LAB 04

9. Constructors of whole classes and parent classes

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

- Aggregates:
  - **Store** aggregates **Media**.
  - **Cart** aggregates **Media**.
  - **CompactDisc** aggregates **Track**.
- **Store Class**
  - **Attributes**: Likely contains a collection of Media objects.
  - **Constructor**: Initializes the list of Media.
  - **Aggregation**: The Store class aggregates Media objects because Media can exist independently of the Store.
- **Cart Class**
  - **Attributes**: Likely contains a collection of Media objects.
  - **Constructor**: Initializes the list of Media.
  - **Aggregation**: The Cart class aggregates Media objects for the same reason as Store.
- **Disc Class**
  - **Attributes**: May contain additional details like length and director.
  - **Constructor**: Sets properties for Disc, and indirectly via inheritance, initializes Media attributes.
  - **Aggregation**: Aggregates no separate objects but inherits from Media.
- **CompactDisc Class**
  - **Attributes**: Contains a List<Track> and an artist.
  - **Constructor**: Likely initializes the List<Track>.
  - **Aggregation**: The CompactDisc aggregates Track because Track instances can exist independently of a CompactDisc.
- **Track Class**
  - **Attributes**: Title and length.
  - **Constructor**: Initializes these properties.
  - **Aggregation**: Not an aggregate class since it contains no other objects.

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

10. If the passing object is not an instance of Media, what happens?

- If the object passed to **equals()** is not an instance of **Media** or **Track**, I will returns false. This ensures type safety and avoids **ClassCastException** (in case of not handling type checking).

12. Sort media in the cart

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?

- The **Media** class should implement the **Comparable interface** since we want to define a default ordering for **media objects**.

- 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?

- No, it's very hard because the **Comparable** interface allows only one natural ordering for a class. If we need multiple ordering rules (like sorting by title or cost), I think we must replace with **Comparator** instead.

- 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?

- As DVDs has different ordering rule from other media types, I think we can override **compareTo()** method in **DigitalVideoDisc** class. After that, DVDs will be compare by the overridden **compareTo()** method.

AimsProject > src > hust > soict > dsai > aims > media > DigitalVideoDisc.java > Language Support for Java(TM) by Red Hat > DigitalVideoDisc >

```java
public class DigitalVideoDisc extends Disc implements Playable {

    // Handle compare for DVD with difference ordering rule      You, 1 second ago • Uncommitted changes
    @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());
    }
}
```

Test compare:

```java
package hust.soict.dsai.test.compare;

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.Book;
import hust.soict.dsai.aims.media.CompactDisc;
import hust.soict.dsai.aims.media.DigitalVideoDisc;

public class CompareTest {
    public static void main(String[] args) {
        Cart cart = new Cart();

        // create some media here
        // for example: cd, dvd, book
        CompactDisc cd1 = new CompactDisc("Son Tung MTP", "Album Special", "Classic", "Son Tung MTP", 24.55f);
        CompactDisc cd2 = new CompactDisc("Amee");

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
                "Animation", "Roger Allers", 87, 19.95f);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
                "Science Fiction", "George Lucas", 87, 24.95f);

        Book b1 = new Book("1 Van Cau Hoi Vi sao", "Truyen", 80.25f);
        Book b2 = new Book("So Dua", "Truyen Co Tich", 40.5f);

        cart.addMedia(cd1);
        cart.addMedia(cd2);
        cart.addMedia(dvd1);
        cart.addMedia(dvd2);
        cart.addMedia(b1);
        cart.addMedia(b2);

        // Sort by Title, then Cost
        cart.sortByTitleCost();

        System.out.println("");

        // Sort by Cost, then Title
        cart.sortByCostTitle();
    }
}
```

```
Compact Disc-Amee---0: 0.0$
Book-1 Van Cau Hoi Vi sao-Truyen: 80.25$
Compact Disc-Son Tung MTP-Classic-Son Tung MTP-0: 24.55$
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$

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$
Compact Disc-Son Tung MTP-Classic-Son Tung MTP-0: 24.55$
DVD-The Lion King-Animation-Roger Allers-87: 19.95$
Compact Disc-Amee---0: 0.0$
```