|  |
| --- |
| **Day-22 Assignment**  **By**  **Bhanu Rama Krishna Prakash Jakkamsetti**  **22/2/2022** |

|  |
| --- |
| Final Project On C# |
| Code for DLL: |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DataAccesLibrary  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* author:bhanu rama krishna prakash jakkamsetti  \* prpose:3 layer archtecture  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  public static class EmployeeDAL  {  public static string Filepath = "C:\\Krish\\c#\\NHTraining\\Day22-Bhanu Rama Krishna Prakash [FianlProject]\\EmployeeData.txt";  /// <summary>  /// adding all emp details  /// </summary>  /// <param name="empId"></param>  /// <param name="empName"></param>  /// <param name="empSalary"></param>  /// <param name="empAge"></param>  /// <returns>bool</returns>  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 e)  {  return false;  }  }  /// <summary>  /// get emp details by using id  /// </summary>  /// <param name="id"></param>  /// <returns>string</returns>  public static List<string> GetEmployeesById(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;  }  /// <summary>  /// get emp details by using name  /// </summary>  /// <param name="name"></param>  /// <returns>string</returns>  public static List<string> GetEmployeesByNAme(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;  }  /// <summary>  /// get all emp details  /// </summary>  /// <returns>string</returns>  public static string[] GetAllEmployees()  {  var allEmployees=File.ReadAllLines(Filepath);  return allEmployees;  }  }  } |
| Code for BLL: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAccesLibrary;  namespace BiginessLogicLibrary  {  public class EmployeeBLL  {  /// <summary>  /// call dal in bll for add details  /// </summary>  /// <param name="empId"></param>  /// <param name="empName"></param>  /// <param name="empSalary"></param>  /// <param name="empAge"></param>  /// <returns></returns>  public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  // TO DO validation  if (empId < 0)  {  Console.WriteLine("id should not be -ve number");  }  for (int i = 1; i <= empId; i++)  {  if (empId == i)  {  Console.WriteLine("id not same as old value");  }  }  if (empName.Length < 3)  {  Console.WriteLine("name must >3 characters");  }  if (empSalary < 10000)  {  Console.WriteLine("salary more than 10000");  }  if (empAge >= 18 && empAge <= 58)  {  Console.WriteLine("age only between 18-58");  }  //If all validatins are succesful then call DLL  var result =EmployeeDAL.AddEmployee(empId, empName, empSalary, empAge);  return result;  }  /// <summary>  /// call dal in bll for get em datails by using id  /// </summary>  /// <param name="id"></param>  /// <returns></returns>  public static List<string> GetEmployeesById(int id)  {    var result=EmployeeDAL.GetEmployeesById(id);  return result;  }  /// <summary>  /// call dal in bll for get em datails by using name  /// </summary>  /// <param name="name"></param>  /// <returns></returns>  public static List<string> GetEmployeesByNAme(string name)  {  var result = EmployeeDAL.GetEmployeesByNAme( name);  return result;  }  /// <summary>  /// call dal in bll for get all emp details  /// </summary>  /// <returns></returns>  public static string[] GetAllEmployees()  {  var result= EmployeeDAL.GetAllEmployees();  return result;  }  }  } |
| Code for console: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BiginessLogicLibrary;  namespace MyClientApp  {  internal class Program  {  static void Main(string[] args)  {  string choice;  int ch;  do  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("Employee Management Application");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("1.Add Employee details:");  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:  AddEpmloyee();  break;  case 2:  SearchEmployeeById();  break;  case 3:  SearchEmployeeByName();  break;  case 4:  DisplayAllEmployees();  break;  default:  Console.WriteLine("invadid option");  break;  }  Console.WriteLine("Do you want to continue (y/n):");  choice = Console.ReadLine();  }  while (choice.Equals("y"));  Console.ReadLine();  }  /// <summary>  /// read all emp data  /// </summary>  public static void AddEpmloyee()  {  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  var result = EmployeeBLL.AddEmployee(id,name, salary, age);    if (result)  Console.WriteLine("employee details saved succesfully");  else  Console.WriteLine("error");  }  /// <summary>  /// search emp data by id  /// </summary>  public static void SearchEmployeeById()    {  int id;  Console.WriteLine("enter id");  id = Convert.ToInt32(Console.ReadLine());  //call BLL  var result = EmployeeBLL.GetEmployeesById(id);  if (result.Count==0)  Console.WriteLine("no records found");  else  {  foreach (var item in result)  {  Console.WriteLine(item);  }  }  }  /// <summary>  /// search emp data by name  /// </summary>  public static void SearchEmployeeByName()  {  string name;  Console.WriteLine("enter name");  name =Console.ReadLine();  //Call BLL  var result=EmployeeBLL.GetEmployeesByNAme(name);  foreach (var item in result)  {  Console.WriteLine(item);  }  }  /// <summary>  /// display all employee details  /// </summary>  public static void DisplayAllEmployees()  {  //Call BLL  var result=EmployeeBLL.GetAllEmployees();  foreach (var item in result)  {  Console.WriteLine(item);  }  }      }  } |
| Output for text file: |
|  |
| Output for add employee details: |
