

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 **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	Размер исходной матрицы
Count NUD.Minimum	2
Count NUD.Maximum	10
Count NUD.Value	2
Matr L.Text	Исходная матрица
Matr_DGV.AllowUserToAddRows	false
Matr_DGV.AllowUserToDeleteRows	false
Matr_DGV.ColumnHeadersVisible	false
Matr_DGV.RowHeadersVisible	false
Calc B.Text	Расчет
Mas L.Text	Полученный массив
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();
        }
    }
}
```

```

//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(Mat_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.

```