**Course Project   
DeVry University  
College of Engineering and Information Sciences**

***Screenshot of program running:***

A screenshot of a computer

Description automatically generated

***Form code (only the code for the form and classes, not program.cs):***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Rodriguez\_CourseProject\_Part2

{

public partial class PayrollButton : Form

{

private object sr;

private object parts;

private string healthIns;

private object frmupdate;

public PayrollButton()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void AddButton\_Click(object sender, EventArgs e)

{

// add item to the Employee listbox

InputForm frmInput = new InputForm();

using (frmInput)

{

DialogResult result = frmInput.ShowDialog();

// see if the input form was cancelled

if (result == DialogResult.Cancel)

return; // end the method since cancelled

// get user's input and create Employee object

string fName = frmInput.FirstNameTextBox.Text;

string lName = frmInput.LastNameTextBox.Text;

string ssn = frmInput.SSNTextBox.Text;

string date = frmInput.HireDateTextBox.Text;

DateTime hireDate = DateTime.Parse(date);

string healthIns = frmInput.HealthInsuranceTextBox.Text;

int lifeIns = int.Parse(frmInput.LifeInsuranceTextBox.Text);

int vacation = int.Parse(frmInput.VacationDaysTextBox.Text);

Benefits ben = new Benefits(healthIns, lifeIns, vacation);

Employee emp = new Employee(fName, lName, ssn, hireDate, ben);

// add the Employee object to the Employee listbox

EmployeesListBox.Items.Add(emp);

// write all date to the file

WriteEmpsToFile();

}

}

private void WriteEmpsToFile()

{

string filename = "Employees.csv";

StreamWriter sw = new StreamWriter(filename);

foreach (Employee emp in EmployeesListBox.Items)

{

sw.WriteLine(emp.FirstName + ','

+ emp.LastName + ','

+ emp.SSN + ','

+ emp.HireDate.ToShortDateString() + ','

+ emp.BenefitsEmp.HealthInsuraance + ','

+ emp.BenefitsEmp.LifeInsurance + ','

+ emp.BenefitsEmp.Vacation);

}

sw.Close();

}

private void RemoveButton\_Click(object sender, EventArgs e)

{

// remove the selected item from the employee listbox

int itemNumber = EmployeesListBox.SelectedIndex;

if (itemNumber > -1)

{

EmployeesListBox.Items.RemoveAt(itemNumber);

WriteEmpsToFile();

}

else

{

MessageBox.Show("Please select employee to remove.");

}

}

private void DisplayButton\_Click(object sender, EventArgs e)

{

// clear the Employee listbox

EmployeesListBox.Items.Clear();

// read employees from the file

string filename = "Employees.csv";

StreamReader sr = new StreamReader(filename);

while( sr.Peek() > -1 )

{

string line = sr.ReadLine();

string[] parts = line.Split(',');

string firstName = parts[0];

string lastName = parts[1];

string ssn = parts[2];

DateTime hireDate = DateTime.Parse(parts[3]);

string empName = parts[4];

int lifeIns = int.Parse(parts[5]);

int vacation = int.Parse(parts[6]);

Benefits ben = new Benefits(healthIns, lifeIns, vacation);

Employee emp = new Employee(firstName, lastName,

ssn, hireDate, ben);

EmployeesListBox.Items.Add(emp);

}

sr.Close();

}

private void PrintPaychecksButton\_Click(object sender, EventArgs e)

{

MessageBox.Show("Printing paychecks for all employees...");

}

private void EmployeesListBox\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void EmployeesListBox\_DoubleClick(object sender, EventArgs e)

{

// get the selected Employee object

Employee emp = EmployeesListBox.SelectedItem as Employee;

// show the Input/Update form with the Employee info

InputForm frmUpdate = new InputForm();

frmUpdate.FirstNameTextBox.Text = emp.FirstName;

frmUpdate.LastNameTextBox.Text = emp.LastName;

frmUpdate.SSNTextBox.Text = emp.SSN;

frmUpdate.HireDateTextBox.Text = emp.HireDate.ToString();

frmUpdate.HealthInsuranceTextBox.Text = emp.BenefitsEmp.HealthInsuraance;

frmUpdate.LifeInsuranceTextBox.Text = emp.BenefitsEmp.LifeInsurance.ToString();

frmUpdate.VacationDaysTextBox.Text = emp.BenefitsEmp.Vacation.ToString();

