

## 循环的嵌套和break等关键字的特性

嵌套循环写法：

外层循环每执行一次

内层循环执行5次

```
1      for (int i = 0; i < 5; i++)
2      {
3          Console.WriteLine("外层循环");
4          for (int j = 0; j < 5; j++)
5          {
6              Console.WriteLine("内层循环");
7          }
8      }
```

嵌套循环知识要领：

外层循环执行一次 内层循环全部执行

break写在内层循环 只能退出内层循环

外层循环和内层循环不能出现同名变量

例：使用双重for循环打印4行5列的\*

用外层循环控制行数 内层循环控制列数

```
1  for (int i = 0; i < 4; i++)
2  {
3      for (int j = 0; j < 5; j++)
4      {
5          Console.Write("*");
6      }
7      Console.WriteLine();
8  }
```

例：

一个饭店的3桌客人轮流点5个菜

那么饭店需要给每桌客人做菜和上菜

```
1  for (int i = 1; i <=3; i++)
2  {
3      Console.WriteLine("第" + i + "桌客人点菜");
4      for (int j = 1; j <= 5; j++)
5      {
6          Console.WriteLine("做第" + j + "菜 上菜");
7      }
8  }
```

输入一个 $\leq 100$ 的正整数x 并计算 $1/1+1/2+1/3+\dots+1/x$ 的结果

```
1  Console.WriteLine("请输入一个整数");
2  int x = int.Parse(Console.ReadLine());
3
4  double sum = 0;
5
6  for (int i = 1; i <= x; i++)
7  {
8      sum += 1.0 / i;
9  }
10 Console.WriteLine(sum);
```

1 牌照的前两位数字 相同

2 牌照的后两位数字 相同 但是和前两位不同

3 四位的车号刚好是一个整数的平方

```

1      for (int i = 1000; i <= 9999; i++)
2      {
3          int a = i / 1000;
4          int b = i % 1000 / 100;
5          int c = i % 100 / 10;
6          int d = i % 10;
7
8          //满足甲乙两个人的结果
9          if (a == b && c == d && a != c)
10         {
11             for (int j = 0; j <= 99; j++)
12             {
13                 if(j*j==i)
14                 {
15                     Console.WriteLine("车牌是" + i); //7744
16                     break;
17                 }
18             }
19         }
20     }

```

## 百钱买百鸡

鸡翁一值钱5 鸡母一值钱3 鸡雏3值钱1 百钱买百鸡 问翁 母 雏 各多少只？

```

1  for (int x = 0; x <= 100; x++)
2  {
3      for (int y = 0; y <= 100; y++)
4      {
5          for (int z = 0; z <= 100; z++)
6          {
7              if (x + y + z == 100 && 5 * x + 3 * y + z / 3 == 100 &&
z % 3 == 0)
8              {
9                  Console.WriteLine("鸡翁" + x + "鸡母" + y + "鸡
雏" + z);
10             }
11         }

```

```
12     }
13 }
```

水仙花数 153     $1^3 + 5^3 + 3^3 = 153$

```
1  for (int x = 100; x <= 999; x++)
2  {
3      int b = x/100;
4      int c = x / 10 % 10;
5      int d = x % 10;
6
7      if (b * b * b + c * c * c + d * d * d == x)
8      {
9          Console.WriteLine(x);
10     }
11 }
```

九九乘法表

```
1  for (int i = 1; i <= 9; i++)
2  {
3      for (int j = 1; j <= i; j++)
4      {
5          Console.Write(j + "*" + i + "=" + i * j + "\t");
6      }
7      Console.WriteLine();
8  }
```

打印等腰三角形

```
1  Console.WriteLine("请输入三角形行数");
2  int length = int.Parse(Console.ReadLine());
```

```

3
4 //外层循环 控制整体的行数
5 for (int i = 1; i <= length; i++)
6 {
7     //第一个内层循环 控制打印空格的次数 与行数成反比
8     for (int j = 1; j <= length-i; j++)
9     {
10         Console.Write(" ");
11     }
12
13     //第二层内层循环 控制打印*的次数
14     for (int j = 1; j <= i; j++)
15     {
16         Console.Write("* ");
17     }
18     Console.WriteLine();
19 }

```

## 菱形

```

1 Console.WriteLine("请输入三角形行数");
2 int length = int.Parse(Console.ReadLine());
3
4 //外层循环 控制整体的行数
5 //同上
6
7 //外层循环 控制整体的行数
8 for (int i =length-1; i >=1; i--)
9 {
10     //第一个内层循环 控制打印空格的次数 与行数成反比
11     for (int j = 1; j <= length - i; j++)
12     {
13         Console.Write(" ");
14     }
15
16     //第二层内层循环 控制打印*的次数
17     for (int j = 1; j <= i; j++)
18     {
19         Console.Write("* ");

```

```

20         }
21         Console.WriteLine();
22     }
23
24 }

```

## 空心菱形

```

1      Console.WriteLine("请输入三角形行数");
2      int length = int.Parse(Console.ReadLine());
3
4      //外层循环 控制整体的行数
5      for (int i = 1; i <= length; i++)
6      {
7          //第一个内层循环 控制打印空格的次数 与行数成反比
8          for (int j = 1; j <= length-i; j++)
9          {
10             Console.Write(" ");
11         }
12
13         //第二层内层循环 控制打印*的次数
14         for (int j = 1; j <= i; j++)
15         {
16             if (j == 1 || j == i)
17                 Console.Write("* ");
18             else
19                 Console.Write(" ");
20         }
21         Console.WriteLine();
22     }
23
24     //外层循环 控制整体的行数
25     for (int i =length-1; i >=1; i--)
26     {
27         //第一个内层循环 控制打印空格的次数 与行数成反比
28         for (int j = 1; j <= length - i; j++)
29         {
30             Console.Write(" ");
31         }

```

```
32
33      //第二层内层循环 控制打印*的次数
34      for (int j = 1; j <= i; j++)
35      {
36          if (j == 1 || j == i)
37              Console.Write("* ");
38          else
39              Console.Write(" ");
40      }
41      Console.WriteLine();
42  }
```