

Богатов Р.Н.

# Программирование на языке высокого уровня

---

Лекция 5.  
Массивы. Массивы. Массивы.

Кафедра АСОИУ ОмГТУ, 2012

# Массивы

- Что делать, если пользователь должен ввести  $N$  чисел, а  $N$  заранее неизвестно?
- **Массив** – это упорядоченное множество элементов одного типа.
- Чтобы завести массив в C# требуется: **а)** объявить переменную как массив; **б)** инициализировать массив (выделить память и прочее).
- У массива есть **имя**, а его элементы именуются как **имя[индекс]**, где **индекс** = 0, 1, 2, .... Например,  $x[2*i]$  при  $i = 0, 1, 2, \dots$  – это нечётные элементы массива  $x$ .
- Массив занимает непрерывную область памяти компьютера. (Поэтому, например, вырезать из массива какой-нибудь элемент не так просто.)

# Одномерные массивы. Примеры

```
int[] A; // объявление инициализация
A = new int[100]; // инициализация
```

```
double[] B, C;
B = new double[N];
C = new double[M];
```

```
// примеры использования
```

```
for (int i = 0; i < 100; i++)
    A[i] = 2 * i;
```

```
for (int i = 0; i < N; i++)
    B[i] = Math.Sin(0.1 * i);
```

```
for (int i = 0; i < M; i++)
    D[i] = Math.Abs(C[i]);
```

```
// поиск максимума
```

```
int max = A[0];
for (int i = 1; i < 100; i++)
    if (A[i] > max)
        max = A[i];
```

```
label1.Text = Convert.ToString(max);
```

# Псевдослучайные числа

```
Random r = new Random();

for (int i = 0; i < N; i++)
    A[i] = r.Next(-99, 100);

// числа из диапазона от -99 до 99 включительно, т.е. [-99; 100)
```

```
Random r = new Random();

for (int i = 0; i < N; i++)
    B[i] = r.NextDouble();

// вещественные числа из диапазона [0; 1)
```

```
Random r = new Random();

for (int i = 0; i < N; i++)
    B[i] = r.NextDouble() * (X1 - X0) + X0;

// вещественные числа из диапазона [X0; X1)
```

# Одномерные массивы. Ещё примеры

```
// переворот
for (int i = 0; i < N/2; i++)
{
    double x;
    x = B[i];
    B[i] = B[N-i-1];
    B[N-i-1] = x;
}
```

```
// копирование отрицательных
// элементов массива B в массив C
int Z = 0; //фактический размер массива C

for (int i = 0; i < N; i++)
    if (B[i] < 0)
    {
        C[Z] = B[i];
        Z = Z + 1;
    }
```

```
// сортировка по убыванию
for (int i = 0; i < N-1; i++)
    for (int j = i+1; j < N; j++)
        if (B[j] > B[i])
        {
            double x;
            x = B[i];
            B[i] = B[j];
            B[j] = x;
        }
```

# Задача про пятаки и трёшки

```
int A, B; // искомое число пятаков  
  
for (int i = 0; i < 5; i++)  
    if ((n - i * 3) % 5 == 0)  
        B = i;  
  
A = (n - B * 3) / 5;
```

```
int A, B; // искомое число пятаков и трёшек  
  
for (B = 0; ; B++)  
    if ((n - B * 3) % 5 == 0)  
        break;
```

```
int A, B; // искомое число пятаков и трёшек  
  
for (B = 0; (n - B * 3) % 5 != 0; B++) ;  
  
A = (n - B * 3) / 5;
```

```
    }  
    else  
        if (x == 2)  
        {  
            A =  
            B =  
        }  
    else  
        if (  
        {  
        }  
    else  
    {  
        A = (n - 9) / 5;  
        B = 3;  
    }  
}
```

```
int A, B; // искомое число пятаков и трёшек  
  
int[] G = new int[] { 0, 2, 4, 1, 3 };  
B = G[n % 5];  
A = (n - B * 3) / 5;
```

```
        break;  
    default:  
        A = (n - 9) / 5;  
        B = 3;  
        break;  
    }  
}
```

N = 5a+3b

Введите сумму денег (>7):

