# TA (Textual Analysis) for UML MVC CLASS Diagram MODEL

**for**

**Movie Company System**

**View**

**MovieCompanySystem Controller**

# View 1: showSelection():void

1. The **Movie Company** has the following affiliated Artists (maximum 10):

## “Miles Davis”, with 3 Discs:

**Catalog 1: MAX\_ARTISTS** = **10**

1. Disc 1: **Title** “Miles Ahead” with Track “Miles Ahead” that is 3.34 minutes
2. Disc 2: **Title** “Tutu with Tracks: “Tutu”, 4.56, “Portia”, 3.34, “Tomaas”, 3.56
3. Disc 3: **Title** “Kind of Blue” with Tracks “So What”, 4.54, “Freddie Freeloader, 5.25,
4. “Blue in Green”, 2.56

## “John Coltrane”, with 1 Disc:

1. Disc 1: **Title** “Blue Train” with Blue Train” that is 5.15 minutes; “Moments Notice”,

10 5.52, “Locomotion”, 6.18

# Catalog

11

# Catalog 6: numberOfDiscs: int Catalog 5: numberOfArtists: int Catalog 4: Disc[MAX\_DISCS] discs

**Catalog 3: Artist[MAX\_ARTISTS] artists**

The **Movie Company** wants to keep a **catalog** of its artists and their discs (maximum

**Artist**

12

# Artist 1: name:String

**Catalog 1: addArtist(**String artist**) void; Controller 1: UCAddArtist(**String artist**) void;**

**Catalog 2: MAX\_DISCS = 30**

30). **Artist**s have a name. The **Movie Company** can ***add an artist*** or get a ***listing of all***

1. ***its artists***.

**Disc**

**Controller 2: UCListArtists() void; Catalog 2: listArtists() void;**

1. Each artist produces music **Disc**s. A Disc is associated with an Artist. A Disc is cataloged **Catalog 3: addDisc(**String artist, String discCatalogNumber, String disc**) void; Controller 3: UCAddDisc(**String artist, String discCatalogNumber, String disc**) void;**

**Disc 2: discCatalogNumber**

**Controller 4: UCListDiscs() void;**

with a disc catalogNumber. The **Movie Company** can ***add a disc*** or get a ***listing of all its***

15

1. ***discs***.

**Track 1: addTrack(Track track) void; Track 2: getTrack(int i): Track;**

**Catalog 4: listDiscs();**

**Disc 3: numberOfTracks**

**Track**

**Track 1: title: String Track 2: time: double**

1. A Disc has a number of **Track**s. The Track have a **title** and a duration **time**. A Disc has

**Catalog 5: addTrack(**String discCatalogNumber, String track, double time**); Controller 5: UCAddTrack(**String discCatalogNumber, String track, double time**);**

1. a catalogNumber , a title, the number of tracks. The **Movie Company** can ***add a track*** or

**Disc 1: MAX\_TRACKS = 10**

1. get a ***listing of all tracks for a given disc***.

**Disc 4: Track[MAX\_TRACKS] tracks**

**Controller 6: UCListTracksForDisc(**String discCatalogNumber**) void; Catalog 6: listTracksForDisc(**String discCatalogNumber**) void;**

# Disc 5: Artist artist

## Controller 7: UCRestoreCatalog() void; Catalog 7: restoreCatalog() void;

**Controller 8: UCSaveCatalog() void; Catalog 8: saveCatalog() void;**



**“has a”**

|  |
| --- |
| **View** |
|  |
| + showSelection( ) : void |

|  |
| --- |
| **Artist** |
| - name: **String //1** |
|  |

|  |  |
| --- | --- |
|  | **Disc** |
| -**MAX\_TRACKS = 10 //1**  -discCatalogNumber: **String //2**  -numberOfTracks: **int //3**   * **tracks**: **Track[MAX\_TRACKS] //4** * **artist**: **Artist //5** |
|  |
| + **addTrack**(**Track** track): void **//1**  + **getTrack(int i)**: **Track //2** |

|  |
| --- |
| **Track** |
| -title: **String //1**  -time: **double //2** |
|  |

