){  
  
 // 1. See a media’s details  
 case 1:  
 *showMediaDetails*();  
 break;  
  
 // 2. Add a media to cart  
 case 2:  
 *addAMediaToCart*();  
 break;  
  
 // 3. Play a media  
 case 3:  
 *playAMedia*();  
 break;  
  
 // 4. See current cart  
 case 4:  
 *seeCurrentCart*();  
 break;  
  
 // 0. Back  
 case 0:  
 return true;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 return false;  
 }  
 return false;  
 }  
  
 public static void viewStore(){  
  
 boolean backToMenuMain = false;  
  
 while (!backToMenuMain) {  
 *store*.printStore();  
 *storeMenu*();  
 backToMenuMain = *solveOptionOfStoreMenuSelected*();  
 }  
 }  
 //============================================================================================//  
  
 //=================================Update store========================================//  
  
 public static void updateStoreMenu(){  
 System.*out*.println("##########Update Store##########");  
 System.*out*.println("--------------------------------");  
 System.*out*.println("1. Add a Media");  
 System.*out*.println("2. Remove a Media");  
 System.*out*.println("0. Back");  
 System.*out*.println("--------------------------------");  
 System.*out*.println("Please choose a number: 0-1-2");  
 }  
  
  
 public static void addAMediaToStoreMenu(){  
 System.*out*.println("What type of media do you want to add?");  
 System.*out*.println("--------------------------------");  
 System.*out*.println("1. Book");  
 System.*out*.println("2. DVD");  
 System.*out*.println("3. CD");  
 System.*out*.println("0. Back");  
 System.*out*.println("--------------------------------");  
 System.*out*.println("Please choose a number: 0-1-2-3");  
 }  
  
 public static void addABookToStore(){  
 System.*out*.println("------Book------");  
 *sc*.nextLine();  
 System.*out*.println("Title: ");  
 String title = *sc*.nextLine();  
 System.*out*.println("Cost: ");  
 float cost = *sc*.nextFloat();  
 List<String> authors = new ArrayList<>();  
 *sc*.nextLine();  
 System.*out*.println("Author (enter 'done' to finish): ");  
 while (true) {  
  
 String author = *sc*.nextLine();  
  
 if (author.equalsIgnoreCase("done")) {  
 break;  
 }  
  
 authors.add(author);  
 }  
 *store*.addMedia(new Book(title, "Book", cost, authors));  
 System.*out*.println("Added Book to Store successfully");  
 System.*out*.println("--------------------------------");  
 }  
 public static void addADVDToStore(){  
 System.*out*.println("------DVD------");  
 *sc*.nextLine();// Consume the newline character  
  
 System.*out*.print("Title: ");  
 String title = *sc*.nextLine();  
  
 System.*out*.print("Cost: ");  
 float cost = *sc*.nextFloat();  
 *sc*.nextLine(); // Consume the newline character  
  
 System.*out*.print("Director: ");  
 String director = *sc*.nextLine();  
  
 System.*out*.print("Length: ");  
 int length = *sc*.nextInt();  
 *sc*.nextLine(); // Consume the newline character  
  
 *store*.addMedia(new DigitalVideoDisc(title, "DVD", director, length, cost));  
 System.*out*.println("Added DVD to Store successfully");  
 System.*out*.println("-------------------------------");  
 }  
 public static void addACDToStore(){  
 *sc*.nextLine(); // Consume the newline character  
 System.*out*.println("------CD------");  
 System.*out*.print("Title: ");  
 String title = *sc*.nextLine();  
  
 System.*out*.print("Cost: ");  
 float cost = *sc*.nextFloat();  
 *sc*.nextLine(); // Consume the newline character  
  
 System.*out*.print("Artist: ");  
 String artist = *sc*.nextLine();  
  
  
 System.*out*.println("Enter Track Information for CD (Enter '0' to finish):");  
 ArrayList<Track> tracks = new ArrayList<>();  
 int trackNumber = 1;  
 while (true) {  
 System.*out*.print("Track " + trackNumber + " Title (Enter '0' to finish): ");  
 String trackTitle = *sc*.nextLine();  
 if (trackTitle.equals("0")) {  
 break;  
 }  
  
