Можно ли не использовать плюсы?

Куда слать отчёты?

|  |  |
| --- | --- |
| Отчёты: @mr\_agalakov |  |

@mr\_agalakov это что?

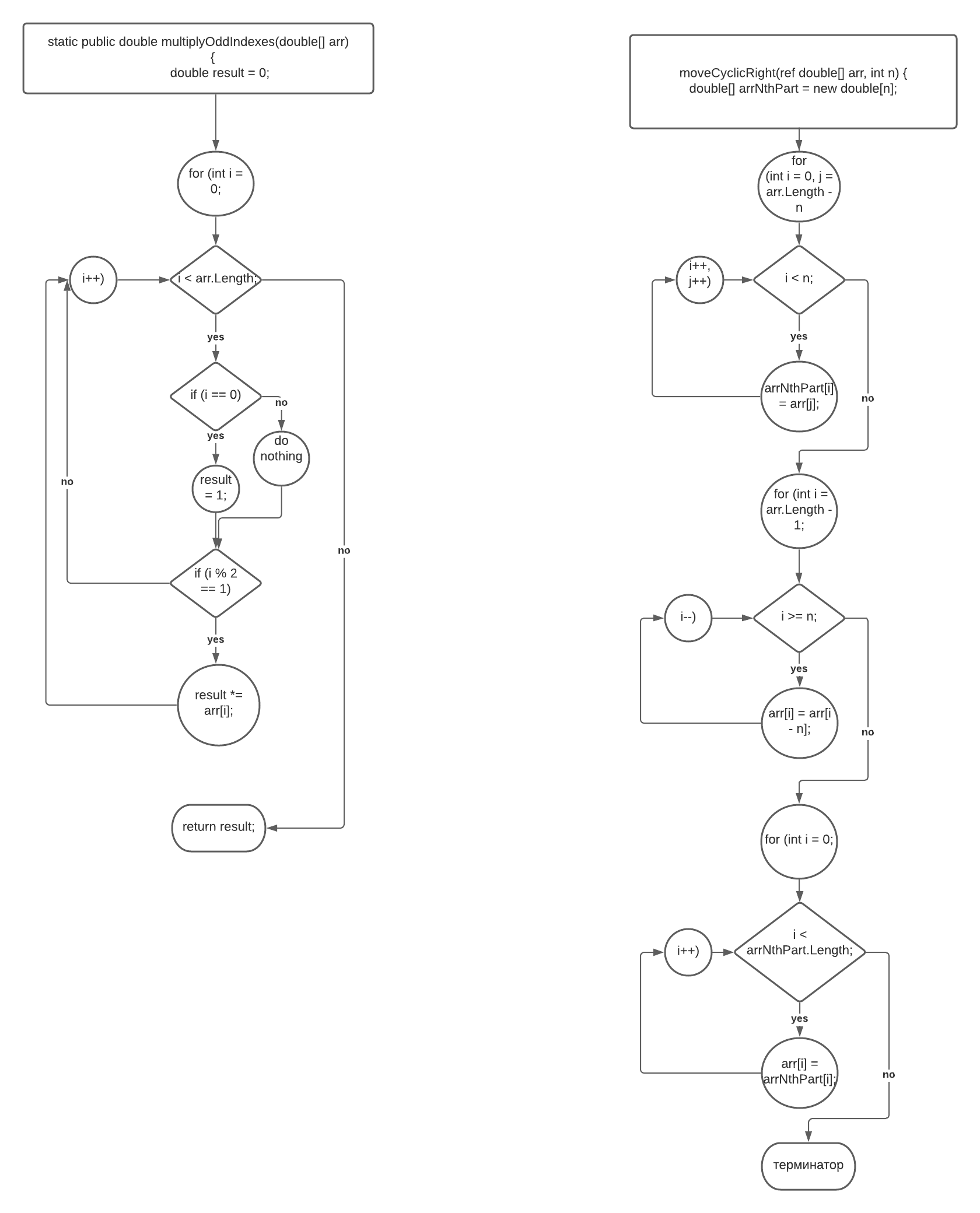
Как оформлять отчет?(где брать титул)

