МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

# по дисциплине «Объектно-ориентированное проектирование и программирование»

на тему: «Операции и перегруженные методы класса»

Выполнил студент гр. ИТИ-21:  
Буякевич Д.О.  
 Принял преподаватель:  
 Чернявская А.Э.

Гомель 2022

Цель: научиться работать с перегрузкой методов класса. Закрепить знания работы с перегрузкой операторов.

Ход работы

На рисунке 1 и 2 показаны требования для выполнения лабораторной работы и вариант с заданиями.

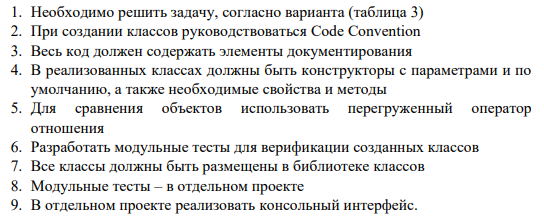


Рисунок 1 – Требования

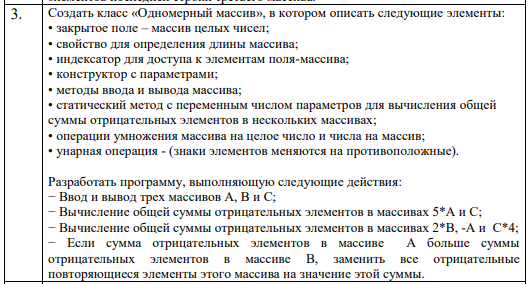


Рисунок 2 – Задачи

ЗАДАЧА 1

На рисунке 3 показано меню задачи 1.

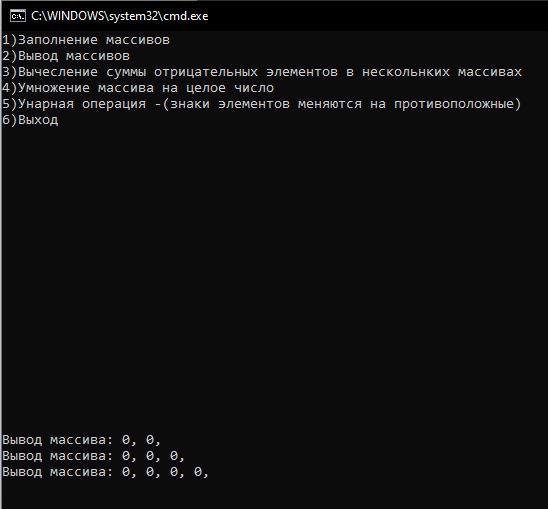
****

Рисунок 3 – Меню задачи 1

Рисунок 4 показывает ввод и вывод массива.

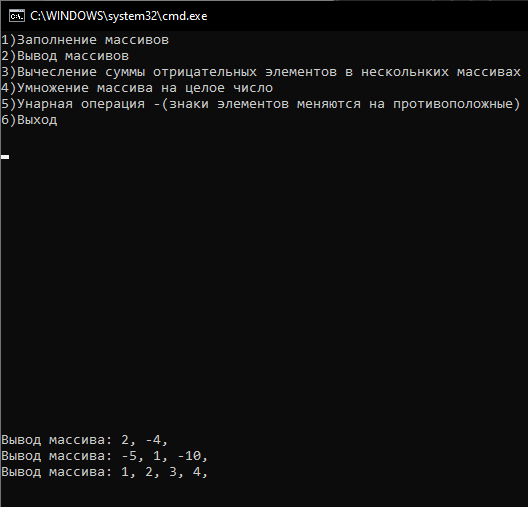


Рисунок 4 – Ввод и вывод массива

Рисунок 5 демонстрирует 3ий пункт меню.

