**MINOR PROJECT-2023**

**EMPLOYEE DATABASE USING CLASS AND OBJECT**

**NAME:-Anit Kumar Nag**

**REGD. NO:-2241014001**

**COURSE:-B.TECH(CSE)-2ND SEMESTAR**

**SEC:-2241023**

**1.For Class Address:-**

The code provided is a Java class called "Address" that represents an address. It has private instance variables (fields) for street, city, state, and pincode. The class has a constructor that accepts these variables as parameters and initializes the corresponding fields.

There are setter methods (setStreet, setCity, setState, and setPinCode) that allow modifying the values of the fields after the object is created.

Getter methods (getStreet, getCity, getState, and getPincode) are provided to retrieve the values of the fields.

Additionally, there is a method called "getAddress" that returns a formatted string representation of the address (street, city, state, and pincode concatenated together with commas).

**2.For Class Date:-**

The provided code is a Java class called "Date" that represents a date. Here's a breakdown of the code:

1. The class starts with the declaration of three private instance variables: day, month, and year. These variables will hold the values for the day, month, and year of a date.
2. The class has a constructor that accepts three parameters: day, month, and year. Inside the constructor, the values of the parameters are assigned to the corresponding instance variables using the this keyword.
3. Following the constructor, there are setter methods: setDay, setMonth, and setYear. These methods allow modifying the values of the day, month, and year variables after the object is created. Each setter method takes a parameter and assigns it to the respective instance variable.
4. Next, there are getter methods: getDay, getMonth, and getYear. These methods return the values of the day, month, and year variables, respectively.
5. Finally, there is a method called getDate which returns a formatted string representation of the date. It concatenates the day, month, and year values together with hyphens.

That's the breakdown of the code. The provided class can be used to create Date objects, set their values using the setter methods, retrieve the values using the getter methods, and obtain a formatted string representation of the date using the getDate method.

**3.For Class Employee:-**

The provided code is a Java class called "Employee" that represents an employee. Here's a breakdown of the code:

1. The class starts with the declaration of several private instance variables: name, empId, salary, hireDate, jobPosition, contactNumber, and address. These variables will hold the employee's information such as name, ID, salary, hire date, job position, contact number, and address.
2. The class has a constructor that accepts parameters corresponding to the instance variables. Inside the constructor, the values of the parameters are assigned to the corresponding instance variables using the this keyword.
3. Following the constructor, there are setter methods for each instance variable. These methods allow modifying the values of the respective instance variables after the object is created. Each setter method takes a parameter and assigns it to the respective instance variable.
4. Next, there are getter methods for each instance variable. These methods return the values of the respective instance variables.
5. The getHireDate method calls the getDate method from the Date class to retrieve a formatted string representation of the hire date.
6. The getAddress method calls the getAddress method from the Address class to retrieve a formatted string representation of the address.
7. Finally, there is a method called getEmployee that returns a formatted string representation of the employee's information. It concatenates the values of the instance variables together with appropriate labels.

That's the breakdown of the code. The provided class can be used to create Employee objects, set their values using the setter methods, retrieve the values using the getter methods, and obtain a formatted string representation of the employee's information using the getEmployee method.

**4.For Class Test:-**

The provided code is a Java class called "Test" that performs various operations on an array of Employee objects. Here's a breakdown of the code:

