

学习笔记

编辑人: 黄明富 日期 2016.10.30.

1.java&c# 变量类型

byte: 单字节类型 -128至127
short: 多字节类型 -32768至32767
(常用) int: 常用数值类型 -2147483648至2147483647
long: 长数值类型 -2^64至(2^64)-1
float: 单精度类型 1.0
(常用) double: 双精度类型 1.00
char: 整个unicode字符集 A-Z 41-59 a-z 61-79
(常用) string: 字符串类型 null
(常用) bool : 布尔类型 多用于判断 当结果为真即true 否着为false

2.java&c# 运算符

1.一元运算符

(): 优先级第一 括号
[] : 方括号
+: 加 | 正
-: 减 | 负
~: 位逻辑运算符
++: 自增运算符
--: 自减运算符
*: 乘
/: 除
%: 取模 同除法相似, 不同的是模取除不尽的数 即 10%3=1 10/3=3.3 所以模取 后面的3
<<: 2进制位左移
>>: 2进制位右移

逻辑运算

!: 非 求反
&: 和 bool类型 &&短路和
|: 或, 与 bool类型 ||短路或, 与

2.二元运算符

大部分一元运算符都可以参加二元运算
关系运算符 关系运算符返回值为布尔类型bool 一般用于判断语句
==: 相等运算符
!=: 相反运算符
<=: 小于等于
>=: 大于等于

+=: 加法赋值 a +=b 即 a = a+b
-=: 减法赋值 a -=b 即 a = a-b
*=: 乘法赋值 a *=b 即 a = a*b
/=: 除法赋值 a /=b 即 a = a/b
%=: 取模赋值 a %=b 即 a = a%b

3.三元运算符

一元和二元大部分都可进行三元运算
?: 条件运算符

```
a>=b ? a:b
当a>=b 为真的时候 取chur 即取a: b 的a
当a>=b 为假的时候 取falas 即取a: b 的b
```

4.赋值符

=: 赋予 通常用于初始化和赋值

3.a++和++a 执行顺序

```
//声明一个变量 a
int a = 10;
a = 20;
a = 2;//赋值只看最后，前面就不管
int b = 3;
int c = 5;
a = (a + b) * c;//25
int d = (++a + a++ + a - b++ + c + b);
        //25+1+26+ 26+1- 3+ 5+4=85
/*
*c++ 视为后妈 在二元计算或者以上的时候下一次计算才增加1
*++c 视为亲妈 在计算时即刻增加1，再进行运算
*c++和++c都是增加1
*%为眸，比如5%2 取眸后就5（5/2=2.5）眸取除不尽的，所以为5
*/
```

4.变量之间的关系

```
//变量类型的转换分为低精度向高精度自动转换和强制转换
byte a = 5;
double c = a + 5;//系统自动视a为byte类型，自动向double转换
byte d =(byte)( 10 + a);//系统发现a属于byte，10属于double，它们运算则
//a向高级double转换然而前面写着byte(强制转换)
//又应为（10+a）属于int类无法向byte转换，所以需要手动强制转换
double /*int改为double*/ e = c + a;//默认需要强制转换c为double
        //a为byte,需要强制转换为int
//或者改变e为double，或者更改c为int或者byte就可以自动向int转换，否着要强制转换。

//控制台输出语句 java为syso
Console.WriteLine(e);//line为控制台输入是否换行同java
console.ReadKey();//控制台等待
```

5.变量之间的转换

```
/*
*变量之间的互换
*1，增加临时变量
*2，不增加变量的互换
*a,b,c,的值为a=b,b=c,c=a.
*/
//声明3个变量 a,b,c
int a = 50;
int b = 20;
int c = 5;

//用匿名变量，即不声明变量实现 a=b,b=c,c=a
/*a = a + b + c;
c = a - b - c;
b = a - b - c;
a = a - b - c;
*/
```

```
//用声明2个变量实现 a=b,b=c,c=a
int d = b;
int e = c;
c = a;b = e;a = d;
```

6. 键盘监听

```
//控制台输出
Console.WriteLine("请输入年龄");
string a=console.readline();//得到的值赋给 a
int b=convert.toInt32(a);//把a的值转为int并赋给b
console.writeline
//把控制台得到的string转为int并加5然后控制台输出
Console.WriteLine(Convert.ToInt32(Console.ReadLine()) + 5);//3句合并为一句话
Console.ReadKey();
```

7. 流程控制结构语句

1. if if为bool类型，所有的都可以判断

```
if（条件）{//语句块1
}
else{//语句块2
}
例：
int a=0;//声明变量a，并初始化
int b=0;//声明变量b，并初始化
//当a<b为真取iaf内的语句输出a<b，否着输出eles内的a>=b
it(a<b){
System.out,println("a<b");
}
else{System.out.println("a>=b");
}
//当if下出现多个if是时，只要匹配的if都进行执行
//当if后都为 eles if时，只匹配一个执行，相当于短路语句
//if运用可以多元化，if下或者别的下都可前嵌套if
```

2. switch 为数值类判断，只可进行函数表达返回式或值，不可进行bool类

```
switch(//表达式){
case value1: //程序判定为真的时候执行 case value1的语句
break; //跳出switch，如果不写，则无条件执行下一个case，直到出现break
case value2: //程序语句
break;
case value3: //程序语句
break;
case value4: //程序语句
break;
default: //当以上case都为假时，执行default内的语句
break;
```

表达式类型:byte short int char string (加"a")

8. 随机数表达式

随机数 random

```
Random a=new Random();//声明一个随机数类型 a
int i=a.Next(9);//把随机数a 赋给i 范围为0-（9-1）
```

```
/*
int i=a.next(10)+1;//把随机数赋予i的同时加1，得到的范围为（0-（10-1））+1为1-10
int x=a.next(6)+5;//把随机数赋予x的同时加5，得到的范围为（0-（6-1））+5为5-10
*/
```

程序直接结束语句

```
return;//当执行到此语句时，所有程序立刻结束
```

9.综合学习案例

9.1输入数字判断周几

```
Console.WriteLine("请输入1-7的数字");//得到数字
string a1 = Console.ReadLine();//把得到的字符串赋给a1
int week=convert.ToInt32(a1);//把a1转换为int赋给week
string str="输入的week是:"+week+"所对应的是:";
switch(week){
case 1:
    str+="星期一";
    break;
case 2:
    str+="星期二";
    break;
case 3:
    str+="星期三";
    break;
case 4:
    str+="星期四";
    break;
case 5:
    str+="星期五";
    break;
case 6:
    str+="星期六";
    break;
case 7:
    str+="星期日";
    break;
default: Console.WriteLine("你就是个SB");
    break;
}

Console.WriteLine(str);
```

9.2输入数字判断月份 全加防误判断

```
static void Main(string[] args)
{
    /*使用键盘监听到用户输入的字符串*/
    Console.WriteLine("欢迎使用季节推荐系统");
    Console.WriteLine("请输入1-12的数字以确定月份");
    string a1 = Console.ReadLine();//声明变量a1以存储用户键入的字符串
    int a;//声明变量a，为下一步字符串转int准备
    /*if 判断当a1的值为1-12的时候，执行string a1转 int a 否着就执行else*/
    if (a1 == "1" || a1 == "2" || a1 == "3" || a1 == "4" || a1 == "5" || a1 == "6" || a1 == "7" || a1 == "8" || a1 == "9" || a1 == "10" || a1 == "11" || a1 == "12")
    {
        a = Convert.ToInt32(a1);//转string a1为int
        /*继续判断用户输入数字为几月以执行*/
        if (a == 3 || a == 4 || a == 5)
        {
            /*当a的值为3,4,5时输出*/
            Console.WriteLine("hello,您输入的月份是春季哟");
            switch (a)//判断int a的具体数字，以执行下一步
            {
                case 3:
                    Console.WriteLine("三月");
                    break;
                case 4:
                    Console.WriteLine("四月");
                    break;
                case 5:
                    Console.WriteLine("五月");
                    break;
            }
        }
    }
    else
    {
        Console.WriteLine("输入的月份不在1-12范围内");
    }
}
```

```

{
    case 3://当int a=3时 执行下面语句
        Console.WriteLine("3月天气很好, 很适合春游哟");
        break;//执行断点, 否着还会继续判断
    case 4:
        Console.WriteLine("4月, 有你关注的日子");
        break;
    case 5:
        Console.WriteLine("5月, 听说5月桂花香, 不知道是不是真的");
        break;
}
}
//判断int a的具体数字, 以执行下一步
else if (a == 6 || a == 7 || a == 8)//在多个条件判断时 else if 和if的区别为 加else相当于短路语句, 满足一个后就不执行i别的f了
{
    Console.WriteLine("hello,您输入的月份是夏季哟");
    switch (a)//switch 中的a 的值必须有结果, 不能是bool类型 a>b或a !=c 都不行
    {
        case 6:
            Console.WriteLine("6月读书好, 可惜有知了");
            break;
        case 7:
            Console.WriteLine("7月天热很正常, 早出晚归好纳凉");
            break;
        case 8:
            Console.WriteLine("8月放假好, 就是快没了");
            break;
    }
}
else if (a == 9 || a == 10 || a == 11)
{
    Console.WriteLine("hello,您输入的月份是秋季哟");
    switch (a)
    {
        case 9:
            Console.WriteLine("9月天气炎热, 建议你喝“喝一口爽半年的百事可乐”");
            break;
        case 10:
            Console.WriteLine("10月, 我的10月就只有国庆节, 你呢? ");
            break;
        case 11:
            Console.WriteLine("11月, 11你来重庆吧, 天天下雨, 哭死你");
            break;
    }
}
else if (a == 12 || a == 1 || a == 2)
{
    Console.WriteLine("hello,您输入的月份是冬季哟");
    switch (a)
    {
        case 12:
            Console.WriteLine("12月天气寒冷, 快回家歇着吧");
            break;
        case 1:
            Console.WriteLine("1月, 天啊, 一月是放寒假的日子, 886");
            break;
        case 2:
            Console.WriteLine("2月, 2月如果你是一个中国人就各回各家各找各妈吧");
            break;
    }
}
}

else//如果int a的值不能转时, 执行
{
    Console.WriteLine("哥们, 我喊你输入1-12啊, 你乱输入干嘛。。。");
}

