50m left

ALL

1

2

EXPERTISE ONLY.

Problem Statement

Create a class Movies with the below attributes:
id-int
name-String
rating- double
genre-String

Write getters, setters and parameterized constructor in the above-mentioned attribute sequence as required

Create a class Solution with the main method.

Implement two static methods - findHighestRatedMovie and searchMovieByName in Solution class.

findHighestRatedMovie method:
Create a static method findHighestRatedMovie in the Solution class. This method will take

Create a static method findHighestRatedMovie in the Solution class. This method will take an array of Movie objects and return the movie details with the highest rating if found else return null if not found.

searchMovieByName
Create a static method searchmoviebyname in the Solutionclass. This method will take an array of Movie objects and a movie name(a string parameter given in the user input), as parameters and return the matching movie if found else return null if not found.

These methods should be called from the main method.

Write code to perform the following tasks:

1. Take the necessary inputs and

1. Take the necessary inputs and call **findHighestRatedMovie.** For this method - The main method should print the Movie object details as it is if the returned value is not null and if it is null then it should print "No Movie found."

2. Take the necessary inputs and call **searchMovieByName.** For this method – the main method should print the Movie object details as it is if the returned value is not null and if it is null, then it should print "No Movie found.", excluding quotes

The above-mentioned static method should be called from the main method. Also write the code for accepting the inputs and printing the outputs. Don't use any static test or formatting for printing the result. Just invoke the method and print the

outputs. Don't use any static test or formatting for printing the result. Just invoke the method and print the result.

Note:
All String comparison needs to be case in-sensitive.
You can use/refer to the below given sample input and output to verify your solution.

Sample Input (below) description:
The 1st input taken in the main section is the number of Movie objects to be added to the list of Movies.
The next set of inputs are ID,name,rating,genre for each Movie object taken one after another and is repeated for the number of Movie objects given in the first line of input.
The last input taken is a String which is a movieName to pass it as a parameter in method 2.

Consider below sample input and

parameter in method 2.

Consider below sample input and output to test your code:

Input:

```
4
101
Kung Fu Panda
8.5
Animation
102
Godzilla
8.7
Sci-Fi
103
Appu
7.7
Animation
104
Migration
9.0
Animation
Migration
```

Output:

49m left

ALL

1

2

Animation
104
Migration
9.0
Animation
Migration

Output:

id- 104
name- Migration
rating- 9.0
genre- Animation
id- 104
name- Migration
rating- 9.0
genre- Animation

Note on using Scanner object:
Sometimes scanner does not read the new line character while invoking methods like nextInt(), nextDouble() etc.
Usually, this is not an issue, but this may be visible while calling nextLine() immediately after those methods.
Consider below input values:

nguage Java 17

Autocomplete Ready

Environment

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {
    public static void main(String args[]) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Movies [] arr= new Movies[n];
        for (int i = 0; i < arr.length; i++) {
            int a = sc.nextInt(); sc.nextLine();
            String b = sc.nextLine();
            double c = sc.nextDouble(); sc.nextLine();
            String d = sc.nextLine();
            arr[i] = new Movies(a, b, c, d);
        }
        String inp = sc.nextLine();
        sc.close();
        Movies obj1 = findHighestRatedMovie(arr);
        if(obj1 != null){
            System.out.println("id- "+obj1.getId());
            System.out.println("name- "+obj1.getname());
```

Line: 43

Test Results | Custom Input | Run Code | Run Tests | Su

Search

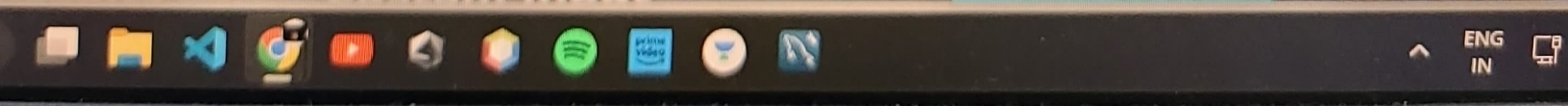
ENG IN

