**Отчёт по практике 10.**

**Студент: Гуртякова А.П.**

**Задача:**Выполнить практические задания на тему перегрузок операций

**Цель задания:**Выполнить эмпирическую практику с практическим заданием.

**Задание:**

1. Описать деревню с характеристиками населения и площади.

Деревни могут складывать с друг другом - превращаться в новую объединенную деревню.

Деревню можно умножать на целое число (умножаются все характеристики)

Деревню можно сравнивать между собой (критерий сравнения - население)

Должен быть переопределен метод ToString(), выводящий население и площадь деревни.

1. Создать класс, представляющий массив.

Объекты такого класса можно складывать, вычитать, умножать, делить, брать остаток от деления.

В результате работы этих операций должен возвращаться новый такой массив.

Если массивы разной длины, то возвращаться должен новый массив с длинной большего из них.

(Над лишними элементами операции не выполняются - они просто помещаются в новый массив)

Написать метод Print, выводящий на консоль все элементы массива.

**Код программы 1:**

using System;

using System.Collections.Generic;

public class Village : IComparable<Village>

{

private int population;

private int area;

public Village(int population, int area)

{

this.population = population;

this.area = area;

}

public int Population

{

get { return population; }

}

public int Area

{

get { return area; }

}

public void Multiply(int factor)

{

population \*= factor;

area \*= factor;

}

public Village Stack(Village other)

{

int stackedPopulation = population + other.population;

int stackedArea = area + other.area;

return new Village(stackedPopulation, stackedArea);

}

public int CompareTo(Village other)

{

return population.CompareTo(other.population);

}

public override string ToString()

{

return $"Население: {population}, Площадь: {area}";

}

public static void Main(string[] args)

{

Village komarovo = new Village(100, 50);

Village byazevo = new Village(200, 75);

Village titarovka = new Village(150, 60);

Console.WriteLine();

// Multiplication

komarovo.Multiply(2);

Console.WriteLine("Результат увеличения деревни Комарово в 2 раза: " + komarovo);

// Stacking

Village stackedVillage = komarovo.Stack(byazevo);

Console.WriteLine("Результат сложения увеличенной в 2 раза деревни Комарово и деревни Бязево: " + stackedVillage);

// Comparison

List<Village> villages = new List<Village> {komarovo, byazevo, titarovka };

villages.Sort();

Console.WriteLine("Результат сортировки деревень по количеству населения и площади:");

foreach (Village village in villages)

{

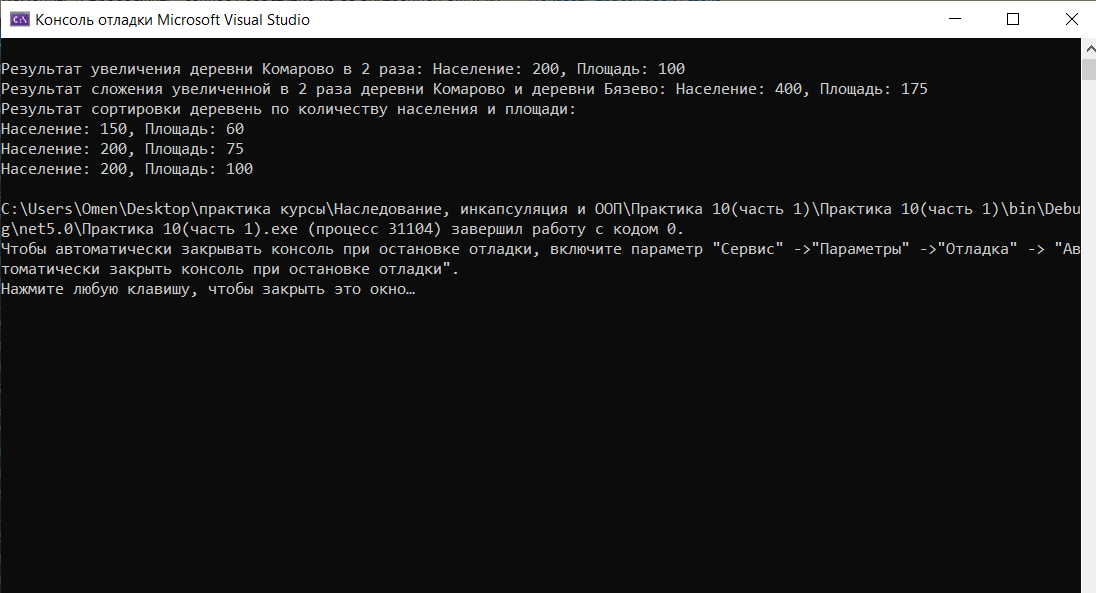
Console.WriteLine(village);

}

}

}

**Скриншот работы программы 1:**



**Код программы 2:**

using System;

public class MyArray

{

private int[] elements;

public MyArray(int[] elements)

{

this.elements = elements;

}

public MyArray Add(MyArray other)

{

int maxLength = Math.Max(elements.Length, other.elements.Length);

int[] result = new int[maxLength];

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

{

int a = i < elements.Length ? elements[i] : 0;

int b = i < other.elements.Length ? other.elements[i] : 0;

result[i] = a + b;

}

return new MyArray(result);

}

public MyArray Subtract(MyArray other)

{

int maxLength = Math.Max(elements.Length, other.elements.Length);

int[] result = new int[maxLength];

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

{

int a = i < elements.Length ? elements[i] : 0;

int b = i < other.elements.Length ? other.elements[i] : 0;

result[i] = a - b;

}

return new MyArray(result);

}

public MyArray Multiply(MyArray other)

{

int maxLength = Math.Max(elements.Length, other.elements.Length);

int[] result = new int[maxLength];

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

{

int a = i < elements.Length ? elements[i] : 0;

int b = i < other.elements.Length ? other.elements[i] : 0;

result[i] = a \* b;

}

return new MyArray(result);

}

public MyArray Divide(MyArray other)

{

int maxLength = Math.Max(elements.Length, other.elements.Length);

int[] result = new int[maxLength];

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

{

int a = i < elements.Length ? elements[i] : 0;

int b = i < other.elements.Length ? other.elements[i] : 1; // Avoid division by zero

result[i] = a / b;

}

return new MyArray(result);

}

public MyArray Remainder(MyArray other)

{

int maxLength = Math.Max(elements.Length, other.elements.Length);

int[] result = new int[maxLength];

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

{

int a = i < elements.Length ? elements[i] : 0;

int b = i < other.elements.Length ? other.elements[i] : 1; // Avoid division by zero

result[i] = a % b;

}

return new MyArray(result);

}

public void Print()

{

Console.WriteLine(string.Join(", ", elements));

}

}

public class Program

{

public static void Main(string[] args)

{

MyArray array1 = new MyArray(new int[] { 1, 2, 3 });

MyArray array2 = new MyArray(new int[] { 4, 5 });

MyArray sum = array1.Add(array2);

sum.Print(); // Output: 5, 7, 3

MyArray difference = array1.Subtract(array2);

difference.Print(); // Output: -3, -3, 3

MyArray product = array1.Multiply(array2);

product.Print(); // Output: 4, 10, 0

MyArray quotient = array1.Divide(array2);

quotient.Print(); // Output: 0, 0, 3

MyArray remainder = array1.Remainder(array2);

remainder.Print(); // Output: 1, 2, 0

}

}

**Скриншот работы программы 2:**

