Data Structures and Algorithms

SuperSet ID:6412063

Exercise 4: Employee Management System

Code:

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;

    }

    public String toString() {

        return "ID: " + employeeId + ", Name: " + name +

               ", Position: " + position + ", Salary: ₹" + salary;

    }

}

public class EmployeeSystem {

    private static final int MAX\_EMPLOYEES = 100;

    private static Employee[] employees = new Employee[MAX\_EMPLOYEES];

    private static int size = 0;

    public static void addEmployee(Employee emp) {

        if (size < MAX\_EMPLOYEES) {

            employees[size++] = emp;

            System.out.println("Employee added.");

        } else {

            System.out.println("Cannot add more employees. Limit reached.");

        }

    }

    public static Employee searchEmployee(int empId) {

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

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

                return employees[i];

            }

        }

        return null;

    }

    public static void displayAllEmployees() {

        if (size == 0) {

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

        } else {

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

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

            }

        }

    }

    public static void deleteEmployee(int empId) {

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

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

                // Shift left to overwrite deleted employee

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

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

                }

                employees[size - 1] = null; // clear last slot

                size--;

                System.out.println("Employee deleted.");

                return;

            }

        }

        System.out.println("Employee not found.");

    }

    public static void main(String[] args) {

        addEmployee(new Employee(1, "Dharshini", "Developer", 50000));

        addEmployee(new Employee(2, "Arun", "Manager", 75000));

        addEmployee(new Employee(3, "Rekha", "HR", 45000));

        System.out.println("\nAll Employees:");

        displayAllEmployees();

        System.out.println("\nSearching for Employee ID 2:");

        Employee found = searchEmployee(2);

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

        System.out.println("\nDeleting Employee ID 1:");

        deleteEmployee(1);

        System.out.println("\nEmployees after deletion:");

        displayAllEmployees();

    }

}

Output:

A screen shot of a computer

Description automatically generated