```

```
        Console.WriteLine("自己重来");
    }

    Console.ReadKey();
}
```

9.3输入3个数判断大小 忘写注释，不喜勿喷

```
static void Main(string[] args)
{
    Console.WriteLine("请输入任意数");
    string a1= Console.ReadLine();
    double a=Convert.ToDouble( a1);
    Console.WriteLine("请输入任意数");
    string b1 = Console.ReadLine();
    double b = Convert.ToDouble(b1);
    Console.WriteLine("请输入任意数");
    string c1 = Console.ReadLine();
    double c = Convert.ToDouble(c1);
    if (a>b&&b>c)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}",a,b,c);
    }
    else if (a>b&&a<c)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}", c, a, b);
    }
    else if (a>c&&c>b)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}", a, c, b);
    }
    else if (b>a&&a>c)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}", b, a, c);
    }

    else if (b>c&&c>a)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}", b, c, a);
    }
    else if (c>b&&b>a)
    {
        Console.WriteLine("顺序为{0}>{1}>{2}", c, b, a);
    }

    else if (c == b && b==a)
    {
        Console.WriteLine("顺序为{0}={1}={2}", c, b, a);
    }

    else if (c > b && b == a)
    {
        Console.WriteLine("顺序为{0}={1}<{2}", a, b, c);
    }
    else if (c < b && b == a)
    {
        Console.WriteLine("顺序为{0}={1}>{2}", a, b, c);
    }

    else if (c == b && b < a)
    {
        Console.WriteLine("顺序为{0}={1}<{2}", b, c, a);
    }
}
```

```

else if (c == b && b > a)
{
    Console.WriteLine("顺序为{0}={1}>{2}", b, c, a);
}

else if (c == a && b < a)
{
    Console.WriteLine("顺序为{0}={1}>{2}", c, a, b);
}
else if (c == a && b > a)
{
    Console.WriteLine("顺序为{0}={1}<{2}", c, a, b);
}

Console.ReadKey();

}

```

9.4 学生成绩录入，注释很少，不喜勿喷

```

static void Main(string[] args)
{
    //声明变量储存键盘输入成绩
    Console.WriteLine("学生成绩录入系统: ");
    Console.WriteLine("请输入语文成绩: ");
    string a1 = Console.ReadLine();
    int a = Convert.ToInt32(a1);
    Console.WriteLine("请输入数学成绩: ");
    string b1 = Console.ReadLine();
    int b = Convert.ToInt32(b1);
    Console.WriteLine("请输入英语成绩: ");
    string c1 = Console.ReadLine();
    int c = Convert.ToInt32(c1);
    Console.WriteLine("请输入历史成绩: ");
    string d1 = Console.ReadLine();
    int d = Convert.ToInt32(d1);
    Console.WriteLine("请输入地理成绩: ");
    string e1 = Console.ReadLine();
    int e = Convert.ToInt32(e1);

    int p = (a + b + c + d + e) / 5; //声明为平均成绩

    Console.WriteLine("您的平均成绩为: " + p);

    if (p <= 60)
    {
        Console.WriteLine("学校给你的评价是: “差”");
        Console.WriteLine("亲你的成绩真的太菜了, 是否需要补课? ");
        Console.WriteLine("如果需要请输入“是”, 如果不需要补课请随意输入");

        Console.WriteLine(); //监听键盘输入
        string xueXi = Console.ReadLine();

        if (xueXi.Equals("是")) //判定是否补课
        {
            Console.WriteLine("请去黄老师处交补课费, 总共需要19800元");
            Console.WriteLine("提示, 请勿去非法的教师处补课");
        }
        else
        {
            Console.WriteLine("如果不补课你就去网吧玩玩吧, 为以后生存打基础");
        }
    }
}

```

```

    }
else if (p > 60 && p <= 80)
{
    Console.WriteLine("你的平均成绩为：“良”");
    Console.WriteLine("需要多多努力");
}
else if (p >= 80 && p <= 95)
{
    Console.WriteLine("你的平均成绩为：“优”");
    Console.WriteLine("亲，还需努力哟");
}
else if (p >= 95 && p <= 100)
{
    Console.WriteLine("你的平均成绩为：“优+”");
    Console.WriteLine("亲，请保持哟");
}

Console.ReadKey();
}

```

9.5 周长面积计算

```

static void Main(string[] args)
{

    Console.WriteLine("请输入a的边长");
    string bol1= Console.ReadLine();//声明一个矩形边a
    int blo1 = Convert.ToInt32(bol1);
    Console.WriteLine("请输入a的边长");
    string bol2 = Console.ReadLine();//声明一个矩形边a1
    int blo2= Convert.ToInt32(bol2);
    Console.WriteLine("请输入b的边长");
    string bol3 = Console.ReadLine();//声明一个矩形边b
    double blo3 = Convert.ToDouble(bol3);
    Console.WriteLine("请输入b的边长");
    string bol4 = Console.ReadLine();//声明一个矩形边b1
    double blo4 = Convert.ToDouble(bol4);
    //计算面积和周长

    double d = blo1 *2+ blo2*2;
    double d1 = blo3*2+blo4*2;
    double s = blo1 * blo2;
    double s1 = blo3 * blo4;
    string kouo = "a周长为=" + d + ", b周长为=" + d1 + ", a面积为=" + s + ", b面积为=" + s1;
    Console.WriteLine(kouo);
    Console.ReadKey();
}

```

9.6 逆序有游戏 关于输入的值是否为数字类型防误判断没有做，因为做不来

```

static void Main(string[] args)
{
    Console.WriteLine("请输入1-999999的数字");//声明得到数字
    int a = Convert.ToInt32(Console.ReadLine());
    if (a > 0 && a < 999999)//防误判断
    {
        Console.WriteLine("你输入的数是{0}", a);
        Console.WriteLine("请问需要逆序吗？如果需要就输入“1”，不需要就输入“2”");

        Console.WriteLine();//声明得到数字
        string m = Console.ReadLine();
        if (m == "1" || m == "2")//防误判断
    }
}

```



```

{
    int b = Convert.ToInt32(m);//把string m转int b

    switch (b)//判定b
    {
        case 1://b为1的时候
            int c;
            int d;
            int e;
            int f;
            int g;
            int h;

            c = a % 10;
            d = (a / 10) % 10;
            e = ((a / 10) / 10) % 10;
            f = (((a / 10) / 10) / 10) % 10;
            g = ((((a / 10) / 10) / 10) / 10) % 10;
            h = (((((a / 10) / 10) / 10) / 10) / 10) % 10;
            if (a < 10) Console.WriteLine("逆序后为: {0}", c);
            else if (a < 100) Console.WriteLine("逆序后为: {0}{1}", c, d);
            else if (a < 1000) Console.WriteLine("逆序后为: {0}{1}{2}", c, d, e);
            else if (a < 10000) Console.WriteLine("逆序后为: {0}{1}{2}{3}", c, d, e, f);
            else if (a < 100000) Console.WriteLine("逆序后为: {0}{1}{2}{3}{4}", c, d, e, f, g);
            else if (a < 1000000) Console.WriteLine("逆序后为: {0}{1}{2}{3}{4}{5}", c, d, e, f, g, h);
            break;
        default:
            Console.WriteLine("那就拜拜咯");
            break;
    }
}
else
{
    Console.WriteLine("请输入“1”或者“2”");
}
}

else
{
    Console.WriteLine("亲输入6位数哦");
}
Console.ReadKey();
}

