

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.01

Выполнил: Ковязин А.М.

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2023

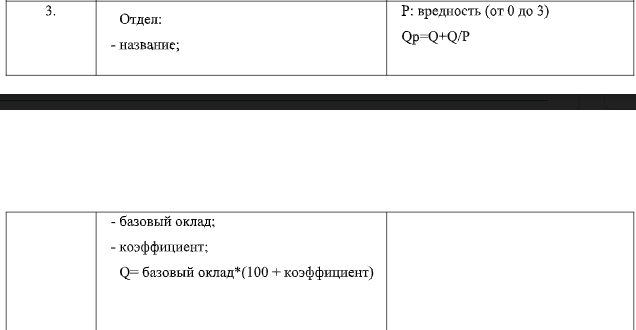
# 1. Задание 3 - Индивидуальное задание (Вариант 3)

## Описание задачи

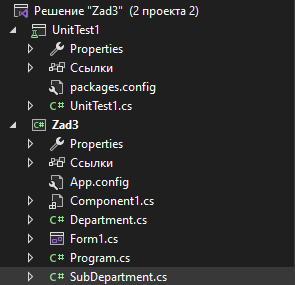
Создать проект для демонстрации работы: ввод-вывод информации об

объектах базового класса и класса потомка.

Разработать модульные тесты к реализованным методам классов



## Структура проекта



## Код разработанных функций

Класс SubDepartmentt

public double Quality (double Harmfulness)

{

double sum = base.Quality() + base.Quality() / Harmfulness;

return sum;

}

public double MinSalaryAll (double Harmfulness)

{

double sum = base.MinSalaryAll()/Harmfulness;

return sum;

}

public static void AddDepartment (List<SubDepartment> list , SubDepartment dept)

{

list.Add(dept);

}

public static void RemoveDepartment (List<SubDepartment> subdeptList, string name)

{

subdeptList.RemoveAll(e => e.Name == name);

}

Класс Department

public double Quality ()

{

double sum = BaseSalary \* (100 + Coefficient);

return sum;

}

public double MinSalaryAll ()

{

double sum = BaseSalary \* NumOfEmployees \* Coefficient;

return sum;

}

public string PrintInfo ()

{

return $"Department Name: {Name}\nBase Salary: {BaseSalary}\nCoefficient: {Coefficient}\nDepartment Head: {Head}\nNumber of Employees: {NumOfEmployees}\nNumber of Vacancies: {NumOfVacancies}\nLocation: {Location}\nPhone: {Phone}";

}

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

{

string name, head, adress, phone;

double baseSalary, koef, harmfulness;

int countEm, countVac;

name = NameBox.Text;

head = HeadBox.Text;

adress = adressBox.Text;

phone = phoneBox.Text;

baseSalary = (double)baseNum.Value;

koef =(double)koefNum.Value;

countEm = (int)countEmNum.Value;

countVac = (int)countVacNum.Value;

harmfulness = (double) HarmfulnessNum.Value;

if(name=="" || head=="" || adress=="" || phone=="" || baseSalary==0 || koef==0 || countEm==0|| countVac == 0)

{

MessageBox.Show("Не все поля заполнены", "Ошибка",MessageBoxButtons.OK,MessageBoxIcon.Error);

return;

}

if(checkBox1.Checked && harmfulness == 0)

{

MessageBox.Show("Не все поля заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (!CheckDept(name))

return;

if (checkBox1.Checked)

{

var subDeptObj1 = new SubDepartment(name, baseSalary, koef, head, countEm, countVac, adress, phone, harmfulness);

SubDepartment.AddDepartment(subDeptList, subDeptObj1);

} else

{

var deptObj1 = new Department(name, baseSalary, koef, head, countEm, countVac);

deptObj1.Location = adress;

deptObj1.Phone = phone;

Department.AddDepartment(deptList, deptObj1);

}

MessageBox.Show("Отдел сохранен", "Инфо", MessageBoxButtons.OK, MessageBoxIcon.Information);

NameBox.Text="";

HeadBox.Text="";

adressBox.Text="";

phoneBox.Text = "";

baseNum.Value=0;

koefNum.Value=0;

countEmNum.Value=0;

countVacNum.Value=0;

HarmfulnessNum.Value = 0;

ComboBoxLoad();

}

public bool CheckDept (string name)

{

bool flag = true;

foreach (var dp in deptList)

{

if (dp.Name == name)

{

MessageBox.Show("Отдел с таким названием уже есть", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

flag = false;

}

}

foreach (var dp in subDeptList)

{

if (dp.Name == name)

