

一维数组

概念:用个连续的内存空间,存储一组相同类型的数据,数组长度不可更改

定义数组的语法

数据类型[] 数组名

```
1      int[] array1 = new int[4]; //仅声明并赋值 (未初始化元素)
2      int[] array2= new int[4] { 6, 7, 8, 9 }; //声明 赋值并初始化元素
3      int[] array3 = { 5, 6, 7 }; //简写
```

访问数组元素

数组名[索引]

```
1 array3[1]; //获取数组中下标为1的元素 即输出6
```

获取数组长度

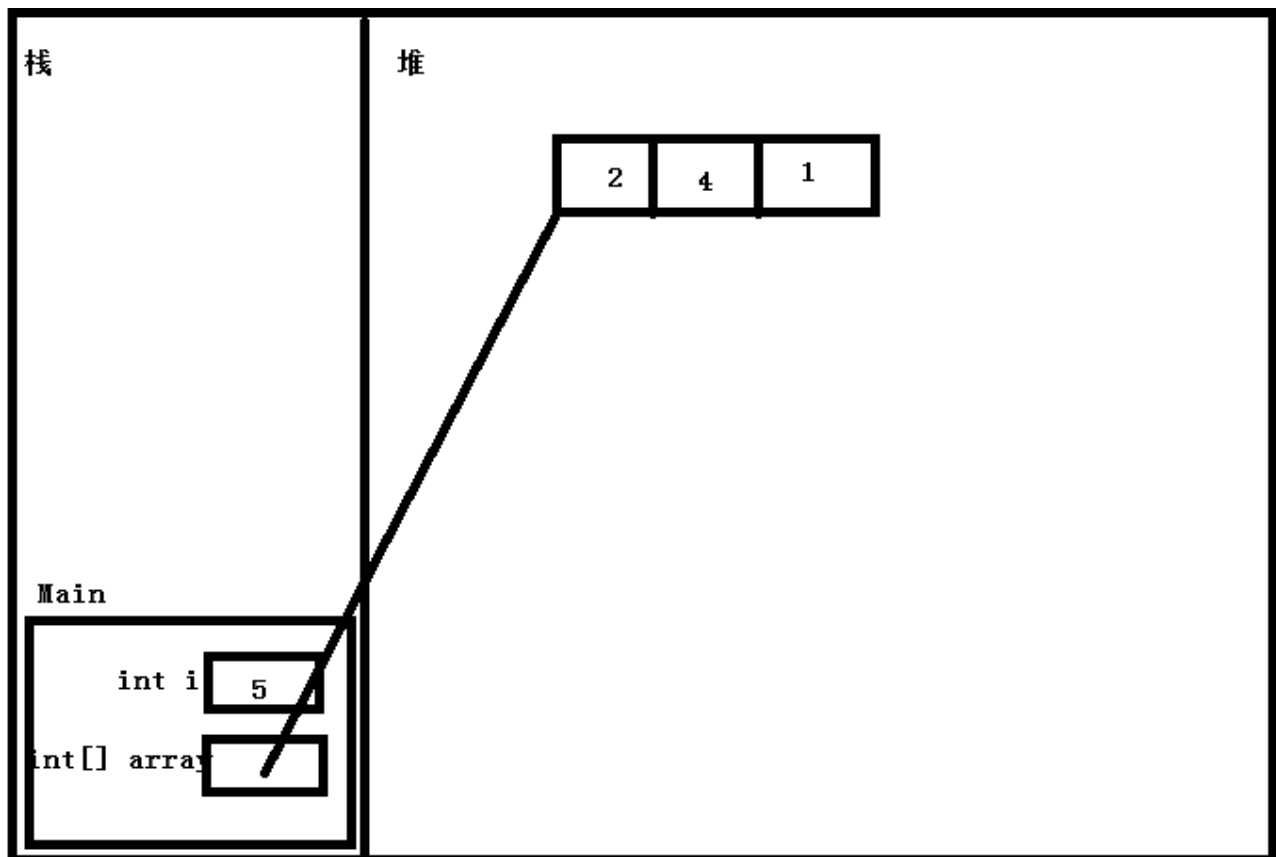
数组名.Length

遍历数组

```
1      //遍历数组 (使用 for循环)
2      for (int i = 0; i < array3.Length ; i++)
3      {
4          Console.WriteLine(array3[i]);
5      }
6      // 遍历数组 (使用foreach)
7      foreach (int item in array3)
8      {
9          Console.WriteLine(item);
10     }
```