```

9.7奇偶数判断

```

static void Main(string[] args)
{
    Console.WriteLine("亲，输入一个数，我就知道是奇数还是偶数哦");
    long a = Convert.ToInt64(Console.ReadLine());
    if (a%2==0)
    {
        Console.WriteLine("这个数为偶数哦");
    }
    else
    {
        Console.WriteLine("这个数为奇数哦");
    }
    Console.ReadKey();
}

```

9.8人物角色平均战斗力

```

static void Main(string[] args)

```

```

{
    string Name;//声明人物名称
    string Id;//任务职业
    int mxa;//人物等级
    int Ad;//物理攻击
    int Ap;//魔法攻击
    int Sd;//移动速度
    int Hp;//生命值
    int Mp;//魔法值
    int Wf;//物理防御
    int Mf;//魔法防御

    Console.WriteLine("请输入人物名称:");//键盘监听
    Name = Console.ReadLine();//监听结果的值赋给Name
    Console.WriteLine("请输入你想要的职业");
    Id = Console.ReadLine();
    Console.WriteLine("请输入你想要的等级");//键盘监听
    string mxa1 = Console.ReadLine();//监听值赋给mxal，用string是为了更好的保存所有类型输出
    mxa = Convert.ToInt32(mxa1);//把string类型转为int类，方便计算平均值
    Console.WriteLine("请输入你想要的物理攻击");
    string Ad1 = Console.ReadLine();
    Ad = Convert.ToInt32(Ad1);
    Console.WriteLine("请输入你想要的魔法攻击");
    string Ap1 = Console.ReadLine();
    Ap = Convert.ToInt32(Ap1);
    Console.WriteLine("请输入你想要的移动速度");
    string Sd1 = Console.ReadLine();
    Sd = Convert.ToInt32(Sd1);
    Console.WriteLine("请输入你想要的hp");
    string Hp1 = Console.ReadLine();
    Hp = Convert.ToInt32(Hp1);
    Console.WriteLine("请输入你想要的mp");
    string Mp1 = Console.ReadLine();
    Mp = Convert.ToInt32(Mp1);
    Console.WriteLine("请输入你想要的wf");
    string Wf1 = Console.ReadLine();
    Wf = Convert.ToInt32(Wf1);
    Console.WriteLine("请输入你想要的mf");
    string Mf1 = Console.ReadLine();
    Mf = Convert.ToInt32(Mf1);
    Console.Clear();//清除控制台显示类容
        //Convert.ToInt32();
    int pingGun = (Ad + Ap + Sd + Hp + Mp + Wf + Mf) / 7;//声明变量保存平均值
    Console.WriteLine("人物名称: " + Name);
    Console.WriteLine("人物职业: " + Id);
    Console.WriteLine("人物等级: " + mxa);
    Console.WriteLine("您的综合战斗力为: {0}", pingGun);

    /*("nnnnnn{0}{1}{2}{3}",a,c,f,g)此方法为占位符*/

    Console.ReadKey();
}

```

9.9输入秒数得到具体年月

```

static void Main(string[] args)
{
    Console.WriteLine("请输入您要测试的秒数: ");
    string s1 = Console.ReadLine();

    long s2 = Convert.ToInt64(s1);//声明时间秒

    long h = s2 / 3600;//声明得到的小时数

```

```

long m1 = s2 % 3600;//得到的分（秒）
long m = m1 / 60;//得到的分
long s = m1 % 60;//得到的秒
long day = h / 24;//得到天数
long yue = day / 30;//得到月数
long lian = yue / 12;//得到年数

if (day != 0) h = (h - day * 24);
if (yue != 0) day = (day - yue * 30);
if (lian != 0) yue = (yue - lian * 12);
if (day == 0) Console.WriteLine("时间为: " + h + "小时" + m + "分" + s + "秒");
else if (day > 0 && yue == 0 && lian == 0) Console.WriteLine("时间为: " + day + "天" + h + "小时" + m + "分" + s + "秒");
else if (yue > 0 && lian == 0) Console.WriteLine("时间为: " + yue + "月" + day + "天" + h + "小时" + m + "分" + s + "秒");
else Console.WriteLine("时间为: " + lian + "年" + yue + "月" + day + "天" + h + "小时" + m + "分" + s + "秒");

Console.ReadKey();
}

```

10 循环语句

1.while循环

1.1 while（条件）{循环体}

条件:可以使**bool**（布尔）类型的值.变量和表达式,还可以是一个结果为**bool**（布尔）的方法

```

int i=1;//初始化循环条件
//循环结束条件，当结束条件为空则死循环
while(i<10){
System.out.println("第"+i+"次循环");
i++;//循环增量
}

```

1.2 do-while:当我们需要循环执行至少一次的时候,及时表达式的值为**false**（假）,do-while也可以执行一次，但是**while**是不能执行的。

```

do{
//循环体
}while( 条件 )

int i=11;
do {
System.out.println("第"+i+"次循环");
i++;
}while(i>10);

```

2. for循环语句

for（初始化①;循环结束条件②;迭代运算③;）{循环体④}

结束条件:必须是**boolean**（布尔）表达式

执行过程:①--②---如果条件为真，则--④--③--②---如果条件为真.....
---如果条件为假，则直接跳出循环，执行后面的语句

流程控制之**break.continue.return**的用法

1.1 break

break:强制当前循环终止

break:跳出当前循环，跳出循环后继续执行后面的代码

```
for(int i=1;i<10;i++){

    System.out.println("这是外循环,当前循环第"+i+"次");

    for(int j=1;j<3;j++){

        System.out.println("这是内循环,当前循环第"+j+"次");

        break;

    }

}
```

1.2 continue

continue:停止本次循环,继续执行剩下的循环。

```
for(int i=0;i<10;i++){
    if(i==6){
        continue;
    }
    System.out.println(i);
}
```

1.3 return

从当前的方法中跳出，# 结束 #所有代码

```
for (int i = 0;i<10;i++) {
    if (i==6) {
        return;
    }
    System.out.println(i);
}
```

11. 加循环综合案例

11.1 1-100的总和2种方法 for

```
static void Main(string[] args)
{
    /*
    *计算1-100的和
    */

    /* int sum = 0; //声明变量 用于存放数字的和
    for (int i = 1; i <= 100; i++)
        sum +=i; //等价于语 old=old+score

    Console.WriteLine("总和"+sum); */
    int sum = 0; //声明变量1-100之和
    for (int i = 1, j = 100; i < j; i++, j--)
    {
        sum += i + j; //等于语 求和

    } //用1+100 2+99 3+98 ....50+51的方法求和
    Console.WriteLine(sum);
}
```

```
Console.ReadKey();
}
```

11.2 输入一个数，求数的和并算出6的倍数 for+while

```
static void Main(string[] args)
{

    Console.WriteLine("输入任意数");
    long a = Convert.ToInt64(Console.ReadLine()); //得到a的值
    long b = 0; //声明变量以存储总值
    long c = 0; //循环变量

    /* for (int i = 0; i <= a ; i++)
    {
        b = i + b;
    } */
    while (c < a) //循环判定c等于大于a的时候停止循环
    {
        c++; //c=c+1
        b = b + c; //得到c+1+b赋给b，然后在计算
        // b += c;

    }
    Console.WriteLine("{0}内的数相加为{1}", a, b);

    long d = b / 6;
    Console.WriteLine("其中是6的倍数有{0}个", d);
    Console.WriteLine("是否显示6的倍数？请输入“1”，否请输入“2”");
    Console.WriteLine();
    string s1 = (Console.ReadLine());
    if (s1 == "1" || s1 == "2")
    {
        int s = Convert.ToInt32(s1);

        switch (s)
        {
            case 1:
                long q = 0; //循环变量
                while (q < b) //循环到q! => b时停止
                {
                    q++;
                    if (q % 6 == 0)
                    {
                        Console.Write(q + ",");
                    }

                }
                break;
            case 2:
                Console.WriteLine("那就按任意键拜拜吧");
                break;
            default:
                Console.WriteLine("请正确输入哟");
                break;

        }
    }
    else
    {
        Console.WriteLine("请正确输入哟");
    }
    Console.ReadKey();
}
```