```
        Movies obj1 = findHighestRatedMovie(arr);
        if(obj1 != null){
            System.out.println("id- "+obj1.getId());
            System.out.println("name- "+obj1.getname());
            System.out.println("rating- "+obj1.getRating());
            System.out.println("genre- "+obj1.getGenre());
        }else{
            System.out.println("No Movie found.");
        }

        Movies obj2 = searchMovieByName(arr, inp);
        if(obj2 != null){
            System.out.println("id- "+obj2.getId());
            System.out.println("name- "+obj2.getname());
            System.out.println("rating- "+obj2.getRating());
            System.out.println("genre- "+obj2.getGenre());
        }else{
            System.out.println("No Movie found.");
        }
    }
    public static Movies findHighestRatedMovie(Movies[]arr){
        if(arr == null || arr.length ==0){
            return null;
        }
```

Li

Results | Custom Input | Run Code | Run Tests



ENG
IN

guage Java 17 Autocomplete Ready

vironment

```
    public static Movies findHighestRatedMovie(Movies[]arr){
        if(arr == null || arr.length ==0){
            return null;
        }
        Movies high = arr[0];
        for (int i = 0; i < arr.length; i++) {
            if(arr[i].getRating() > high.getRating()){
                high = arr[i];
            }
        }return high;
    }
    public static Movies searchMovieByName(Movies[] arr, String inp){
        for (int i = 0; i < arr.length; i++) {
            if(arr[i].getname().equalsIgnoreCase(inp)){
                return arr[i];
            }
        }return null;
    }
}
class Movies{
    int id;
    String name;
    double rating;
    String genre;
```

Line: 43 Col

est Results | Custom Input | Run Code | Run Tests | Submi

Search

ENG IN 06:13 PM 12-05-2024

```java
lass Movies{
    int id;
    String name;
    double rating;
    String genre;
    public Movies(int id, String name, double rating, String genre){
        this.id= id;
        this.name=name;
        this.rating=rating;
        this.genre=genre;
    }
    public int getId(){
        return id;
    }
    public void setId(int id){
        this.id=id;
    }
    public String getname(){
        return name;
    }
    public void setName(String name){
        this.name=name;
    }
```

```
72          return id;
73      }
74      public void setId(int id){
75          this.id=id;
76      }
77      public String getname(){
78          return name;
79      }
80      public void setName(String name){
81          this.name=name;
82      }
83      public double getRating(){
84          return rating;
85      }
86      public void setId(double rating){
87          this.rating=rating;
88      }
89      public String getGenre(){
90          return genre;
91      }
92      public void setGenre(String genre){
93          this.genre=genre;
94      }
95  }
```

Test Results Custom Input

ANY MEANS OF INTENDED PLAGIARISM WHILE ATTEMPTING THIS QUESTION IS LIABLE TO HIGHEST POSSIBLE STRICT HR ACTIONS AS IT IS VIOLATION IF INTEGRITY. HENCE, REFRAIN FROM ANY SUCH MEANS AND ATTEMPT THIS QUESTION AS PER YOUR EXPERTISE ONLY.

Write the main method in the "Solution" class.

You need to read an input number (integer) from the console (test case) and print the average of digits in the number.

For example, consider the sample input and output given below:

Sample Input:
5678

Sample Output:
Average of digits:6.5

Language: Java 15 | Environment | Autocomplete Ready

```java
import java.util.Scanner;

public class Solution {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        double sum = 0;
        int count = 0;
        int temp = n;
        while(temp > 0){
            sum += temp % 10;
            temp /= 10;
            count ++;
        }
        double average = sum/count;
        System.out.println("Average of digits:" + average);

    }
}
```

Test Results | Custom Input | Run Code | Run Tests