

9. Constructors of whole classes and parent classes

- Which classes are collections of other classes? Check all constructors of the main classes to see if they initialize their parts.

Collections:

- Store contains Media.
- Cart contains Media.
- CompactDisc contains Track.

Store Class

- **Attributes:** Likely contains a collection of Media objects.
- **Constructor:** Initializes the list of Media.
- **Aggregation:** The Store class contains Media objects because Media can exist on its own, without the Store.

Cart Class

- **Attributes:** Likely contains a collection of Media objects.
- **Constructor:** Initializes the list of Media.
- **Aggregation:** The Cart class contains Media objects, similar to the Store.

Disc Class

- **Attributes:** May have extra information like length and director.
- **Constructor:** Sets up properties for Disc, and through inheritance, also initializes Media attributes.
- **Aggregation:** Does not contain any other objects but inherits from Media.

CompactDisc Class

- **Attributes:** Contains a list of tracks and an artist.
- **Constructor:** Likely initializes the list of tracks.
- **Aggregation:** The CompactDisc class contains tracks because tracks can exist independently of the CompactDisc.

Track Class

- **Attributes:** Title and length.
- **Constructor:** Initializes these properties.
- **Aggregation:** Not a collection class since it doesn't contain any other objects.

DigitalVideoDisc Class

- **Attributes:** Inherits attributes and methods from Disc.
- **Constructor:** Sets specific properties for DigitalVideoDisc and also initializes inherited properties.
- **Aggregation:** None; it directly inherits from Disc.

10. What happens if the passed object is not an instance of Media?

- If the object passed to the equals() method is not an instance of Media or Track, it will return false. This makes sure the type is correct and avoids ClassCastException (if type checking isn't handled properly).

12. Sorting media in the cart

Question: Instead of using Comparator to compare items in the cart, we can use the Comparable interface and override the compareTo() method. You can check the Java documentation for information on this interface.

Suppose we use the Comparable interface approach.

- **Which class should implement the Comparable interface?**
 - The Media class should implement the Comparable interface because we want a default ordering for media objects.
- **How should you implement the compareTo() method in those classes to reflect the ordering we want?**
- **Can we have two ordering rules for items (like by title then cost, and by cost then title) if we use the Comparable interface?**
 - No, it is very hard because the Comparable interface allows only one natural ordering for a class. If we need multiple ordering rules (such as sorting by title or cost), we should use Comparator instead.

```

@Override
public abstract String toString();

@Override
public boolean equals(Object o) {
    if(o == null || !(o instanceof Media)) {
        return false;
    }
    Media other = (Media) o;
    return this.getTitle() != null && this.title.equals(other.getTitle());
}

public int compareTo(Media other) {
    int titleComparison = this.title.compareTo(other.title);
    if (titleComparison != 0) {
        return titleComparison;
    }
    // If titles are the same, compare by cost (higher cost first)
    return Double.compare(other.cost, this.cost);
}
}

```

-
- If DVDs have a different sorting rule from other media types, which is by title, then decreasing length, then cost, how would you modify the code to support this?
 - Since DVDs have a different sorting rule from other media types, we can override the compareTo() method in the DigitalVideoDisc class. Then, DVDs will be compared using the overridden compareTo() method.

```
1 package compare;
2
3 import hust.soict.dsai.aims.cart.Cart;
4 import hust.soict.dsai.aims.media.Book;
5 import hust.soict.dsai.aims.media.CompactDisc;
6 import hust.soict.dsai.aims.media.DigitalVideoDisc;
7
8 public class CompareTest {
9     public static void main(String[] args) {
10         Cart cart = new Cart();
11
12         // create some media here
13
14         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
15             "Animation", "Roger Allers", 87, 19.95f);
16
17         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
18             "Science Fiction", "George Lucas", 87, 24.95f);
19
20         Book b1 = new Book("1 Van Cau Hoi Vi sao", "Truyen", 80.25f);
21         Book b2 = new Book("So Dua", "Truyen Co Tich", 40.5f);
22
23         // cart.addMedia(cd1);
24         // cart.addMedia(cd2);
25         cart.addMedia(dvd1);
26         cart.addMedia(dvd2);
27         cart.addMedia(b1);
28         cart.addMedia(b2);
29
30         // Sort by Title, then Cost
31         cart.sortByTitleCost();
32
33         System.out.println("");
34
35         // Sort by Cost, then Title
36         cart.sortByCostTitle();
37     }
38 }
```

<terminated> CompareTest [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Dec 2, 2024)

Sorted by Title, then Cost:
DVD-The Lion King-Animation-Roger Allers-87: 19.95\$
DVD-Star Wars-Science Fiction-George Lucas-87: 24.95\$
Book-So Dua-Truyen Co Tich: 40.5\$
Book-1 Van Cau Hoi Vi sao-Truyen: 80.25\$

Sorted by Cost, then Title:
Book-1 Van Cau Hoi Vi sao-Truyen: 80.25\$
Book-So Dua-Truyen Co Tich: 40.5\$
DVD-Star Wars-Science Fiction-George Lucas-87: 24.95\$
DVD-The Lion King-Animation-Roger Allers-87: 19.95\$

```
// Handle compare for DVD with different ordering rule
@Override
public int compareTo(Media other) {
    if (!(other instanceof DigitalVideoDisc)) {
        return super.compareTo(other);
    }

    DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;
    int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());
    if (titleComparison != 0) {
        return titleComparison;
    }

    int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength());
    if (lengthComparison != 0) {
        return lengthComparison;
    }

    return Double.compare(this.getCost(), otherDVD.getCost());
}
```