11.3 石头剪刀布游戏 for+while 以加防误判断 循环执行

```
static void Main(string[] args)
{
    string q = null; //声明变量用于判定第一个for循环是否继续的
    for (;;) //死循环, 循环结束条件2为空
    {
        int w = 0; //声明变量储存3次的输赢值
        for (int i = 0; i < 3; i++)
        {
            Console.WriteLine("请输入“0”石头或“1”剪刀或“2”布");
            string a1 = Console.ReadLine(); //获得键盘输入的数字存储在a1
            int a = 0; //声明变量以存储a1的值
            int b = 0; //声明变量以存储计算随机出的值

            if (a1 == "0" || a1 == "1" || a1 == "2") //防误判断, 输入别的字符不通过
            {
                a = Convert.ToInt32(a1); //把string a1转换为int a
                switch (a) //判定a的具体数字
                {
                    case 0:
                        /*当a==1的时候执行一下*/
                        Console.WriteLine("您出的是石头");
                        break;
                    case 1:
                        Console.WriteLine("您出的是剪刀");
                        break;
                    case 2:
                        Console.WriteLine("您出的是布");
                        break;
                }
                Random b1 = new Random(); //new 一个随机数
                b = b1.Next(3);
                switch (b)
                {
                    case 0: //当计算机输出的为0的时候
                        Console.WriteLine("计算机出的是石头");
                        if (b == a)
                        {
                            Console.WriteLine("哇哦, 你真厉害, 和计算机打成了平手");
                            w += 0; //w=w+0
                        }
                        else if (a > 0 && a > 1)
                        {
                            Console.WriteLine("666, 你居然把计算机都打败了");
                            w = w + 1;
                        }
                        else
                        {
                            Console.WriteLine("嘿嘿, 你输给计算机了哟");
                            w -= 1; //w=w-1
                        }
                        break;
                    case 1: //当计算机输出的为1的时候
                        Console.WriteLine("计算机出的是剪刀");
                        if (b == a)
                        {
                            Console.WriteLine("哇哦, 你真厉害, 和计算机打成了平手");
                            w = w + 0;
                        }
                        else if (a < 1)
                        {
                            Console.WriteLine("666, 你居然把计算机都打败了");
                            w += 1;
                        }
                }
            }
        }
    }
}
```

```

        }
        else
        {
            Console.WriteLine("嘿嘿，你输给计算机了哟");
            w -= 1;
        }
        break;
    case 2://当计算机输出的为2的时候
        Console.WriteLine("计算机出的是布");
        if (b == a)
        {
            Console.WriteLine("哇哦，你真厉害，和计算机打成了平手");
            w = w + 0;
        }
        else if (a < 2 && a > 0)
        {
            Console.WriteLine("666,你居然把计算机都打败了");
            w += 1;
        }
        else
        {
            Console.WriteLine("嘿嘿，你输给计算机了哟");
            w -= 1;
        } break;
    }

}

else//防误判断，当输入不通过时执行
{
    Console.WriteLine("你的机会浪费一次，下次请正确输入");
}

}

if (w < 0) Console.WriteLine("不好意思，3次过后你总体来说“您输了”");
else if (w == 0) Console.WriteLine("运气不错，3次过后你总体来说“是平局”");
else Console.WriteLine("恭喜恭喜，3次过后你总体来说“您赢了”");
Console.WriteLine("你的3次机会已用完，如果还想继续玩请在汇众精英交流群发0.1元红包");
Console.WriteLine("只需0.1元就可以玩三次哟");
Console.WriteLine("如果已发过红了，就按“yes”就可以继续玩3次了，不想玩就随意输入退出吧");
string q1 = Console.ReadLine();//监听后把值赋给q1
if (q1=="yes")//防误判断，当结果为真的时候执行以下
{
    q = "yes";//结果为真，把yes赋给q
}
else//当q的值为null的时候，程序结束
{
    return;
}

}

Console.ReadKey();
}

```

11.*** 经典3角型打印

正

```

for (int i = 1; i <= 9; i++)
{
    for (int b = 1; b <= i; b++) { Console.Write("☆"); }

    Console.WriteLine();
}

```

反

```

for (int i = 1; i <= 9; i++)
{
    for (int b = 9; b <= i; b--) { Console.WriteLine("☆"); }

    Console.WriteLine();
}

```

11.4 最笨的图型打印

```

for (int i = 1; i <= 14; i++)//一共七行
{ //每一列打印13次从1到13，因为条件限制j=7的时候打印@，其余打印""
    for (int j = 1; j <= 14; j++)//一共13列
    { //判断每一列打印（每列打印13次），达成if条件时打印@，否则打印""
        if (i == 1 & j == 8
            | i == 2 & (j == 7 | j == 8 | j == 9)
            | i == 3 & (j == 6 | j == 7 | j == 9 | j == 10)
            | i == 4 & (j == 5 | j == 6 | j == 10 | j == 11)
            | i == 5 & (j == 4 | j == 5 | j == 11 | j == 12)
            | i == 6 & (j == 3 | j == 4 | j == 12 | j == 13)
            | i == 7 & j != 1 | i == 8 & j != 14
            | i == 9 & (j == 2 | j == 3 | j == 11 | j == 12)
            | i == 10 & (j == 3 | j == 4 | j == 10 | j == 11)
            | i == 11 & (j == 4 | j == 5 | j == 9 | j == 10)
            | i == 12 & (j == 5 | j == 6 | j == 8 | j == 9)
            | i == 13 & (j == 6 | j == 7 | j == 8)
            | i == 14 & j == 7)

        {
            Console.WriteLine(" @ "); //去除line 在同一行打印
        }
        else
        {
            Console.WriteLine(" "); //去除line 在同一行打印
        }
    }
    Console.WriteLine(); //换行操作
}
Console.ReadKey();

```

11.5 最笨的金字塔打印

```

for (int i = 1; i <= 7; i++)//一共七行
{ //每一列打印13次从1到13，因为条件限制j=7的时候打印@，其余打印""
    for (int j = 1; j <= 13; j++)//一共13列
    { //判断每一列打印（每列打印13次），达成if条件时打印@，否则打印""
        if (i == 1 & j == 7
            | i == 2 & (j == 6 | j == 7 | j == 8)
            | i == 3 & (j == 5 | j == 6 | j == 7 | j == 8 | j == 9)
            | i == 4 & (j == 4 | j == 5 | j == 6 | j == 7 | j == 8 | j == 9 | j == 10)
            | i == 5 & (j != 1 & j != 2 & j != 12 & j != 13)
            | i == 6 & (j != 1 & j != 13)
            | i == 7)

        {
            Console.WriteLine(" @ "); //去除line 在同一行打印
        }
        else
        {
            Console.WriteLine(" "); //去除line 在同一行打印
        }
    }
    Console.WriteLine(); //换行操作
}

```



```
Console.ReadKey();
```

11.6 金典金字塔打印 手动输入行数和正反判定

```
Console.WriteLine("请输入行数");
int cos = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("需要打印的三角形为正还是反，请输入“正”或者“反”");
string p = Console.ReadLine();
if (p == "正")
{
    for (int i = 1; i <= cos; i++)//控制的一共几行
    {
        for (int a = cos; a >= i; a--) { Console.Write(" "); }//第一个打印5个空格递减
        //在第二个for 内 ☆旁边加空格，把三角形挤成 等腰梯形
        for (int b = 1; b <= i; b++) { Console.Write(" ☆"); }//第二个在5空格后打印一个☆递加
        // for (int c = 1; c < i; c++) { Console.Write("☆"); }//第二

        Console.WriteLine();
    }
}
else if (p == "反")
{
    for (int i = 1; i <= cos; i++)
    {
        for (int b = 1; b <= i; b++) { Console.Write(" "); }
        for (int a = cos; a >= i; a--) { Console.Write(" ☆"); }

        Console.WriteLine();
    }
}
else { return; }

Console.ReadKey();
```

11.7 for 拓展 买鸡训练 无注释

公鸡 3元 母鸡 2元 小鸡 1元3只 问100元如何正好花完

```
int a, b, c;
for (a = 1; a < 33; a++)
{
    for (b = 1; b < 50; b++)
    {
        for (c = 1; c <= (100 - a - b); c++)
        {
            if (100 == 3 * a + 2 * b + c / 3)
            {
                Console.WriteLine("有" + a + "只公鸡" + b + "只母鸡" + c
                    + "只小鸡");
            }
        }
    }
}
Console.ReadKey();
```