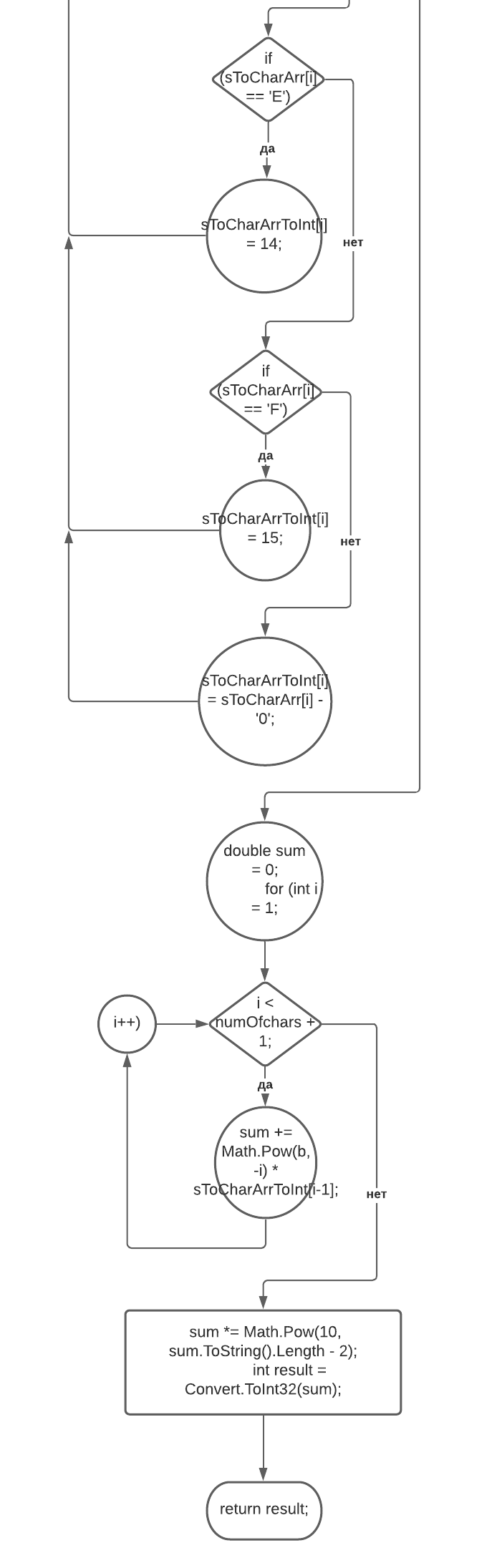
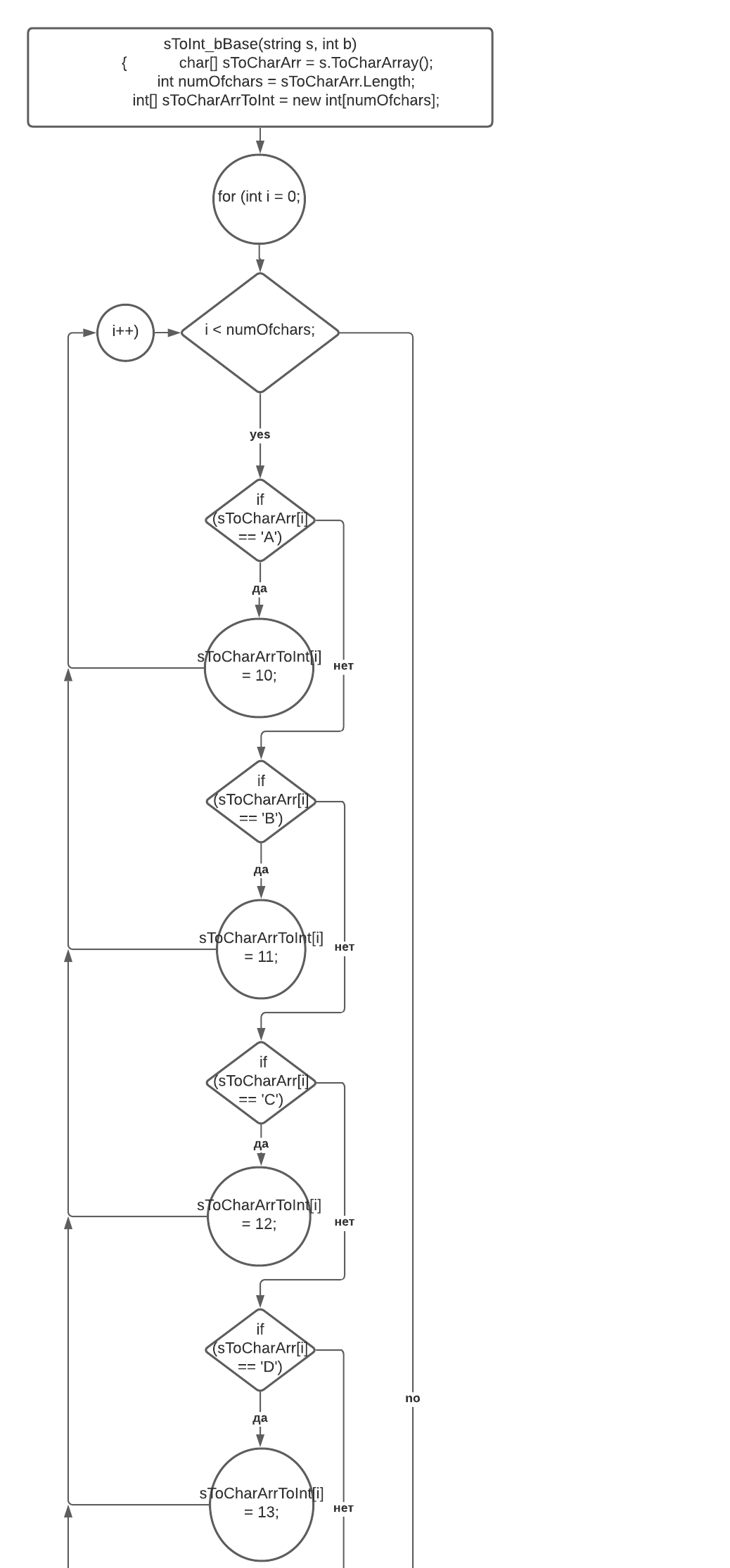
**Задание Вариант 1**

