Báo cáo thực hành OOP Lab 04

Lục Minh Đức – 20225810

1. Tạo lớp Book:

```
import java.util.*;
public class Book {
    private int id;
    private String title;
    private String category;
    private float cost;
    private List<String> authors = new ArrayList<String>();

    public Book() {
    }

    public void addAuthor (String authorName)
    {
        if ( !authors.contains(authorName))
        {
            authors.add(authorName);
            System.out.println("Successfully added");
        }
        else System.out.println("Already listed");
    }

    public void removeAuthor (String authorName)
    {
        if ( authors.contains(authorName))
        {
            authors.remove(authorName);
            System.out.println("Successfully removed");
        }
        else System.out.println("Not on the list");
    }
```

2. Tạo lớp trừu tượng Media:

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

public abstract class Media {
    private int id;
    private String title;
    private String category;
    private float cost;

    public Media(int id, String title, String category, float cost) {
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }
    public int getId() {
        return id;
    }
    public String getTitle() {
        return title;
    }
    public String getCategory() {
        return category;
    }
    public float getCost() {
        return cost;
    }
    public void setId(int id) {
        this.id = id;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public void setCost(float cost) {
        this.cost = cost;
    }
}
```

Sửa lại lớp Book:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class Book extends Media {

    private List<String> authors = new ArrayList<String>();

    public Book(int id, String title, String category, float cost,
            List<String> authors) {
        super(id, title, category, cost);
        this.authors = authors;
    }

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

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

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

    public void addAuthor (String authorName)
    {
        if ( !authors.contains(authorName))
        {
            authors.add(authorName);
            System.out.println("Successfully added");
        }
        else System.out.println("Already listed");
    }

    public void removeAuthor (String authorName)
    {
        if ( authors.contains(authorName))
        {
            authors.remove(authorName);
            System.out.println("Successfully removed");
        }
        else System.out.println("Not on the list");
    }
}
```

Sửa lại lớp DigitalVideoDisc:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class DigitalVideoDisc extends Media {
    private String director;
    private int length;
    private static int nbDigitalVideoDiscs = 0;

    public DigitalVideoDisc(String director, int length, int id, String title,
            String category, float cost) {
        super(id, title, category, cost);
        this.director = director;
        this.length = length;
    }

    public DigitalVideoDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }

    public String getDirector() {
        return director;
    }

    public int getLength() {
        return length;
    }

    public void setDirector(String director) {
        this.director = director;
    }

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

    @Override
    public String toString() {
        return "DVD - " + this.getTitle() + " - " + this.getCategory()
                + " - " + this.getDirector() + " - " + this.getLength()
                + ": " + this.getCost()+"$";
    }
}
```

3. Tạo lớp CompactDisc:

3.1. Tạo lớp Disc kế thừa lớp Media:

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

public class Disc extends Media {
    private float length;
    private String director;

    public Disc(int id, String title,
            String category, float cost, float length, String director) {
        super(id, title, category, cost);
        this.length = length;
        this.director = director;
    }

    public Disc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }

    public float getLength() {
        return length;
    }

    public String getDirector() {
        return director;
    }
}
```

Sửa lại lớp DigitalVideoDisc:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class DigitalVideoDisc extends Disc {
    private static int nbDigitalVideoDiscs = 0;

    public DigitalVideoDisc(int id, String title, String category, float cost,
            float length, String director) {
        super(id, title, category, cost, length, director);
        nbDigitalVideoDiscs++;
    }

    public DigitalVideoDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
        nbDigitalVideoDiscs++;
    }

    @Override
    public String toString() {
        return "DVD - " + this.getTitle() + " - " + this.getCategory()
                + " - " + this.getDirector() + " - " + this.getLength()
                + ": " + this.getCost()+"$";
    }
}
```

3.2. Tạo lớp Track:

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

public class Track {
    private String title;
    private int length;

    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }

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

}
```