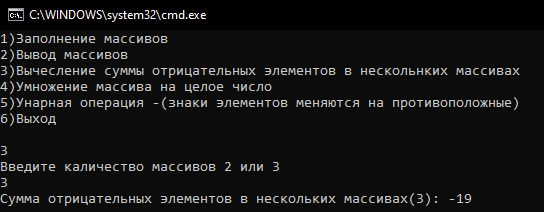


Рисунок 5 – Сумма отрицательных элементов

Рисунок 6 показывает умножение массива на целое число

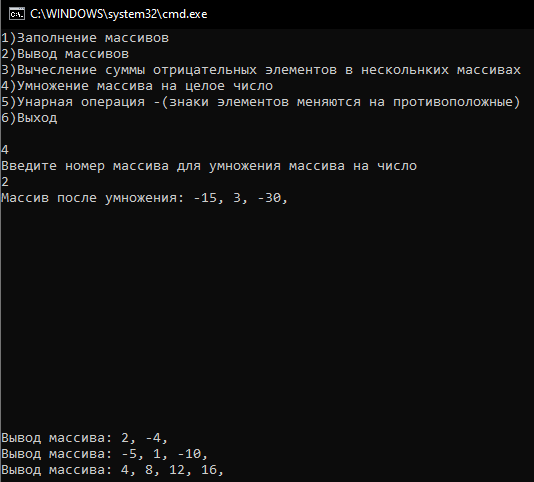


Рисунок 6 – Пункт 4

Рисунок 7 и 8 показывает унарную операцию минус.

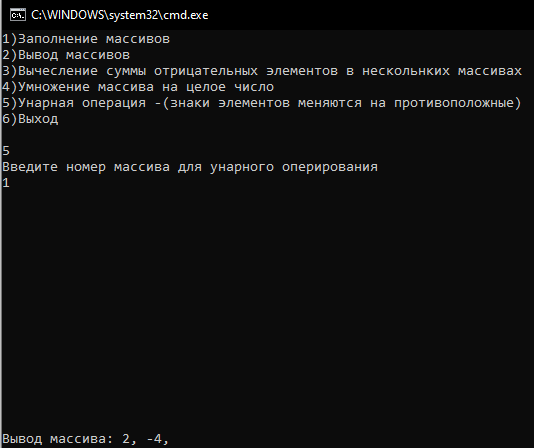


Рисунок 7 – Ввод пункта

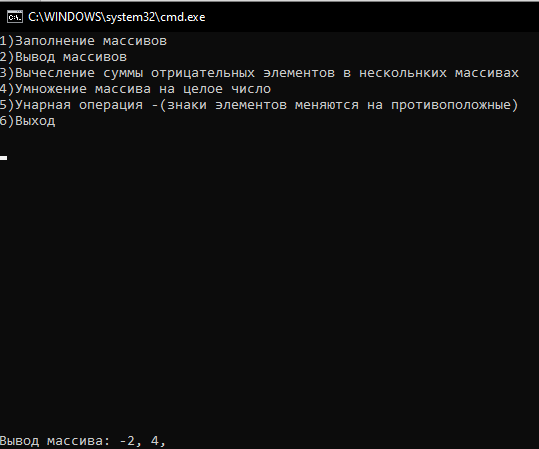


Рисунок 8 – Результат

ЗАДАЧА 2

На рисунке 9 меню задачи 2.

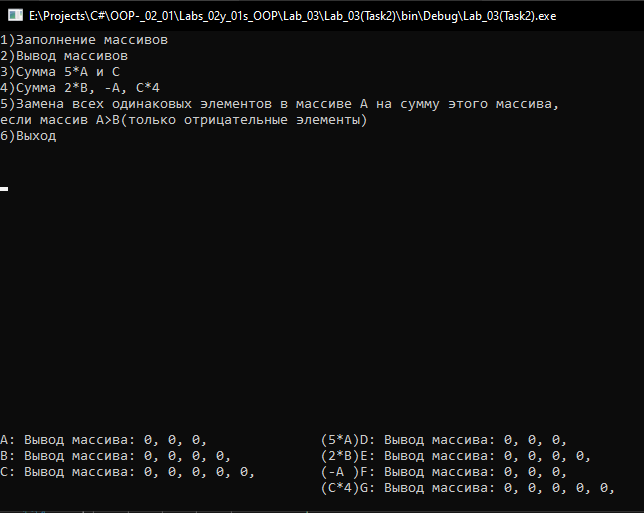


Рисунок 9 – Меню задачи 2

Массивы после заполнения и их вывод на рисунке 10.