Функция получает одномерный массив вещественных переменных. Вычисляет и возвращает произведение значений компонентов массива с нечетными значениями индексов.

Функция получает одномерный массив вещественных переменных и целое – параметр сдвига. Функция изменяет массив циклическим сдвигом значений его элементов вправо на число позиций, равное параметру сдвига.

Функция получает целое число b – основание системы счисления и строку s, содержащую представление дробной части числа в системы счисления с основанием b. Функция формирует и возвращает из строки s целое число.

**УГП**



Листинг прогаммы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace STP\_01\_changingTheTypeToDotNetFramework

{

public class MyClass

{

//tests lab 1

static void Main(string[] args)

{

launchFirstLab();

Console.ReadKey();

}

static public void launchFirstLab()

{

Console.WriteLine("First lab of testing tasks");

double[] arr = { 1, 2.3, 3.1, 8.6 };//2.3\*8.6=19,78

Console.WriteLine("The resulting number of multiplication of numbers with odd indexes is: " +

multiplyOddIndexes(arr) + "\n");

Console.WriteLine("Original array: ");

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

{

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

}

moveCyclicRight(ref arr, 3);

Console.WriteLine("\nMoved array: ");

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

{

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

}

sToInt\_bBase("58", 16);

}

static public double multiplyOddIndexes(double[] arr)

{

double result = 0;

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

{

if (i == 0) result = 1;

if (i % 2 == 1)

{

result \*= arr[i];

}

Console.WriteLine("result = " + result);

}

return result;

}

static public void moveCyclicRight(ref double[] arr, int n)

{//cyclic shift of the elements to the right

double[] arrNthPart = new double[n];

for (int i = 0, j = arr.Length - n; i < n; i++, j++)

{//store the last n elements in an additional array

arrNthPart[i] = arr[j];

}

for (int i = arr.Length - 1; i >= n; i--)

{//shift arr's elements n positions right

arr[i] = arr[i - n];

}

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

{//add the remaining elements

arr[i] = arrNthPart[i];

