

IKI10100I: Data Structures & Algorithms • 2015-16
Faculty of Computer Science, Universitas Indonesia
Tutorial 2 (Week 2): Abstract Data Types

A common feature that is always present in every media player is the ability to keep a list of music tracks to be played, commonly called a *playlist*. A user can add, remove, or arrange music tracks in a playlist.

Your task is to represent the concept of a “**music track**” and “**playlist**” as described in this document into ADT classes. Implement these classes so that the operations that are performed on the playlist works as described.

1. Track Class Description

A music track is represented by the `Track` class with three fields: `title`, `artist`, `genre`, and `is_fav`. The `title`, `artist`, and `genre` fields are self-explanatory and consist of alphanumeric characters. The `is_fav` field is a boolean value that represents whether the track is favoured by the user or not. Each property should have an accessor and mutator method. The class is not yet available and it is your task to create this class.

2. Playlist Class Description

The playlist is represented by the `Playlist` class with a single field: `queue`. This list will contain instances of the `Track` class. The class is provided in the code template and has 7 unimplemented methods that you must complete. The methods are: `add(track)`, `delete_title(title)`, `delete_position(pos)`, `move(n,m)`, `count_favourite()`, `count_genre(genre)`, and `print_playlist()`.

The description for each method is as follows:

`add(track)`: This method accepts an instance of a `Track` object and inserts it into `queue`. The addition must not allow multiple tracks with the same name. The method returns `True` if the track was successfully inserted into the list, `False` otherwise. This method is called when the program reads a string preceded by **ADD** from the input. The string format of the **ADD** operation is “**ADD,w,x,y,z**” where **w**, **x**, and **y** is a string that represents the title, artist, and the genre of a track, and **z** is a string that determines whether the track is favoured by the user. The possible values of **z** are either “**True**” or “**False**”.

`delete_title(title)`: This method deletes an instance of a `Track` object from the list by its title. The method returns `True` if the track was successfully deleted from the list, and `False` otherwise. This method is called when the program reads a string preceded by **DELTITLE** from the input. The string form at of the **DELTITLE** operation is “**DELTITLE,x**” where **x** is a string that represents the title of a track that will be removed from the playlist.

delete_position(pos) : This method deletes an instance of a **Track** object from the list by its index position **pos** in the list. The method returns **True** if the track was successfully deleted, and **False** otherwise. This method is called when the program reads a string preceded by **DELPOS** from the input. The string format of **DELPOS** operation is "**DELPOS, n**" where **n** is an integer that represents the track's position in the list with a possible value within the range from 0 to **len** where **len** is the number of tracks in the list.

move(n, m) : This method moves an instance of a **Track** object in the list from position **n** to position **m**. If **n < m**, then all tracks from position **m** until **n+1** will be shifted to the left (assuming list is visualised as a horizontal array) to leave an empty space for track from **n**. Otherwise, if **n > m**, then all tracks starting from the beginning of the list until **n-1** track will be shifted to the right. All tracks from position **m** to **len-1** where **len** is the number of tracks in the list will be shifted to the right before the track from position **n** is inserted into position **m**. The method always returns **True**. This method is called when the program reads a string preceded by **MOVE** from the input. The string format of the **MOVE** operation is "**MOVE, n, m**" where **n** and **m** are integers that represents the position of a track in the playlist. **n** and **m** is a valid index position of the list starting from 0 to **len-1** where **len** is the number of tracks in the list.

count_favourite() : This method counts the number of **Track** objects in the list that is favoured by the user. It returns the number of favoured tracks in the list. This method is called when the program reads a string preceded by **COUNTFAV** from the input. The string format of **COUNTFAV** operation is simply "**COUNTFAV**".

count_genre(genre) : This method counts the number of **Track** objects in the list having the same genre as given in the method parameter **genre**. It returns the number of tracks that share the same genre. This method is called when the program reads a string preceded by **COUNTGENRE** from the input. The string format of **COUNTGENRE** operation is "**COUNTGENRE, x**" where **x** is the name of genre whose tracks we want to count in the list.

print_playlist() : This method prints the title of each **Track** objects in the list into standard output. Each titles are separated by a new line. This method is called when the program reads a string preceded by **PRINT** from the input. The string format of **PRINT** operation is "**PRINT**" string.

