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 SalaryCalculation

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int n;

double[,] A;

List<Woker> wokers = new List<Woker>(); // Список рабочих

class Woker // Класс "Рабочий"

{

public string type { get; set; } // Тип работника - "бригадир", "основной работник", "вспомогательный работник"

public int number { get; set; } // Табельный номер

public double salary { get; set; } // Зарплата

public double all\_salary { get; set; } // Оклад

public double prepay { get; set; } // Аванс

public double main\_tax { get; set; } // Процент отчисления по налогам

public double other\_tax { get; set; } // Прочие отчисления (либо долг перед предприятием)

// Определим конструктор без параметров

public Woker()

{

type = "";

number = 0;

all\_salary = 0.0;

salary = 0.0;

prepay = 0.0;

main\_tax = 0.0;

other\_tax = 0.0;

}

// Определим конструктор с параметрами

public Woker(string w\_type, int w\_number, double w\_allsalary, double w\_salary, double w\_prepay, double w\_maintax, double w\_othertax)

{

type = w\_type;

number = w\_number;

all\_salary = w\_allsalary;

salary = w\_salary;

prepay = w\_prepay;

main\_tax = w\_maintax;

other\_tax = w\_othertax;

}

// Определим конструктор с параметрами

public Woker(string w\_type, int w\_number, double w\_allsalary, double w\_maintax)

{

type = w\_type;

number = w\_number;

all\_salary = w\_allsalary;

salary = 0.0;

prepay = 0.0;

main\_tax = w\_maintax;

other\_tax = 0.0;

}

// Функция для вычисления "чистой" зарплаты

public double GetNetSalary(double w\_allsalary, double w\_maintax)

{

return w\_allsalary \* (1 - w\_maintax);

}

// Функция для вычисления зарплаты

public void CalcSalary(double w\_allsalary, double w\_maintax, double w\_prepay)

{

salary = w\_allsalary \* (1 - w\_maintax) - w\_prepay;

}

}

// Функция, добавляющая n столбцов в элемент управления dgw.

private void AddColumns(int n, DataGridView dgw)

{

DataGridViewColumn column;

for (int i = 0; i < n; i++)

{

column = new DataGridViewTextBoxColumn();

column.DataPropertyName = "Column" + i.ToString();

column.Name = "Column" + i.ToString();

dgw.Columns.Add(column);

}

}

// Функция, добавляющая m строк в элемент управления dgw.

private void AddRows(int m, DataGridView dgw)

{

for (int i = 0; i < m; i++)

{

dgw.Rows.Add();

dgw.Rows[i].HeaderCell.Value = "row" + i.ToString();

}

}

private void button1\_Click(object sender, EventArgs e)

{

bool brigadier\_exist = false; // Есть ли бригадир

Random rnd = new Random(); // Создание объекта для генерации чисел

double ndfl = 0.13;

string[] woker\_types = new string[3] { "Бригадир", "Основной рабочий", "Вспомогательный рабочий" };

double[] salaries = new double[3] { 35400, 27540, 19390 };

bool success = Int32.TryParse(textBox1.Text, out n);

if (success && n > 1)

{

A = new double[n, 2]; // Массив для сортировки рабочих

for (int i = 0; i < n; i++)

{

Woker man = new Woker();

man.number = i + 1; // Табельный номер - по счётчику i

A[i, 0] = i;

if (!brigadier\_exist) // Если нет бригадира

{

int j = rnd.Next(0, 3);

man.type = woker\_types[j];

if (man.type == "Бригадир")

brigadier\_exist = true;

man.all\_salary = salaries[j];

A[i, 1] = man.all\_salary;

man.main\_tax = ndfl;

double percenrt = rnd.Next(300, 400);

double net\_salary = man.GetNetSalary(man.all\_salary, man.main\_tax);

man.prepay = net\_salary \* percenrt / 1000;

int resh = rnd.Next(0, 1);

if (resh == 0)

{

man.other\_tax = 0.0;

}

else

{

man.other\_tax = rnd.Next(10000, 2000000) / 100;

}

}

else // Если бригадир уже есть

{

int j = rnd.Next(1, 3);

man.type = woker\_types[j];

man.all\_salary = salaries[j];

A[i, 1] = man.all\_salary;

man.main\_tax = ndfl;

double percenrt = rnd.Next(300, 400);

double net\_salary = man.GetNetSalary(man.all\_salary, man.main\_tax);

man.prepay = net\_salary \* percenrt / 1000;

int resh = rnd.Next(0, 2);

if (resh == 0)

{

man.other\_tax = 0.0;

}

else

{

man.other\_tax = rnd.Next(10000, 2000000) / 100;

}

}

wokers.Add(man);

}

int k = 0;

k = dataGridView1.ColumnCount;

if (k != 0)

{

for (int i = 0; i < k; i++)

{

dataGridView1.Columns.RemoveAt(0);

