Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт/Факультет - ИИТиАД

**Название работы -Свой тип**

Отчет по лабораторной работе № 3

по дисциплине

Технология программирования

Выполнил cтудент ИСТб-20-2 М.А. Пигарев

номер группы подпись И. О. Фамилия

21.09.21

дата

Принял Доцент З.А. Бахвалова

21.09.21 Должность подпись И. О. Фамилия

дата

Иркутск –2021

Оглавление

[Постановка задачи 3](#_Toc86091947)

[Словесное описание алгоритма 3](#_Toc86091948)

[Таблица спецификаций 3](#_Toc86091949)

[Таблица тестов 4](#_Toc86091950)

[Код программы 5](#_Toc86091951)

[Результаты тестов 10](#_Toc86091952)

[Ссылка на Git 11](#_Toc86091953)

### Постановка задачи

Множество целых чисел

* объединение двух множеств (как сумма множеств),
* пересечение двух множеств (как произведение множеств),
* разность двух множеств,
* добавление элемента к множеству (как сумма с числом),
* удаление элемента из множества (как разность с числом)

### Словесное описание алгоритма

Модуль OpPlus

1. Создаём новый массив.
2. Перебираем каждый элемент двух массивов и записываем их элементы в новый массив.
3. Вызываем модуль Stroka для вывода ответа.

Модуль Union

1. Создаём новый массив.
2. Перебираем два исходных массива, если их элементы равны, то присваиваем это значение элементу нового массива.
3. Вызываем модуль Stroka для вывода ответа.

Модуль difference

1. Создаём новый массив.
2. Перебираем два исходных массива, если два элемента не равны, то записываем значение элемента первого множества в новый массив.
3. Вызываем модуль Stroka для вывода ответа.

Модуль Stroka

1. Сортируем массив.
2. Удаляем повторяющиеся элементы.
3. Перебираем массив и записываем его элементы в строку.
4. Возвращаем строку.

### Таблица спецификаций

Модуль OpPlus

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| a | Хранение первого множества | Положительные целые числа |
| b | Хранение второго множества | Положительные целые числа |
| Выходные величины | | |
| - | - | - |

Модуль Union

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| a | Хранение первого множества | Положительные целые числа |
| b | Хранение второго множества | Положительные целые числа |
| Выходные величины | | |
| s | Хранение ответа | Строка |

Модуль difference

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| a | Хранение первого множества | Положительные целые числа |
| b | Хранение второго множества | Положительные целые числа |
| Выходные величины | | |
| s | Хранение ответа | Строка |

Модуль Stroka

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| c | Хранение множества | Положительные целые числа |
| Выходные величины | | |
| s | Хранение массива, преобразованного в строку | Строка |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Сложение множеств | A= {17,2,3,4,5}  B= {2,3,8,4,10} | 2 3 4 5 8 10 17 |
| 2 | Умножение множеств | A= {17,20,0,4,5}  B= {20,0,8,5,10} | 0 5 20 |
| 3 | Разность множеств | A= {17,20,0,4,5}  B= {20,0,8,5,10} | 0 4 17 |
| 4 | Добавление элемента | A= {17,2,3,4,5}  B= {8} | 2 3 4 5 8 17 |
| 5 | Удаление элемента | A= {17,20,2,4,5}  B= {20} | 2 4 5 17 |

### Код программы

Код main:

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 \_3\_laba

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

textBox1.Text = Properties.Settings.Default.s1.ToString();

textBox3.Text = Properties.Settings.Default.s2.ToString();

textBox2.Text = Properties.Settings.Default.outText.ToString();

}

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

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

textBox2.Focus();

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

button1\_Click(sender, e);

}

}

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

{

// int c;

string s1,s2;

s1 = textBox1.Text;

s2 = textBox2.Text;

if(s1[s1.Length-1]==' ')

{

s1 = s1.Substring(0, s1.Length - 1);

}

if (s2[s2.Length - 1] == ' ')

{

s2 = s2.Substring(0, s2.Length - 1);