3.3. Sửa lớp CompactDisc:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class CompactDisc extends Disc{
    private String artist;
    private ArrayList<Track> tracks;
    public String getArtist() {
        return artist;
    }
    public CompactDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }
    public CompactDisc(int id, String title, String category, float cost, int length,
            String director, String artist, ArrayList<Track> tracks) {
        super(id, title, category, cost, length, director);
        this.artist = artist;
        this.tracks = tracks;
    }
    public void addTrack( Track tck){
        if ( tracks.contains(tck)){
            System.out.println("Already in the list!");
        }
        else{
            tracks.add(tck);
            System.out.println("Successfully added!");
        }
    }
    public void removeTrack( Track tck){
        if ( tracks.contains(tck)){
            tracks.remove(tck);
            System.out.println("Successfully removed");
        }
        else{
            System.out.println("Not on the list!");
        }
    }
    @Override
    public int getLength(){
        int lengthSum = 0;
        for ( Track x : tracks){
            lengthSum += x.getLength();
        }
        return lengthSum;
    }
}
```

4. Tạo interface Playable:

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

Thêm Playable vào lớp DigitalVideoDisc:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class DigitalVideoDisc extends Disc implements Playable{
    private static int nbDigitalVideoDiscs = 0;

    public DigitalVideoDisc(int id, String title, String category, float cost,
            int length, String director) {
        super(id, title, category, cost, length, director);
        nbDigitalVideoDiscs++;
    }

    public DigitalVideoDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
        nbDigitalVideoDiscs++;
    }

    @Override
    public String toString() {
        return "DVD - " + this.getTitle() + " - " + this.getCategory()
                + " - " + this.getDirector() + " - " + this.getLength()
                + ": " + this.getCost()+"$";
    }

    @Override
    public void play()
    {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
}
```

Thêm Playable vào lớp Track:

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

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

    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }

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

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

}
```

Thêm Playable vào lớp CompactDisc:

```
package hust.soict.hedspi.aims.media;
import java.util.*;
public class CompactDisc extends Disc implements Playable{
    private String artist;
    private ArrayList<Track> tracks;
    public String getArtist() {
        return artist;
    }
    public CompactDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }
    public CompactDisc(int id, String title, String category, float cost, int length,
            String director, String artist, ArrayList<Track> tracks) {
        super(id, title, category, cost, length, director);
        this.artist = artist;
        this.tracks = tracks;
    }
    public void addTrack( Track tck){
        if ( tracks.contains(tck)){
            System.out.println("Already in the list!");}
        else{
            tracks.add(tck);
            System.out.println("Successfully added!");}
    }
    public void removeTrack( Track tck){
        if ( tracks.contains(tck)){
            tracks.remove(tck);
            System.out.println("Successfully removed");}
        else{
            System.out.println("Not on the list!");}
    }
    @Override
    public int getLength(){
        int lengthSum = 0;
        for ( Track x : tracks){
            lengthSum += x.getLength();
        }
        return lengthSum;
    }
    @Override
    public void play() {
        for ( Track x : tracks)
        {
            x.play();
        }
    }
}
```

5. Cập nhật lớp Cart để hoạt động với Media:

```
package hust.soict.hedspi.aims.cart;
import hust.soict.hedspi.aims.media.DigitalVideoDisc;
import hust.soict.hedspi.aims.media.Media;
import java.util.*;

public class Cart {
    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();

    public Cart() {
    }

    public void addMedia(Media med) {
        if ( itemsOrdered.contains(med))
        {
            System.out.println("Already in the cart!");
        }
        else
        {
            itemsOrdered.add(med);
            System.out.println("Successfully added!");
        }
    }

    public void removeMedia(Media med) {
        if ( itemsOrdered.contains(med))
        {
            itemsOrdered.remove(med);
            System.out.println("Successfully removed!");
        }
        else
        {
            System.out.println("Not on the list!");
        }
    }

