DAILY TASK - 2

NAME : SANJEEV N
DATE : 06-08-2025

EMPLOYEE INFORMATION

```
1. View All Employees
2. Add Employee
3. Add Manager
4. Exit
Select an option (1-4): 1

All Employees:
ID: 1, Name: Sanjeev, Age: 21, Role: AI
ID: 2, Name: Eren, Age: 18, Role: Data Science
ID: 3, Name: Mikasa, Age: 21, Role: BlackBaud
ID: 4, Name: Armin, Age: 31, Role: BA
ID: 5, Name: Levi, Age: 21, Role: Tester
ID: 6, Name: Erwin, Age: 41, Role: Team Lead
ID: 7, Name: Grisha, Age: 21, Role: Senior Developer
ID: 8, Name: Zeke, Age: 21, Role: Intern
ID: 9, Name: Attack, Age: 21, Role: Senior Developer
ID: 10, Name: Founding, Age: 21, Role: Skip Level Manager
```

```
All Managers:
1. View All Employees
2. Add Employee
3. Add Manager
4. Exit
Select an option (1-4): 2

Enter Employee ID: 12
Enter Name: Historia
Enter Age: 20
Enter Role: Leader
Employee added successfully!
1. View All Employees
2. Add Employee
3. Add Manager
4. Exit
Select an option (1-4): 1

All Employees:
ID: 1, Name: Sanjeev, Age: 21, Role: AI
ID: 2, Name: Eren, Age: 18, Role: Data Science
ID: 3, Name: Mikasa, Age: 21, Role: BlackBaud
ID: 4, Name: Armin, Age: 31, Role: BA
ID: 5, Name: Levi, Age: 21, Role: Tester
ID: 6, Name: Erwin, Age: 41, Role: Team Lead
ID: 7, Name: Grisha, Age: 21, Role: Senior Developer
ID: 8, Name: Zeke, Age: 21, Role: Intern
ID: 9, Name: Attack, Age: 21, Role: Senior Developer
ID: 10, Name: Founding, Age: 21, Role: Skip Level Manager
ID: 12, Name: Historia, Age: 20, Role: Leader
```

PROGRAM.CS

```
using System;
using Simple.Models;
using Simple.Repo;
using System.Collections.Generic;

namespace Simple
{
    class Program
    {
        static void Main(string[] args)
        {
            EmployeeRepo<Employee> employeeRepo = new EmployeeRepo<Employee>();
            List<Manager> managers = new List<Manager>();

            employeeRepo.Add(new Employee(1, "Sanjeev", 21, "AI"));
```

```
employeeRepo.Add(new Employee(2, "Eren", 18, "Data Science"));
employeeRepo.Add(new Employee(3, "Mikasa", 21, "BlackBaud"));
employeeRepo.Add(new Employee(4, "Armin", 31, "BA"));
employeeRepo.Add(new Employee(5, "Levi", 21, "Tester"));
employeeRepo.Add(new Employee(6, "Erwin", 41, "Team Lead"));
employeeRepo.Add(new Employee(7, "Grisha", 21, "Senior Developer"));
employeeRepo.Add(new Employee(8, "Zeke", 21, "Intern"));
employeeRepo.Add(new Employee(9, "Attack", 21, "Senior Developer"));
employeeRepo.Add(new Employee(10, "Founding", 21, "Skip Level Manager"));

bool running = true;

while (running)
{
    Console.WriteLine("1. View All Employees");
    Console.WriteLine("2. Add Employee");
    Console.WriteLine("3. Add Manager");
    Console.WriteLine("4. Exit");
    Console.Write("Select an option (1-4): ");

    string? input = Console.ReadLine();
    Console.WriteLine();

    switch (input)
    {
        case "1":
            Console.WriteLine("All Employees:");
            foreach (var emp in employeeRepo.GetAll())
            {
                Console.WriteLine(emp.GetInfo());
            }

            Console.WriteLine("\nAll Managers:");
            foreach (var mgr in managers)
            {
                Console.WriteLine(mgr.GetInfo());
            }
            break;

        case "2":
            Console.Write("Enter Employee ID: ");
            int id = int.Parse(Console.ReadLine()!);

            Console.Write("Enter Name: ");
```

```
                string name = Console.ReadLine()!;

                Console.Write("Enter Age: ");
                int age = int.Parse(Console.ReadLine()!);

                Console.Write("Enter Role: ");
                string role = Console.ReadLine()!;

                employeeRepo.Add(new Employee(id, name, age, role));
                Console.WriteLine("Employee added successfully!");
                break;

            case "3":
                Console.Write("Enter Manager ID: ");
                int mid = int.Parse(Console.ReadLine()!);

                Console.Write("Enter Name: ");
                string mname = Console.ReadLine()!;

                Console.Write("Enter Age: ");
                int mage = int.Parse(Console.ReadLine()!);

                Console.Write("Enter Role: ");
                string mrole = Console.ReadLine()!;

                Console.Write("Enter Team Size: ");
                int teamSize = int.Parse(Console.ReadLine()!);

                managers.Add(new Manager(mid, mname, mage, mrole, teamSize));
                Console.WriteLine("Manager added successfully!");
                break;

            case "4":
                running = false;
                Console.WriteLine("Exiting...");
                break;

            default:
                Console.WriteLine("Invalid option. Please try again.");
                break;
        }
    }
  }
}
```

```
}
```

EMPLOYEE.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Simple.Models
{
    internal class Employee
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int Age { get; set; }
        public string Role { get; set; }

        public Employee(int id, string name, int age, string role)
        {
            Id = id;
            Name = name;
            Age = age;
            Role = role;
        }

        public virtual string GetInfo()
        {
            return $"ID: {Id}, Name: {Name}, Age: {Age}, Role: {Role}";
        }
    }
}
```

MANAGER.CS

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;
```

```
namespace Simple.Models
{
    internal class Manager : Employee
    {
        public int TeamSize { get; set; }

        public Manager(int id, string name, int age, string role, int teamSize): base(id, name, age,
role)
        {
            TeamSize = teamSize;
        }

        public override string GetInfo()
        {
            return $" ID: {Id}, Name: {Name}, Age: {Age}, Role: {Role}, Team Size: {TeamSize}";
        }
    }
}
```

EMPLOYEEREPO.CS

```
using System.Collections.Generic;
using System.Linq;
using Simple.Models;

namespace Simple.Repo
{
    internal class EmployeeRepo<T> : IEmployeeRepo<T> where T : Employee
    {
        private readonly List<T> employees = new List<T>();

        public void Add(T emp)
        {
            employees.Add(emp);
        }

        public List<T> GetAll()
        {
            return employees;
        }

        public T GetById(int id)
        {
```

```
            return employees.FirstOrDefault(e => e.Id == id);
        }
    }
}
```

IEMPLOYEEREPO.CS

```
using Simple.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Simple.Repo
{
    internal interface IEmployeeRepo<T> where T : Employee
    {
        void Add(T emp);
        List<T> GetAll();
        T GetById(int id);
    }
}
```