**“uses”**

|  |
| --- |
| **MovieCompanySystem** |
| * **control: Controller** * **view**: **View**   +artist: **String**  +disc: **String**  +track:: **String**  +discCatalogNumber: **String**  +time: **double**  +choice: **int** |
| + main ( ) |

|  |
| --- |
| **Controller** |
| - **catalog**: **Catalog** |
| + **UCAddArtist**(String artist): void // UC1  + **UCListArtists**(): void // UC2  + **UCAddDisc**(String artistName, String discCatalogNumber, String discTitle): void // UC3  + **UCListDiscs**(): void // UC4  + **UCAddTrack**(String discCatalogNumber, String trackTitle, double time): void // UC5  + **UCListTracksForDisc**(String discCatalogNumber): void // UC6  + **UCRestoreCatalog**(): void // UC7  + **UCSaveCatalog**(): void // UC8 |

|  |
| --- |
| **Catalog** |
| -**MAX\_ARTISTS = 10 //1**  -**MAX\_DISCS = 30 //2**   * **artists**: **Artist[MAX\_ARTISTS] //3** * **discs**: **Disc[MAX\_DISCS] //4**   #numberOfArtists: **int //5**  #numberOfDiscs: **int //6** |
| + **addArtist**(String artist): void // 1  + **listArtists**(): void // 2  + **addDisc**(String artistName, String discCatalogNumber, String discTitle): void // 3  + **listDiscs**(): void // 4  + **addTrack**(String discCatalogNumber, String trackTitle, double time): void // 5  + **listTracksForDisc**(String discCatalogNumber): void // 6  + **restoreCatalog**(): void // 7  + **saveCatalog**(): void // 8 |

|  |
| --- |
| **MovieCompanySystem** |
| * **control: Controller** * **view**: **View**   +artist: **String**  +disc: **String**  +track:: **String**  +discCatalogNumber: **String**  +time: **double**  +choice: **int** |
| + main ( ) |

public static void main ( ) {

This method will create the **Controller** and **View objects** that allows the Actor **User** to use the **MovieCompanySystem**.

Create the **instance** of the **Controller** Class

## Controller control = new Controller();

Call the **UC 7 control.UCRestoreCatalog()** of the **Controller** Class to read the Catalog from the “**Catalog.txt**” file into the **MovieCompanySystem** memory

Create the **instance** of the **View** Class

## View view= new View();

Call the **view.showSelection()** of the **View** Class to display the menu of options for Actor **User**

Create **local variables** choice: **int**

artist: **String** disc: **String** track: **String**

discCatalogNumber: **String**

time: **double**

Read the choice from the keyboard Case choice of

1: Prompt for Artist Name artist

Call the Controller method **UC1 control.UCAddArtist(artist);**

2: Call the Controller method **UC2 control.UCListArtist();**

3: Prompt for Artist Name artist

Prompt for Disc Catalog Number Name discCatalogNumber Prompt for Disc Name disc

Call the Controller method **UC3 control.UCAddDisc(**artist,discCatalogNumber,disc**);**

4: Call the Controller method **UC4 control.UCListDiscs();**

5: Prompt for Artist Name artist

Prompt for Disc Catalog Number Name discCatalogNumber Prompt for Disc Name disc

Call the Controller method **UC5 control.UCAddDisc(**artist,discCatalogNumber,disc**);**

6: Prompt for Disc Catalog Number Name discCatalogNumber Call the Controller method **UC6**

**control.UCListTracksForDisc(**discCatalogNumber**);**

Call the **UC 8 control.UCSaveCatalog()** of the **Controller** class to write the Catalog in the “**Catalog.txt**” file

|  |
| --- |
| **View** |
|  |
| + showSelection( ) : void |

public void showSelection( ) {

Display the string “Welcome the Movie Company System” Display the string “Enter 1 for adding Artist:”

