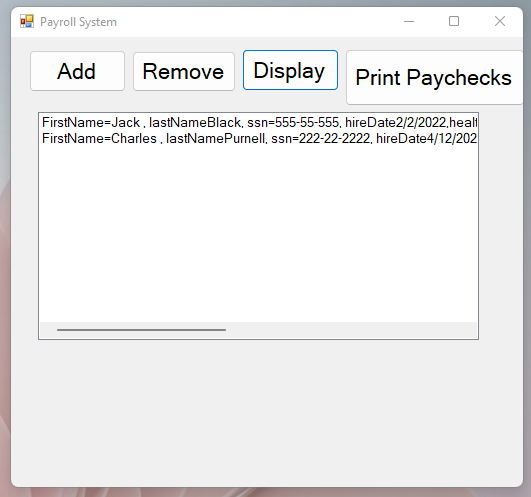
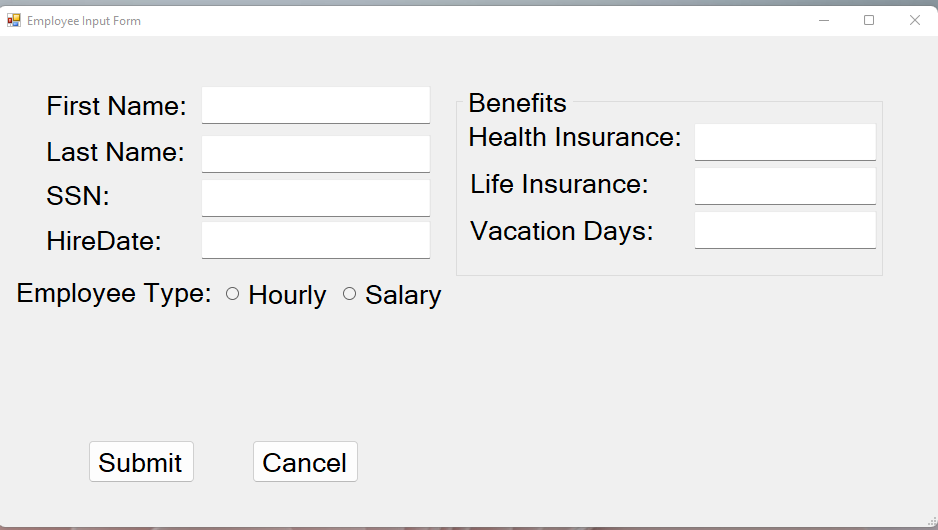
**Course Project   
DeVry University  
College of Engineering and Information Sciences**

***Screenshot of program running:***





***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.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Purnell\_CourseProject\_Part2

{

[Serializable]

public partial class MainForm : Form

{

// class level references

private const string FILENAME = "Employee.dat";

public MainForm()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

// check to see if file exists

if (File.Exists(FILENAME) && new FileInfo(FILENAME).Length > 0)

{

// create a pipe fromt the and create translator

FileStream fs = new FileStream(FILENAME, FileMode.Open);

BinaryFormatter formatter = new BinaryFormatter();

// read the generic list from the file

List<Employee> list = (List<Employee>)formatter.Deserialize(fs);

// close the pipe

fs.Close();

// clear the Employee listbox

EmployeesListBox.Items.Clear();

foreach (Employee emp in list)

EmployeesListBox.Items.Add(emp);

}

}

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 input form was cancelled

if (result == DialogResult.Cancel)

return; // end the methos 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 = null; // empty reference

if (frmInput.SalaryRadioButton.Checked)

{

double salary = double.Parse(frmInput.SalaryTextBox.Text);

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

}

else if (frmInput.HourlyRadioButton.Checked)

{

double hourlyRate = double.Parse(frmInput.HourlyRateTextBox.Text);

double hoursWorked = double.Parse(frmInput.HoursWorkedTextBox.Text);

emp = new Hourly(fName, lName, ssn, hireDate, ben, hourlyRate, hoursWorked);

}

else

{

MessageBox.Show("Error. Please select an employee type.");