 System.*out*.print("Track " + trackNumber + " Length: ");  
 int trackLength = *sc*.nextInt();  
 *sc*.nextLine(); // Consume the newline character  
  
 Track track = new Track(trackTitle, trackLength);  
 tracks.add(track);  
 trackNumber++;  
 }  
  
 *store*.addMedia(new CompactDisc(title, "CD", cost, artist, tracks));  
 System.*out*.println("Added CD to Store successfully");  
 System.*out*.println("------------------------------");  
 }  
  
 public static boolean solveOptionOfAddAMediaToStoreSelected(){  
 switch (*readOption*()){  
  
 // 1. Add a Book  
 case 1:  
 *addABookToStore*();  
 break;  
  
 // 2. Add a DVD  
 case 2:  
 *addADVDToStore*();  
 break;  
  
 // 3. Add a CD  
 case 3:  
 *addACDToStore*();  
 break;  
  
 // 0. Back  
 case 0:  
 return true;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 return false;  
 }  
 return false;  
 }  
 public static void addAMediaToStore(){  
  
 boolean backToUpdateStoreMenu = false;  
  
 while (!backToUpdateStoreMenu) {  
 *addAMediaToStoreMenu*();  
 backToUpdateStoreMenu = *solveOptionOfAddAMediaToStoreSelected*();  
 }  
 }  
 public static void removeAMediaFromStore(){  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media you want to remove: ");  
 System.*out*.print(">>>>>");  
 Media tmp = *store*.searchMedia(*sc*.nextLine());  
 if(tmp!=null){  
 System.*out*.println("\n>>>>>>>>>>>>>=======<<<<<<<<<<<<");  
 *store*.removeMedia(tmp);  
 System.*out*.println(">>>>>>>>>>>>>=======<<<<<<<<<<<<\n");  
 }  
 else {  
 System.*out*.println("=================================================");  
 System.*out*.println("===>>>The Media does not exist in the store<<<===");  
 System.*out*.println("=================================================");  
 }  
 }  
 public static boolean solveOptionOfUpdateStoreMenuSelected(){  
 switch (*readOption*()){  
  
 // 1. Add a Media  
 case 1:  
 *addAMediaToStore*();  
 break;  
  
 // 2. Remove a Media  
 case 2:  
 *removeAMediaFromStore*();  
 break;  
  
 // 0. Back  
 case 0:  
 return true;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 return false;  
 }  
 return false;  
 }  
 public static void updateStore(){  
  
 boolean backToMenuMain = false;  
  
 while (!backToMenuMain) {  
 *updateStoreMenu*();  
 backToMenuMain = *solveOptionOfUpdateStoreMenuSelected*();  
 }  
  
 }  
  
 //=====================================================================================//  
  
 //===============================See current cart====================================//  
  
 public static void filterMediaInCart(){  
  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media you want to filter: ");  
 System.*out*.print(">>>>>");  
 String title = *sc*.nextLine();  
  
 ArrayList<Media> itemsOrdered = *order*.getOrderedList();  
  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 System.*out*.println("Ordered Items by Title:");  
 int i = 0;  
  
 String lowerCaseKeyword = title.toLowerCase();  
 for (Media media : itemsOrdered){  
  
 String lowerCaseTitle = media.getTitle().toLowerCase();  
 if (lowerCaseTitle.contains(lowerCaseKeyword)){  
 System.*out*.printf("#%d: %s. %.2f $\n", ++i, media.getTitle(), media.getCost());  
 }  
  
