**Cabrera, Jen Jade B.**

**BSCS – 2nd Year**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WeeklySalaryCalculator {

class EmployeeClass {

public string FirstName { get; set; }

public string MiddleName { get; set; }

public string LastName { get; set; }

public int HoursWorked { get; set; }

public double RatePerHour { get; set; }

public double Deduction { get; set; }

public string Position { get; set; }

private double \_allowance;

private double \_grossPay;

private double \_netPay;

public static Dictionary<string, int> PositionAllowanceMap =

new Dictionary<string, int> {

{"Regular", 1000}, { "Probationary", 500}, {"Part-time", 1000}

};

public double ComputeAllowance() {

\_allowance = PositionAllowanceMap[Position];

return \_allowance;

}

public double ComputeGrossPay() {

\_grossPay = (RatePerHour \* HoursWorked) + \_allowance;

return \_grossPay;

}

public double ComputeNetPay() {

\_netPay = ComputeGrossPay() - Deduction;

return \_netPay;

}

}

}

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 WeeklySalaryCalculator {

public partial class WeeklySalaryCalculatorForm : Form {

EmployeeClass employee = new EmployeeClass();

private double \_allowance;

private double \_grossPay;

private double \_netPay;

public WeeklySalaryCalculatorForm() {

InitializeComponent();

}

private void WeeklySalaryCalculatorForm\_Load(object sender, EventArgs e) {

foreach (KeyValuePair<string, int> dictionary in EmployeeClass.PositionAllowanceMap) {

cmbPosition.Items.Add(dictionary.Key);

}

}

private void ComputeOutputFields() {

\_allowance = employee.ComputeAllowance();

\_grossPay = employee.ComputeGrossPay();

\_netPay = employee.ComputeNetPay();

}

private void FetchInputInformation() {

employee.FirstName = txtFirstName.Text;

employee.MiddleName = txtMiddleName.Text;

employee.LastName = txtLastName.Text;

employee.HoursWorked = int.Parse(txtHoursWorked.Text);

employee.RatePerHour = Double.Parse(txtRatePerHour.Text);

employee.Deduction = Double.Parse(txtDeduction.Text);

employee.Position = cmbPosition.Text;

}

private void SetOutputFields() {

ComputeOutputFields();

txtAllowance.Text = \_allowance.ToString();

txtGrossPay.Text = \_grossPay.ToString();

txtNetPay.Text = \_netPay.ToString();

}

private string GetAllEmployeeInformation() {

FetchInputInformation();

ComputeOutputFields();

string employeeFullName = $"{employee.FirstName} {employee.MiddleName} {employee.LastName}";

string employeeInformation = String.Concat(

"Employee name: ", employeeFullName, "\n",

"Position: ", employee.Position, "\n",

"Hours worked: ", employee.HoursWorked, "\n",

"Rate per hour: ", employee.RatePerHour, "\n",

"Allowance: ", \_allowance.ToString(), "\n",

"Deduction: ", employee.Deduction, "\n",

"Gross pay: ", \_grossPay.ToString(), "\n",

"Net pay: ", \_netPay.ToString(), "\n"

);

return employeeInformation;

}

private void btnCompute\_Click(object sender, EventArgs e) {

FetchInputInformation();

SetOutputFields();

}

private void btnPreview\_Click(object sender, EventArgs e) {

FetchInputInformation();

MessageBox.Show(GetAllEmployeeInformation());

}

private void btnClose\_Click(object sender, EventArgs e) {

this.Close();

}

}

}