Display the string “Enter 2 for listing Artists:” Display the string “Enter 3 for adding Disc:” Display the string “Enter 4 for listing Discs:”

Display the string “Enter 5 for adding Track for Disc:” Display the string “Enter 6 for listing Tracks for Disc:”

}

|  |
| --- |
| **Controller** |
| - **catalog**: **Catalog** |
| + **UCAddArtist**(String artist): void // UC1  + **UCListArtists**(): void // UC2  + **UCAddDisc**(String artistName, String discCatalogNumber, String discTitle): void // UC3  + **UCListDiscs**(): void // UC4  + **UCAddTrack**(String discCatalogNumber, String trackTitle, double time): void // UC5  + **UCListTracksForDisc**(String discCatalogNumber): void // UC6  + **UCRestoreCatalog**(): void // UC7  + **UCSaveCatalog**(): void // UC8 |

Create the **instance** of the **Catalog** Class

## Catalog catalog = new Catalog();

public void **UCAddArtist**(String **artist**){ **catalog.addArtist(artist);**

} // UC1

public void **UCListArtists**(){

## catalog.listArtists();

} // UC2

public void **UCAddDisc**(String artistName, String discCatalogNumber, String discTitle){

## catalog.addDisc(artistName, discCatalogNumber, discTitle);

}

public void **UCListDiscs**(){

## catalog.listDiscs();

// UC3

}

// UC4

public void **UCAddTrack**(String discCatalogNumber, String trackTitle, double time){

## catalog.addTrack(discCatalogNumber, trackTitle, time);

} // UC5

public void **UCListTracksForDisc**(String discCatalogNumber){

## catalog.listTracksForDisc(discCatalogNumber);

}

public void **UCRestoreCatalog**(){

## catalog.restoreCatalog();

// UC6

}

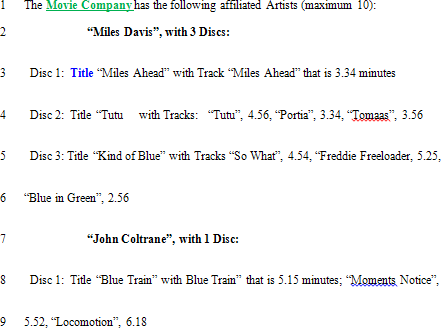
public void **UCSaveCatalog**(){

## catalog.saveCatalog();

// UC7

Since the **catalog** is **private** we need to call a **public** method in the **catalog** to operate on the **private** instance variables of the **catalog**

} // UC8



## artists



**name = ”John Coltrane”**

**:Artist**

**name = “Miles Davis”**

**:Artist**

[0] [1] [2] **……** [8] [9]

|  |
| --- |
| **Catalog** |
| -**MAX\_ARTISTS = 10 //1**  -**MAX\_DISCS = 30 //2**   * **artists**: **Artist[MAX\_ARTISTS] //3** * **discs**: **Disc[MAX\_DISCS] //4**   #numberOfArtists: **int //5**  #numberOfDiscs: **int //6** |
| + **addArtist**(String artist): void // 1  + **listArtists**(): void // 2  + **addDisc**(String artistName, String discCatalogNumber, String discTitle): void // 3  + **listDiscs**(): void // 4  + **addTrack**(String discCatalogNumber, String trackTitle, double time): void // 5  + **listTracksForDisc**(String discCatalogNumber): void // 6  + **restoreCatalog**(): void // 7  + **saveCatalog**(): void // 8 |



**public** void **addArtist**(String artistName){

If the artistName is not already in the **catalog**

if (numberOfArtists **< MAX\_ARTISTS**) **artists[**numberOfArtists**] = new Artist(**artistName**);** numberOfArtists**++;**