}

string[] subs = s1.Split(' ');

string[] subs\_1 = s2.Split(' ');

int[] a = new int[subs.Length];

int[] b = new int[subs\_1.Length];

try

{

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

{

a[i]= int.Parse(subs[i]);

}

for (int i = 0; i < subs\_1.Length; i++)

{

b[i]=int.Parse(subs\_1[i]);

}

}

//catch (FormatException)

catch (Exception exc)

{

// сообщение об ошибке

MessageBox.Show("Некорректный ввод. В поля можно вводить только числа и пробелы между ними (в конце и в начале пробелы запрещены).", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

return; // а затем прерываем обработчик

}

switch (comboBox1.Text)

{

case "сумма":

textBox3.Text = Many.OpPlus(a, b);

break;

case "пересечение":

textBox3.Text = Many.Union(a, b);

break;

case "разность":

textBox3.Text = Many.difference(a, b);

break;

}

Properties.Settings.Default.s1 = s1;

Properties.Settings.Default.s2 = s2;

Properties.Settings.Default.outText = textBox3.Text;

Properties.Settings.Default.Save();

// textBox3.Text = outText;

}

}

}

Код класса:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3\_laba

{

public class Many

{

private int[] arr;

public Many(string s) //конструктор

{

string[] subs = s.Split(' ');

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

{

arr[i] = int.Parse(subs[i]);

}

}

public static string OpPlus(int[] a ,int[] b) //объединение двух множеств

{

// string s = "";

int[] c = new int[a.Length+b.Length];

// int[] c;

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

{

c[i] = a[i];

}

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

{

c[i + a.Length] = b[i];

}

/\*

Array.Sort(c);

int[] ch = c.Distinct().ToArray();

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

{

s = s + ch[i]+" ";

}

// s= c.Distinct().ToArray().ToString();

s = s.Substring(0, s.Length - 1);

return s;

\*/

return Stroka(c,true);

}

public static string Union(int[] a, int[] b)

{

string s = "";

int k = 0;

bool check = false,check\_b=false;

int[] c = new int[a.Length + b.Length];

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

{

for (int j = 0; j < b.Length; j++)

{

if (a[i] == b[j])

{

c[k] = a[i];

k++;

}

if (b[j] == 0)

{

check\_b = true;

}

}

if (a[i] == 0)

{

check = true;

}

}

/\*

Array.Sort(c);

int[] ch = c.Distinct().ToArray();

if ((check == true) & (check\_b ==true))

{

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

{

s = s + ch[i] + " ";

}

}

else

{

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

{

s = s + ch[i] + " ";

}

}

s = s.Substring(0, s.Length - 1);

return s;

\*/

if ((check == true) & (check\_b == true))

{

s=Stroka(c, true);

}

else

{

s= Stroka(c, false);

}

return s;

}

public static string difference(int[] a, int[] b)

{

string s = "";

int k = 0;

bool check = false, check\_a = false,check\_b=false;

int[] c = new int[a.Length];

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

{

check = false;

for (int j = 0; j < b.Length; j++)

{

if (a[i] == b[j])

{

check = true;

}

if (b[j] == 0)

{

check\_b = true;

}

}

if (check == false)

{

c[k] = a[i];

k++;

}

if (a[i] == 0)

{

check\_a = true;

}

}

if ((check\_a == true) & (check\_b == true))

{

s = Stroka(c, true);

}

else

{

s = Stroka(c, false);

}

// s = Stroka(c, true);

return s;

}

public static string Stroka(int[] c, bool check)

{

string s = "";

Array.Sort(c);

int[] ch = c.Distinct().ToArray();

if (check == true)

{

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

{

s = s + ch[i] + " ";

}

}

else

{

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

{

s = s + ch[i] + " ";

}

}

s = s.Substring(0, s.Length - 1);

return s;

}

}

}

## Результаты тестов

|  |  |
| --- | --- |
| №Теста | Скриншот результата |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

## Ссылка на Git

https://github.com/Bert-10/3\_laba