Итого: 19 пятаков + 4 трёшек

# Многомерные массивы

- Массивы могут быть двумерными, трёхмерными и ...-мерными.
- Т.к. память компьютера одномерна, многомерные массивы в памяти развёрнуты в одномерные. Например, как если бы из кинотеатра стали выносить кресла на улицу, ряд за рядом, и выстраивать в одну линейку.
- В С# существует два способа задать двумерный (и по аналогии любой многомерный) массив: **а)** как **прямоугольный массив** или **б)** как **«рваный» массив** (массив массивов).

```
// прямоугольный массив  
int[,] F;  
F = new int[N, M];
```

=

```
// рваный массив  
double[][] F;  
F = new double[N][];  
for (int i = 0; i < N; i++)  
    B[i] = new double[M];
```

- Рваные массивы хороши тем, ...что они рваные :o/  
Например, позволяют сделать треугольный массив и сэкономить память.

# Двумерные массивы. Примеры

```
// полное обнуление
for (int i = 0; i < N; i++)
    for (int j = 0; j < M; j++)
        F[i, j] = 0;

// верхняя треугольная матрица
for (int i = 0; i < N; i++)
    for (int j = i; j < M; j++)
        F[i, j] = 0;

// нижняя треугольная матрица
for (int i = 0; i < N; i++)
    for (int j = 0; j <= i; j++)
        F[i, j] = 0;

// под побочной диагональю
for (int i = 1; i < N; i++)
    for (int j = M-i; j < M; j++)
        F[i, j] = 0;
```

```
// поиск максимума
int max = A[0];
for (int i = 1; i < 100; i++)
    if (A[i] > max)
        max = A[i];

label1.Text = Convert.ToString(max);
```

```
// поиск максимума
double max = double.MinValue;
for (int i = 0; i < N; i++)
    for (int j = 0; j < M; j++)
        if (F[i, j] > max)
            max = F[i, j];

label1.Text = Convert.ToString(max);
```



# Цикл `foreach`

- **foreach** – это упрощённый вариант **for** для перебора всех элементов массива любой размерности.

- Обход массива **ИМЯ** с элементами типа **ТИП**:

```
foreach(тип переменная in ИМЯ)
```

```
    тело_цикла
```

(В теле цикла **переменная** будет принимать значения всех элементов массива по очереди.)

- Одномерный пример:

```
int sum = 0;
foreach (int x in A)
    sum = sum + x;
```

- Многомерный пример:

```
// поиск максимума
double max = double.MinValue;
for (int i = 0; i < N; i++)
    for (int j = 0; j < M; j++)
        if (F[i, j] > max)
            max = F[i, j];
```

==

```
// поиск максимума
double max = double.MinValue;
foreach (double x in F)
    if (x > max)
        max = x;
```

## Домашнее задание

Подсчитать сумму граничных элементов прямоугольной матрицы (лежащих по периметру). Отдельно решить ту же задачу для квадратной матрицы.

Подсказки:

- можно сделать четыре цикла (пробегающих по каждой стороне)
- а лучше – два (для обхода по горизонтали и вертикали)
- а в случае квадратной матрицы хватит и одного

# Упражнения на одномерные массивы

- Заполнить массив натуральными числами, значениями  $\sin x$ , случайными числами
- Подсчитать в массиве количество отрицательных чисел
- Найти сумму элементов массива, максимум, минимум, среднее...
- Найти произведение элементов массива (с прерыванием цикла, если встретится ноль)
- Найти элемент, наиболее близкий к заданному числу
- Каких элементов больше: отрицательных или неотрицательных?
- Заполнить массив по образцу: 1, 3, 5, ...,  $N-2$ , ...,  $N-1$ , ..., 6, 4, 2
- Проверить, является ли массив упорядоченным по возрастанию
- Повышенной сложности: найти медиану массива – элемент, разделяющий массив на две наиболее близких по сумме части

## Упражнения на двумерные массивы

- Заполнить массив натуральными числами, функцией двух переменных (например,  $-2x^4+x^3+8x^2-2y^4+y^3+8y^2$ ), случайными числами
- Подсчитать в массиве количество чисел, близких к нулю (с заданной точностью)
- Найти максимум, минимум, среднее, суммы главной и побочной диагоналей. Вычесть из матрицы значение минимума
- Транспонировать квадратную матрицу
- Получить заданный минор (удалить заданные строку и столбец)
- Переставить четверти по часовой стрелке
- Повышенной сложности: заполнить массив натуральными числами а) по спирали, б) змейкой, в) массив 8x8 – прыжками шахматного коня