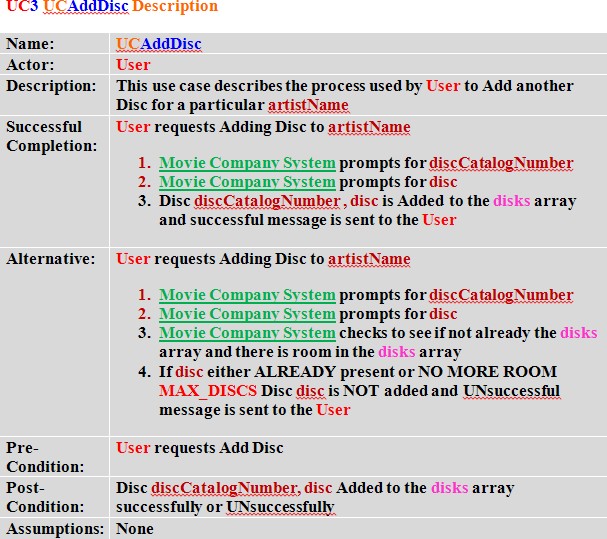
} // 1



**public** void **listArtists**(){

for (int i=0;i<numberOfArtists;i++) display **artists**[i].**getName**());

} // 2

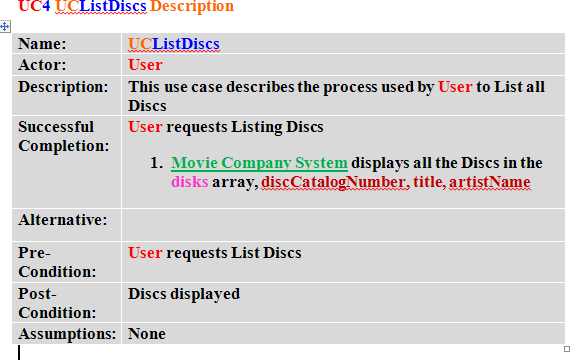


**public** void **addDisc**(String artistName, String discCatalogNumber, String discTitle){ get the artist for artistName **Artist result** = **getArtistForName**(artistName);

create a new **Disc** object **Disc theDisc** = **new Disc**(**result**, discCatalogNumber, discTitle);

if (numberOfDiscs < **MAX\_DISCS**) **discs**[numberOfDiscs] = **theDisc**; numberOfDiscs++;

} // 3



**public** void **listDiscs**(){

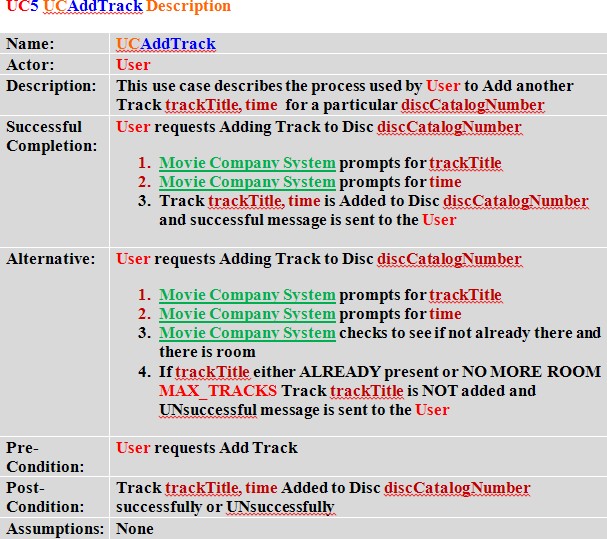
for (int i=0;i<numberOfDiscs;i++)

display **discs**[i].g**etDiskCatalogNumber**() + ", "

+ **discs**[i].**getTitle()** + ", by "

## + discs[i].getArtist().getName()

} // 4



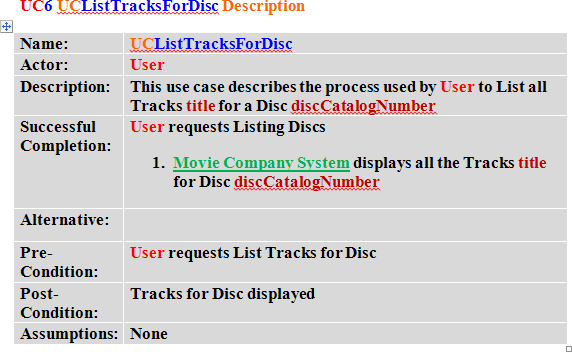
**public** void **addTrack**(String discCatalogNumber, String trackTitle, double time){

get the disc for discCatalogNumber **Disc result** =

**getDiscForCatalogNumber**(discCatalogNumber);

if (**result**.**getNumberofTracks**() < **Disc**.**MAX\_TRACKS**) **Track theTrack** = **new Track**(trackTitle, time); **result**.**addTrack**(**theTrack**);