{

MessageBox.Show("Отдел с таким названием уже есть", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

flag = false;

}

}

return flag;

}

public void ComboBoxLoad ()

{

comboBox1.Items.Clear();

foreach(var dp in deptList)

{

comboBox1.Items.Add(dp.Name);

}

foreach (var dp in subDeptList)

{

comboBox1.Items.Add(dp.Name);

}

}

private void comboBox1\_SelectedIndexChanged (object sender, EventArgs e)

{

if (comboBox1.Text != "")

{

listBox1.Items.Clear();

foreach(var dp in deptList)

{

if (dp.Name == comboBox1.Text)

{

ListBoxAdd(dp);

return;

}

}

foreach (var dp in subDeptList)

{

if (dp.Name == comboBox1.Text)

{

ListBoxAdd(dp);

return;

}

}

}

}

public void ListBoxAdd (Department dp)

{

listBox1.Items.Add($"Name department: {dp.Name}");

listBox1.Items.Add($"Base Salary: {dp.BaseSalary}");

listBox1.Items.Add($"Coefficient: {dp.Coefficient}");

listBox1.Items.Add($"Department Head: {dp.Head}");

listBox1.Items.Add($"Number of Employees: {dp.NumOfEmployees}");

listBox1.Items.Add($"Number of Vacancies: {dp.NumOfVacancies}");

listBox1.Items.Add($"Location: {dp.Location}");

listBox1.Items.Add($"Phone: {dp.Phone}");

listBox1.Items.Add($"Quality: {dp.Quality()}");

listBox1.Items.Add($"MinSalaryAll: {dp.MinSalaryAll()}");

}

public void ListBoxAdd (SubDepartment dp)

{

listBox1.Items.Add($"Name department: {dp.Name}");

listBox1.Items.Add($"Base Salary: {dp.BaseSalary}");

listBox1.Items.Add($"Coefficient: {dp.Coefficient}");

listBox1.Items.Add($"Department Head: {dp.Head}");

listBox1.Items.Add($"Number of Employees: {dp.NumOfEmployees}");

listBox1.Items.Add($"Number of Vacancies: {dp.NumOfVacancies}");

listBox1.Items.Add($"Location: {dp.Location}");

listBox1.Items.Add($"Phone: {dp.Phone}");

listBox1.Items.Add($"QualityP: {dp.Quality(dp.Harmfulness)}");

listBox1.Items.Add($"Harmfulness: {dp.Harmfulness}");

listBox1.Items.Add($"MinSalaryAllP: {dp.MinSalaryAll(dp.Harmfulness)}");

}

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

{

if (comboBox1.Text == "")

{

MessageBox.Show("Выберите отдел который хотите удалить", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

Department.RemoveDepartment(deptList,comboBox1.Text);

SubDepartment.RemoveDepartment(subDeptList, comboBox1.Text);

ComboBoxLoad();

listBox1.Items.Clear();

comboBox1.Text = "";

}

private void checkBox1\_CheckedChanged (object sender, EventArgs e)

{

if (checkBox1.Checked)

{

label18.Visible = true;

HarmfulnessNum.Visible = true;

} else

{

label18.Visible = false;

HarmfulnessNum.Visible = false;

}

}

private void NameBox\_KeyPress (object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar) == true)

return;

e.Handled = true;

}

private void phoneBox\_KeyPress (object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar) == true)

return;

e.Handled = true;

}

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

{

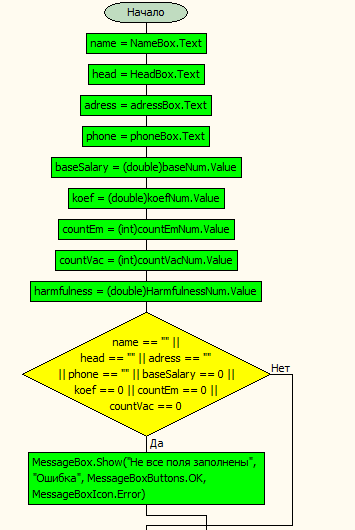
Close();