11.8 for 拓展 搬砖训练

男人 4块 女人 3块 2小孩 1块 问36块砖 36个人 如何分配一次搬完

```
int a, b, c;
```

```

for (a = 0; a <= 9; a++)//男人最大9个
{
    for (b = 0; b <=12; b++)//女人最大12个
    {
        for (c = 0; c <= (36 - a - b); c++)//小孩最多为36-男-女
        {
            if (4 * a + 3 * b + c / 2 == 36&&36==a+b+c)//控制男女小孩==36个人，且他们一共要搬36块砖
            {
                if (c % 2 != 1)//小崽子不能为单数
                {
                    Console.WriteLine("{0}男{1}女{2}小孩", a, b, c);
                }
            }
        }
    }
}
Console.ReadKey();

```

11.9 for 拓展 9*9乘法表

"\t"为制表符 即为空格

```

for (int i = 1; i <=9; i++)//9行
{
    for (int j = 1; j <=i; j++)//列随着行++，列每次++
    {
        Console.Write("{0}*{1}={2}\t",j,i,(i*j));//    \t为制表符 就是空格
    }
    Console.WriteLine();
}
Console.ReadKey();

```

12 函数基础

I.提高代码的赋用性

写函数必要的三个思考

- 1.需要返回值么？
- 2.需要未知元素参加么？
- 3.思考参与元素类型和返回值类型 和 函数返回结果的处理有函数调用者处理和函数本身无关

II.函数如何使用？

- 1.c#是通过主函数（main方法）进行调用
- 2.如何调用函数： 函数名和（）；有有参数传入参数 并注意参数类型
- 3.函数和函数之间的关系？函数和函数一般是调用关系

III.函数的重载

- 1.函数可以重命名么？可以但是必须是函数重载
- 2.函数重载和函数不同在于哪儿？参数数量，参数类型，参数顺序。和函数返回值类型无关
- 3.重载函数可以进一步提高代码的扩展性和复用性

12.1 例

```

static void Main(string[] args)
{
    getS(1,2);
}

```

```

        getS(1.0, 2);
        getS(1, 2, 2);
        Console.ReadKey();
    }
    static double getS(double down, double height)
    { // 三角形面积计算
        double s = down * height / 2;
        Console.WriteLine("三角形面积为{0}", s);
        return s;
    }
    static double getS(int wihti, int height)
    { // 长方形面积计算
        double s = wihti * height;
        Console.WriteLine("长方形面积为{0}", s);
        return s;
    }
    static double getS(double down, double up, double height)
    { // 梯形面积计算
        double s = (up + height) * height / 2;
        Console.WriteLine("梯形面积为{0}", s);
        return s;
    }
}

```

13 数组

存储在堆里的一组数组 数组是进行索引 进行 增删改查

空指针异常是指 引用方出现为空 数组始终存在 即在栈内存内建立一个数组名然后指向堆内存内的数组

13.1 创建数组的3个方法

```

数组类型 +[]+数组名=new+ 数组类型[数组长度length]{数组初始化值}
int +[]+array=new int[9] //直接new一个未初始化的数组
int +[]+array=new int[] {1,2,3,4} //new 一个数组 声明长度 并赋值
int +[]+array={1,2,3,7,4,0} //声明一个数组并初始化

```

13.2 直接打印数组的话是一个哈希值 数组最后一个的角标为 length-1

13.3 数组如果没有赋值的话也可以使用 int类默认为0,string类为null, bool为false

13.4 数组工具类 案例1

数组生成

```

static int[] caar(int a)
{ // 数组生成
    int[] num = new int[a]; // 声明一个未初始化数组
    for (int i = 0; i < num.Length; i++)
    {
        num[i] = i;
    }
    return num;
}

```

数组遍历

```

static void prArray(int [] array)
{ // 数组遍历

```

```

Console.WriteLine("开始遍历数组");
Console.Write("["); //打印[
//数组循环打印次数为数组长度
for (int i = 0; i < array.Length; i++)
{
    //当打印数组最后一个时，打印以下
    if (i == array.Length - 1)
    { //打印数组最后一个值+】
        Console.Write(array[i] + "】");
    }
    else
    { //打印出每个数组的值+，
        Console.Write(array[i] + ",");
    }
}
Console.WriteLine("数组遍历结束");
Console.ReadKey();
}

```

数组复制

```

static int[] printArrey(int[] arrey) //数组复制
{
    int[] arrey1 = new int[arrey.Length];
    arrey1 = arrey;

    return arrey1;
}

```

数组手动输入赋值

```

static string[] prclay() {
    //内存在堆内存存着
    Console.WriteLine("输入数组长度");
    int length = Convert.ToInt32(Console.ReadLine());
    string[] a = new string[length]; //声明string类型的数组，长度为10
    for (int i = 0; i < a.Length; i++)
    {
        Console.WriteLine("请存入第{0}角标的值", i);

        a[i] = Console.ReadLine();
        Console.Clear();
    }

    Console.ReadKey();
    return a;
}

```

找出数组最大值

```

static int getMax(int[] a)
{ //把数组最大值找出来并返回

    int sum = 0; //声明变量以存储最大值

    for (int i = 0; i < a.Length; i++)
    { //遍历，sum>第一个角标的值的时候等于本身 否者等于大于的角标值
        if (sum > a[i]) sum = sum;
        else sum = a[i];
    }

    return sum;
}

```

数组逆序

```
static int[] prxxoo(int[] a)
{
    //数组逆序
    int[] b = new int[a.Length]; //声明一个空数组，用于储存逆序后的数组
    for (int i = a.Length - 1, j = 0; i >= 0 & j < a.Length; i--, j++)
    {
        b[j] = a[i];
    }

    //返回值为int[] b
    return b;
}
```

二维数组便历

```
static void bianLi(int[][] a)
{
    for (int i = 0; i < a.Length; i++)
    {
        Console.WriteLine("[");
        for (int j = 0; j < a[i].Length; j++)
        {
            Console.Write(a[i][j] + " ");
        }
        Console.WriteLine("]");
        Console.WriteLine();
    }
}
```

13.5 名字匹配并输出角标

```
static void getChacun(string[] a)
{
    Console.WriteLine("输入要查询人的名字");
    string b = Console.ReadLine();
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i].Equals(b))
        {
            Console.WriteLine("true");
            Console.WriteLine("按任意键退出");
            Console.WriteLine("角标为a[{0}]", i);

            Console.ReadKey();
        }
        else
        {
            Console.WriteLine("找不到他【fals】");
            Console.ReadKey();
            return;
        }
    }
}
```

char 方法判断键盘输入的大小写和字符

```
public class Test {
    public static void main(String[] args) {
        //new 一个扫描仪
        Scanner scanner = new Scanner(System.in);
        System.out.println("输入字符:");
    }
}
```

```

        //获得键盘输入的字符串，并赋给str
        String str = scanner.next();
        //把字符串转换为char数组内的值的方法
        char[] ch = str.charAt(0);
        //判断char中角标下的char数字，得出大小写
        if (ch[0] >= 'A' && ch[0] <= 'Z') {
            System.out.println("输入字符是大写");
        } else if (ch >= 'a' && ch <= 'z') {
            System.out.println("输入字符是小写");
        }
        //ps: 本文没有写for循环，只是片段示例代码!
    }
}

```

14.数组逆序

14.1方法一

```

static int[] nixu(int[] a)
{
    int[] b = new int[a.Length];
    for (int i = 0, j = a.length - 1; i < a.length & j >= 0; i++, j--)
    {
        b[i] = a[j];
    }
    return b;
}

```

14.2方法二

```

static int[] nixu(int[] a)
{
    int[] b = new int[a.Length];
    for (int i = 0; i < a.Length; i++)
    {
        b[i] = a[a.Length - 1 - i];
    }
    return b;
}

```

*****程序延时输出

using System.Threading; //把这一行添加进去

Thread.Sleep(200); //间隔200毫秒

15.学生成绩统计表（待修改）

```

static void xx(string[] a, int[] b)                //因为要用到两个数组，所以全部传进来
{
    string[] x = new string[a.Length];
    for (int i = 0; i < a.Length; i++)
    {
        if (b[i] == 100)                            //判定满分结果
        {
            x[i] = a[i] + "满分";                    //名字+判定结果
        }
        else if (b[i] >= 60 & b[i] < 100)            //判定及格结果
        {

```

```

        x[i] = a[i]+"及格";                //名字+判定结果
    }
    else
    {
        x[i] = a[i]+"考的西撒! ";          //剩下的就是王尼玛这个辣鸡，居然不及格
    }
}
shuchu(x);                                //输出最后的结果，这个函数不用我再复制一次了吧.....
}

```

数组冒泡算法

Arrey.rost(一个数组);//系统封包好的，直接调用 数组排序

15.冒泡算法 比邻比较

```

static void  getMax(int[] a)
{
    int temp;//声明一个临时变量
    for(int i=0; i<a.length-1;i++)//减1是防止内部循环超出界限
    {
        for(int j=i+1;j<a.length;j++)//每次i和i+1比较
        {
            //if判断i和j的值 当i>j的时候 把小的赋给前面，大的赋给后面
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
}

