**Практическая работа № 6**

**Работа с компонентом DataGridView**

**Студент:** Тустугашев Яков Витальевич

**Группа:** ИС-19

**Задание:** Дана целочисленная квадратная матрица **А**. Получить массив **B**, *i*-ый элемент которого содержит среднее значение элементов *i*-ой строки матрицы **A**. Отсортировать массив **B** по возрастанию.

Будем считать, что начальное и минимально число строк и столбцов матрицы равно двум, а максимальное – 10.

Для решения поставленной задачи потребуются компоненты:

*Count\_L* класса *Label* для подписи компонента *Count\_NUD*;

*Count\_NUD* класса *NumericUpDown* для задания размерности матрицы **А**;

*Matr\_L* класса *Label* для подписи компонента *Matr\_DGV*;

*Matr\_DGV* класса *DataGridView* для ввода матрицы **A**;

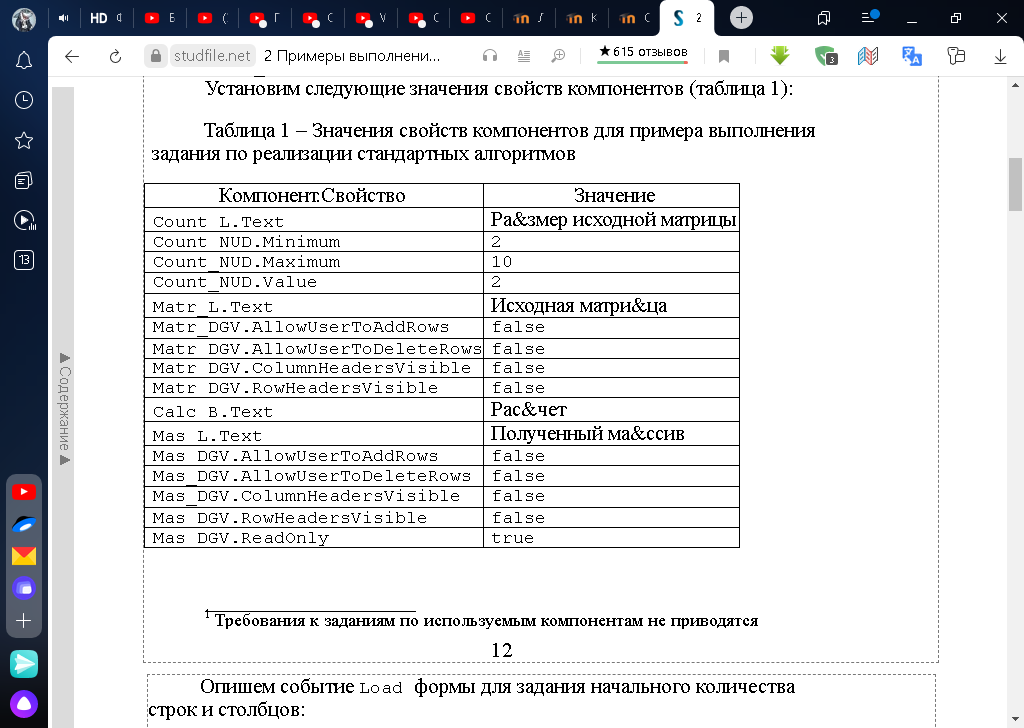
*Calc\_B* класса *Button* для активизации расчета;

*Mas\_L* класса *Label* для подписи компонента *Mas\_DGV*;

*Mas\_DGV* класса *DataGridView* для вывода массива **В**.

Установим следующие значения свойств компонентов (таблица 1):

Таблица 1 – Значения свойств компонентов для примера выполнения задания по реализации стандартных алгоритмов.



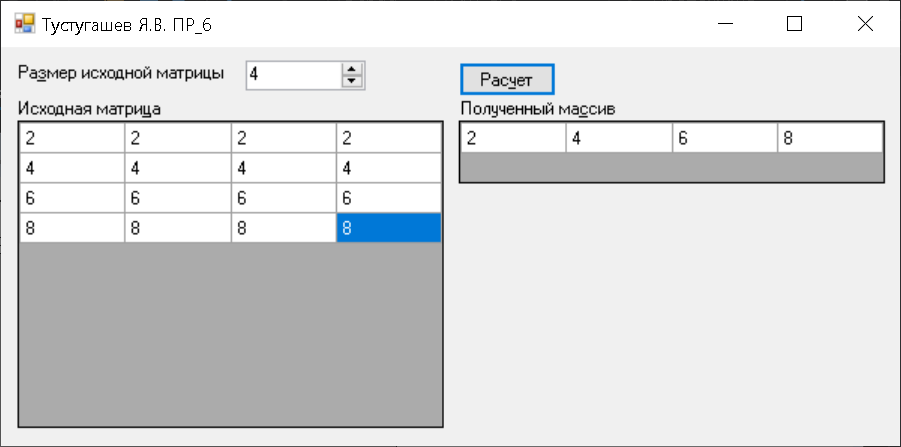


Рисунок – 1

namespace PR\_6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Опишем событие Load формы для задания начального количества строк и столбцов:

private void Form1\_Load(object sender, EventArgs e)

{

Matr\_DGV.ColumnCount = Convert.ToInt32(Count\_NUD.Value);

Matr\_DGV.RowCount = Convert.ToInt32(Count\_NUD.Value);

//Размерность матрицы и количество столбцов массива одинаковые

Mas\_DGV.ColumnCount = Convert.ToInt32(Count\_NUD.Value);

Mas\_DGV.RowCount = 1;

}

//Опишем событие ValueChanged компонента Count\_NUD:

private void Count\_NUD\_ValueChanged(object sender, EventArgs e)

{

Matr\_DGV.ColumnCount = Convert.ToInt32(Count\_NUD.Value);

Matr\_DGV.RowCount = Convert.ToInt32(Count\_NUD.Value);

Mas\_DGV.ColumnCount = Convert.ToInt32(Count\_NUD.Value);

}

//Опишем событие Click кнопки Calc\_B:

private void Calc\_B\_Click(object sender, EventArgs e)

{

double[] mas = new double[Mas\_DGV.ColumnCount];

int[,] matr = new int[Matr\_DGV.RowCount,Matr\_DGV.ColumnCount];

int i, j;

//Заполнение матрицы

for (i = 0; i < Matr\_DGV.RowCount; i++)

for (j = 0; j < Matr\_DGV.ColumnCount; j++)

matr[i, j] = Convert.ToInt32(Matr\_DGV[j, i].Value);

//Перебор строк матрицы

for (i = 0; i < Matr\_DGV.RowCount; i++)

{

//Для каждой строки находится сумма ее элементов

mas[i] = 0;

for (j = 0; j < Matr\_DGV.ColumnCount; j++)

mas[i] += matr[i, j];

//Для нахождения среднего сумма элементов строки делится на их количество

mas[i] /= Matr\_DGV.ColumnCount;

}

//Сортировка массива

i = 0;

while (i < Matr\_DGV.ColumnCount - 1)

if (mas[i] > mas[i + 1])

{

double temp = mas[i];

mas[i] = mas[i + 1];

mas[i + 1] = temp;

if (i > 0)

i--;

else

i++;

}

else

i++;

//Вывод массива

for (i = 0; i < Matr\_DGV.ColumnCount; i++)

Mas\_DGV[i, 0].Value = mas[i];

}

}

}

//BY TUSTUGASHEV Y.V.