WEEK – 1

Exercise 3: Employee Management System

Nipuna Amanapu

superset id : 6432842

**Exercise 4: Employee Management System**

**Scenario:**

You are developing an employee management system for a company. Efficiently managing employee records is crucial.

**Steps:**

1. **Understand Array Representation:**
   * Explain how arrays are represented in memory and their advantages.
2. **Setup:**
   * Create a class Employee with attributes like **employeeId**, **name**, **position**, and **salary**.
3. **Implementation:**
   * Use an array to store employee records.
   * Implement methods to **add**, **search**, **traverse**, and **delete** employees in the array.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, search, traverse, delete).
   * Discuss the limitations of arrays and when to use them.

Solution:

1)employee.java

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;

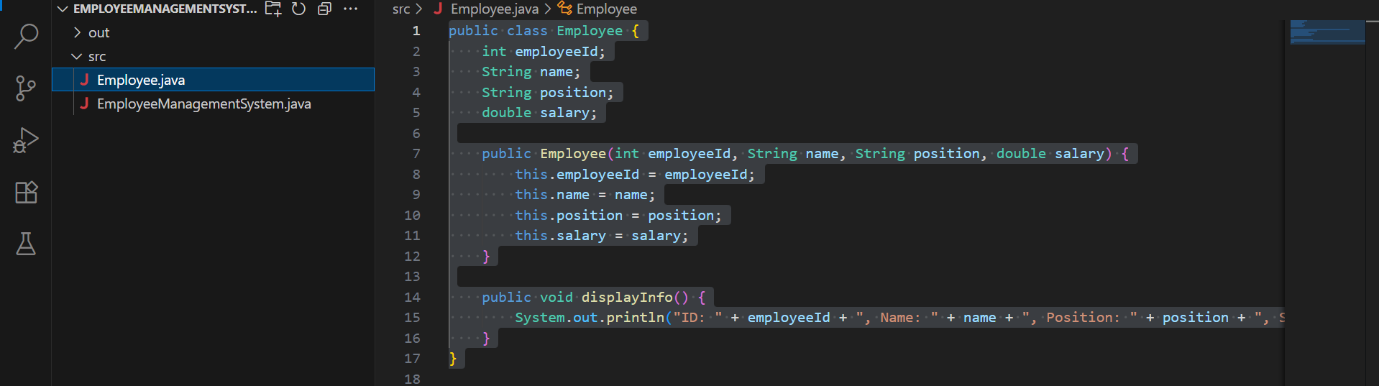
    }

    public void displayInfo() {

        System.out.println("ID: " + employeeId + ", Name: " + name + ", Position: " + position + ", Salary: " + salary);

    }

}



2)EmployeeManagementSystem.java

import java.util.Scanner;

public class EmployeeManagementSystem {

    private static final int MAX\_EMPLOYEES = 100;

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

    private static int count = 0;

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int choice;

        do {

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

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

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

            System.out.println("3. Display All Employees");

            System.out.println("4. Delete Employee");

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

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

            choice = scanner.nextInt();

            switch (choice) {

                case 1 -> addEmployee(scanner);

                case 2 -> searchEmployee(scanner);

                case 3 -> displayEmployees();

                case 4 -> deleteEmployee(scanner);

                case 5 -> System.out.println("Exiting program.");

                default -> System.out.println("Invalid choice.");

            }

        } while (choice != 5);

    }

    private static void addEmployee(Scanner scanner) {

        if (count >= MAX\_EMPLOYEES) {

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

            return;

        }

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

        int id = scanner.nextInt();

        scanner.nextLine();

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

        String name = scanner.nextLine();

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

        String position = scanner.nextLine();

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

        double salary = scanner.nextDouble();

        employees[count++] = new Employee(id, name, position, salary);

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

    }

    private static void searchEmployee(Scanner scanner) {

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

        int id = scanner.nextInt();

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

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

                employees[i].displayInfo();

                return;

