**MODULE-2**

**DATA STRUCTURE AND ALGORITHM**

SUPERSET ID:6407550

**Exercise 4: Employee Management System**

package employee;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        EmployeeManager manager = new EmployeeManager(100);

        Scanner sc = new Scanner(System.in);

        int choice;

        do {

            System.out.println("\n--- Employee Management ---");

            System.out.println("1. Add Employee");

            System.out.println("2. Search Employee");

            System.out.println("3. Delete Employee");

            System.out.println("4. Display All");

            System.out.println("5. Exit");

            System.out.print("Enter choice: ");

            choice = sc.nextInt();

            switch (choice) {

                case 1:

                    System.out.print("ID: ");

                    int id = sc.nextInt();

                    sc.nextLine();

                    System.out.print("Name: ");

                    String name = sc.nextLine();

                    System.out.print("Position: ");

                    String pos = sc.nextLine();

                    System.out.print("Salary: ");

                    double salary = sc.nextDouble();

                    manager.addEmployee(new Employee(id, name, pos, salary));

                    break;

                case 2:

                    System.out.print("Enter ID to search: ");

                    int sid = sc.nextInt();

                    Employee found = manager.searchEmployee(sid);

                    System.out.println(found != null ? found : "Not found.");

                    break;

                case 3:

                    System.out.print("Enter ID to delete: ");

                    int did = sc.nextInt();

                    boolean deleted = manager.deleteEmployee(did);

                    System.out.println(deleted ? "Deleted successfully." : "Employee not found.");

                    break;

                case 4:

                    manager.displayAll();

                    break;

                case 5:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice.");

            }

        } while (choice != 5);

        sc.close();

    }

}

**Employee.java**

package employee;

public class Employee {

    int employeeId;

    String name;

    String position;

    double salary;

    public Employee(int employeeId, String name, String position, double salary) {

        this.employeeId = employeeId;

        this.name = name;

        this.position = position;

        this.salary = salary;

    }

    @Override

    public String toString() {

        return employeeId + " | " + name + " | " + position + " | ₹" + salary;

    }

}

**EmployeeManager.java**

package employee;

public class EmployeeManager {

    private Employee[] employees;

    private int size;

    public EmployeeManager(int capacity) {

        employees = new Employee[capacity];

        size = 0;

    }

    // Add employee

    public boolean addEmployee(Employee emp) {

        if (size < employees.length) {

            employees[size++] = emp;

            return true;

        }

        return false; // Array full

    }

    // Search by ID

    public Employee searchEmployee(int id) {

        for (int i = 0; i < size; i++) {

            if (employees[i].employeeId == id) {

                return employees[i];

            }

        }

        return null;

    }

    // Traverse all employees

    public void displayAll() {

        if (size == 0) {

            System.out.println("No employee records found.");

        } else {

            for (int i = 0; i < size; i++) {

                System.out.println(employees[i]);

            }

        }

    }

    // Delete by ID

    public boolean deleteEmployee(int id) {

        for (int i = 0; i < size; i++) {

            if (employees[i].employeeId == id) {

                // Shift elements left

                for (int j = i; j < size - 1; j++) {

                    employees[j] = employees[j + 1];

                }

                employees[--size] = null;

                return true;

            }

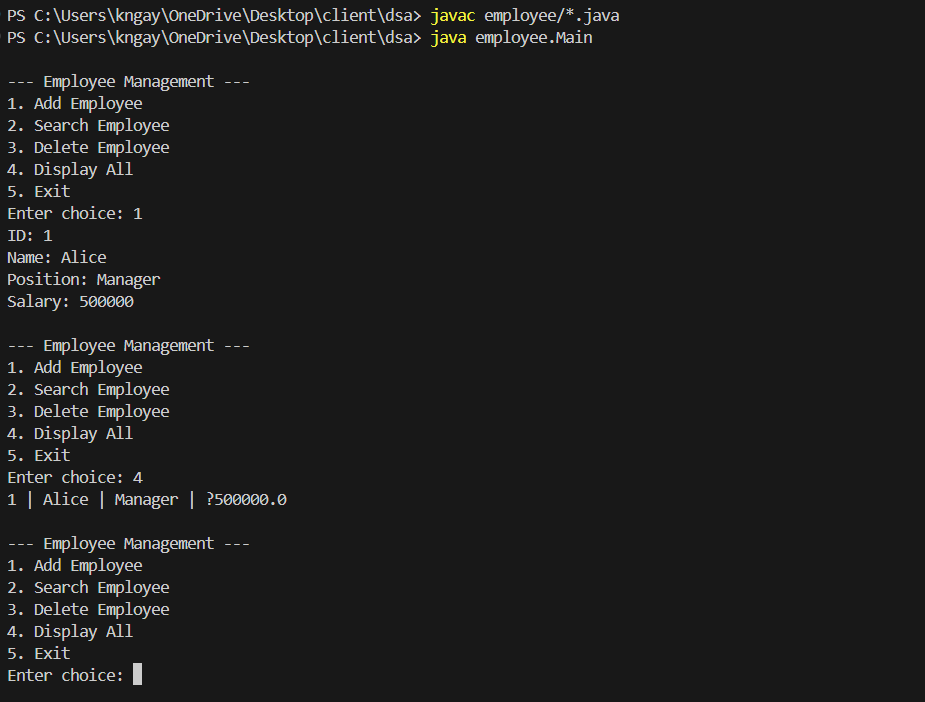
        }

        return false;

    }

}

**OUTPUT:**

****