```

16.打擂比较

```

static void  getMax(int[] a)
{
    int temp;//声明一个临时变量
    for(int i=1; i<a.length;i++)//1
    {
        for(int j=0;j<a.length-1;j++)//每次i和i+1比较
        {
            //if判断j和j+1的值 当j>j+1的时候 把小的赋给前面，大的赋给后面
            //内部每次从j和j+1比较
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}

```

17.增强for循环

只做寻找数组和数组相关的操作 而不做其他的

```

static void Main(string[] args)
{
    int[] a = { 1, 3, 5, 7, 9 };
}

```

```

        int b=0;
//for 循环的加强 直接以数组角标0开始循环至角标结束
//此处i和for的i不一样，for循环的i是循环的条件必备语句，而此处的i是代表数组角标
        foreach (var i in a)
        {
            b += i;//此处i是代表数组内角标上的值
        }
        Console.WriteLine(b);
        Console.ReadKey();
    }

```

18.二维数组

18.1二维数组的声明

```

int[][] a = new int[0][];//前面是行，后面是列，行必须写长度

int[,] a1 = new int[1, 2];//用逗号分割行和列

//交错数组 每一个列的长度不一样 必须用new关键字
int[][] a3 = { new[] { 1, 2, 3 }, new[] { 1, 2, 3, 4 },new[] {1,2} };
//或者这样写交错数组
int[] a={1,2};int[] b={1,3,3};int[] c={1,2,3,4};int[] d={1,4,2,3,5};
int[][] array={a,b,c,d};

```

18.2二维数组的和

```

static double print(int[][] a)
{
    //得到数组的和
    double tepm = 0;//声明临时变量

    for (int i = 0; i < a.Length; i++)
    {
        for (int j = 0; j < a[i].Length; j++)
        {
            tepm += a[i][j];
        }
    }
    return tepm;
}

```

18.3二维数组求指定组（b）的平均值

```

static double print(int[][] a, int b)
{
    //得到数组的和
    double tepm = 0;//声明临时变量

    for (int i = 0; i < a.Length; i++)
    {
        for (int j = 0; j < a[i].Length; j++)
        {
            tepm += a[i][j];
        }
    }
    tepm = tepm / a[b - 1].Length;//得到指定组的平均值
    return tepm;
}

```

19 学生管理系统案例（java版） c_sharp版以封包处理

```

public static void main(String[] args) {

```



```

System.out.println("你好，欢迎进入成绩录入系统");
String[] a = prclay(); // 名字
@SuppressWarnings("unused")
int b = a.length; // 人数

String[] keMu = prclay1(); // 科目
float[][] keMucji = prclay1(a, keMu); // 每个人的科目成绩
double[] pinJun = prdouble(keMucji); // 每个人的平均分

double[] zhongfen = prdouble1(keMucji); // 每个人的总分

paiMing(pinJun, keMucji, zhongfen, a);

// 打印成绩单
prArray(keMucji, keMu, a, pinJun, zhongfen);
guoXian(zhongfen, a);

System.out.println();
}

static String[] prclay() {
    // 手动输入赋值数组
    // 内存在堆内存着
    @SuppressWarnings("resource")
    Scanner scanner = new Scanner(System.in);
    System.out.println("输入参加测试数学生的人数");
    int length = scanner.nextInt();
    String[] a = new String[length]; // 用于存储学生名字

    for (int i = 0; i < a.length; i++) {
        @SuppressWarnings("resource")
        Scanner scaneer = new Scanner(System.in);
        System.out.printf("请存入第%s个人的名字", i + 1);

        a[i] = scaneer.next();
    }

    return a;
}

static String[] prclay1() {
    @SuppressWarnings("resource")
    Scanner scanner = new Scanner(System.in);
    System.out.println("请输入测试的科目总数");
    int kk = scanner.nextInt();
    String[] keMu = new String[kk];
    for (int i = 0; i < kk; i++) {
        @SuppressWarnings("resource")
        Scanner Scanner1 = new Scanner(System.in);
        System.out.printf("请存入测试的第%s个科目", i + 1);
        keMu[i] = Scanner1.next();
    }

    return keMu;
}

static float[][] prclay1(String[] name, String[] keMu) {
    // 手动输入赋值数组
    // 内存在堆内存着

    int length = name.length;
    int length1 = keMu.length;

    float[][] a = new float[length][length1]; // 用于保存学生的每科成绩

    for (int i = 0; i < name.length; i++) {

```

```

        for (int j = 0; j < keMu.length; j++) {
            @SuppressWarnings("resource")
            Scanner xxoo = new Scanner(System.in);
            System.out.printf("请存入%s的%s成绩", name[i], keMu[j]);

            a[i][j] = xxoo.nextFloat();
        }
    }

    return a;
}

static float[] prclay(int b, String[] c) {
    // 手动输入赋值数组
    // 内存在堆内存着

    int length = b;

    float[] a = new float[length]; // 用于存储学生成绩

    for (int i = 0; i < a.length; i++) {
        @SuppressWarnings("resource")
        Scanner xxoo = new Scanner(System.in);
        System.out.printf("请存入%s成绩", c[i]);
        a[i] = xxoo.nextFloat();
    }

    return a;
}

static void prArray(float[][] array, String[] a, String[] b, double[] c,
    double[] d) { // 数组遍历

    System.out.println("成绩打印, 排名按总分计算");

    for (int i = 0; i < array.length; i++) {
        System.out.printf("第%s名", i + 1); // 打印名次
        System.out.print(b[i] + ":" + "\t"); // 打印名字

        for (int j = 0; j < array[i].length; j++) {
            System.out.print(a[j] + ":" + "\t"); // 打印科目
            System.out.print(array[i][j] + "分" + "\t"); // 打印科目分数
        }

        System.out.print("总分: " + d[i] + " " + "\t"); // 打印总分
        System.out.print("平均分: " + c[i] + "\t"); // 打印平均分

        System.out.println();
    }
}

static void prArray(String[] array) { // 数组遍历
    System.out.println("开始遍历");

    // 数组循环打印次数为数组长度
    for (int i = 0; i < array.length; i++) {
        // 当打印数组最后一个时, 打印以下
        System.out.print(array[i] + " ");
        System.out.println();
    }
}
}

```

```

// 得到平均值
static double[] prdouble(float[][] a) {
    double[] tepm = new double[a.length]; // 声明临时变量，用于保存平均值
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[i].length; j++) {
            tepm[i] += a[i][j];
        }
    }

    for (int i = 0; i < tepm.length; i++) {
        tepm[i] = tepm[i] / a[i].length;
    }
    return tepm;
}

static double[] prdouble1(float[][] a) {
    double[] tepm = new double[a.length]; // 声明临时变量，用于保存平均值
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[i].length; j++) {
            tepm[i] += a[i][j];
        }
    }
    return tepm;
}

static void paiMing(double[] pinJun, float[][] zhiLiao, double[] zhongFen,
    String[] name) {
    // 冒泡排序 比邻排序
    float[] temp; // 临时变量

    double temp1; // 临时变量

    String temp2; // 临时变量

    double temp3; // 临时变量

    for (int i = 0; i < pinJun.length - 1; i++) // -1是防止内部for角标值超出界限
    {
        for (int j = i + 1; j < pinJun.length; j++) // 相当于0和1比 1和2比
        {
            if (pinJun[i] < pinJun[j]) // 当0>1时
            { // 总的来说 外部for循环一次内部循环一组
                temp3 = pinJun[i];
                pinJun[i] = pinJun[j];
                pinJun[j] = temp3;

                temp = zhiLiao[i];
                zhiLiao[i] = zhiLiao[j];
                zhiLiao[j] = temp;

                temp1 = zhongFen[i];
                zhongFen[i] = zhongFen[j];
                zhongFen[j] = temp1;

                temp2 = name[i];
                name[i] = name[j];
                name[j] = temp2;
            }
        }
    }
}

static void guoXian(double[] a, String[] b) {
    System.out.println("关于分数提档问题，达到180分可以提档");
    for (int i = 0; i < a.length; i++) {

```

```
    if (a[i]>180)System.out.printf("%s的分数以可以提档了哟",b[i]);
        else System.out.printf("%s的分数以距离提档还有点距离哟",b[i]);
    System.out.println();
}
```

20.面向对象基础

函数进阶思想-函数就是最小单元封装，内部黑箱（具体的执行过程私有化）-外部查询结果

```
int a=55;
float b=a;(把a装箱)
float c=b;(把b拆箱)
float c = float.Parse("字符串");//调用parse方法把字符串转float
string v = Convert.ToString(int);//调用convert函数把int转为string
```