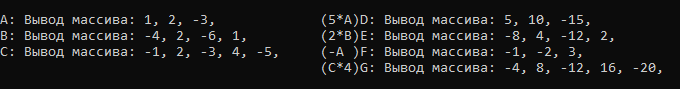


Рисунок 10 – Ввод и вывод массивов

На рисунках 11 и 12 показаны пункты 3 и 4 соответственно.

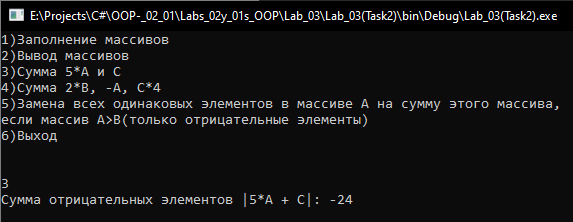


Рисунок 11 – Пункт 3

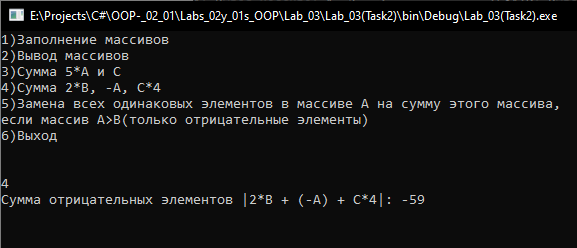


Рисунок 12 – Пункт 4

На рисунках 13 и 14 показан процесс пункта 5 при соблюдении условия.

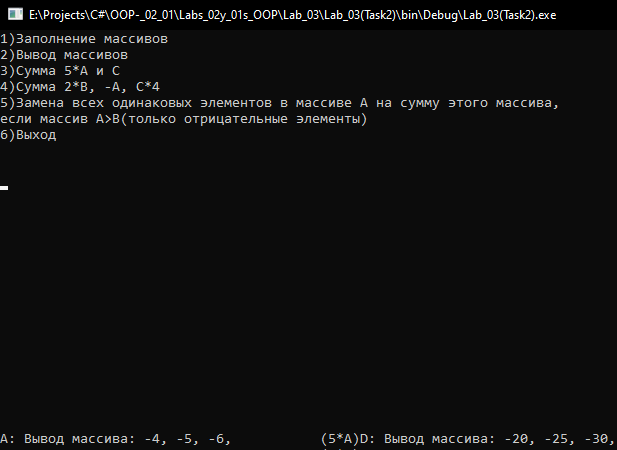
****

Рисунок 13 – До проверки

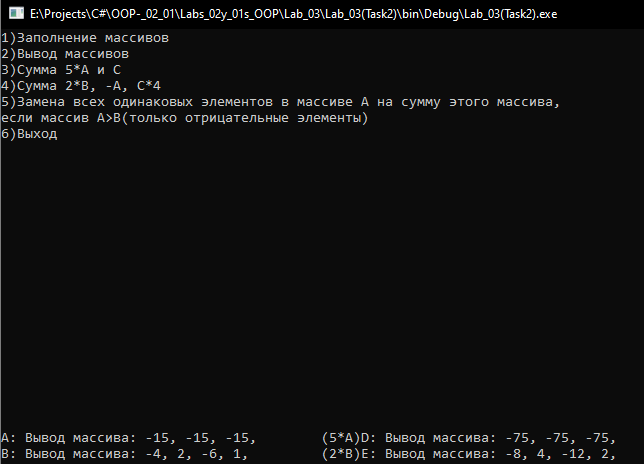
****

Рисунок 14 – После проверки

Вывод: познакомились с перегрузкой методов и закрепили знания о перегрузке операторов.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

**ЗАДАЧА 1**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_03

{

internal class Program

{

static void Main(string[] args)

{

// СДЕЛАТЬ МЕНЮ ЧЕРЕЗ SWITCH CASE С ПУНКТАМИ ЛАБ РАБОТЫ

// OneDimensionalArray - Одномерный массив

OneDimensionalArray array1 = new OneDimensionalArray(2, 2);