 }  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 }  
  
 public static void sortMediasInCart() {  
 System.*out*.println(">>>>>>>Sort by: \n1. Cost\n2. Title");  
 switch (*readOption*()) {  
 case 1:  
 *order*.sortByCost();  
 break;  
 case 2:  
 *order*.sortByTitle();  
 break;  
 }  
 }  
  
 public static void removeAMediaFromCart(){  
  
 *sc*.nextLine();  
 System.*out*.println(">>>>>Enter the title of the media you want to remove: ");  
 System.*out*.print(">>>>>");  
 Media tmp = *order*.searchMedia(*sc*.nextLine());  
 if(tmp!=null){  
 System.*out*.println("\n>>>>>>>>>>>>>=======<<<<<<<<<<<<");  
 *order*.removeMedia(tmp);  
 System.*out*.println(">>>>>>>>>>>>>=======<<<<<<<<<<<<\n");  
 }  
 else {  
 System.*out*.println("=================================================");  
 System.*out*.println("===>>>The Media does not exist in the cart<<<===");  
 System.*out*.println("=================================================");  
 }  
 }  
  
 public static void placeOrder(){  
 if(*order*.getOrderedList().isEmpty()){  
 System.*out*.println("!!!! Your order is EMPTY !!!!");  
 }  
 else{  
 System.*out*.println("\*\*\*\* Order is created \*\*\*\*");  
 }  
 }  
  
 public static boolean solveOptionOfCartMenuSelected(){  
 switch (*readOption*()){  
  
 // 1. Filter medias in cart  
 case 1:  
 *filterMediaInCart*();  
 break;  
  
 // 2. Sort medias in cart  
 case 2:  
 *sortMediasInCart*();  
 break;  
  
 // 3. Remove media from cart  
 case 3:  
 *removeAMediaFromCart*();  
 break;  
  
 // 4. Play a media  
 case 4:  
 *playAMedia*();  
 break;  
  
 // 5. Place order  
 case 5:  
 *placeOrder*();  
 break;  
  
 // 0. Back  
 case 0:  
 return true;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 return false;  
 }  
 return false;  
 }  
  
 public static void seeCurrentCart(){  
  
 *order*.printCart();  
 boolean back= false;  
  
 while (!back) {  
 *cartMenu*();  
 back = *solveOptionOfCartMenuSelected*();  
 }  
  
  
 }  
 //===================================================================================//  
  
 //===================================Menu Main==========================================//  
 public static void solveOptionSelected(){  
 switch (*readOption*()){  
  
 // 1. View store  
 case 1:  
 *viewStore*();  
 break;  
  
 // 2. Update store  
 case 2:  
 *updateStore*();  
 break;  
  
 // 3. See current cart  
 case 3:  
 *seeCurrentCart*();  
 break;  
  
 // 0. Exit  
 case 0:  
 System.*out*.println("Are you sure you want to exit the system ? (y/n)");  
 if(*sc*.next().charAt(0) == 'y'){  
 System.*exit*(0);  
 *sc*.close();  
 }  
 else {  
 break;  
 }  
 break;  
  
 default:  
 System.*out*.println("========================================");  
 System.*out*.println("===>>>This option is not available<<<===");  
 System.*out*.println("========================================");  
 break;  
 }  
 }  
  
 public static int readOption(){  
 return *sc*.nextInt();  
 }  
  
 public static void initData(){  
  
 // Book  
 Book initBook = new Book("abc", "Book", 100.00f,  
 Arrays.*asList*("h", "ani"));  
  
 // Book  
 Book initBook2 = new Book("xyz", "Book", 85.00f,  
 Arrays.*asList*("a", "ki"));  
  
 // Book  
 Book initBook3 = new Book("xyz2", "Book", 197.00f,  
 Arrays.*asList*("u", "ai"));  
  
 // DVD  
 DigitalVideoDisc initDVD = new DigitalVideoDisc("DVD 1", "DVD",  
 "abc ", 150, 35.88f);  
  
 // CD  
 ArrayList<Track> tracks = new ArrayList<>();  
 tracks.add(new Track("GD", 3));  
 tracks.add(new Track("MCK", 4));  
 CompactDisc initCD = new CompactDisc("IU", "CD", 90.00f,  
 "YG", tracks);  
  
 *store*.addMedia(initBook);  
 *store*.addMedia(initBook2);  
 *store*.addMedia(initBook3);  
 *store*.addMedia(initCD);  
 *store*.addMedia(initDVD);  
 }  
  
 //===================================================================================//  
  
 public static void main(String[] args) {  
  
 *initData*();  
 while(true){  
 *showMenu*();  
 *solveOptionSelected*();  
 }  
 }  
}