对象就是忽视实现的细节，重视结果。

面向对象思维的转折

I.JAVA 万物皆对象 C# 思想的转变 就是所谓类 所谓对象

II.类就是概念的集合 对象就是概念集合的实例化

III.面向对象里把类当做图纸，把对象当做示例 类就是模板 实例就是 我们操作和使用研究目标

类的关键字

```
class
class Uuiuls关键字写出来的; 对象 new uiuls();
```

描述事物2种即类 方式

1. 静态 字段（属性） 成员
2. 动态 函数（方法） 行为

字段的封装

- 1.在使用private关键字修饰成员时为私有化成员
- 2.必须使用Get（取得） Set（赋予） 方法来实现

例：

```
private int age;//私有化年龄age
public void Age()
{
    Get {return age}
    Set {age=value}
}
```

ps: 简写方法 public void Age(){ Get; Set; }

类的来源和实现

我们描述1个事物，思想来源于现实世界
生物都具有，自己特征
即我们class 的类里面一般包含（静态（属性），动态（行为））

自定义的类 对象的具体调用方法

```
Uiuls a = new Uiuls();
```

“=”左右侧的划分 右侧是对象，左侧是对象的引用
只要是用new 关键字创建的 实体就是对象

构造函数

- 1.在new一个类中 除了属性和行为 其默认有一个无参构造函数
- 2.构造函数就是 函数命名和类名完全相同
- 3.构造函数可以有未知元素参加 即重载多参构造函数

21.类的继承

I.继承的提出

- 1.我们创建一个类 其他类以此类为模版
- 2.其他类和继承的父类可以差异化
- 3.继承后的类可以把属性和行为传递下去

II.类的关系

- 1.类和类之间的关系可以是调用 也可以是继承
- 2.子类继承父类 父类继基类 子类就是父类的派生类
- 3.obejie是所有类的基类

III.属性与修饰符

private : 私有成员，在类的内部才可以访问。
protected : 保护成员，该类内部和继承类中可以访问。
public : 公共成员，完全公开，没有访问限制。
internal: 在同一命名空间内可以访问。

ps: 属性是一个方法或一方法，但在调用它的代码看来，它是一个字段
ps: 对于私有化属性的命名规范 名字前加短线 即 private _temp

IV.方法重写

I.virtual 父类的函数加入关键字 虚方法
II.override 子类的函数加入关键字 方法重写或者叫方法覆盖

ps: 方法的重写 系统自动完成复写 复写的方法 : 必须保持 方法签名一致
ps: 不写override 但是函数名相同也可以方法重写 因为默认使用new 关键字

示例：

父类：

```
class ParClass //父类ParClass
{
    protected string _textVal = ""; //注意修饰符为protected

    public ParClass(string textVal)//多参构造函数
    {
        this.textVal = textVal;
    }
    public ParClass()//无参构造函数
    {
        this.textVal = "not name";
    }
}
```

```
}

public virtual cat()//虚方法
{
    Console.WriteLine("吃东西的虚方法")
}
}
```

子类:

```
class SubClass : ParClass //继承格式就是子类: 父类
{
    //重写的cat方法 用的override
    public override Cat() {
    }

    //子类构造方法需要调用父类同样参数类型的构造方法, 用base关键字代表父类
    public SubCalss(string textVal) : base(textVal)
    {
    }
}

ps: 继承的优点是能使得所有子类公共的部分都放在了父类, 使得代码得到共享。
ps: 缺点是父类变则子类不得不变, 另外继承会破坏包装, 父类实现细节暴露给子类。
```

ps: baes的作用域是指向父类空间

22.单例设计模式

即某一个类new出的对象都是同一个对象即单例设计模式 2种方法实现

- 1.私有化构造函数
- 2.创建私有的静态对象
- 3.创建获取对象的方法

I. 饿汉式

```
//声明一个本类的对象 并私有化
private Student (本类对象) a = new Student();

//声明一个共有的 静态的 返回值为本类对象的方法
public static Student Instance ()
{
    rutrun a; //返回私有化的本类对象
}
```

II. 懒汉式

```
//声明一个本类为空的对象 并私有化
private Student a = null ;
//声明一个共有的 静态的 返回值为本类对象的方法
public static Student Instance()
{
    //当第一次a为空的时候就new 一个对象 并返回
    if(a==null) { a = new Student; }
    rutrun a;
}
```

23.抽象类简介

有时候如果有些类和方法本身是抽象的，那这种类其实根本不可能实例化。

例如：Food类，食物是一个抽象的名词，没有具体的对象与之对应。
此时可以将virtual修饰符改为abstract，使之成为抽象方法。

C#里面允许把这种类和方法声明为abstract，即抽象类和抽象方法。

PS:

1. 抽象类不能实例化；
2. 抽象方法是必须被子类重写的方法；
3. 如果类中包含抽象方法，那么类就必须定义为抽象类，不论是否还包含其他一般方法。

24.里式转换法

子类对象实体指向父类引用 即 里式转换法的核心

关键字：**as** 转换类型 **is** 判断转换：

```
A is B; //A可以向上转型为B吗?
//Animal是父类
Monkey m2=new Monkey(5,"大圣");
//判断m2可以向Animal转换吗 可以 true
if (m2 is Animal)
{
    //把m2 向上转型为 Animal类型的 并赋予 a1
    Animal a1=m2 as Animal;

    a1.show();//a1调用的是父类的show方法
}
```

ps: 子类对象默认调用父类构造函数 最Default就是 子类无参掉父类无参

24.1 as is 代码案例

向上转型 和向下转型 up/down:

is的 用法

```
static void Main(string[] args)
{
    //里式转换第一原则 提高子类地位
    Animal a = null; // 声明一个父类对象为空
    Animal a1= null; // 声明一个父类对象为空

    Dog g = new Dog();//声明一个狗的对象
    Yang y=new Yang();//声明一个羊的对象

    //体现里式转换法
        //true        //true
    if (g is Animal&& y is Animal)
    {
        a = g;
        a1= y;
        //把 子类转为父类对象 利用隐式转换 按低精度向精度转换
    }
    //这个 里式转换 从狗变成了动物 子类对象提升地位
    a.show();
    a1.show();
}
```

as的 用法

```
static void Main(string[] args)
{
    //里式转换第一原则 提高子类地位
    Animal a = new Dog() as Animal;

    Animal a1 = new Yang() as Animal;

    //is 先判断 后转换

    //as 又判断 又转换

    a.show();
    a1.show();
}
```

ps:

1. 向上转型 是从子类转型父类（必须有继承关系）
可自动完成也可以手动操作 //参考单精度转高精度的自动换方式
2. 向下转型是强制类型转换 提示作用域减小 所以需要我们手动操作
3. 无论转型方式 强制在：继承关系中完成 ，转型后根据 类型调用其作用域的成员

25. windows窗体应用程序 像素鸟

1.创建windows窗口应用程序

窗体程序会自动生成Form1.cs的类
在Form1.cs[设计]中更改窗口属性
BackColor: 更改窗体的颜色
BackgroundImage: 更改窗体地背景
Maximumsize: 更改窗体最大宽高
timer: 计时器

load: 窗体闪烁的方法
Paint: 窗体描绘的方法
keyDown: 键盘控制的方法
MouseDown: 鼠标控制的方法

pictureBox: 部分窗体

ps: 资源加载 Resources.resx

ps:解决闪烁具体代码

```
//解决闪烁问题
private void Form1_Load(object sender, EventArgs e)
{

    this.BackgroundImageLayout = ImageLayout.Stretch;
    //解决闪烁的代码
    this.SetStyle(ControlStyles.OptimizedDoubleBuffer |
        ControlStyles.ResizeRedraw | ControlStyles.AllPaintingInWmPaint, true);
}
```

2.Form1.cs程序设计

思想：在程序执行时 外观 Paint 描绘出小鸟
在事件中双击paint 生成 描绘的具体方法

所以我们需要一个小鸟的类
为了更好的扩展性 所以我们创建GameObject类 作为小鸟父类
小鸟类需要描绘时的具体坐标 X Y 和 宽 高 作为抽取父类

3.小鸟父类

1.声明小鸟的坐标 X Y
2.声明小鸟的宽高 W H
3.声明描绘小鸟的虚方法 abstract 让小鸟类自己具体化
ps: Graphics graphics 必须让画笔参加具体执行
4.重载多参构造函数
ps: 只要有虚方法 这个类就必须改为虚方法类 并且 谁继承虚类就必须重写虚方法

4.小鸟类

1.重载多参构造函数 调用父类多参 虚方法
2.声明一个public static Image[] 数组
ps: 用于保存要描绘的小鸟图片
3.重写父类的描绘小鸟的方法
PS:画笔调用描绘图片的方法 并传入需要描绘 图片和坐标
ps: graphics.DrawImage(AImage[0],this.X,this.Y);