OneDimensionalArray array2 = new OneDimensionalArray(3, 3);

OneDimensionalArray array3 = new OneDimensionalArray(4, 4);

int choose = 0;

while (choose != 6)

{

Console.WriteLine("1)Заполнение массивов");

Console.WriteLine("2)Вывод массивов");

Console.WriteLine("3)Вычесление суммы отрицательных элементов в нескольнких массивах");

Console.WriteLine("4)Умножение массива на целое число");

Console.WriteLine("5)Унарная операция -(знаки элементов меняются на противоположные)");

Console.WriteLine("6)Выход");

Console.SetCursorPosition(0, 25);

array1.OutputArray();

array2.OutputArray();

array3.OutputArray();

Console.SetCursorPosition(0, 7);

choose = Convert.ToInt32(Console.ReadLine());

switch (choose)

{

case 1:

Console.WriteLine("Введите массив для заполнения от 1 до 3");

int chooseArray1 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray1)

{

case 1:

array1.InputArray();

break;

case 2:

array2.InputArray();

break;

case 3:

array3.InputArray();

break;

default:

Console.WriteLine("Такого массива нет!");

break;

}

break;

case 2:

Console.WriteLine("Введите массив для вывода от 1 до 3");

int chooseArray2 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray2)

{

case 1:

array1.OutputArray();

Console.ReadKey();

break;

case 2:

array2.OutputArray();

Console.ReadKey();

break;

case 3:

array3.OutputArray();

Console.ReadKey();

break;

default:

Console.WriteLine("Такого массива нет!");

Console.ReadKey();

break;

}

break;

case 3:

Console.WriteLine("Введите каличество массивов 2 или 3");

int chooseArray3 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray3)

{

case 2:

Console.WriteLine("Введите 2 массива от 1 до 3");

int ar1 = Convert.ToInt32(Console.ReadLine());

int ar2 = Convert.ToInt32(Console.ReadLine());

if ((ar1 == 1 && ar2 == 2) || (ar2 == 1 && ar1 == 2))

{

OneDimensionalArray.SumMinusElements(array1, array2);

Console.ReadKey();

}

if ((ar1 == 1 && ar2 == 3) || (ar2 == 1 && ar1 == 3))

{

OneDimensionalArray.SumMinusElements(array1, array3);

Console.ReadKey();

}

if ((ar1 == 2 && ar2 == 3) || (ar2 == 2 && ar1 == 3))

{

OneDimensionalArray.SumMinusElements(array2, array3);

Console.ReadKey();

}

break;

case 3:

OneDimensionalArray.SumMinusElements(array1, array2, array3);

Console.ReadKey();

break;

default:

Console.WriteLine("Такой пары массивов нет!");

break;

}

break;

case 4:

Console.WriteLine("Введите номер массива для умножения массива на число");

int chooseArray4 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray4)

{

case 1:

array1.MultiplyArray();

Console.ReadKey();

break;

case 2:

array2.MultiplyArray();

Console.ReadKey();

break;

case 3:

array3.MultiplyArray();

Console.ReadKey();

break;

default:

break;

}

break;

case 5:

Console.WriteLine("Введите номер массива для унарного оперирования");

int chooseArray5 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray5)

{

case 1:

array1 = -array1;

break;

case 2:

array2 = -array2;

break;

case 3:

array3 = -array3;

break;

default:

break;

}

break;

case 6:

break;

default:

break;

}

Console.Clear();

}

}

}

}

OneDimensionalArray**.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_03

{

class OneDimensionalArray

{

private int \_numberOfMultiply;

//сделать диномическим количество элементов массива

private int[] \_array;

public int ArrayLength { get => \_array.Length; }

public int this[int index]

{

get => \_array[index];

set => \_array[index] = value;

}

public OneDimensionalArray(int numberOfMultiply, int arrayLength)

{

\_numberOfMultiply = numberOfMultiply;

\_array = new int[arrayLength];

}

public void InputArray()

{

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

{

Console.Write($"Введите {i + 1} элемент массива: ");

\_array[i] = Convert.ToInt32(Console.ReadLine());

}

}

public void OutputArray()

