Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Задание 3 – Индивидуальное задание**»

Выполнил: Белозёров Клим Андреевич

Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

Используя методы Linq и коллекции (использовать не менее 2-х видов

коллекций)

Описать базовый класс с указанными в задании полями и методами:

● Конструктор;

● Добавить 2 своих поля;

● Функция, которая определяет качество объекта – Q по заданной

формуле;

● Вывод информации об объекте.

● Реализуйте и проверьте методы для добавления и удаления

объекта базового класса (сделать 3 перегрузки!);

Построить класс потомок, который содержит:

● Дополнительное поле Р;

● Добавить еще 3 свойства

● Функция, которая определяет «качество» объекта класса потомка

– Qp, которая перекрывает функцию качества базового класса,

выполняя вычисления по новой формуле.

 Добавить свою функцию для класса потомка

Создать проект для демонстрации работы: ввод-вывод информации об

объектах базового класса и класса потомка.

Разработать модульные тесты к реализованным методам классов

**Входные и выходные данные**

**Double shirina,dlina,massa,Q,p**

**Листинг программы (если есть)**

namespace ZD3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Требуемый метод для поддержки конструктора

}

Qp road = new Qp(); // Экземпляр класса Qp

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

{

bool proverka = true; //переменная типа bool

if (textBox2.Text != "" && textBox2.Text != " ") // условие на проверку пустоты в текстбоксе

{

foreach (var con in textBox2.Text)

{

if (char.IsWhiteSpace(con) || char.IsDigit(con))//проверка на пробелы и цифры

{

proverka = false; // Переменной proverka присваивается значение false

break;// остановка цикла foreach

}

}

}

else

MessageBox.Show("Пустое поле Страны");// Сообщение если поле Страны пустое

for (int i = 0; i < road.List.Count; i++)

{

if (road.List[i].Name.ToUpper() == textBox1.Text.ToUpper())//проверка на повтор названия дороги

{

proverka = false;// Переменной proverka присваивается значение false

break; // остановка цикла for

}

}

if (proverka&&textBox1.Text!=""&&textBox1.Text!=" ")// условие на проверку пустоты в текстбоксах

{

road.Addobject(num1, num2, num3, num4, textBox2, textBox1);//Добавление

MessageBox.Show("Добавлено");// Выводит сообщение "Добавлено"

road.Listrestart(listBox1); // Обновляет listBox

}

else

{

MessageBox.Show("Неверный формат или такое имя дороги уже есть");

// Выводит сообщение "Неверный формат или такое имя дороги уже есть"

}

}

private void button2\_Click(object sender, EventArgs e) //Удаление по индексу

{

int d = listBox1.SelectedIndex;// Переменная d присваивает индекс выбраной строки в listBox

if (d!=-1)

{

road.Listrestart(d,listBox1);//Метод удаляет элемент из listBox по индексу

}else

{

MessageBox.Show("Выберите строчку");

//Выводит сообщение "Выберите строчку"

}

}

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

{

//Метод вызываемый при загрузке Формы

}

private void button3\_Click(object sender, EventArgs e) // Удаление по названию дороги

{

bool pr = false;// Переменная pr присваивает значение false (проверка)

int d = -1; // Переменная d присваивает значение -1 (индекс)

int check = 0; // Переменная check присваивает 0 (проверка)

if (textBox3.Text != "" && textBox3.Text != " ") // Проверка на пустое поле

{

var search = from t in road.List //

select t; ////Запрос Linq

foreach (var sr in search) // Цикл foreach для проверки на существование названия дороги

{

d++; // Увеличивается индекс на 1

if (sr.Name.ToUpper() == textBox3.Text.ToUpper()) // Проверка на то что в листе есть название дороги которое нужно удалить

{

check = 1; // Переменная check присваивает значение 1

break; // Остановка цикла

}

}

if (check==1)

{

road.Listrestart(d, listBox1);// Удаляет из листа по индексу d

pr = true; // Переменная pr присваивает значение true

}

if (pr)

{

MessageBox.Show("Удалено");

// Выводит сообщение "Удалено"

}else

{

MessageBox.Show("Названия дороги не найдено");

//Выводит сообщение "Название дороги не найдено"

}

}else

{

MessageBox.Show("Пустое поле");

//Выводит сообщение "Пустое поле"

