

## Задание 1. Эффективность языка C#. Умножение матриц (простые типы, работа с массивами)

Напишите на языке C# программу, которая умножает две матрицы, заданные в виде прямоугольных двумерных массивов (практически написана в примере на лекции) и в виде вложенных массивов, и засекает время, необходимое для выполнения умножения матриц на данной машине. Размеры матриц должны задаваться как параметры командной строки или с помощью констант. Программа должна распечатать время (в миллисекундах), требуемое для умножения матриц, а также производительность процедуры умножения матриц на вашем компьютере (в ГФлопс, т.е. миллиардах вещественных операций в секунду). Инициализация матрицы должна проводиться случайным образом (время инициализации во время умножения матриц не включать).

### Примечания:

1. Для преобразования строки в целое число можно использовать статический метод

```
int int.Parse(string s)
```

Например:

```
string Chislo = "3000";  
int i = int.Parse(Chislo);
```

2. Для генерации случайных чисел можно использовать класс **Random** в пространстве имен System и его метод

```
double Random.NextDouble(),
```

который возвращает случайное вещественное число в диапазоне от 0 до 1.

Пример:

```
Random rnd = new Random();  
double d = rnd.NextDouble();
```

3. Для замера времени можно использовать класс Stopwatch из пространства имен System.Diagnostics, его методы **Start()**, **Stop()**, **Reset()** и свойство **ElapsedMilliseconds**, возвращающее суммарное количество миллисекунд, прошедшее от Start() до Stop()

Например:

```
Stopwatch sw = new Stopwatch();  
sw.Start();  
/* вычисления */  
sw.Stop();  
long duration = sw.ElapsedMilliseconds;
```

Поскольку вычисления на современных машинах выполняются достаточно быстро, для значительного отличия размер матрицы должен быть большим (скажем, 1000 x 1000), чтобы погрешность была маленькой. Количество операций при умножении матриц (n x n) считается примерно как  $2 * n^3$ . Для получения производительности в ГФлопс нужно разделить количество операций на время (в секундах).

### Дополнительно для желающих:

Если интересно, реализуйте похожую функциональность на C/C++. Сравните производительность умножения матриц.