} // 5



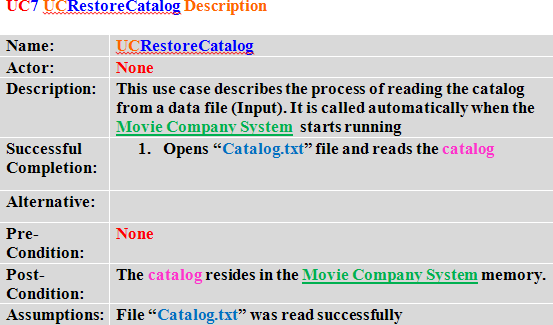
**public** void **listTracksForDisc**(String discCatalogNumber){

get the disc for discCatalogNumber **Disc result** =

**getDiscForCatalogNumber**(discCatalogNumber); for (int i=0;i<**result**.**getNumberofTracks**();i++)

display **result**.**getTrack**(i).**getTitle**());

} // 6



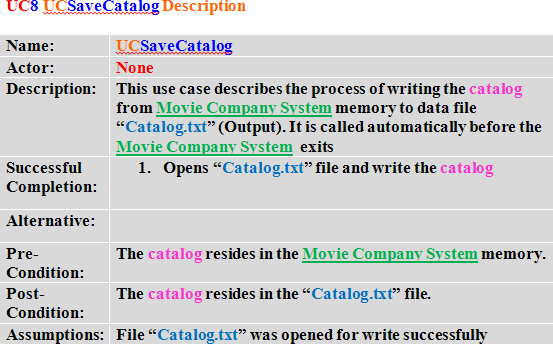
**public** void **restoreCatalog**(){

**Catalog catalog** = **new Catalog**(); **catalog**.artists = **artists**; **catalog**.discs = **discs**;

open file to write **File** file = **new** File("**Catalog.txt**");

## out.writeObject(catalog);

} // 7



**public** void **saveCatalog**(){

open file to read **File** file = **new** File("**Catalog.txt**");

## catalog = (Catalog)in.readObject();

**artists** = **catalog**.**artists**; **discs** = **catalog**.**discs**;

// set number of artists and discs

while (**artists**[numberOfArtists] != null) numberOfArtists++;

while (**discs**[numberOfDiscs] != null) numberOfDiscs++;

} // 8

// UTILITIES //

/\*\*

* getArtistForName() : **Artist** - returns an **Artist** object for
* a specified artist name

\*

* @param name string containing the
* name of an Artist
* @return the **Artist** object
* with the specified name

\*/

**private Artist getArtistForName**(String name) {

int i = 0;

int found = 0;

while (i<numberOfArtists && found==0)

{

compare the names if (name.**compareTo**(**artists**[i].**getName()**) == 0) found = 1;

i++;

}

if (found == 1)

return the **Artist** object return **artists**[i-1]; else return

null;

}

/\*\*

* **getDiscForCatalogNumber()** - returns a **Disc** object for
* a specified catalog number

\*

* @param discCatalogNumber string containing
* the catalog number of a disc
* @return the Disc object with the specified name

\*/

**private Disc getDiscForCatalogNumber**(String discCatalogNumber)

{

int i = 0;

int found = 0;

while (i<numberOfDiscs && found==0)

{

compare the discCatalogNumbers

if (discCatalogNumber.**compareTo**(**discs**[i].**getDiskCatalogNumber()**) == 0) found = 1;

i++;

}

if (found == 1)

return the **Disc** object return **discs**[i-1];

else

}

return null;