1. The code includes the necessary import statement to use the Scanner class.
2. The class contains a method named "employeeDatebase" that creates an array of Employee objects based on user input. It prompts the user to enter details for each employee, such as name, employee ID, salary, hire date, job position, contact number, and address. The method uses the provided constructors of the Employee, Date, and Address classes to create the objects and assigns them to the array.
3. The class includes a method named "arrangeEmployeeBySalary" that sorts the array of Employee objects based on their salary in decreasing order using a bubble sort algorithm.
4. The class includes a method named "getEmployeesByJobPosition" that takes an array of Employee objects and a job position as input. It displays the details of employees who have the specified job position.
5. The class includes a method named "getEmployeesByHireDate" that takes an array of Employee objects and two Date objects representing the start and end dates. It displays the details of employees who were hired between the specified dates.
6. The class includes a method named "foreignEmployeeCount" that takes an array of Employee objects and counts the number of employees whose contact number does not start with "+91", assuming that "+91" represents the country code of India.
7. The class includes a method named "getEmployeesBySalary" that takes an array of Employee objects and two salary values as input. It displays the details of employees whose salary falls within the specified range.
8. The main method of the class serves as the entry point of the program. It prompts the user to enter the number of employees, creates the employee database using the "employeeDatebase" method, and performs various operations on the employee data, such as sorting by salary, filtering by job position, hire date, counting foreign employees, and filtering by salary range. The results are displayed on the console.

That's the breakdown of the code. The provided class can be executed to interact with the user, create an array of Employee objects, perform different operations on the employee data, and display the results.

**/\*The above Descriptions are for the code of Employee Database\*/**

**\*\*\*OUTPUT\*\*\***

*/\*  
----> Data for Testing purpose  
employeeDatebase[0] = new Employee("java user\_1", 2001, 1000, new Date(10, 5 ,2023), "manager", "+91123456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[1] = new Employee("java user\_2", 2002, 2000, new Date(11, 5, 2023), "Software\_Developer", "+91987456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[2] = new Employee("java user\_3", 2003, 3000, new Date(13, 5, 2023), "Graphic\_Designer", "+85258963", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[3] = new Employee("java user\_4", 2004, 804000, new Date(14, 5, 2023), "manager", "+652789654", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[4] = new Employee("java user\_5", 2005, 5000, new Date(15, 5, 2023), "Software\_Developer", "+91789654", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[5] = new Employee("java user\_6", 2006, 6000, new Date(16, 5, 2023), "manager", "+85941256", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[6] = new Employee("java user\_7", 2007, 7000, new Date(17, 5, 2023), "Data\_Scientist", "+78951236", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[7] = new Employee("java user\_8", 2008, 8000, new Date(18, 5, 2023), "Software\_Developer", "+32456987", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[8] = new Employee("java user\_9", 2009, 9000, new Date(19, 5, 2023), "manager", "+91852456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[9] = new Employee("java user\_10", 2010, 10000, new Date(20, 5, 2023), "Cyber\_Security", "+91456852", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
  
----->For Salary Testing  
 for (int i=0 ; i<n ; i++) {  
 System.out.println("------------> Employee " + (i+1) + " : ");  
 System.out.println(e[i].getEmployee());  
 }  
 System.out.println("---->intput Done <----");  
 arrangeEmployeeBySalary(e);  
 for (int i=0 ; i<n ; i++) {  
 System.out.println("------------------------------------------------------> Employee " + (i+1) + " : ");  
 System.out.println(e[i].getEmployee());  
 }  
 System.out.println("----> Sorting Done <----");  
  
