

1h 4m left

BETA

Can't read the text? Switch theme



1. JAVA

ALL



1

2

Create a class Employee with below attributes:

id - int

name - String

salary-double

address-String

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

Create class Solution with main method

Implement two static methods

- **findEmployeeWithMaximumSalary** and **searchEmployeeByName** in Solution class.

findEmployeeWithMaximumSalary

Create a static method

findEmployeeWithMaximumSalary in

the Solution class. This method will take array of Employee objects and returns the

Employee object having the maximum salary

HackerRank Question - Java
hackerrank.com/test/38jebepmf70/questions/bii99134c1e
the Solution class. This method will take array
of Employee objects and returns the
Employee object having the maximum salary
if found else return null if not found.

1h 3m left



ALL



1

2

`searchEmployeeByName`

Create a static method

`searchEmployeeByName` in the Solution

class. This method will take array of Employee
objects and Name as input and returns the
Employee object having the mentioned Name
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 necessary input variable and
call `findEmployeeWithMaximumSalary`. For
this method - The main method should print
the Employee object with maximum of
mentioned attribute as it is if the returned
value is not null, or it should print
"No Employee found with mentioned
attribute."

2. Take necessary input variable and call
`searchEmployeeByName`. For this method -
The main method should print the Employee
object details as it is, if the returned value is

hackerrank.com/test/38jebepmf70/questions/b19913...
searchEmployeeByName. For this method -
The main method should print the Employee
object details as it is, if the returned value is
not null, or it should print "No
Employee found with mentioned attribute."

1h 3m left



ALL



1

2

Note:

The above mentioned static methods 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 result

Note :All String comparison needs to be case in-sensitive.

You can use/refer the below given sample input and output to verify your solution.

The 1st input taken in the main section is the number of Employee objects to be added to the list of Employee.

The next set of inputs are id,name,salary,address for each Employee object taken one after other and is repeated for number of Employee objects given in the first line of input.

The last line of inputs will be the arguments

HackerRank
hackerrank.com/test/38jebepmf70/questions/
object taken one after other and is repeated
for number of Employee objects given in the
first line of input.

1h 3m left

ALL

1

2

Consider below sample input and output to
test your code:

Sample Input 1 -

4

101

Allen

2500

New York

102

Jack

5000

Boston

103

John

3450

Washington

104

Claire

4553

California

Claire

1h 3m left



ALL



1

2

101
Allen
2500
New York
102
Jack
5000
Boston
103
John
3450
Washington
104
Claire
4553
California
Claire



Sample Output 1

id-102
name-Jack
salary-5000
address-Boston
id-104
name-Claire
salary-4553
address-California

1h 3m left

Sample Input 2

4
111
Ajay
ALL

112
Ram
35000
Delhi
113
Gopal
53000
Bangalore

114
Virat
43000
Pune
Virat

Sample Output 2

id-113
name-Gopal
salary-53000
address-Bangalore
id-114



1h 3m left

111

Ajay

10000

Mumbai



112

Ram

35000

ALL

Delhi

113

Gopal



53000

Bangalore

114

1

Virat

43000

Pune

2

Virat



Sample Output 2

id-113

name-Gopal

salary-53000

address-Bangalore

id-114

name-Virat

salary-43000

address-Pune

```
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();
        Employee [] arr = new Employee[n];
        for (int i = 0; i < arr.length; i++) {
            int a = sc.nextInt();
            String b = sc.next();
            double c = sc.nextDouble(); sc.nextLine();
            String d = sc.next(); sc.nextLine();
            arr[i] = new Employee(a, b, c, d);
        }
        String inp = sc.nextLine();
        sc.close();

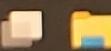
        Employee obj1 = findMaxSalary(arr);
        if(obj1 != null){
            System.out.println("Id-" +obj1.getId());
        }
    }
}
```

Test Results

Custom Input

Run Code

Run Tests





```
Employee obj1 = findMaxSalary(arr);
if(obj1 != null){
    System.out.println("id-"+obj1.getId());
    System.out.println("name-"+obj1.getName());
    System.out.println("salary-"+obj1.getSalary());
    System.out.println("address-"+obj1.getAddress());
}else{
    System.out.println("No Employee found with mentioned attribute.");
}
```

```
Employee obj2 = searchName(arr, inp);
if(obj2 != null){
    System.out.println("id-"+obj2.getId());
    System.out.println("name-"+obj2.getName());
    System.out.println("salary-"+obj2.getSalary());
    System.out.println("address-"+obj2.getAddress());
}else{
    System.out.println("No Employee found with mentioned attribute.");
}
```

```
public static Employee findMaxSalary(Employee[] arr){
    Employee maxx = arr[0];
    for (int i = 0; i < arr.length; i++) {
```

Language Java 7

Environment

Autocomplete Ready

```
System.out.println("No Employee found with mentioned attr")
```

```
}
```

```
}
```

```
public static Employee findMaxSalary(Employee[] arr){
```

```
    Employee maxx = arr[0];
```

```
    for (int i = 0; i < arr.length; i++) {
```

```
        if(arr[i].getSalary() > maxx.getSalary()){
```

```
            maxx = arr[i];
```

```
        }
```

```
    }
```

```
    return maxx;
```

```
}
```

```
public static Employee searchName(Employee [] arr, String inp){
```

```
    for (int i = 0; i < arr.length; i++) {
```

```
        if(arr[i].getName().equalsIgnoreCase(inp)){
```

```
            return arr[i];
```

```
        }
```

```
    }
```

```
    return null;
```

```
}
```

```
class Employee{
```

```
    int id;
```

```
    String name;
```

```
    double salary;
```

```
}

class Employee{
    int id;
    String name;
    double salary;
    String address;
    public Employee(int id, String name, double salary, String address){
        this.id=id;
        this.name=name;
        this.salary=salary;
        this.address=address;
    }
    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;
    }
}
```

```
    return id;
}
public void setId(int id){
    this.id=id;
}
public String getName(){
    return name;
}
public void setName(String name){
    this.name=name;
}
public double getSalary(){
    return salary;
}
public void setSalary(double salary){
    this.salary=salary;
}
public String getAddress(){
    return address;
}
public void setAddress(String address){
    this.address=address;
}
}
```

2. Print character on odd position

You will be creating a function called as `printCharacterAtOddPosition`. This function will take one string parameter as argument. You need to print the character in the odd position. Please check the below input for the better understanding.

Note: Index start with the 0th position. So first character will start with the 0th index

Sample input

hello

Sample output

e

l

Sample input

mobile

sample output

o

sunny

Language Java 15

Autocomplete Ready

Environment

```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8     public static void main(String args[]) throws Exception {
9         Scanner sc = new Scanner(System.in);
10        String input = sc.nextLine();
11        sc.close();
12        printCharacterAtOddPosition(input);
13    }
14    public static void printCharacterAtOddPosition(String str){
15        for(int i=1; i< str.length(); i += 2){
16            System.out.println(str.charAt(i));
17        }
18    }
19 }
```

Test Results

Visual Studio Code

Custom Input

Run Code

Run Tests



Search