}

}

static public int sToInt\_bBase(string s, int b)

{//Функция получает целое число b – основание системы счисления и строку

//s, содержащую представление дробной части числа в системе счисления с

//основанием b. Функция формирует и возвращает из строки s целое число. Буду считать, что возвращает тип int, b = [2...16]

//https://math.semestr.ru/inf/drob.php

//const int eleven

char[] sToCharArr = s.ToCharArray();

int numOfchars = sToCharArr.Length;

int[] sToCharArrToInt = new int[numOfchars];//decimal representations of chars of s

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

{

if (sToCharArr[i] == 'A')

sToCharArrToInt[i] = 10;

else if (sToCharArr[i] == 'B')

sToCharArrToInt[i] = 11;

else if (sToCharArr[i] == 'C')

sToCharArrToInt[i] = 12;

else if (sToCharArr[i] == 'D')

sToCharArrToInt[i] = 13;

else if (sToCharArr[i] == 'E')

sToCharArrToInt[i] = 14;

else if (sToCharArr[i] == 'F')

sToCharArrToInt[i] = 15;

else sToCharArrToInt[i] = sToCharArr[i] - '0';//it's kinda ugly way of converting char to int

}

double sum = 0;

for (int i = 1; i < numOfchars + 1; i++)

{

sum += Math.Pow(b, -i) \* sToCharArrToInt[i - 1];

}

Console.WriteLine("\n\nthe decimal part is " + sum);

sum \*= Math.Pow(10, sum.ToString().Length - 2);

int result = Convert.ToInt32(sum);

Console.WriteLine("result = " + result);

return result;

}

static void moveCyclicLeft(ref double[] arr, int n)

{

double[] arrCloned = (double[])arr.Clone();

double[] arrNthPart = new double[n];

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

{//store the first n elements in an additional array

arrNthPart[i] = arr[i];

}

for (int i = 0; i < arr.Length - n; i++)

{//shift arr's elements n positions right

arr[i] = arr[i + n];

}

for (int i = arr.Length - n, j = 0; j < arrNthPart.Length; i++, j++)

{//add the remaining elements

arr[i] = arrNthPart[j];

}

}

//static int

}

}

**Тесты**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using STP\_01\_changingTheTypeToDotNetFramework;

namespace MytProject\_1\_Tests

{

[TestClass]

public class MyClassTests

{

[TestMethod]

public void Shift\_Right\_result\_2()

{

//arrange(обеспечить)

double[] result = new double[] { 1, 2 };//Исходный массив.

int p = 1;

double[] expected = new double[] { 2, 1 };//Ожидаемое значение.

//act(выполнить)

MyClass.moveCyclicRight(ref result, p);

//assert(доказать)

CollectionAssert.AreEqual(expected, result);//Оракул

}

[TestMethod]

public void multiplyOddIndexes\_result\_19\_78()

{

//arrange(обеспечить)

double[] arr = { 1, 2.3, 3.1, 8.6 };//2.3\*8.6=19,78//Исходный массив.

double expected = 19.78d;//Ожидаемое значение.

//act(выполнить)

double result = MyClass.multiplyOddIndexes(arr);

//assert(доказать)

Assert.AreEqual(result, expected, 0.00000001d, "maybe delta isn't small enough");

}

[TestMethod]

public void sToInt\_bBase\_58\_16\_34375()

{

//arrange(обеспечить)

string s = "58";

int b = 16;

int expected = 34375;

int result = MyClass.sToInt\_bBase(s, b);

//assert(доказать)

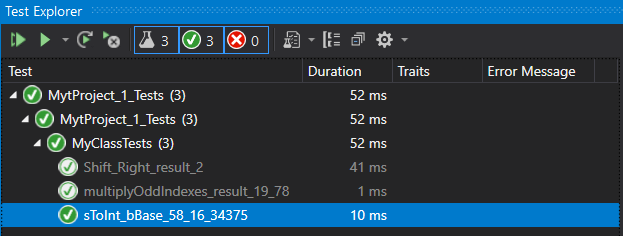
Assert.AreEqual(expected, result);

}

}

}

**Результаты тестирования**



**Вывод**

Научился создавать тесты в среде Visual Studio.