|  |
| Output for search employee details by id: |
|  |
| Output for search employee details by name: |
|  |
| Output for display all employee details: |
|  |

|  |
| --- |
| Using Validations |
| Code For DAL: |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DataAccesLibrary  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* author:bhanu rama krishna prakash jakkamsetti  \* prpose:3 layer archtecture  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  public static class EmployeeDAL  {  public static string Filepath = "C:\\Krish\\c#\\NHTraining\\Day22-Bhanu Rama Krishna Prakash [FianlProject]\\EmployeeData.txt";  /// <summary>  /// adding all emp details  /// </summary>  /// <param name="empId"></param>  /// <param name="empName"></param>  /// <param name="empSalary"></param>  /// <param name="empAge"></param>  /// <returns>bool</returns>  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 e)  {  return false;  }  }  /// <summary>  /// get emp details by using id  /// </summary>  /// <param name="id"></param>  /// <returns>string</returns>  public static List<string> GetEmployeesById(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;  }  /// <summary>  /// get emp details by using name  /// </summary>  /// <param name="name"></param>  /// <returns>string</returns>  public static List<string> GetEmployeesByNAme(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;  }  /// <summary>  /// get all emp details  /// </summary>  /// <returns>string</returns>  public static string[] GetAllEmployees()  {  var allEmployees=File.ReadAllLines(Filepath);  return allEmployees;  }  }  } |
| Code for BLL: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAccesLibrary;  namespace BiginessLogicLibrary  {  public class EmployeeBLL  {  /// <summary>  /// call dal in bll for add details  /// </summary>  /// <param name="empId"></param>  /// <param name="empName"></param>  /// <param name="empSalary"></param>  /// <param name="empAge"></param>  /// <returns></returns>  public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  // TO DO validation  if (empId < 0)  {  Console.WriteLine("id should not be -ve number");  }  for (int i = 1; i <= empId; i++)  {  if (empId == i)  {  Console.WriteLine("id not same as old value");  }  }  if (empName.Length < 3)  {  Console.WriteLine("name must >3 characters");  }  if (empSalary < 10000)  {  Console.WriteLine("salary more than 10000");  }  if (empAge >= 18 && empAge <= 58)  {  Console.WriteLine("age only between 18-58");  }  //If all validatins are succesful then call DLL  var result =EmployeeDAL.AddEmployee(empId, empName, empSalary, empAge);  return result;  }  /// <summary>  /// call dal in bll for get em datails by using id  /// </summary>  /// <param name="id"></param>  /// <returns></returns>  public static List<string> GetEmployeesById(int id)  {    var result=EmployeeDAL.GetEmployeesById(id);  return result;  }  /// <summary>  /// call dal in bll for get em datails by using name  /// </summary>  /// <param name="name"></param>  /// <returns></returns>  public static List<string> GetEmployeesByNAme(string name)  {  var result = EmployeeDAL.GetEmployeesByNAme( name);  return result;  }  /// <summary>  /// call dal in bll for get all emp details  /// </summary>  /// <returns></returns>  public static string[] GetAllEmployees()  {  var result= EmployeeDAL.GetAllEmployees();  return result;  }  }  } |
| Code for console: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BiginessLogicLibrary;  namespace MyClientApp  {  internal class Program  {  static void Main(string[] args)  {  string choice;  int ch;  do  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("Employee Management Application");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("1.Add Employee details:");  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:  AddEpmloyee();  break;  case 2:  SearchEmployeeById();  break;  case 3:  SearchEmployeeByName();  break;  case 4:  DisplayAllEmployees();  break;  default:  Console.WriteLine("invadid option");  break;  }  Console.WriteLine("Do you want to continue (y/n):");  choice = Console.ReadLine();  }  while (choice.Equals("y"));  Console.ReadLine();  }  /// <summary>  /// read all emp data  /// </summary>  public static void AddEpmloyee()  {  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  var result = EmployeeBLL.AddEmployee(id,name, salary, age);    if (result)  Console.WriteLine("employee details saved succesfully");  else  Console.WriteLine("error");  }  /// <summary>  /// search emp data by id  /// </summary>  public static void SearchEmployeeById()    {  int id;  Console.WriteLine("enter id");  id = Convert.ToInt32(Console.ReadLine());  //call BLL  var result = EmployeeBLL.GetEmployeesById(id);  if (result.Count==0)  Console.WriteLine("no records found");  else  {  foreach (var item in result)  {  Console.WriteLine(item);  }  }  }  /// <summary>  /// search emp data by name  /// </summary>  public static void SearchEmployeeByName()  {  string name;  Console.WriteLine("enter name");  name =Console.ReadLine();  //Call BLL  var result=EmployeeBLL.GetEmployeesByNAme(name);  foreach (var item in result)  {  Console.WriteLine(item);  }  }  /// <summary>  /// display all employee details  /// </summary>  public static void DisplayAllEmployees()  {  //Call BLL  var result=EmployeeBLL.GetAllEmployees();  foreach (var item in result)  {  Console.WriteLine(item);  }  }      }  } |
| Output: |
|  |