

Семинар 7. Массивы и коллекции

7.1 Массивы

7.1.1 Простые массивы

Когда в программе всего несколько переменных или объектов, запутаться в них, как и заблудиться в трех соснах, чрезвычайно сложно. Другое дело, если переменных или объектов сотни. На одно придумывание названий для них уйдет целый день. К счастью, в программировании можно собирать любое количество однотипных элементов в один пакет. Название будет только у пакета, а объекты в пакете будут различаться по номерам.

Такие пакеты программисты называют массивами. Работать с ними достаточно просто, объявляются они с помощью служебных слов «array» и «new». Создадим массив для десяти переменных целого типа.

```
Int32[] array = new Int32[10];  
array[0] = 99;
```

Обращаться к элементам массива следует по номеру, нумерация массива начинается с нуля, поэтому первый элемент будет иметь номер 0, второй — 1, а последний будет под номером 9. При необходимости элементы массива можно задать при его объявлении, для этого их нужно перечислить в фигурных скобках через запятую

```
Int32[] array = new Int32[5] {45, 78, 54, 4, 9};  
Console.Write(array[2]);
```

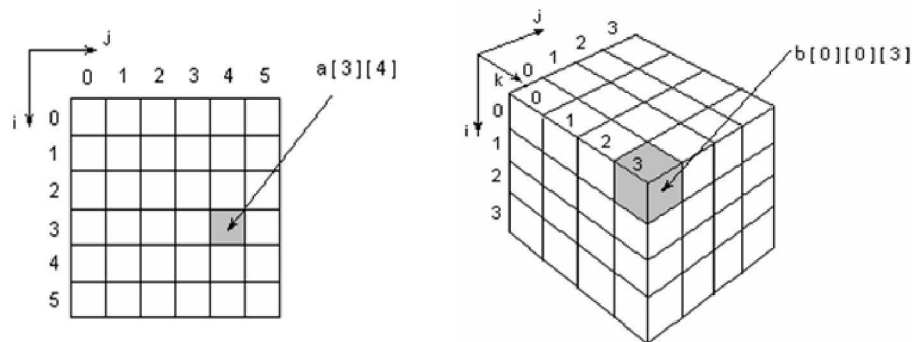
Для массивов можно применять следующие команды:

- Clear – очистка массива от имеющихся данных
- Copy – копирование массива
- Sort – сортировка массива
- Length – получение количества элементов в массиве (его длины)

7.1.2 Многомерные массивы

В приведенных выше примерах массивы содержали только «одну строку» – такие массивы называются одномерными. Но в программировании можно создавать массивы с любой размерностью.

К примеру, двумерный массив можно представить себе в виде таблицы, каждая ячейка которой задается двумя индексами – номерами столбца и строки. Трехмерный массив можно представить в виде куба, в котором уже три индекса – столбик, строчка и ряд. Данные «кубики» можно раздать одноклассникам, и получится 4–мерный массив, в котором новый индекс – это порядковый номер одноклассника в журнале. Классов в школе много, поэтому еще одним индексом может быть номер класса, получается уже 5–мерный массив. И так можно продолжать бесконечно долго, лишь бы памяти хватило.



**Рис. 7-4 Графическое представление
двухмерных и трехмерных массивов**

Для того чтобы объявить многомерный массив, количество элементов нужно указать отдельно для каждого измерения. Чтобы обратиться к отдельному элементу, нужно знать его точное местоположение.

Общий вид:

```
тип [, ,] имя = new тип [размер_1, размер_2 ... размер_n];
```

Пример:

```
Int32[,] a = new Int32[6, 6];
a[3, 4] = 10;
Console.WriteLine(a[3][4]);
```

Бывают так же «ломанные» многомерные массивы, у которых, к примеру, строки имеют разную длину.

```
Int32[][] a = new Int32[4][];
a[0] = new int[7];
a[1] = new int[3];
a[2] = new int[9];
a[3] = new int[12];
a[3][4] = 8;
Console.WriteLine(a[3][4]);
```

7.2 Коллекции

Безусловно, массивы представляют собой достаточно удобное средство для хранения информации, но даже им присущи недостатки и ограничения. Например,

- размер массива задается при его создании и не может быть переопределен
- в середину массива невозможно добавить новый элемент или удалить его

Для решения всех этих проблем в языке C# существуют «коллекции» – объекты, которые позволяют хранить любое количество других объектов и предоставляют программисту полную свободу действий над ними.

Для использования коллекций в начало программы нужно обязательно добавить строку «using System.Collections.ObjectModel;». Сами коллекции объявляются в тексте, как переменные, но только с использованием слова «Collection» и с указанием типа элементов в угловых скобках.

В общем виде:

```
Collection<тип> имя_коллекции = new Collection<тип>();
```

Пример:

```
Collection<String> stringcollection = new Collection<String>();
```

Изначально коллекция пуста. Для того чтобы добавить в нее элемент, нужно воспользоваться функцией «Add();» После этого к элементу можно будет обратиться также, как и в массивах, через индекс, указав его в квадратных скобках.

```
stringcollection.Add(«первая строка в коллекции»);  
Console.Write(stringcollection[0]);  
stringcollection[0] = «первая строка изменена»;  
Console.Write(stringcollection[0]);
```

Объекты коллекций предоставляют большое количество свойств и методов для манипуляции своими элементами. Мы рассмотрим только самые важные из них.