思考：如何让小鸟切换图片

所以：

- 1.增加一个 int 数来作为索引
- 2.描绘小鸟方法需要if 或者 switch 判断索引
- 3.增加索引的自增方法
- 4.增加索引自增后在正确int范围内：

```
//Birdsinxs: 声明的索引 int值
switch (this.Birdsinxs)
{
    case 0:
        graphics.DrawImage(AImage[0], this.X, this.Y);
        break;
    case 1:
        graphics.DrawImage(AImage[1], this.X, this.Y);
        break;
    case 2:
        graphics.DrawImage(AImage[2], this.X, this.Y);
        break;
}
//索引++
this.Birdsinxs++;
//索引上限
if (Birdsinxs >= 3)
{
    Birdsinxs = 0;
}
```

5.中介单例类

好处：方便鸟类调用多参构造函数 和 属性的存取 并实现了小鸟的单例
懒汉式

- 1.私有化构造函数
- 2.私有化 静态的 本类的 对象引用
- 3.共有的 静态的 返回值为本类的方法

思考：如何直接调用鸟类方法 和 实例对象

所以：

- 1.声明一个 公有化的 鸟类对象引用

- 2.声明一个 公有化的 方法 未知参数为一个鸟类对象实例 并赋予鸟类对象引用
- 3.声明一个 公有化的 描绘鸟的方法 增加参数为Graphics 画笔

6.Form1.cs 程序具体

思考：在程序执行时在windows窗体中用画笔描绘出我们需要的小鸟

所以：

- 1.我们声明一个初始化程序的方法 方法内 调用单例取得方法并执行new 鸟类的方法 new对象时使用多参构造函数，用于确定鸟儿的位置和宽高
- 2.在paint 内 调用单例取得方法 用鸟儿对象的引用执行鸟类的 描绘方法
`SingObject.BirdsInstance1().draw(e.Graphics);`

思考:如何使小鸟切换图片

所以：利用计时器重绘，让小鸟扑腾

添加Time组件(按用户定义的间隔引发事件)，更改Enabled改为True；

在time组件内添加代码：

```
this.Invalidate();  
//使空间的整个图画无效并导致重绘 Invalidate 结束
```

7.小鸟的移动

思考：小鸟的自动下坠

所以：

需要添加一个重力类 全部为static

- 1.声明一个 float g 的自然重力 初始值为9.8f；
- 2.声明一个方法 getBirdsHeight 使用算法 $v*t+1/2*g*t^2$ 未知元素为 v t
- 3.在鸟类中 声明float 速度 v float 时间 t 初始化为 100 0f
- 4.我们需要一个timer计时器 在内部封装代码
 - 1.声明float h 取得下降的高度 调用getBirdsHeight并传入鸟的v, t t需要*0.001f；
 - 2.声明一个int变量 int y =取得鸟坐标Y+下降的高度 h
 - 3.限定int y 的范围 $y = y >= 382 ? 382 : y$ ；
 - 4.把int y赋值给鸟的Y坐标实现掉落
 - 5.设置鸟的掉落速度 鸟的v = 鸟的v+鸟的t*自然重力 g *0.001f；
 - 6.在键盘输入中封装 当按下键时 鸟的速度v = 0f；

思考：小鸟的移动

所以：调用键盘输入方法 keydown

- 1.声明if判断方法确定X,Y轴的值的变化
- 2.限定X,Y轴的值最大，最小：

```
if (e.KeyCode.Equals(Keys.W))  
{  
    //y轴的减小  
    birds3.Y -= 15;  
    if (birds3.Y <= 0)  
    {  
        //y轴的最小值  
        birds3.Y = 0;  
    }  
    //速度重置的封装  
    SingObject.BirdsInstance().birds.v = 9.8f;  
}
```

ps: 多只小鸟

I.把中介单例模式 改变为 中介N例模式

单例 变 2例

- 1.增加一个私有化 静态的 本类对象 引用
- 2.增加一个公有化 静态的 返回值为本类对象实例的 方法
- 3.增加一个鸟类引用
- 4.增加一个返回值为鸟类实例的 方法 并返回给新增的鸟类引用

ps: 2例完成 n例就增加n次

PS:最后要在游戏初始化方法内 增加引用实例 并在print 方法内调用实例对象的画鸟方法

ps: 注意实例鸟类对象时 鸟的位置

II.不使用中介单例类

- 1.直接在Form.cs里面声明私有化 鸟类对象的实例 使用多参构造函数

```
private Birds birds1 = new Birds(70, 100, 0);
```
- 2.直接在画鸟的 print 方法内 用鸟类对象实例的引用调用鸟类本身的画鸟方法

```
birds1.draw(e.Graphics, 0);
```

ps: 需要几只鸟就实例化几个鸟类对象 注意鸟的位置和鸟类实例对象的引用名

8.游戏物体类 管道

思考: 管道需要继承gameobjiet类么?

所以: 需要 因为管道也有需要坐标和宽高 和 move 和 draw 方法

- 1.声明 static Image类型 AImage= 需要的图片; //Resources.piping_2 管道

思考: 因为管道的高度是变化的, 所以需要一个随机数

所以: 我们创建一个工具类 Uutil

```
//管道的随机
public static int getRandom(int a,int b)
{
    //a 和 b是管道的最大和最小范围 保证随机出来后不会超出界限
    return new Random().Next(a ,b);
}
//炸弹的Y 随机范围
internal static int getzadanHeight()
{
    return new Random().Next(0,382);
}
```

- 2.重写父类的move 和draw

```
//随机的管道的Y坐标 赋给a
public int a = Uutil.getRandom(-280, -40);
draw:
//一组管道障碍
graphics.DrawImage(downImage, X, a);
graphics.DrawImage(upImage, X, 400 + a);
move:
//管道x坐标的向左移动
this.X -= 3;
if (X <= -50)
{
    this.X = 310;
    //重新获取随机值 赋予a 当出了屏幕后重新随机Y值
    a = Uutil.getRandom(-280, -40);
}
```

9.游戏物体类 炸弹

- 1.声明 static Image类型 Image= 需要的图片; //Resources.zhadan_2 炸弹

```
//随机的炸弹的Y坐标 赋给b
public int b = Uutil.getzadanHeight();
```

- 2.2. 重写父类的move 和draw

```
draw:
//调用 随机的y坐标b
graphics.DrawImage(zadan, X,b);
move:
this.X -= 11;
//当砸蛋X坐标为-10 的时候回到x=350
if (X <= -10)
{
    X = 350;
```

```
        //重新取得随机值 并赋给 b 即赋给砸蛋坐标Y  
        b = Util.getzadanHeight();  
    }  
}
```

10.修改form.cs

- 1.新增2组管道对象实例
- 2.新增2个炸弹对象实例

3.在paint 内写入4个对象的 各自的draw方法

4.新增计时器 写入4个对象的 各自的move方法

思考：如何增加碰撞

所以：需要使用系统自带方法 Rectangle

```
//增加所有绘制方法的矩形A B C D E F G  
Rectangle A = new Rectangle(SingObject.BirdsInstance().birds.X + 7, SingObject.BirdsInstance().birds.Y + 9, 29, 28);  
Rectangle B = new Rectangle(p1.X, p1.a, p1.Width, p1.Heigth);  
Rectangle C = new Rectangle(p1.X, p1.a + 400, p1.Width, p1.Heigth);  
Rectangle D = new Rectangle(p2.X, p2.a, p2.Width, p2.Heigth);  
Rectangle E = new Rectangle(p2.X, p2.a + 400, p2.Width, p2.Heigth);  
//炸弹的矩形  
Rectangle F = new Rectangle(b1.X+5, b1.b+12, b1.Width-10, b1.Heigth-20);  
Rectangle G = new Rectangle(b2.X+5, b2.b+12, b2.Width-10, b2.Heigth-20);  
//绘制出F的矩形 颜色为black 黑色  
//e.Graphics.DrawRectangle(Pens.Black,F);  
  
//矩形触碰时的反应代码  
if (A.IntersectsWith(B) || A.IntersectsWith(C) || A.IntersectsWith(D) || A.IntersectsWith(E) || A.IntersectsWith(F) || A.I  
{  
    //MessageBox.Show("游戏结束!", "", MessageBoxButtons.OK);  
    //游戏直接结束  
    Environment.Exit(0);  
}
```

ps:

new rectangle的时候括号内写入 矩形的坐标和宽高

在写入坐标Y的时候注意，因为Y是移动的 所以需要写入的Y 为 随机的 a 或者 b