**Содержание модуля "practicLib":**

using System.Data;

namespace practicLib

{

public static class practicLib

{

/// <summary>

/// Ввести двухзначное число. Определить: одинаковы ли его цифры.

/// </summary>

/// <param name="number"></param>

/// <returns></returns>

public static bool equalOfNumbers(int number)

{

// Инициализация необходимых переменных

bool equal;

int first\_number;

int second\_number;

first\_number = number / 10; // Получение первого числа

second\_number = number % 10; // Получение второго числа

if (first\_number == second\_number) // Если первое и второе число совпадают, вернуть true

{ // В противном случае вернуть false

equal = true;

} else { equal = false; }

return equal;

}

/// <summary>

/// Ввести три целых числа. Возвести в квадрат те из них, значения которых неотрицательны.

/// </summary>

/// <param name="a"></param>

/// <param name="b"></param>

/// <param name="c"></param>

/// <param name="a\_pow"></param>

/// <param name="b\_pow"></param>

/// <param name="c\_pow"></param>

public static void threeNumbersPow(double a, double b, double c, out double a\_pow, out double b\_pow, out double c\_pow)

{

// Инициализация необходимых переменных

a\_pow = 0;

b\_pow = 0;

c\_pow = 0;

if (a > 0) // В случае если число больше 0, возвести его в квадрат

{

a\_pow = Math.Pow(a, 2);

} else { a\_pow = a; }

if (b > 0)

{

b\_pow = Math.Pow(b, 2);

} else { b\_pow = b; }

if (c > 0)

{

c\_pow = Math.Pow(c, 2);

} else { c\_pow = c; }

}

/// <summary>

/// Составьте программу вычисления в массиве суммы всех чисел, кратных 3.

/// </summary>

/// <param name="array"></param>

/// <returns></returns>

public static int sumOfNumbers(int[] array)

{

// Инициализация необходимых переменных

int sum = 0;

for (int i = 0; i < array.GetLength(0); i++) // Перебор всех элементов массива

{

if (array[i] % 3 == 0) // В случае если элемент кратен 3, сложить его с текущим значением суммы

{

sum += array[i];

}

}

return sum;

}

/// <summary>

/// Дана целочисленная матрица размера M \* N. Найти номер первого из её столбцов, содержащих максимальное количество одинаковых элементов.

/// </summary>

/// <param name="matrix"></param>

/// <returns></returns>

public static int columnMaxEqualPos(int[,] matrix)

{

// Инициализация необходимых переменных

int m = matrix.GetLength(0), n = matrix.GetLength(1), max\_value, dup\_count = 0, cur\_column = -1;

int[] array = new int[n];

for (int j = 0; j < n; j++) // Перебор столбцов матрицы

{

for (int i = 0; i < m - 1; i++)

{

if (matrix[i, j] == matrix[i + 1, j]) // Если найдено повторение, увеличить количество повторений

{

dup\_count += 1;

}

}

array[j] = dup\_count; // Добавить количество повторений в отдельный массив

dup\_count = 0; // Обнуление счетчика повторений

}

max\_value = array[0];

for (int i = 0; i < n; i++) // Поиск самого большого значения

{

if (max\_value < array[i])

{

max\_value = array[i];

}

}

for (int i = 0; i < n; i++) // Поиск первого столбца с самым большим значением

{

if (array[i] == max\_value) // Когда было найдено, закончить выполнение цикла

{

cur\_column = i;

break;

}

}

return cur\_column; // Возвращение индекса первого столбца с самым большим значением

}

public static DataTable ToDataTable<T>(this T[] arr)

{

var res = new DataTable();

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

{

res.Columns.Add("col" + (i + 1), typeof(T));

}

var row = res.NewRow();

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

{

row[i] = arr[i];

}

res.Rows.Add(row);

return res;

}

//Метод для двухмерного массива

public static DataTable ToDataTable<T>(this T[,] matrix)

{

var res = new DataTable();

for (int i = 0; i < matrix.GetLength(1); i++)

{

res.Columns.Add("col" + (i + 1), typeof(T));

}

for (int i = 0; i < matrix.GetLength(0); i++)

{

var row = res.NewRow();

for (int j = 0; j < matrix.GetLength(1); j++)

{

row[j] = matrix[i, j];

}

res.Rows.Add(row);

}

return res;

}

}

}