|  |
| --- |
| Day-22 Morning Assignment  By  U.Joshna  [22-2-2022] |

|  |
| --- |
| Write a C# Code for Employee Management Application.For Adding Employee,Search Employee & Display All Employees. |
| Code: |
| Creating a Solution name as “JoshnaFinalProject” Which Consists of 3 Layers of UI/Presentation Layer(JoshClientApp{ConsoleApp}).  Business Logic Layer(BusinessLogicLibrary{ClassLibrary}).  Data Access Layer(DataAccessLibrary{ClassLibrary}). |
|  |
| DataAccessLibrary: |
| using System;  using System.IO;  using System.Collections.Generic;  namespace DataAccessLibrary  {  public class EmployeeDAL  {  public static string filepath = "E:\\NHTraining1\\Day-22\\Employees.txt";  private static int id;    public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  try  {  String textContent = String.Concat(empId, ",", empName, ",", empSalary, ",", empAge, ",");  File.AppendAllText(filepath, textContent + Environment.NewLine);  return true;  }  catch (Exception )  {  return false;  }  }  public static List<string> GetEmployeesById(int empId)  {  var allEmployees = File.ReadAllLines(filepath);  // bool isFound = false;  List<string> employeeFound = new List<string>();    foreach (String employee in allEmployees)  {  var empDetails = employee.Split(',');  if (Convert.ToInt32(empDetails[0]) == id)    {  // isFound = true;  employeeFound.Add(employee);  break;  }  }  return employeeFound;  }  public static List<string> GetEmployeesByName(string empName)  {  var allEmployees = File.ReadAllLines(filepath);    List<string> employeeFound = new List<string>();    foreach (String employee in allEmployees)  {  var empDetails = employee.Split(',');  if (empDetails[1].Contains(empName))  {    employeeFound.Add(employee);    }    }  return employeeFound;    }  public static String[] GetAllEmployees()  {  var allEmployees = File.ReadAllLines(filepath);  return allEmployees;  }      }  } |
| BusinessLogicLibrary: |
| using System;  using System.Collections.Generic;  using DataAccessLibrary;    namespace BusinessLogicLibrary  {  public class EmployeeBLL  {  public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  //To Do Add Validations    //If All Validations are successful than call DAL  var result = EmployeeDAL.AddEmployee(empId, empName, empSalary, empAge);  return result;  }  public static List<String> GetEmployeesById(int id)  {  var result = EmployeeDAL.GetEmployeesById(id);  return result;  }  public static List<String> GetEmployeesByName(string name)  {  var result = EmployeeDAL.GetEmployeesByName(name);  return result;    }  public static String[] GetAllEmployees()  {  var result = EmployeeDAL.GetAllEmployees();  return result;  }    public static object AddEmployee(int id, object name, int salary, int age)  {  throw new NotImplementedException();  }  }  }       |  | | --- | | JoshClientApp: | | using System;  using BusinessLogicLibrary;  namespace JoshClientApp  {  public class Program  {  private static int id;    static void Main(string[] args)  {  int ch;  string choice;    do  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("Employee Management Application");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("1.Add Employee:");  Console.WriteLine("2.Search Employee By Id:");  Console.WriteLine("3.Search Employee By Name:");  Console.WriteLine("4.Display All Employees:");  Console.WriteLine("Enter your choice:");  ch = Convert.ToInt32(Console.ReadLine());    switch(ch)  {  case 1:  AddEmployee();  break;  case 2:  SearchEmployeeById();  break;  case 3:  SearchEmployeeByName();  break;  case 4:  DisplayAllEmployees();  break;  default:  Console.WriteLine("Invalid Option");  break;    }  Console.WriteLine("Do you Want to continue (y/n):");  choice = Console.ReadLine();  } while (choice.Equals("y"));  }  public static void AddEmployee()  {  int Id, Salary, Age;  string name;  Console.WriteLine("Enter Id:");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Salary:");  Salary = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Age:");  Age = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Name:");  name =( Console.ReadLine());  //call BLL Method  var result = EmployeeBLL.AddEmployee(id, name, Salary, Age);    if (result)  Console.WriteLine("Employee Details Saved Successfully");  else  Console.WriteLine("Some error occured");    }  public static void SearchEmployeeById()  {  int Id;  Console.WriteLine("Enter Id:");  Id = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.GetEmployeesById(Id);  if (result.Count == 0)  Console.WriteLine("No records exists with this Id");  else  {  result.ForEach(p => Console.WriteLine(p));    }    }  public static void SearchEmployeeByName()  {  string Name;  Console.WriteLine("Enter Name:");  Name=Console.ReadLine();  List<string> result = EmployeeBLL.GetEmployeesByName(Name);  if (result.Count == 0)  Console.WriteLine("The Employees Details,Found with Given Name");  else  {  result.ForEach(p => Console.WriteLine(p));    }  }  public static void DisplayAllEmployees()  {  var employees = EmployeeBLL.GetAllEmployees();  Console.WriteLine("Displaying All Employee Details in the Record");  foreach (var employee in employees)  {  Console.WriteLine(employee);  }  Console.ReadLine();  }  }  } | | Output: | |  | |  | |