

## Music Collection

### Class CD

Implement the CD class which defines the properties of objects that we will use later to represent CDs. The table below gives the UML description of the CD class. Recall that, in UML, the following notation:

```
+ setTitle(title : String) : void
```

indicates that `setTitle` is a public (- means private) method that accepts a `String` parameter called `title` and returns nothing (`void`).

CD
- title : String - artist : String - cost : double - tracks : int
+ CD (title : String, artist : String, cost : double, tracks : int) + setTitle(title : String) : void + setArtist(artist : String) : void + setCost(cost : double) : void + setTracks(tracks : int) : void + getTitle() : String + getArtist() : String + getCost() : double + getTracks() : int + equals(cd : CD) : boolean + compareTo(cd : CD) : int + toString() : String

The CD class defines four instance variables. It has one constructor that initializes the instance variables using the passed parameters. It also defines appropriate setters and getters as noted in the UML representation above. Note that your code must observe the naming convention specified above. For example, in the `setTitle` method, the name of the parameter must be `title`.

The CD class defines a `compareTo` method that compares two CDs according to their titles first and their artists next. It returns a negative number (e.g., -1) if the current CD has a (`title`, `artist`) pair that is *smaller* than that of the passed CD object, a positive number (e.g., +1) if it is *larger*, and 0 otherwise. It has an `equals` method that also only uses the `title` and `artist` fields to decide if two CD objects have the same content. Finally the CD class defines a `toString` method that generates a printable representation of the current CD's information in a nice format (all information on one line—check the sample output below).

The driver class `CDDriver` is provided to you and can be used to test your CD class.

### Sample Run

```
C:> java CDDriver
CD1: $14.95      10      Storm Front      Billy Joel
CD2: $17.95      33      Soundtrack      Les Miserables
CD3: $19.95      22      Soundtrack      Chicago
CD4: $19.95      24      Soundtrack      Moulin Rouge
```

```
CD1 >= CD2
CD2 >= CD3
CD3 < CD4
```

### Classes CDCollection & CDCollDriver

Implement the `CDCollection` class described in the UML diagram below.

CDCollection
- collection : CD[] // initial size is 2 - count : int

- totalCost : double
+ CDCollection()
+ CDCollection(size: int)
+ add(title : String, artist : String, cost : double, tracks : int) : void
+ add(cd : CD) : void
+ toString() : String
- increaseSize() : void

An instance of the `CDCollection` class contains an array of `CD` objects to store the CDs in the collection. It maintains a count of the CDs in the collection (in the `count` field) and their combined value (in the `totalCost` field). It uses the `CD` count to decide when to create a larger array if too many CDs are added to the collection.

- The collection array is instantiated in two different ways: (1) using the `CDCollection` no-arg constructor which initializes the collection array to have a size of 2 or (2) using the `CDCollection` one argument constructor which initializes the collection array and sets its size to the passed value.
- Every time a `CD` is added to the collection (using one of the `add` methods), a new `CD` object is created and a reference to it is stored in the first available slot in the collection array. Each time a `CD` is added to the collection, we check to see if we have reached the current capacity of the collection array. If the current capacity has been reached, the private `increaseSize` method is invoked, which doubles the size of the collection array.

The `toString` method of the `CDCollection` class returns an entire report summarizing the collection. The report should be created using implicit calls to the `toString` method of each `CD` object stored in the collection.

For example, in the `CDCollDriver` driver class (which is provided to you), when `System.out.println (music);` is called for the first time, the following output should be displayed:

**Sample Run**

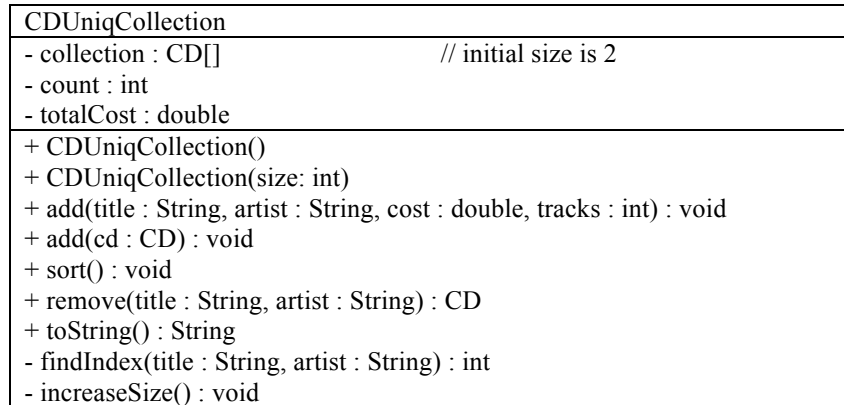
```
C:> java CDCollDriver
My CD Collection
```

```
Number of CDs: 5
Total value:  $80.74
Average cost: $16.15
```

```
CD List:
$14.95  10      Storm Front      Billy Joel
$14.95  16      Come On Over    Shania Twain
$17.95  33      Soundtrack      Les Miserables
$13.90  11      Graceland       Paul Simon
$18.99  18      Ella in Rome    Ella Fitzgerald
```

### Classes CDUniqCollection & Tunes

Implement the CDUniqCollection class described in the UML diagram below.



The CDUniqCollection class is very similar to the CDCollection class and differs from it as follows:

- It does not add a CD to the collection if it already has one with the same title and artist. The private method `findIndex` can be used for this purpose.
- It has a private method `findIndex` that can be used to check if the CD already exists in the collection. It accepts as parameters a CD title and artist and then searches the collection array looking for a CD with the same title and artist. If such a CD were to exist, it returns the index of that CD in the collection array, otherwise it returns -1.
- The `sort` method of the CDCollection class sorts all the CDs in the collection according to their (title, artist) fields in ascending order.
- The `remove` method of the CDCollection class accepts the title and author of a certain CD, removes the corresponding CD from the collection, and returns the removed CD. Before removing the CD, you need to make sure that the CD exists in the collection. The private method `findIndex` can be used for this purpose (similar to what was described in the `add` method). Note that when you remove a CD from the collection array, all CDs that are to its right must be shifted one place to left. Also, the current size of the collection array must be decreased by 1.

The Tunes driver class is provided to you. Its sample run is shown below.

#### Sample Run

```
C:> java Tunes
My Unique CD Collection

Number of CDs: 4
Total value: $61.75
Average cost: $15.44

CD List:
$14.95 16      Come On Over      Shania Twain
$13.90 11      Graceland         Paul Simon
$17.95 33      Soundtrack        Les Miserables
$14.95 10      Storm Front       Billy Joel

Remove 1 CD: $14.95      10      Storm Front      Billy Joel
Add 3 CDs

My Unique CD Collection

Number of CDs: 6
Total value: $98.54
```

Average cost: \$16.42

CD List:

\$14.95	16	Come On Over	Shania Twain
\$13.90	11	Graceland	Paul Simon
\$17.95	33	Soundtrack	Les Miserables
\$15.80	13	Cedarland	Walid Keirouz
\$19.99	26	Double Live	Garth Brooks
\$15.95	13	Greatest Hits	Wassim El-Hajj

Remove: \$15.80 13 Cedarland Walid Keirouz

My Unique CD Collection

Number of CDs: 5

Total value: \$82.74

Average cost: \$16.55

CD List:

\$14.95	16	Come On Over	Shania Twain
\$19.99	26	Double Live	Garth Brooks
\$13.90	11	Graceland	Paul Simon
\$15.95	13	Greatest Hits	Wassim El-Hajj
\$17.95	33	Soundtrack	Les Miserables