{

Console.Write($"Вывод массива: ");

foreach (var element in \_array)

{

Console.Write($"{element}, ");

}

Console.WriteLine();

}

public static void SumMinusElements(OneDimensionalArray array1, OneDimensionalArray array2)

{

int Sum = 0;

for (int i = 0; i < array1.ArrayLength; i++)

{

if (array1[i] < 0)

{

Sum += array1[i];

}

}

for (int i = 0; i < array2.ArrayLength; i++)

{

if (array2[i] < 0)

{

Sum += array2[i];

}

}

Console.WriteLine($"Сумма отрицательных элементов в нескольких массивах(2): {Sum}");

}

public static void SumMinusElements(OneDimensionalArray array1, OneDimensionalArray array2, OneDimensionalArray array3)

{

int Sum = 0;

for (int i = 0; i < array1.ArrayLength; i++)

{

if (array1[i] < 0)

{

Sum += array1[i];

}

}

for (int i = 0; i < array2.ArrayLength; i++)

{

if (array2[i] < 0)

{

Sum += array2[i];

}

}

for (int i = 0; i < array3.ArrayLength; i++)

{

if (array3[i] < 0)

{

Sum += array3[i];

}

}

Console.WriteLine($"Сумма отрицательных элементов в нескольких массивах(3): {Sum}");

}

public void MultiplyArray()

{

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

{

\_array[i] = \_array[i] \* \_numberOfMultiply;

}

Console.Write($"Массив после умножения: ");

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

{

Console.Write(\_array[i] + ", ");

}

Console.WriteLine();

}

// Умножение массива на число ( Как сделать умножение числа на массив?, просто поэлементно умножить массив на число?)

public static OneDimensionalArray operator -(OneDimensionalArray a1)

{

OneDimensionalArray rez = new OneDimensionalArray(a1.ArrayLength, a1.ArrayLength);

for (int i = 0; i < a1.ArrayLength; ++i)

rez[i] = -a1[i];

return rez;

}

}

}

**ЗАДАЧА 2**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_03\_Task\_2\_

{

internal class Program

{

static void Main(string[] args)

{

ArrayTask A = new ArrayTask(3);

ArrayTask B = new ArrayTask(4);

ArrayTask C = new ArrayTask(5);

ArrayTask D = 5 \* A;

ArrayTask E = 2 \* B;

ArrayTask F = -A;

ArrayTask G = C \* 4;

int choose = 0;

int SumA = A.Sum();

int SumB = B.Sum();

while (choose != 6)

{

SumA = A.Sum();

SumB = B.Sum();

D = 5 \* A;

E = 2 \* B;

F = -A;

G = C \* 4;

Console.WriteLine("1)Заполнение массивов");

Console.WriteLine("2)Вывод массивов");

Console.WriteLine("3)Сумма 5\*A и C");

Console.WriteLine("4)Сумма 2\*B, -A, C\*4");

Console.WriteLine("5)Замена всех одинаковых элементов в массиве A на сумму этого массива,\nесли массив A>B(только отрицательные элементы)");

Console.WriteLine("6)Выход");

Console.SetCursorPosition(0, 25);

Console.Write("A: "); A.OutputArray();

Console.Write("B: "); B.OutputArray();

Console.Write("C: "); C.OutputArray();

Console.SetCursorPosition(40, 25);

Console.Write("(5\*A)D: "); D.OutputArray();

Console.SetCursorPosition(40, 26);

Console.Write("(2\*B)E: "); E.OutputArray();

Console.SetCursorPosition(40, 27);

Console.Write("(-A )F: "); F.OutputArray();

Console.SetCursorPosition(40, 28);

Console.Write("(C\*4)G: "); G.OutputArray();

Console.SetCursorPosition(0, 9);

choose = Convert.ToInt32(Console.ReadLine());

switch (choose)

{

case 1:

Console.WriteLine("Введите массив для заполнения от 1 до 3");

int chooseArray1 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray1)

{

case 1:

A.InputArray();

break;

case 2:

B.InputArray();

break;

case 3:

C.InputArray();

break;

default:

Console.WriteLine("Такого массива нет!");

break;

}

break;

case 2:

Console.WriteLine("Введите массив для вывода от 1 до 3");

int chooseArray2 = Convert.ToInt32(Console.ReadLine());

switch (chooseArray2)

{

case 1:

A.OutputArray();

Console.ReadKey();

break;

case 2:

B.OutputArray();

Console.ReadKey();

break;

case 3:

C.OutputArray();

Console.ReadKey();

break;

default:

Console.WriteLine("Такого массива нет!");

Console.ReadKey();

break;

}

break;

case 3:

Console.Write("Сумма отрицательных элементов |5\*A + C|: ");

int Sum1 = D.Sum() + C.Sum();

Console.WriteLine(Sum1);

Console.ReadKey();

break;

case 4:

Console.Write("Сумма отрицательных элементов |2\*B + (-A) + C\*4|: ");

int Sum2 = E.Sum() + F.Sum() + G.Sum();

Console.WriteLine(Sum2);

Console.ReadKey();

break;

case 5:

// брал по модулю т.е |-10|<|-16|

if (SumA < SumB)

{

for (int i = 0; i < A.Length; i++)

{

for (int j = i + 1; j <= A.Length; j++)

{

if (A[j - 1] < 0)

A[j - 1] = SumA;

}

}

}

Console.Write("Массив после последнего задания: ");

for (int i = 0; i < A.Length; i++)

{

Console.Write(A[i] + ", ");

}

Console.WriteLine();

break;

case 6:

break;

default:

break;

}

Console.Clear();

}

}

}

}

**ArrayTask.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_03\_Task\_2\_

{

class ArrayTask

{

public int Length { get { return \_array.Length; } }

private int[] \_array;

public ArrayTask(int length) => \_array = new int[length];

public void InputArray()

{

for (int i = 0; i < \_array.Length; i++)

{

Console.Write($"Введите {i + 1} элемент массива");

\_array[i] = Convert.ToInt32(Console.ReadLine());

}

}

public void OutputArray()

{

Console.Write($"Вывод массива: ");

foreach (var element in \_array)

{

Console.Write($"{element}, ");

}

Console.WriteLine();

}

public int Sum()

{

int Sum = 0;

for (int i = 0; i < \_array.Length; i++)

{

if (\_array[i] < 0)

Sum += \_array[i];

}

return Sum;

}

public int this[int i]

{

get

{

if (i >= 0 && i < \_array.Length) return \_array[i];

else throw new Exception("Плохой индекс " + i);

}

set

{

if (i >= 0 && i < \_array.Length) \_array[i] = value;

else throw new Exception("Плохой индекс " + i);

}

}

public static ArrayTask operator +(ArrayTask array1, ArrayTask array2)

{

int Length1, Length2, k;

if (array1.Length > array2.Length)

{

Length1 = array1.Length; Length2 = array2.Length; k = 1;

}

else

{

Length1 = array2.Length; Length2 = array1.Length; k = 2;

}

ArrayTask TemporaryArray = new ArrayTask(Length1);

for (int i = 0; i < Length1; ++i)

{

if (i < Length2) TemporaryArray[i] = array1[i] + array2[i];

else TemporaryArray[i] = (k == 1) ? array1[i] : array2[i];

}

return TemporaryArray;

}

public static ArrayTask operator \*(int number, ArrayTask array1)

{

ArrayTask TemporaryArray = new ArrayTask(array1.Length);

for (int i = 0; i < array1.Length; ++i)

TemporaryArray[i] = array1[i] \* number;

return TemporaryArray;

}

public static ArrayTask operator \*(ArrayTask array1, int number)

{

ArrayTask TemporaryArray = new ArrayTask(array1.Length);

for (int i = 0; i < array1.Length; ++i)

TemporaryArray[i] = array1[i] \* number;

return TemporaryArray;

}

public static ArrayTask operator -(ArrayTask array1)

{

ArrayTask TemporaryArray = new ArrayTask(array1.Length);

for (int i = 0; i < array1.Length; ++i)

TemporaryArray[i] = -array1[i];

return TemporaryArray;

}

}

}