|  |
| --- |
| Day22(22 Feb) Assignment  By Ramakrishna |

|  |
| --- |
| Create Employee Management Application |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.IO;  using System.Text;  using System.Threading.Tasks;  namespace DataAccessesLibrary  {  public static class EmployeeDal  {  public static string Filepath = "C:\\Final project data\\Employee Data.txt";  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 ex)  {  return false;  }  }  public static List<string> GetEmployesById(int id)  {  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> GetEmployesByName(String name)  {  var allEmployees = File.ReadAllLines(Filepath);    List<String> employeeFound = new List<String>();  foreach (string employee in allEmployees)  {  var empDetails = employee.Split(',');  if (empDetails[1].Contains(name))  {    employeeFound.Add(employee);    }  }  return employeeFound;  }  public static string[] GetAllEmployees()  {  var allEmployees = File.ReadAllLines(Filepath);  return allEmployees;  }  }  } |
| BusinessLogicLibrary: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAccessesLibrary;  namespace BusinessLogicLibrary  {  public class EmployeeBLL  {  public static bool AddEmployee(int empid, string empname, int empsalary, int empage)  {  // to Do validations  //iF all validations are sucessful then call DAL  var result = EmployeeDal.AddEmployee(empid, empname, empsalary, empage);  return result;  }  public static List<string> GetEmployesById(int id)  {  var result = EmployeeDal.GetEmployesById(id);  return result;  }  public static List<string> GetEmployesByName(String name)  {  var result = EmployeeDal.GetEmployesByName(name);  return result;  }  public static String[] GetAllEmployees( )  {  var result = EmployeeDal.GetAllEmployees();  return result;    }  }  } |
|  |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BusinessLogicLibrary;  namespace MyClientApp  {  internal class Program  {  static void Main(string[] args)  {  int ch;  string choice;    do  {  Console.WriteLine("$\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*$");  Console.WriteLine("Employee Managment Appplication");  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 name:");  name = Console.ReadLine();  Console.WriteLine("Enter salary:");  salary = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter age:");  age = Convert.ToInt32(Console.ReadLine());  //call BLL Method  var result = EmployeeBLL.AddEmployee(id, name, salary, age);  if (result)  Console.WriteLine("Employee details saves 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.GetEmployesById(id);  if(result.Count == 0)  Console.WriteLine("no records exists with this id");  else  {  result.ForEach(p => Console.WriteLine(p.ToString()));  }  }  public static void SearchEmployeeByName()  {  string name;  Console.WriteLine("Enter name");  name= Console.ReadLine();  var result = EmployeeBLL.GetEmployesByName(name);  foreach(var item in result)  {  Console.WriteLine(item);  }  }  public static void DisplayAllEmployees()  {  var result = EmployeeBLL.GetAllEmployees();  foreach(var item in result)  {  Console.WriteLine(item);  }  }  }  } |

|  |
| --- |
| Output: |
| **AddEmployee:** |
| **SearchEmployeeByID:** |
| **SearchEmployeeByName:** |
| **Display AllEmployees:** |
|  |