 OUTPUT  
Enter the Number of Employees : 10  
----> Employee Database Creation Started :  
|  
v  
employeeDatebase[0] = new Employee("java user\_1", 2001, 1000, new Date(10, 5 ,2023), "manager", "+91123456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[1] = new Employee("java user\_2", 2002, 2000, new Date(11, 5, 2023), "Software\_Developer", "+91987456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[2] = new Employee("java user\_3", 2003, 3000, new Date(13, 5, 2023), "Graphic\_Designer", "+85258963", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[3] = new Employee("java user\_4", 2004, 804000, new Date(14, 5, 2023), "manager", "+652789654", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[4] = new Employee("java user\_5", 2005, 5000, new Date(15, 5, 2023), "Software\_Developer", "+91789654", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[5] = new Employee("java user\_6", 2006, 6000, new Date(16, 5, 2023), "manager", "+85941256", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[6] = new Employee("java user\_7", 2007, 7000, new Date(17, 5, 2023), "Data\_Scientist", "+78951236", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[7] = new Employee("java user\_8", 2008, 8000, new Date(18, 5, 2023), "Software\_Developer", "+32456987", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[8] = new Employee("java user\_9", 2009, 9000, new Date(19, 5, 2023), "manager", "+91852456", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
employeeDatebase[9] = new Employee("java user\_10", 2010, 10000, new Date(20, 5, 2023), "Cyber\_Security", "+91456852", new Address("jagamara", "Bhubaneswar", "Odisha", 751089));  
|  
v  
----> Sorting the Employee Database According to Salary in Decreasing Order :  
----> Sorting is Done <----  
----> For Displaying the Detail of Employee (of the Entered Job Position ) :  
|  
v  
Enter Job Position : manager  
--------> Employee Detail <-------- :  
Name : java user\_4  
Employee Id : 2004  
Salary : 804000.0  
Hire Date : 14-5-2023  
Job Position : manager  
Contact Number : +652789654  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_9  
Employee Id : 2009  
Salary : 9000.0  
Hire Date : 19-5-2023  
Job Position : manager  
Contact Number : +91852456  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_6  
Employee Id : 2006  
Salary : 6000.0  
Hire Date : 16-5-2023  
Job Position : manager  
Contact Number : +85941256  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_1  
Employee Id : 2001  
Salary : 1000.0  
Hire Date : 10-5-2023  
Job Position : manager  
Contact Number : +91123456  
Address : jagamara , Bhubaneswar , Odisha , 751089  
----> For Displaying the Detail of the Employee Between the Following Dates :  
|  
v  
Enter starting Hire Date : 13 5 2023  
Enter Ending Hire Date : 18 5 2023  
|  
v  
--------> Employee Detail <-------- :  
Name : java user\_4  
Employee Id : 2004  
Salary : 804000.0  
Hire Date : 14-5-2023  
Job Position : manager  
Contact Number : +652789654  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_8  
Employee Id : 2008  
Salary : 8000.0  
Hire Date : 18-5-2023  
Job Position : Software\_Developer  
Contact Number : +32456987  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_7  
Employee Id : 2007  
Salary : 7000.0  
Hire Date : 17-5-2023  
Job Position : Data\_Scientist  
Contact Number : +78951236  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_6  
Employee Id : 2006  
Salary : 6000.0  
Hire Date : 16-5-2023  
Job Position : manager  
Contact Number : +85941256  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_5  
Employee Id : 2005  
Salary : 5000.0  
Hire Date : 15-5-2023  
Job Position : Software\_Developer  
Contact Number : +91789654  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_3  
Employee Id : 2003  
Salary : 3000.0  
Hire Date : 13-5-2023  
Job Position : Graphic\_Designer  
Contact Number : +85258963  
Address : jagamara , Bhubaneswar , Odisha , 751089  
----> Display the Foreign Employees :  
The Number of Foreign Employees are : 5  
----> Displaying the Employee Detail between the range of the following Salary :  
|  
v  
Enter the salary value of S1 : 3000  
Enter the salary value of s2 : 7000  
--------> Employee Detail <-------- :  
Name : java user\_7  
Employee Id : 2007  
Salary : 7000.0  
Hire Date : 17-5-2023  
Job Position : Data\_Scientist  
Contact Number : +78951236  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_6  
Employee Id : 2006  
Salary : 6000.0  
Hire Date : 16-5-2023  
Job Position : manager  
Contact Number : +85941256  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_5  
Employee Id : 2005  
Salary : 5000.0  
Hire Date : 15-5-2023  
Job Position : Software\_Developer  
Contact Number : +91789654  
Address : jagamara , Bhubaneswar , Odisha , 751089  
--------> Employee Detail <-------- :  
Name : java user\_3  
Employee Id : 2003  
Salary : 3000.0  
Hire Date : 13-5-2023  
Job Position : Graphic\_Designer  
Contact Number : +85258963  
Address : jagamara , Bhubaneswar , Odisha , 751089  
 \*/*

**--------------------------------------------------------------------------------------------**

**THE END**

**--------------------------------------------------------------------------------------------------------------------------**