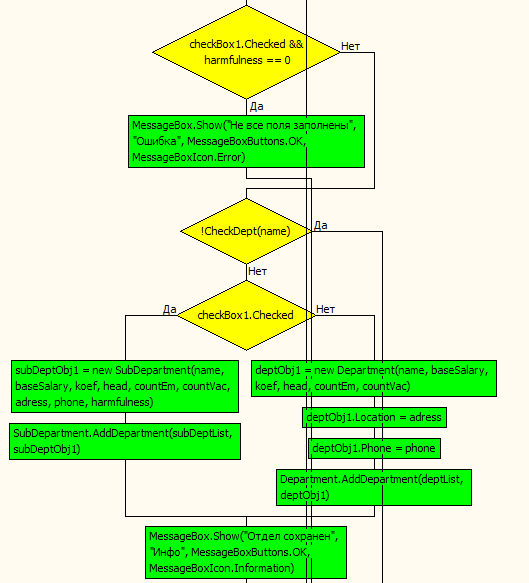
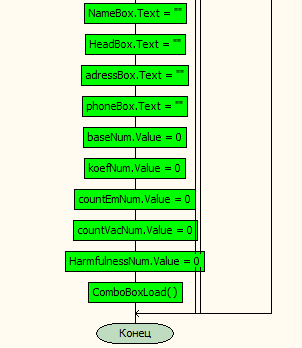
## }

## 1.4 Алгоритм решения

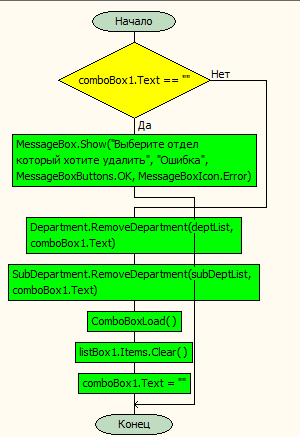
## Блок-схемы:

Метод addBut\_Click

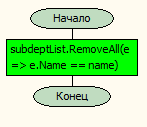


Метод deleteBut\_Click



Метод RemoveDepartment



Метод

## 1.5 Используемые библиотеки

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;

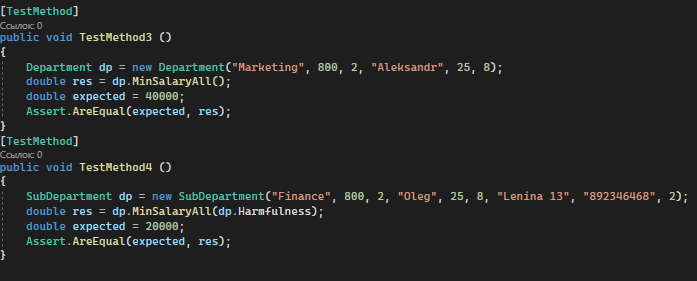
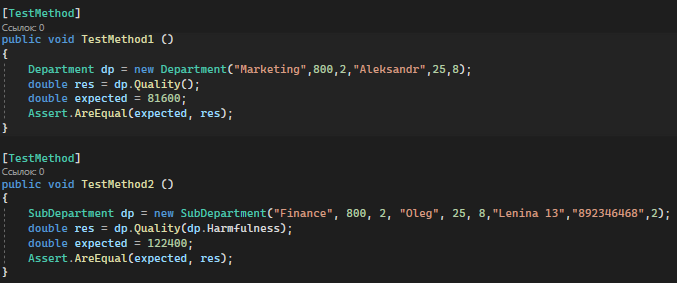
using static System.Windows.Forms.MonthCalendar;

using System.Xml.Linq;

using System.Text.RegularExpressions;

using System.Security.Cryptography;

## 1.6 Тестовые случаи



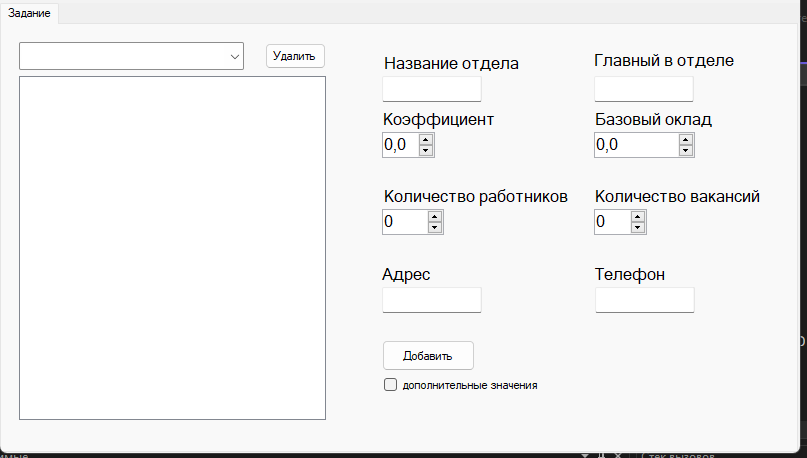
## 1.7 Используемые инструменты

Язык программирования - C#

Среда разработки - Visual Studio

Фреймворк - .net

## 1.8 Приложение (pr screen экранов)

****