}

// add the Employee object to the Employee listbox

EmployeesListBox.Items.Add(emp);

// write all date to the file

WriteEmpToFile();

}

}

private void WriteEmpToFile()

{

// convert the ListBox items to a generic list

List<Employee> emplist = new List<Employee>();

foreach (Employee emp in EmployeesListBox.Items)

{

emplist.Add(emp);

}

// open a pipe to the file and create a translator

FileStream fs = new FileStream(FILENAME, FileMode.Create);

BinaryFormatter formatter = new BinaryFormatter();

// write the generic list to the file

formatter.Serialize(fs, emplist);

// close the pipe

fs.Close();

}

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

{

// remove the selected itme from the Employee listbox

int itemNumber = EmployeesListBox.SelectedIndex;

if (itemNumber > -1)

{

EmployeesListBox.Items.RemoveAt(itemNumber);

WriteEmpToFile();

}

else

{

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

}

}

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.ToShortDateString();

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

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

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

// check to see if emp is a Salary or Hourly object

if (emp is Salary)

{

frmUpdate.HourlyRateLabel.Visible = false;

frmUpdate.HourlyRateTextBox.Visible = false;

frmUpdate.HoursWorkedLabel.Visible = false;

frmUpdate.SalaryLabel.Visible = true;

frmUpdate.SalaryTextBox.Visible = true;

// mark the Salary radiobutton as checked

frmUpdate.SalaryRadioButton.Checked = true;

// convert the Employee reference to a Salary object

Salary sal = (Salary)emp;

// show the Salary information

frmUpdate.SalaryTextBox.Text = sal.AnnualSalary.ToString("F2");

}

else if (emp is Hourly)

{

if (emp is Hourly)

{

frmUpdate.HourlyRateLabel.Visible = true;

frmUpdate.HourlyRateTextBox.Visible = true;

frmUpdate.HoursWorkedLabel.Visible = true;

frmUpdate.SalaryLabel.Visible = false;

frmUpdate.SalaryTextBox.Visible = false;

// mark the Hourly radiobutton as checked

frmUpdate.HourlyRadioButton.Checked = true;

// convert the Employee reference to a Hourly object

Hourly hrly = (Hourly)emp;

// show the Hourly information

frmUpdate.HourlyRateTextBox.Text = hrly.HourlyRate.ToString("F2");

frmUpdate.HoursWorkedTextBox.Text = hrly.HoursWorked.ToString("F2");

}

else

{

MessageBox.Show("Error. Invalid employee type found.");

return; // end the method

}

DialogResult result = frmUpdate.ShowDialog();

// if cancelled, stop the mehtod

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 update Information

Employee newEmp = null;

string fName = frmUpdate.FirstNameTextBox.Text;

string lName = frmUpdate.LastNameTextBox.Text;

string ssn = frmUpdate.SSNTextBox.Text;

DateTime hireDate = DateTime.Parse(frmUpdate.HireDateTextBox.Text);

string healthInsurance = frmUpdate.HealthInsuranceTextBox.Text;

int lifeInsurance = int.Parse(frmUpdate.LifeInsuranceTextBox.Text);

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

Benefits ben = new Benefits(healthInsurance, lifeInsurance, vacation);

if (frmUpdate.SalaryRadioButton.Checked)

{

double salary = double.Parse(frmUpdate.SalaryTextBox.Text);

newEmp = new Salary(fName, lName, ssn, hireDate, ben, salary);

}

else if (frmUpdate.HourlyRadioButton.Checked)

{

double hourlyRate = double.Parse(frmUpdate.HoursWorkedTextBox.Text);

double hoursWorked = double.Parse(frmUpdate.HoursWorkedTextBox.Text);

newEmp = new Hourly(fName, lName, ssn, ben, hourlyRate, hoursWorked);

}

else

{

MessageBox.Show("Error. Invalid employee type.");

return; // end the method

}

// add the new employee to the listbox

EmployeesListBox.Items.Add(newEmp);

}

}

}

}