Code – Challenge

Pan,Chia-Kai

Song Cache

Steps for Creating App

1. I create an interface for SongCache. Then I create 4 methods, which are recordSongPlays, getPlaysForSong, getTopNSongsPlayed, and printSong.

2. recordSongPlays is for recording Number of plays for a song, getPlaysForSong is for fetching the number of plays for a song, getTopNSongsPlayed is for returning the top n songs played and return in descending order of numbers of plays, and printSong is for printing the song in the songList.

3. I create a class SongCacheImpl to implement SongCache interface. I override 4 methods then do further implementation. I create a hashmap field for the SongCacheImpl class for storing songs.

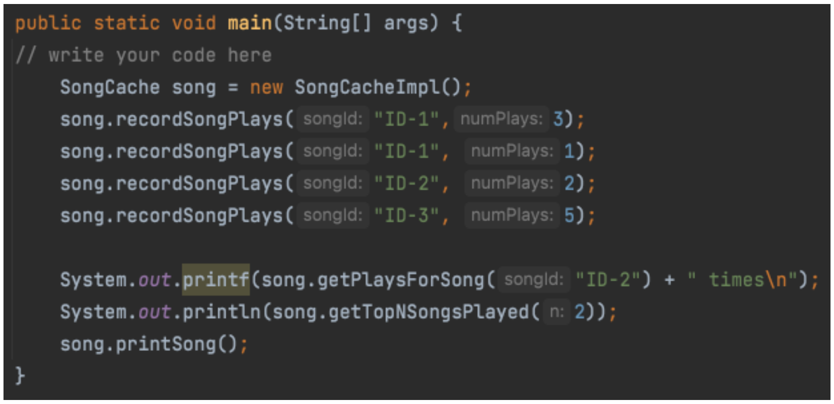
    - For recordSongPlays, I wrote songList.put(songId,songList.getOrDefault(songId,0)+numPlays) for calcaulate numbers played for the song.

    - For getPlaysForSong, I wrote if the entry we want to get is null we return -1, otherwise we return the number of the song played.

    - For getTopNSongsPlayed, I create a minheap for keep top n song played and 2 arraylist for displaying the data in descending order. Corner case for this method I wrote if n is equal to zero or input is null, I return an empty list, and if hashmap size is smaller than n I throw new Illegal Argument Exception.

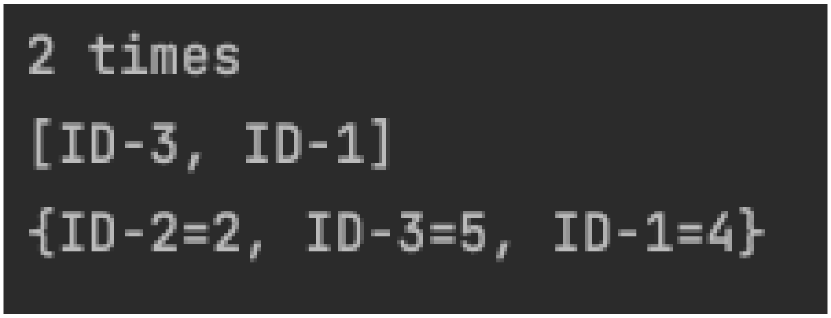
    - For printSong, print out all the song in hashmap.

4. The following 2 image is how I instantiate song object and the result I want to show. First line is to show how many times is ID-2 being played. Second line is the top 2 songs being played in descending order which is [ID – 3, ID – 1], because ID-3 is 5 and ID-1 is 4. The last line is the song and times played in the song object.



```java
public static void main(String[] args) {
// write your code here
    SongCache song = new SongCacheImpl();
    song.recordSongPlays( songId: "ID-1", numPlays: 3);
    song.recordSongPlays( songId: "ID-1",  numPlays: 1);
    song.recordSongPlays( songId: "ID-2",  numPlays: 2);
    song.recordSongPlays( songId: "ID-3",  numPlays: 5);

    System.out.printf(song.getPlaysForSong( songId: "ID-2") + " times\n");
    System.out.println(song.getTopNSongsPlayed( n: 2));
    song.printSong();
}
```



```
2 times
[ID-3, ID-1]
{ID-2=2, ID-3=5, ID-1=4}
```