            }

        }

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

    }

    private static void displayEmployees() {

        if (count == 0) {

            System.out.println("No employees to display.");

            return;

        }

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

            employees[i].displayInfo();

        }

    }

    private static void deleteEmployee(Scanner scanner) {

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

        int id = scanner.nextInt();

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

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

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

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

                }

                employees[--count] = null;

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

                return;

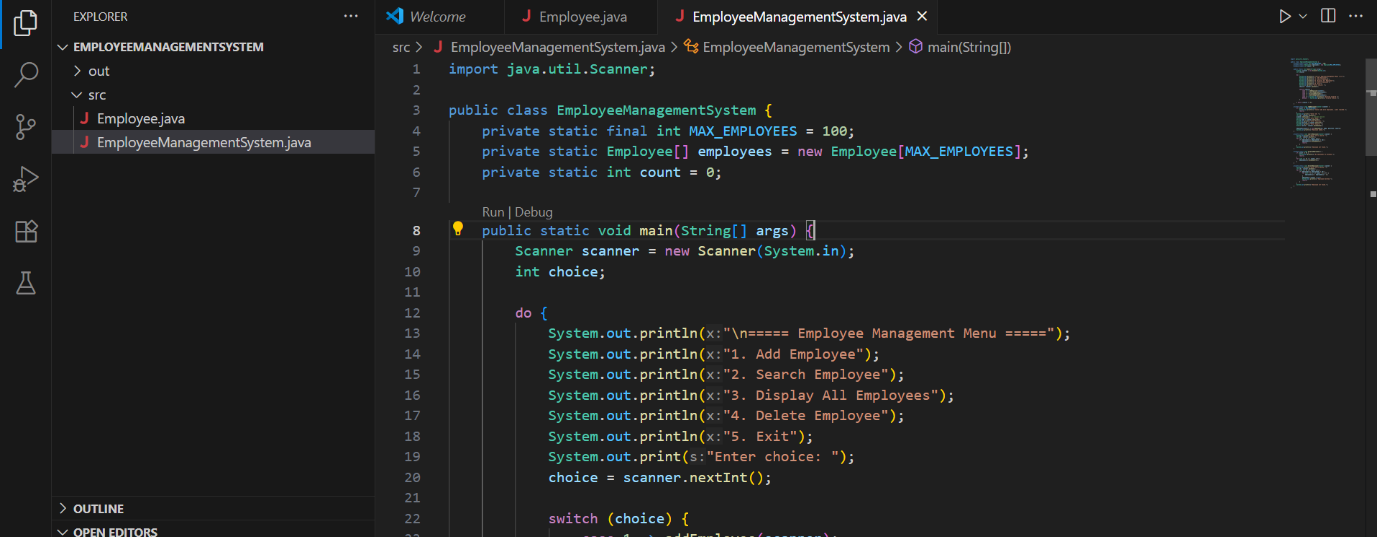
            }

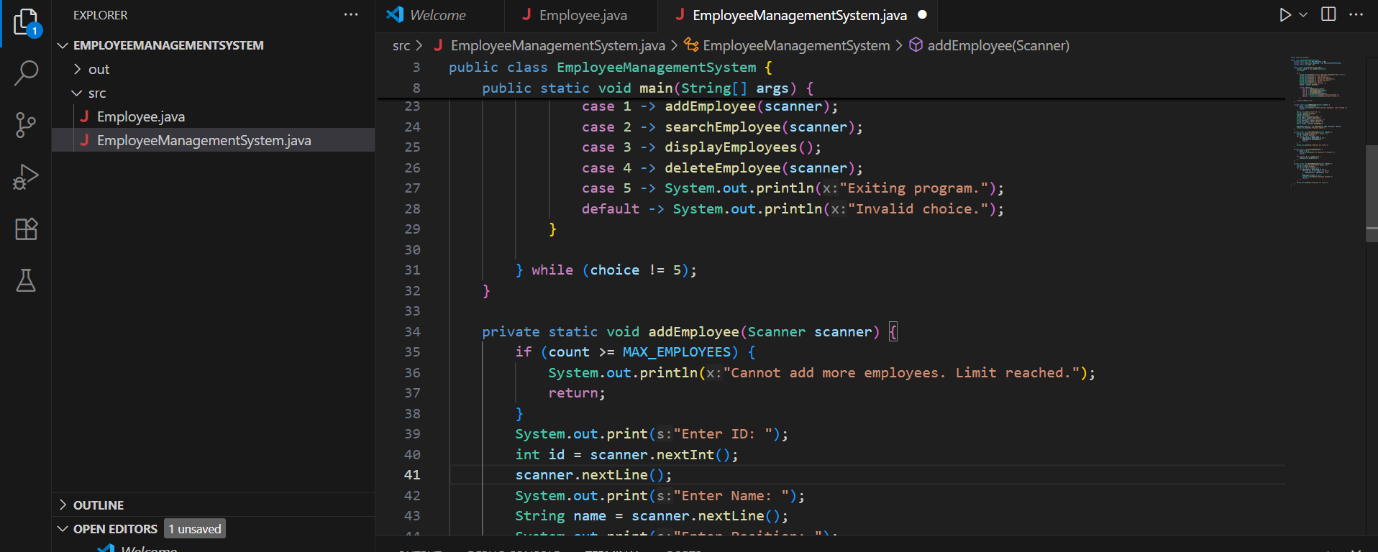
        }

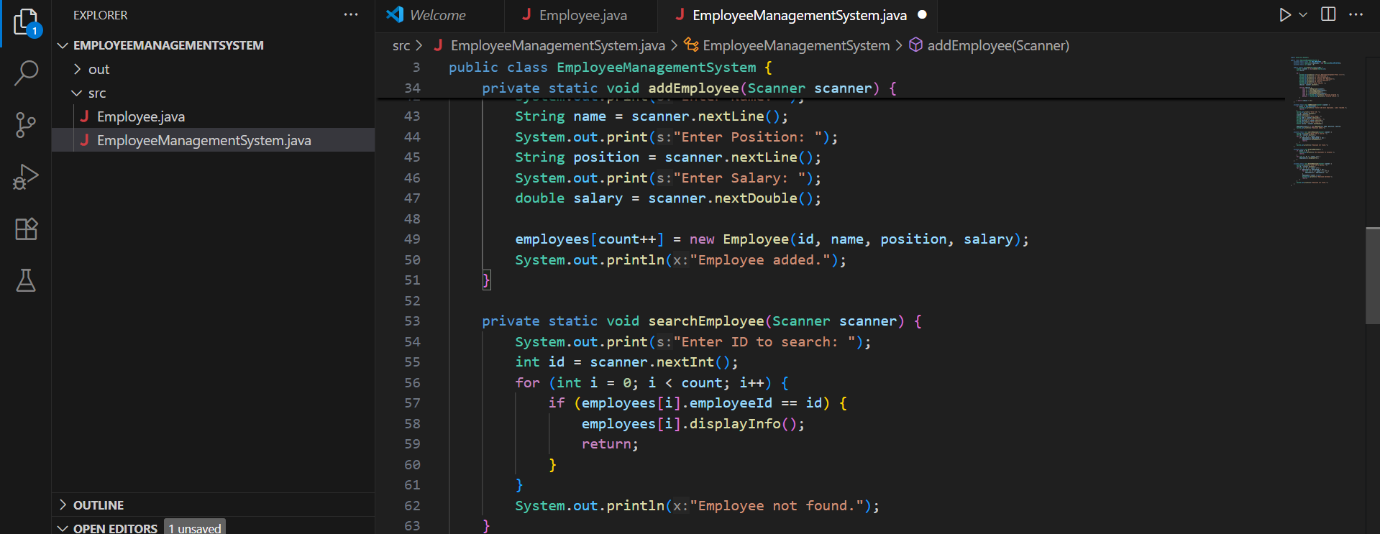
        System.out.println("Employee not found.");

    }

}







Output:

