Практическая работа № 6

Работа с компонентом DataGridView

Студент: Тустугашев Яков Витальевич

Группа: ИС-19

Задание: Дана целочисленная квадратная матрица **A**. 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 A;

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 ${\bf B}$.

Установим следующие значения свойств компонентов (таблица 1):

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

Компонент:Свойство	Значение
Count L.Text	Ра&змер исходной матрицы
Count NUD.Minimum	2
Count_NUD.Maximum	10
Count_NUD.Value	2
Matr_L.Text	Исходная матри&ца
Matr_DGV.AllowUserToAddRows	false
Matr DGV.AllowUserToDeleteRows	false
Matr DGV.ColumnHeadersVisible	false
Matr DGV.RowHeadersVisible	false
Calc B.Text	Рас&чет
Mas L.Text	Полученный ма&ссив
Mas DGV.AllowUserToAddRows	false
Mas_DGV.AllowUserToDeleteRows	false
Mas_DGV.ColumnHeadersVisible	false
Mas DGV.RowHeadersVisible	false
Mas DGV.ReadOnly	true

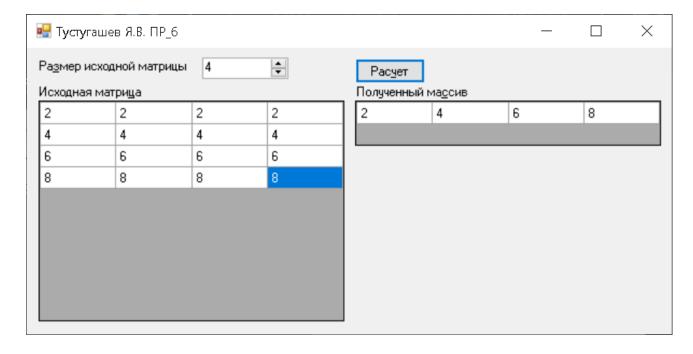


Рисунок – 1

```
namespace PR_6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
             InitializeComponent();
        }
}
```

```
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++)</pre>
                for (j = 0; j < Matr DGV.ColumnCount; j++)</pre>
                    matr[i, j] = Convert.ToInt32(Matr_DGV[j, i].Value);
        //Перебор строк матрицы
            for (i = 0; i < Matr_DGV.RowCount; i++)</pre>
        //Для каждой строки находится сумма ее элементов
                mas[i] = 0;
                for (j = 0; j < Matr_DGV.ColumnCount; j++)</pre>
                    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)</pre>
                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++)</pre>
                    Mas_DGV[i, 0].Value = mas[i];
            }
      }
//BY TUSTUGASHEV Y.V.
```

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