    public float totalCost ()
    {
        float sum = 0;
        for ( Media x : itemsOrdered)
        {
            sum += x.getCost();
        }
        return sum;
    }
}
```

6. Cập nhật lớp Store để hoạt động với Media:

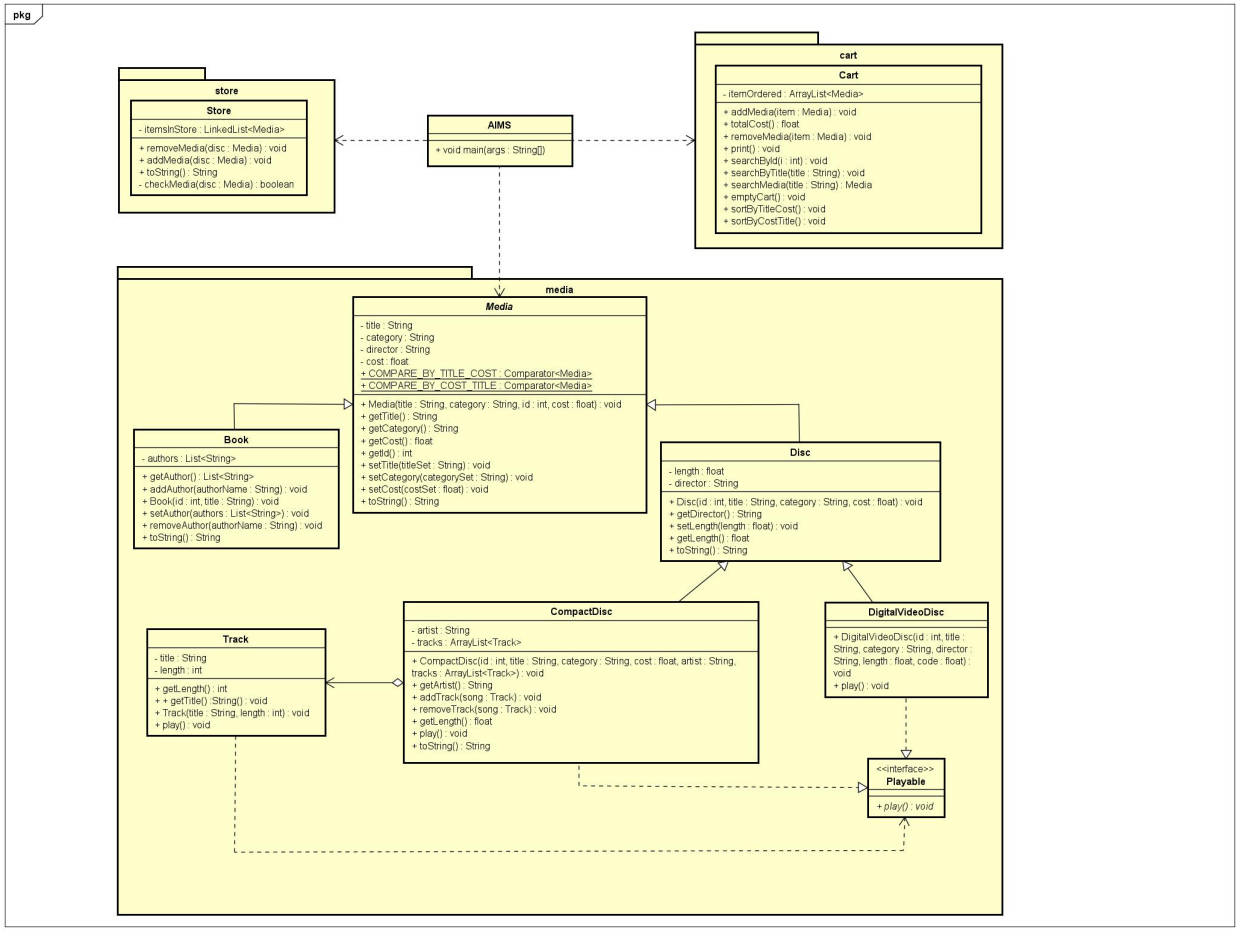
```
package hust.soict.hedspi.aims.store;
import hust.soict.hedspi.aims.media.Media;
import java.util.*;
public class Store {
    private ArrayList<Media> itemsInStore = new ArrayList<Media>();

    public Store() {
    }

    public void addMedia(Media med) {
        if ( itemsInStore.contains(med))
        {
            System.out.println("Already in the store!");
        }
        else
        {
            itemsInStore.add(med);
            System.out.println("Successfully added!");
        }
    }

    public void removeMedia(Media med) {
        if ( itemsInStore.contains(med))
        {
            itemsInStore.remove(med);
            System.out.println("Successfully removed!");
        }
        else
        {
            System.out.println("Not in the store!");
        }
    }
}
```

7. Cập nhật lại UML class diagram:



8. Item độc nhất trong danh sách:

Overriding boolean equals(Object o) của Media:

```
@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Media other = (Media) obj;
    if (!Objects.equals(this.title, other.title)) {
        return false;
    }
    return true;
}
```

Overriding boolean equals(Object o) của Track:

```
@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Track other = (Track) obj;
    if (this.length != other.length) {
        return false;
    }
    if (!Objects.equals(this.title, other.title)) {
        return false;
    }
    return true;
}
```

Câu hỏi: If the passing object is not an instance of Media, what happens?

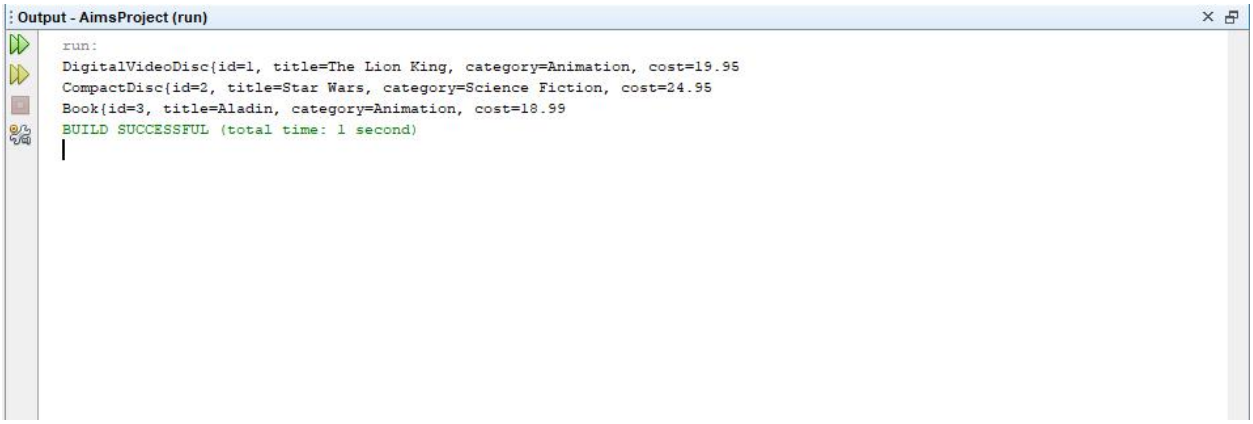
Hàm có thể sẽ hoạt động sai. Nên ta cần kiểm tra lớp của đối tượng trước khi so sánh.

9. Phương thức toString():

Sample code;

```
public static void main(String[] args) {
List <Media> mediaList = new ArrayList<Media>();
DigitalVideoDisc dvd = new DigitalVideoDisc(1, "The Lion King",
        "Animation", 19.95f);
CompactDisc cd = new CompactDisc(2, "Star Wars",
        "Science Fiction", 24.95f);
Book book = new Book(3, "Aladin",
        "Animation", 18.99f);
mediaList.add(dvd);
mediaList.add(cd);
mediaList.add(book);
for ( Media m : mediaList)
{
    System.out.println(m.toString());
}
}
```

Kết quả:



10. Sắp xếp Media trong giỏ hàng (cart):

Tạo lớp MediaComparatorByCostTitle:

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

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {

    @Override
    public int compare(Media t, Media t1) {
        if (Double.compare(t.getCost(), t1.getCost()) !=0 )
        {
            return Double.compare(t.getCost(), t1.getCost());
        }
        else
        {
            return t.getTitle().compareTo(t1.getTitle());
        }
    }
}
```

Tạo lớp MediaComparatorByTitleCost:

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

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {

    @Override
    public int compare(Media t, Media t1) {
        if (t.getTitle().compareTo(t1.getTitle()) !=0 )
        {
            return t.getTitle().compareTo(t1.getTitle());
        }
        else
        {
            return Double.compare(t.getCost(), t1.getCost());
        }
    }
}
```

Thêm vào lớp Media:

```
public abstract class Media {
    private int id;
    private String title;
    private String category;
    private float cost;

    public static final Comparator<Media> COMPARE_BY_COST_TITLE =
            new MediaComparatorByCostTitle();

    public static final Comparator<Media> COMPARE_BY_TITLE_COST =
            new MediaComparatorByTitleCost();
```

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?
Answer: Lớp nên triển khai giao diện Comparable là lớp chứa đối tượng mà bạn muốn so sánh, trong trường hợp này, là abstract class “Media”

- In those classes, how should you implement the compareTo()method be to reflect the ordering that we want?
Answer: Để triển khai phương thức compareTo() thì cần so sánh các thuộc tính của đối tượng. Đối với yêu cầu sắp xếp theo title sau đó là cost, có thể triển khai như sau:

```
@Override
public int compareTo(Media other) {
if (!this.getTitle().equals(other.getTitle())) {
    return this.getTitle().compareTo(other.getTitle());
} else {
    return Double.compare(this.getCost(), other.getCost());
}
```

- 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?
Answer: Không, với giao diện Comparable, bạn chỉ có thể có một quy tắc sắp xếp cho mỗi lớp. Điều này là do phương thức compareTo() chỉ trả về một giá trị int.

- 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?
Answer: Có thể tạo một lớp riêng cho DVD và triển khai phương thức compareTo():

```
public int compareTo(DigitalVideoDisc other) {
    if (this.getTitle().equals(other.getTitle())) {
        return this.getTitle().compareTo(other.getTitle());
    } else if (this instanceof DigitalVideoDisc && other instanceof DigitalVideoDisc ) {
        return Integer.compare(((DigitalVideoDisc ) other).getLength(), ((DigitalVideoDisc ) this).getLength());
    } else {
        return Double.compare(this.getCost(), other.getCost());
    }
}
```

11. Create a complete console application in the Aims class:

```
package hust.soict.hedspi.aims;

import hust.soict.hedspi.aims.cart.Cart;
import hust.soict.hedspi.aims.media.DigitalVideoDisc;
import hust.soict.hedspi.aims.media.Book;
import hust.soict.hedspi.aims.media.CompactDisc;
import hust.soict.hedspi.aims.media.Media;
import hust.soict.hedspi.aims.media.Track;
import hust.soict.hedspi.aims.store.Store;
import java.util.*;

public class Aims {

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Store str = new Store();
      Cart crt = new Cart();
      showMenu(str, crt);
   }



   public static void showMenu(Store str, Cart crt)
   {
      while (true)
      {
```

```
Scanner sc = new Scanner(System.in);
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");

int x = sc.nextInt();
switch(x)
{
   case 1:
   {
      //view store
      storeMenu(str, crt);
   }
   case 2:
   {
      //update store
      System.out.println("Options: ");
      System.out.println("--------------------------------");
      System.out.println("1. Add media");
      System.out.println("2. Delete media");
      System.out.println("--------------------------------");
      System.out.println("Please choose a number: 0-1-2");
      int option = sc.nextInt();
switch (option) {
   case 1: {
      System.out.println("1. DigitalVdieoDisc");
      System.out.println("2. CompactDisc");
      System.out.println("3. Book");
      int option2 = sc.nextInt();
      System.out.print("Enter id: ");
      int id = sc.nextInt();
      sc.nextLine();
      System.out.print("Enter title: ");
      String title = sc.nextLine();
      System.out.print("Enter category: ");
```

```
String category = sc.nextLine();
System.out.print("Enter cost: ");
float cost = sc.nextFloat();
sc.nextLine();
switch (option2) {
   case 1:
   {
      System.out.print("Enter director's name: ");
      String director = sc.nextLine();
      System.out.print("Enter dvd's length: ");
      int length = sc.nextInt();
      sc.nextLine();
      str.addMedia(new DigitalVideoDisc(id,title,category,cost,length,director));
   }
   case 3: {
      System.out.print("Enter author's name (Enter nothing to skip): ");
      StringBuilder author = new StringBuilder(sc.nextLine());
      ArrayList<String> authors = new ArrayList<String>();
      while (!authors.isEmpty()) {
         authors.add(author.toString());
         System.out.print("Enter author's name (Enter nothing to skip): ");

      }
      str.addMedia(new Book(id,title,category,cost,authors));
   }
   case 2: {
      System.out.print("Enter artist's name: ");
      StringBuffer artist = new StringBuffer(sc.nextLine());
      System.out.print("Enter number of track: ");
      int nbTrack = sc.nextInt();sc.nextLine();
      ArrayList<Track> tracks = new ArrayList<Track>();
      StringBuilder name = new StringBuilder();
      for(int i = 0;i< nbTrack;i++) {
         System.out.print("Enter Track[" + i + "]'s name: ");
         name.replace(0,name.length(),sc.nextLine());
         System.out.print("Enter Track[" + i + "]'s length: ");
         int length = sc.nextInt();
         tracks.add(new Track(name.toString(), length));
         sc.nextLine();
```