DialogResult result = frmUpdate.ShowDialog();

// if can cancelled, stop the method

if (result == DialogResult.Cancel)

return; // end the method

// delete the selected object

int position = EmployeesListBox.SelectedIndex;

EmployeesListBox.Items.RemoveAt(position);

// create new employee using the updated information

Employee newEmp = new Employee();

newEmp.FirstName = frmUpdate.FirstNameTextBox.Text;

newEmp.LastName = frmUpdate.LastNameTextBox.Text;

newEmp.SSN = frmUpdate.SSNTextBox.Text;

newEmp.HireDate = DateTime.Parse(frmUpdate.HireDateTextBox.Text);

newEmp.BenefitsEmp.HealthInsurance = frmUpdate.HealthInsuranceTextBox.Text;

newEmp.BenefitsEmp.LifeInsurance = int.Parse(frmUpdate.LifeInsuranceTextBox.Text);

newEmp.BenefitsEmp.Vacation = int.Parse(frmUpdate.VacationDaysTextBox.Text);

// add the new employee to the list box

EmployeesListBox.Items.Add(newEmp);

}

}

}

Note: InputForm.cs Code

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Rodriguez\_CourseProject\_Part2

{

public partial class InputForm : Form

{

public InputForm()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void SubmitButton\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.OK;

this.Hide();

}

private void ExitButton\_Click(object sender, EventArgs e)

{

this.DialogResult= DialogResult.Cancel;

this.Hide();

}

private void HireDateTextBox\_TextChanged(object sender, EventArgs e)

{

}

}

}

Employee.cs Code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Runtime.ExceptionServices;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

namespace Rodriguez\_CourseProject\_Part2

{

public class Employee

{

// attributes

private string firstName;

private string lastName;

private string ssn;

private DateTime hireDate;

private Benefits benefits;

// constructors

public Employee()

{

firstName = "N/A";

lastName = "N/A";

ssn = "N/A";

hireDate = new DateTime();

benefits = new Benefits();

}

public Employee(string firstname, string lastname,

string ssn, DateTime hireDate, Benefits benefits)

{

this.firstName = firstname;

this.lastName = lastname;

this.ssn = ssn;

this.hireDate = hireDate;

this.benefits = benefits;

}

// behaviors

public override string ToString()

{

return "firstname=" + firstName

+ ", lastName=" + lastName

+ ", ssn=" + ssn

+ ", hireDate=" + hireDate.ToShortDateString()

+ ", healthInsurance=" + benefits.HealthInsuraance

+ ", lifeInsurance=" + benefits.LifeInsurance

+ ", vacation=" + benefits.Vacation;

}

public double CalculatePay()

{

return 0;

}

// properties

public string FirstName

{

get { return firstName; }

set { firstName = value; } // emp.FirstName = "Bob";

}

public string LastName

{

get { return lastName; }

set { lastName = value; }

}

public string SSN

{

get { return ssn; }

set { ssn = value; }

}

public DateTime HireDate

{

get { return hireDate; }

set { hireDate = value; }

}

public Benefits BenefitsEmp

{

get { return benefits; }

set { benefits = value; }

}

}

}

Benefits.cs Code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Rodriguez\_CourseProject\_Part2

{

public class Benefits

{

// attributes

private string healthInsuraance;

private int lifeInsurance;

private int vacation;

// constructors

public Benefits()

{

healthInsuraance ="N/A";

lifeInsurance = 0;

vacation = 0;

}

public Benefits(string healthInsuraance, int lifeInsurance, int vacation)

{

this.healthInsuraance = healthInsuraance;

this.lifeInsurance = lifeInsurance;

this.vacation = vacation;

}

// behaviors

public override string ToString()

{

return "healthInsurance=" + healthInsuraance

+ ", lifeInsurance=" + lifeInsurance

+ ", vacation=" + vacation;

}

// properties

public string HealthInsuraance

{

get { return healthInsuraance; }

set { healthInsuraance = value; }

}

public int LifeInsurance

{

get { return lifeInsurance; }

set { lifeInsurance = value;}

}

public int Vacation

{

get{ return vacation; }

set { vacation = value; }

}

public string HealthInsurance { get; internal set; }

}

}