}

}

}

}

namespace ZD3

{

class Q // Класс Q

{

string name; // Доп поле 1

string country; // Доп поле 2

double roadwidth; // Ширина дороги

double dlina; // Длина

double massa; // масса

double q; //Q

public Q()// Конструктор

{

}

public double Qmath(double w, double d, double m) // Функция Q

{

double result = w \* d \* (m / 1000); // Формула

q = result;

return result;

}

public double Roadwidth{ //Свойство

get { return roadwidth; } set { roadwidth = value; }

}

public double Dlina { //Свойство

get { return dlina; } set { dlina = value; }

}

public double Massa // Свойство

{

get { return massa; } set { massa = value; }

}

public double QQ //Свойство

{

get { return q; }

set { q = value; }

}

public string Name //Свойство

{

get { return name; }

set { name = value; }

}

public string Country //Свойство

{

get { return country; }

set { country = value; }

}

}

}

namespace ZD3

{

class Qp:Q// Класс Qp наследник от класса Q

{

double p; // Поле коэфициент

double qp;

List<Qp> list = new List<Qp>();// Лист

public List<Qp> List// Свойство

{

get { return list; }

}

public void Qmath(Qp ex)// Метод для расчета Qp

{

double result=ex.Qmath(ex.Roadwidth,ex.Dlina,ex.Massa); // В переменную result возвращает расчет из метода Qmath

ex.QQ = result; // в поле q в базовом классе присваивает значение result

if (ex.p >= 5 && ex.p <= 8) ///////

{ //

ex.qp = result \* 1.1; //

//

//

//

}else //

if (ex.p == 3 || ex.p == 4 || ex.p == 9 || ex.p == 10) /////////// Если P=3, 4, 9, 10 то Qp=Q\*1,6 иначе Qp = 1,9 \* Q

{ //

//

ex.qp = result \* 1.6; //

//

//

//

} //

else ///////

{

ex.qp = result \* 1.9;

}

}

public void Addobject(NumericUpDown num1, NumericUpDown num2, NumericUpDown num3, NumericUpDown num4,TextBox txt,TextBox txt2)

{

//Метод добавления

Qp road1 = new Qp();// Создает экземпляр Qp

road1.Roadwidth = Convert.ToDouble(num1.Value);// в поле ширина присваивается num1

road1.Dlina = Convert.ToDouble(num2.Value);// в поле dlina присваивается num2

road1.Massa = Convert.ToDouble(num3.Value);// в поле massa присваивается num3

road1.p = Convert.ToDouble(num4.Value);// в поле p присваивается num4

road1.Country = txt.Text;// в поле country присваивается txt

road1.Name = txt2.Text;// в поле name присваивается txt2

road1.Qmath(road1); // Вызываеться метод Qmath

list.Add(road1); // Экземпляр добавляется в лист

}

public void Listrestart(ListBox list1) // обновляет listBox

{

list1.Items.Clear(); //Очищает лист

for (int i = 0; i < list.Count; i++)

{

list1.Items.Add($"Страна {list[i].Country} Название дороги {list[i].Name} Ширина {list[i].Roadwidth} Длина {list[i].Dlina} Масса {list[i].Massa} Коэф прочности {list[i].p} Q = {list[i].QQ} Qp = {list[i].qp}");

//добавляет в listbox элемент

}

}

public void Listrestart(int d, ListBox list1)//перегрузка метода Listrestart которая удаляет из listBox элемент

{

list1.Items.RemoveAt(d);// удаляет элемент из listBox по индексу

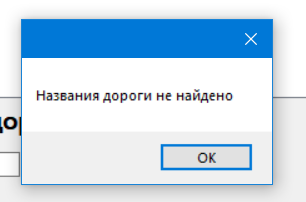
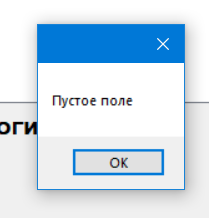
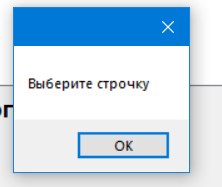
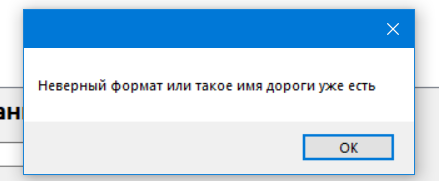
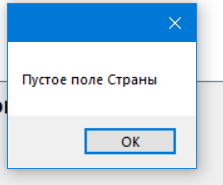
list.RemoveAt(d); // удаляет из List<Qp> list экзмепляр по индексу

}

}

}

**Тестовые ситуации**

****

**Вывод**

