# visualstudio快捷键

## 快捷注释：crtl+k+c

## 取消快捷注释：ctrl+k+u

## 快捷选中：shift+方向键

# 矩阵数组和交错数组

1. //这是矩形数组，在堆里面是一个数组对象

2. int[,,] data = new int[,,]

3. //这是交错属于组,在堆里面有四个数组对象

4. int[][] ints = new int[3][3]

5.

# String学习

String的split分割如果连续遇到多个分隔符会用空格代替，添加StringSplitOptions.RemoveEmptyEntries来去掉空格

# 数组

数组克隆会在堆里面生成新对象

# 格式化字符串输出

1. int[] intar = { 1, 2, 3, 4, 5};

2. int[] intar2 = (int[])intar;

3. Console.WriteLine("{0}and{1}", intar2[0], intar2[1]);

4. //输出1and2{index，alignment:format}

5. Console.WriteLine($"{intar[0]}and{intar[1]}");

6. //输出1and2

1. int a = 666;

2. Console.WriteLine("|{0,10}|",a);/\*右对齐\*/

3. Console.WriteLine("|{0,-10}|", a);/\*左对齐\*/

4.

# 委托学习

委托就相当于有一个或几个方法的对象，但是委托和典型的对象不同。执行委托，会执行他特有的方法，委托是引用类型，指向了一个调用列表，可以在列表里面增加（+=）方法和移除（-=）方法

1. using System.Diagnostics;

2. using System.IO;

3.  //声明委托

4. delegate void MyDel(int value);

5. internal class Program

6. {

7. void PrintLow(int value)

8. {

9. Console.WriteLine($"{value}Low Value");

10. }

11. void PrintHight(int value)

12. {

13. Console.WriteLine($"{value}Hight Value");

14. }

15. public static void Main(string[] args )

16. {

17. Program program = new Program();

18. MyDel del;

19.

20. Random rand = new Random();

21. int randValue =rand.Next(99);

22. del = randValue < 50 ? new MyDel(program.PrintLow): new MyDel(program.PrintHight);

23. del(randValue);

24. }

25. }

26.

# 匿名方法

1. OtherDel dell=delegate(int x)

2. {

3. return x;

4. };

5.

# Lamda表达式

用lamda表达式来替代匿名方法

1. MyDel dell = delegate (int x) { return x + 1; };/\*匿名对象\*/

2. MyDel lell= (int x) => { return x - 1; };/\*lamda表达式\*/

3.