3. Input

The input is read from the provided test case text files and consists of several lines. The lines consist of non-empty alphabetic strings that represents an operation and its parameters if exists. The string format for all possible operations are defined in their respective method specification in the **Playlist** class description above.

4. Output

The output will be written to the standard output each time the program processes an operation read from file. The output format for each operation are defined as follows:

ADD: The string format of successful **ADD** execution is "**x - y added to playlist.**", where **x** and **y** is the artist and title of the inserted music track. When the operation fails, the format is "**x - y already in playlist.**"

DELTITLE: The format of successful **DELTITLE** execution is "**x deleted from playlist.**" where **x** is the title of deleted music track. When the execution fails, the format is "**x not found in playlist.**"

DELPOS: The format of successful **DELPOS** execution is "**Track n deleted from playlist.**" where **n** is the index position of the deleted music track. When the execution fails, the format is "**Cannot delete track at position n.**"

MOVE: The format of a successful **MOVE** execution is "**x - y moved into position m.**", where **x** and **y** is the artist and title of the moved music track and **m** is the new index position.

COUNTFAV: The format of **COUNTFAV** execution is "**Number of favourite tracks: n**" where **n** is the number of favourited tracks in the list.

COUNTGENRE: The format of **COUNTGENRE** execution is "**Number of x tracks: n**", where **x** is the name of the genre and **n** is the number of tracks having genre **x**.

PRINT: The format of **PRINT** operation is **n** lines where each lines is the name of track in **n**-th position.

Sample Input 1

```
ADD,Silver Soul,Beach House,Dream Pop,True
ADD,No One Knows,Queens of the Stone Age,Rock,False
ADD,Machine Gun,Noisia,DnB,False
ADD,Hotline Bling,Drake,Hip Hop,False
ADD>About You,XXYYXX,Electro,True
COUNTFAV
```

Sample Output 1

```
Beach House - Silver Soul added to playlist.
Queens of the Stone Age - No One Knows added to playlist.
Noisia - Machine Gun added to playlist.
Drake - Hotline Bling added to playlist.
XXYYXX - About You added to playlist.
Number of favourited tracks: 2
```

Sample Input 2

ADD,Get Away,The Internet,Soul,True
ADD,Ice Cream,Battles,Experimental,True
ADD,Dopamine,DIIV,Shoegaze,True
DELTITLE,Dopamine
ADD,Mostly Hair And Bones Now,Gaza,Hardcore,False
ADD,Alright,Kendrick Lamar,Hip-hop,False
DELPOS,2

Sample Output 2

The Internet - Get Away added to playlist.
Battles - Ice Cream added to playlist.
DIIV - Dopamine added to playlist.
DIIV - Dopamine deleted from playlist.
Gaza - Mostly Hair And Bones Now added to playlist.
Kendrick Lamar - Alright added to playlist.
Track 2 deleted from playlist.

Sample Input 3

COUNTGENRE,Punk
COUNTFAV
ADD,Coalition,Iceage,Punk,True
ADD,Begginning To End,Ta-ku,Hip-hop,False
COUNTGENRE,Punk
ADD,Absolution,Ghost,Metal,True
ADD,Saeglopur,Sigur Ros,Post-rock,False
ADD,Kool Thing,Sonic Youth,Rock,True
PRINT
MOVE,3,1
PRINT

Sample Output 3

Number of Punk tracks: 0
Number of favourited tracks: 0
Iceage - Coalition added to playlist.
Ta-ku - Begginning To End added to playlist.
Number of Punk tracks: 1
Ghost - Absolution added to playlist.
Sigur Ros - Saeglopur added to playlist.
Sonic Youth - Kool Thing added to playlist.

Iceage - Coalition
Ta-ku - Begginning To End
Ghost - Absolution
Sigur Ros - Saeglopur
Sonic Youth - Kool Thing
Ghost - Absolution moved into position 1.
Ghost - Absolution
Iceage - Coalition
Ta-ku - Begginning To End
Sigur Ros - Saeglopur
Sonic Youth - Kool Thing

5. Submission

Compress your .py files into a ZIP file named according to the following file naming format: YourNPM_YourName_Tutorial2.zip. Upload the ZIP file to the provided submission slot in SCellE. The submission deadline is 22nd February 2016 18:00.

Good Luck :~)