}

}

AddColumns(8, dataGridView1);

AddRows(n, dataGridView1);

dataGridView1.Columns[0].Name = "Табельный номер";

dataGridView1.Columns[1].Name = "Должность";

dataGridView1.Columns[2].Name = "Оклад без вычета НДФЛ";

dataGridView1.Columns[3].Name = "С вычетом НДФЛ";

dataGridView1.Columns[4].Name = "Величина НДФЛ";

dataGridView1.Columns[5].Name = "Аванс";

dataGridView1.Columns[6].Name = "З/п";

dataGridView1.Columns[7].Name = "Долг";

for (int i = 0; i < n; i++)

{

dataGridView1.Rows[i].Cells[0].Value = wokers[i].number.ToString();

dataGridView1.Rows[i].Cells[1].Value = wokers[i].type;

string str = String.Format("{0:f2}", wokers[i].all\_salary);

dataGridView1.Rows[i].Cells[2].Value = str;

str = String.Format("{0:f2}", wokers[i].GetNetSalary(wokers[i].all\_salary, wokers[i].main\_tax));

dataGridView1.Rows[i].Cells[3].Value = str;

dataGridView1.Rows[i].Cells[4].Value = (wokers[i].main\_tax \* 100).ToString() + " %";

str = String.Format("{0:f2}", wokers[i].prepay);

dataGridView1.Rows[i].Cells[5].Value = str;

wokers[i].CalcSalary(wokers[i].all\_salary, wokers[i].main\_tax, wokers[i].prepay);

str = String.Format("{0:f2}", wokers[i].salary);

dataGridView1.Rows[i].Cells[6].Value = str;

if (wokers[i].other\_tax == 0.0)

{

dataGridView1.Rows[i].Cells[7].Value = "Долга нет!";

}

else

{

str = String.Format("{0:f2}", wokers[i].other\_tax);

dataGridView1.Rows[i].Cells[7].Value = str;

}

}

}

else

{

MessageBox.Show("Неверное количество рабочих!");

}

}

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

{

AddColumns(8, dataGridView1);

dataGridView1.Columns[0].Name = "Табельный номер";

dataGridView1.Columns[1].Name = "Должность";

dataGridView1.Columns[2].Name = "Оклад без вычета НДФЛ";

dataGridView1.Columns[3].Name = "С вычетом НДФЛ";

dataGridView1.Columns[4].Name = "Величина НДФЛ";

dataGridView1.Columns[5].Name = "Аванс";

dataGridView1.Columns[6].Name = "З/п";

dataGridView1.Columns[7].Name = "Долг";

}

private void button2\_Click(object sender, EventArgs e)

{

// Сортируем рабочих по возрастанию зп

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n - 1; j++)

{

if (A[j, 1] > A[j + 1, 1])

{

double x = A[j, 0];

double y = A[j, 1];

A[j, 0] = A[j + 1, 0];

A[j, 1] = A[j + 1, 1];

A[j + 1, 0] = x;

A[j + 1, 1] = y;

}

}

}

// Обновляем таблицу

for (int i = 0; i < n; i++)

{

int j = Convert.ToInt32(A[i, 0]);

dataGridView1.Rows[i].Cells[0].Value = wokers[j].number.ToString();

dataGridView1.Rows[i].Cells[1].Value = wokers[j].type;

string str = String.Format("{0:f2}", wokers[j].all\_salary);

dataGridView1.Rows[i].Cells[2].Value = str;

str = String.Format("{0:f2}", wokers[j].GetNetSalary(wokers[j].all\_salary, wokers[j].main\_tax));

dataGridView1.Rows[i].Cells[3].Value = str;

dataGridView1.Rows[i].Cells[4].Value = (wokers[j].main\_tax \* 100).ToString() + " %";

str = String.Format("{0:f2}", wokers[j].prepay);

dataGridView1.Rows[i].Cells[5].Value = str;

wokers[j].CalcSalary(wokers[j].all\_salary, wokers[j].main\_tax, wokers[j].prepay);

str = String.Format("{0:f2}", wokers[j].salary);

dataGridView1.Rows[i].Cells[6].Value = str;

if (wokers[j].other\_tax == 0.0)

{

dataGridView1.Rows[j].Cells[7].Value = "Долга нет!";

}

else

{

str = String.Format("{0:f2}", wokers[j].other\_tax);

dataGridView1.Rows[i].Cells[7].Value = str;

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

double salary\_sum = 0.0;

double prepay\_sum = 0.0;

for (int i = 0; i < n; i++)

{

salary\_sum += wokers[i].all\_salary;

prepay\_sum += wokers[i].prepay;

}

salary\_sum \*= 0.87;

prepay\_sum /= salary\_sum;

prepay\_sum \*= 100;

string result = String.Format("{0:f2}", prepay\_sum);

textBox2.Text = result + " %";

}

}

}