```
                }
                str.addMedia(new CompactDisc(id,title,category,cost,artist.toString(),tracks));
            }
        }
    }
    case 2: {
        System.out.println("Enter item's title: ");
        sc.nextLine();
        String title = sc.nextLine();
        Iterator<Media> iter = str.getItemsInStore().iterator();
        while (iter.hasNext()) {
            Media item = iter.next();
            if(item.getTitle().equals(title)) {
                iter.remove();
                System.out.println(item.getClass().getSimpleName() + " " + item.getTitle() + "'ve
been deleted from the store !");
            }
        }
    }
    case 3:
    {
        // see current cart
        crt.printCart();
        cartMenu(str, crt);
    }
    case 0:
    {
        return;
    }
}
    }
}
}

}
    public static void storeMenu(Store str, Cart crt)
{
    while(true)
    {
    Scanner sc = new Scanner(System.in);
```

```
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");
int option = sc.nextInt();
sc.nextLine();
if ( option == 1)
   {
      System.out.println("Enter the title: ");
      String mediaTitle = sc.nextLine();
      Media med = str.searchMedia(mediaTitle);
      if ( med == null)
      {
         System.out.println(mediaTitle+" not in the store!");
      }
      else
      {
         med.toString();
         mediaDetailsMenu(str, crt, med);
      }
   }
else if ( option == 2)
   {
      System.out.println("Enter the title: ");
      String mediaTitle = sc.nextLine();
      Media med = str.searchMedia(mediaTitle);
      if ( med == null)
      {
         System.out.println(mediaTitle+" is not valid!");
      }
      else
      {
         crt.addMedia(med);
         if ( med instanceof DigitalVideoDisc)
         System.out.println(crt.getNumberDVDs());
```

```
            }
        }
    else if ( option == 3)
        {
            System.out.println("Enter the title: ");
            String mediaTitle = sc.nextLine();
            Media med = str.searchMedia(mediaTitle);
            if ( med == null)
            {
                System.out.println(mediaTitle+" is not valid!");
            }
            else
            {
                if ( med instanceof DigitalVideoDisc)
                {
                    ((DigitalVideoDisc) med).play();
                }
                if ( med instanceof CompactDisc)
                {
                    ((CompactDisc) med).play();
                }
            }
        }
    else if ( option == 4)
        {
            crt.printCart();
            cartMenu(str, crt);
        }
    else if ( option == 5)
        {
            return;
        }

    }
}

public static void mediaDetailsMenu(Store store, Cart crt, Media med) {
    while(true)
    {
    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");
    Scanner sc = new Scanner(System.in);
    int option = sc.nextInt();
    if ( option == 1)
            {
               crt.addMedia(med);
               System.out.println(med.getTitle()+" has been added to cart!");
            }
    else if ( option == 2)
            {
               if ( med instanceof DigitalVideoDisc)
               {
                  ((DigitalVideoDisc) med).play();
               }
               if ( med instanceof CompactDisc)
               {
                  ((CompactDisc) med).play();
               }
            }
    else if ( option == 0)
            {
               return;
            }
    }
}

public static void cartMenu(Store str, Cart crt) {
    while(true)
    {
    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");
Scanner sc = new Scanner(System.in);
int option = sc.nextInt();
if ( option == 3)
   {
      System.out.println("Enter the title: ");
      String mediaTitle = sc.nextLine();
      Media med = str.searchMedia(mediaTitle);
      if ( crt.getItemsOrdered().contains(med))
      {
         crt.removeMedia(med);
         System.out.println(mediaTitle+" has been removed from cart!");
      }
      else
      {
         System.out.println(mediaTitle+" not in the cart!");
      }
   }
else if ( option == 4)
   {
      System.out.println("Enter the title: ");
      String mediaTitle = sc.nextLine();
      Media med = str.searchMedia(mediaTitle);
      if ( crt.getItemsOrdered().contains(med))
      {
         if ( med instanceof DigitalVideoDisc)
         {
            ((DigitalVideoDisc) med).play();
         }
         if ( med instanceof CompactDisc)
         {
            ((CompactDisc) med).play();
         }
      }
   }
else if ( option == 5)
   {
```

```
            System.out.println("Order has been placed!");
            crt.getItemsOrdered().removeAll(crt.getItemsOrdered());
        }
      else if ( option == 0)
        {
            return;
        }
      }
   }
}
```