数组在内存中的存储方式

```
1 static void Main(string[] args)//以这段代码为例画图
2 {
3     int i = 5;
4     int[] array = { 2, 4, 1 };
5 }
```



二维数组

定义二维数组的语法

```
1 static void Main(string[] args)
2 {
3     //定义二维数组
4     int[,] arr = new int[3, 3] //声明 赋值并初始化元素
5     {
6         { 1, 5, 3 },
7         { 2, 9, 7 },
```

```

8         { 2,9,7}
9     };
10    //简写方式:
11    int[,] arr1 =
12    {
13        {1,5,3 },
14        { 2,9,7}
15    };
16 }

```

访问二维数组元素

```

1 //访问数组元素
2     Console.WriteLine(arr[1, 1]);

```

取对应维度上得长度

```

1     arr.GetLength(0);//row 行数
2     arr.GetLength(1);//column 列数

```

遍历二维数组

```

1     for (int r = 0; r < arr.GetLength(0); r++)//遍历行
2     {
3         for (int c = 0; c < arr.GetLength(1); c++)//遍历列
4         {
5             Console.Write(arr[r,c]);
6         }
7         Console.WriteLine();
8     }

```

交错数组(数组的数组)

概念:外层数组的每一个元素都是数组

```

1      static void Main(string[] args)
2      {
3          int[][] arr = new int[2][]; //定义一个可以存放2个一维数组的交错数组
4          arr[0] = new int[3] { 2,5,6}; //为第一个一维数组赋值
5          arr[1] = new int[2] { 1,9}; //为第二个一维数组赋值
6          //访问元素
7          Console.WriteLine(arr[0][1]); //输出5
8          //遍历
9          for (int i = 0; i < arr.Length; i++) //外层
10         {
11             for (int j = 0; j < arr[i].Length; j++) //内层
12             {
13                 Console.Write(arr[i][j]);
14             }
15             Console.WriteLine(); //换行
16         }
17     }

```

排序算法

冒泡排序

核心思想:每一位和后面的数字比较,如果比后面的大,就交换位置

```

1      int[] arr = { 6, 1, 8, 2, 7, 0 };
2
3      for (int j = 0; j < arr.Length-1; j++)
4      {
5          for (int i = 0; i < arr.Length -1-j; i++)
6          {
7              if (arr[i] > arr[i + 1])
8              {
9                  int temp = arr[i];
10                 arr[i] = arr[i + 1];
11                 arr[i + 1] = temp;
12             }
13         }
14     }

```

选择排序

核心思想:从剩下的里面选最小的放到最前面

```
1 static void Main(string[] args)
2 {
3     int[] arr = { 1, 2, 8, 5, 6, 7 };
4
5     for (int i = 0; i < arr.Length; i++)//决定内层循环的开始位置
6     {
7         int minIndex = i;//minIndex是最小元素的索引
8         for (int j = i + 1; j < arr.Length; j++)
9         {
10             if (arr[j] < arr[minIndex])
11             {
12                 minIndex = j;
13             }
14         }
15         //交换
16         int temp = arr[i];
17         arr[i] = arr[minIndex];
18         arr[minIndex] = temp;
19     }
20 }
21
```

作业:

- 1.输出九九乘法口诀表
- 2.求数组最大值
- 3.判断某一数是否在数组中,输出true或false即可
- 4.倒序输出数组 {8,4,6,1}

输出: 1,6,4,8

- 5.把一下两个数组合并成一个一维数组,并对合并后的数组进行排序,不能使用API

数组1: {3,5,7,2,1}

数组2: {9,12,4}

1×1=1
1×2=2
1×3=3
1×4=4
1×5=5
1×6=6
1×7=7
1×8=8
1×9=9

2×2=4
2×3=6
2×4=8
2×5=10
2×6=12
2×7=14
2×8=16
2×9=18

3×3=9
3×4=12
3×5=15
3×6=18
3×7=21
3×8=24
3×9=27

4×4=16
4×5=20
4×6=24
4×7=28
4×8=32
4×9=36

5×5=25
5×6=30
5×7=35
5×8=40
5×9=45

6×6=36
6×7=42
6×8=48
6×9=54

7×7=49
7×8=56
7×9=63

8×8=64
8×9=72

9×9=81

GOOD LIFE GEAR™



The illustration shows two children, a boy and a girl, standing in a field of red flowers. The boy is on the left, wearing a white shirt and blue shorts, holding a green umbrella. The girl is on the right, wearing a blue dress and red boots, holding an orange umbrella. They are both smiling. The background is a blue sky with white clouds and several blue stars of different sizes.