- Add – вставляет элемент в конец коллекции
- Clear – очищает коллекцию
- Insert – вставляет элемент в коллекцию под определенным номером
- Remove – удаляет элемент с определенным номером
- RemoveAt – удаляет все элементы до конца, начиная с некоторого
- SetItem – заменяет элемент с определенным номером на новый
- Count – возвращает количество элементов в массиве (его длину)

7.3 Массивы vs коллекции

Если коллекции так хороши, то зачем тогда нужны массивы? Все очень просто. Массивы представляют собой более простую структуру данных, их длина практически не ограничена. Кроме того, при операциях с массивами не происходит лишних циклов, шерстящих набор данных, поэтому они работают значительно быстрее коллекций.

Выбор той или иной конструкции определяется условием поставленной задачи. Если набор данных невелик, тогда целесообразнее использовать коллекции, т.к. в этом случае у нас появляются дополнительные команды для работы с данными. Если требуется повысить эффективность обработки большого объема данных, тогда следует остановить свой выбор на массивах и реализовать все необходимые функции вручную, избавив их от лишних проверок.

7.4 Циклическая конструкция foreach

С массивами обычно не работают вручную, для этого пользуются циклами. Можно использовать любой из известных тебе конструкций, но так как в теле цикла необходимо иметь счетчик для его подстановки в массив в качестве индекса, наиболее удобным является цикл for.

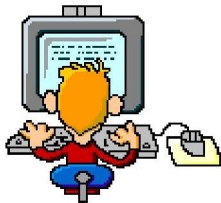
```
Int32[] fibarray = new int[] { 0, 1, 2, 3, 5, 8, 13 };  
foreach (Int32 i=0; i<7; i++)  
{  
    System.Console.WriteLine(i);  
}
```

Спустя некоторое время разработчики пришли к выводу, что указанная выше конструкция является настолько часто используемой, что ее целесообразно определить на уровне языка, так появился цикл foreach. Он является случаем цикла for, и позволяет повторить группу вложенных операторов для каждого элемента массива или коллекции объектов.

```
Int32[] fibarray = new int[] { 5, 2, 54, 87, 5, 8, 2 };
foreach (Int32 i in fibarray)
{
    System.Console.WriteLine(i);
}
```

Многомерные массивы обычно обрабатываются во вложенных циклах. В этом случае управляющая переменная внешнего цикла – первый индекс массива (изменяется медленно), а управляющая переменная внутреннего цикла – второй индекс массива (изменяется быстро). Например, построим таблицу умножения:

```
Int32[,] MutlyTab = new Int32[10, 10];
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        MutlyTab[i, j] = i * j;
    }
}
Console.WriteLine(MutlyTab[5, 5]);
Console.ReadLine();
```



Задания

- #1** Напиши программу, которая не только заполняет таблицу умножения, но и выводит потом ее на экран.
- #2** Напиши программу, цель которой - в квадратной матрице размером 5x5 элементов, заполненной случайными значениями, поменять местами главную и побочную диагонали
- #3** Напиши программу, в ходе выполнения которой с клавиатуры вводится одномерный массив из 7 целых чисел и выводится количество ненулевых элементов. Перед вводом каждого элемента на экране должна появляться подсказка с его номером.
- #4** Напиши программу, которая выводит наименьший элемент введенного с клавиатуры одномерного массива целых положительных чисел. Если пользователь будет вводить число 0, программа должна потребовать ввести число еще раз.
- #5** Напиши программу, которая вычисляет среднее арифметическое ненулевых элементов введенного с клавиатуры одномерного массива. С клавиатуры вводится количество элементов и сами элементы массива целых чисел.
- #6** Напиши программу, которая проверяет, находится ли введенное пользователем число в заранее определенном массиве чисел.
- #7** Напиши программу, которая проверяет, образуют ли элементы введенного с клавиатуры целочисленного массива возрастающую последовательность.
- #8** Напиши программу, которая переставляет элементы введенного с клавиатуры массива в убывающем порядке.
- #9** В программе есть двухмерный массив, в строках – имена од-

ноклассников, в столбцах – месяцы, а в ячейках – количество трафика, скачанного за месяц.

Напиши программу, вычисляет сумму элементов отдельно по строкам (сколько всего скачано учеником) и столбцам (сколько было скачано всеми за месяц).

	Январь	Февраль	Март	Апрель	Итого
Иванов	25	30	27	10	92
Петров	32	20	45	14	111
Сидоров	26	28	29	15	98
Итого	83	78	101	39	301

#10 Напиши программу, которая меняет местами строки квадратной матрицы, переворачивая ее сверху вниз.

#11 Напиши игру «Угадай число». Правила следующие: компьютер загадывает случайное трехзначное число, а пользователю нужно его отгадать. На каждом шаге угадывающий делает предположение, а компьютер выводит, сколько цифр угадано и сколько из них находятся на своем месте.

Отгадывайте число

- Ваш вариант: 123
- Угадано: 0 На своих местах: 0
- Ваш вариант: 654
- Угадано: 2 На своих местах: 2
- Ваш вариант: 954
- Угадано: 1 На своих местах: 1
- Ваш вариант: 658
- Вы выиграли !!!!!!!