

כתבו תוכנית לניהול מאגר שירים.
לכל שיר נשמור את שם השיר, האמן שכתב את השיר והאמן האחרון שביצע את השיר.
לכל אמן נשמור את שם האמן, רשימת השירים שכתב ורשימת השירים שביצע.

דגשים כלליים:

במידה והמערך מלא יש להגדיל אותו במקום אחד בלבד אלא אם כן צוין אחרת..
שימו לב היטב לקשרים בין המחלקות בעת הוספת שיר.

המחלקות בתוכנית:

1. המחלקה אמן Artist, כתבו את המחלקה לפי הדרישות הבאות:

Attributes

- - name: String
- - writtenSongs: Song[]
- - performedSongs: Song[]

Constructors

- + Artist (String name) set 2 arrays to size 0

Methods

- + String getName()
- + boolean hasWroteSong(Song theSong)
- + boolean addWrittenSong(Song newSong)
- + boolean hasPerformedSong(Song song)
- + boolean addPerformedSong (Song song)
- + toString(): String

דגשים והבהרות:

השיטה addWrittenSong : הוספת שיר תתבצע רק אם השיר לא קיים, שיר מזוהה לפי שמו.

השיטה addPerformedSong : הוספת שיר תתבצע רק אם השיר לא קיים, שיר מזוהה לפי שמו.

2. המחלקה שיר Song, כתבו את המחלקה לפי הדרישות הבאות:

Attributes – מאפיינים

- -name: String
- - writer Artist
- - lastPerformer: Artist

Constructors – בנאים

- + Song (String name, Artist writer, Artist lastPerformer)
lastPerformer could be null

Methods – מתודות

- + String getName()
- + setLastPerformer(Artist artist)
- + toString(): String

דגשים והבהרות:

בבנאי יש לשים לב להוסיף את השיר לרשימת השירים שהאמן כתב וכן גם לרשימת השירים שביצע במידה והועבר שיר ולא null.

3. המחלקה שיר SongsRepository, כתבו את המחלקה לפי הדרישות הבאות:

Attributes – מאפיינים

- + eAddSongStatus: enum
Success, SongAlreadyExist, WriterDoesntExist, PerformerDoesntExist
- - allArtists: Artist[]
- - numOfArtists: int
- - allSongs: Song[]
- - numOfSongs: int

Constructors – בנאים

- + SongsRepository () set 2 arrays to size 2

Methods – מתודות

- + Artist getArtistByName(String artistName)
- + Song getSongByName(String songName)
- + boolean addArtist(String artistName)
- + eAddSongStatus addSong(String songName, String artistName, String performerName)
- + toString(): String

דגשים והבהרות:

בשיטה addSong, במידה ואין אומן שביצע את השיר, תועבר מחרוזת ריקה. השיטה תנסה להוסיף את השיר למאגר ותצליח רק אם השיר לא במאגר, האמן שכתב את השיר קיים במאגר והאומן שביצע אחרון את השיר (במידה והועבר) קיים במאגר. כמו כן, תגדיל את מאגר שירים במידה ואין מספיק מקום. השיטה תדאג להוסיף את השיר לאמן שכתב את השיר, ואם הועבר אמן שביצע אחרון את השיר אז גם תעדכן את רשימת השירים שלו. השיטה תחזיר ערך מהטיפוס eAddSongStatus בהתאם לאופי הכישלון או ההצלחה.

ניהול התוכנית:

דוגמא ל-main:

```
public class Main {
    public static void main(String[] args) {
        SongsRepository repository = new SongsRepository();
        String[] artists = {"Yael", "Adam", "Rina", "Yael"};
        System.out.println("Init Artists");
        for (String a: artists) {
            System.out.println("Try to add Artist " + a);
            if (repository.addArtist(a)){
                System.out.println("\tArtist " + a + " added
successfully");
            } else {
                System.out.println("\tFailed to add Artist " + a);
            }
        }
        System.out.println("\nInit Songs");
        String[][] songs = {
            {"Corona Hey!", "Yael", "Yael"},
            {"Antigen sucks", "Yael", "Adam"},
            {"Antigen sucks", "Yael", "Adam"},
            {"PCR rocks!", "Yael", "Adam"},
            {"Just", "Yosi", "Adam"},
            {"Be Healthy Choopie-Doop", "Adam", "Yosi"},
            {"Doing", "Yael", ""}
        };
        for (String[] a: songs) {
            System.out.println("Try to add song " + a[0]);
            SongsRepository.eAddSongStatus res =
repository.addSong(a[0],a[1],a[2]);
            if (res.equals(SongsRepository.eAddSongStatus.Success)){
                System.out.println("\tSong " + a[0] + " added
successfully");
            } else {
                System.out.println("\tFailed to add Song " + a[0] + ",
" + res);
            }
        }
        System.out.println(repository);

        System.out.println("Repository after change:");
        Song song = repository.getSongByName("Corona Hey!");
        song.setLastPerformer(repository.getArtistByName("Adam"));

        System.out.println(repository);
    }
}
```