# 01

Console.WriteLine(“要打印的内容 ”);//把内容输出到控制台。

Console.ReadKey();//等用户按下任意键。

Console:控制台

Console.Readline（）读入

# 02

1. 驼峰命名法：要求变量名单词的首字母要小写，其余每个字母的首字母要大写。多用于给变量命名。
2. 帕斯卡命名法：要求每个单词的首字母都要大写，其余字母小写。多用于给类或者方法命名。

# 03

Ctrl+z:撤销

Ctrl+s：保存

Ctrl+k+c：注释所选代码

Ctrl+k+u：取消对所选代码的注释

# region和#endregion：折叠冗余代码

# 04

两个不兼容类型转换

代码：

Convert.ToInt32();

Convert.ToDouble();

# 05

数组初始化：

数组类型[] 数组名=new 数据类型 [数组长度]

New:创建数组实例

Int []age=new int [10]；

Int []nember={0，1，2，3，4，5}；

NULL:意思为根本没有在内存当中开辟空间。

空：意思为开了空间只是没有储存值。

String []name=new string[3]//初始值为null

Bool []a=new bool[5]//初始值为False

Int为0

# 06

**Out:一个函数中如果返回多个不同类型的值，需要用到out参数（传的是参数）**

**要点：函数外可以不为变量赋值，而函数内必须为其赋值。**

**语法：形参与实参前面都要加OUT关键字**

# 07

**Ref：将一个变量传入一个函数中进行“处理”，“处理”完成后，再将“处理”后的值带出函数。（传的是参数）**

**要点：**

# 08

重载：函数的名称相同，但是参数列表不同。

调用该函数的时候，会根据不同的参数，自动选择合适的函数重载形式。

1. **如果参数的个数相同，那么参数的类型就不能相同；**
2. **如果参数的类型相同，那么参数的个数就不能相同；**
3. **注意：函数的返回值和重载没有关系**

# 09

属性名：和字段名一样，只不过首字母要大写

Get：{return ;}当通过属性名取值的时候，会自动调用Get中的代码；

Set：{ =value;}当通过属性名给字段赋值的时候，会自动调用set中的代码

Value：也是系统关键字，代表赋给属性的值；

Get和Set本质上就是两个函数

快捷键:ctrl+R+E

# 10

在调用实例成员的时候，需要使用对象名。实例成员;

在调用静态成员的时候，需要使用类名.静态成员名。

总结：静态成员必须使用类名去调用，而实例成员使用对象名调用。

静态函数中，只能访问静态成员，不允许访问实例成员。

实例函数中，既可以使用静态成员，也可以使用实例成员

静态类中只允许有静态成员，不允许出现实例成员

# 11

1构造函数没有返回值，连void也不能写

2构造海曙的名称必须跟类名一样。

**例如：public Man（）{}**

**this： 代表当前类的对象**

# 12

Number.ToUpper():转换成大写；

Number.ToLower():转换成小写；

Equals（）：作用：比较两个字符是否相同。相同返回真，不相同返回假。

Split（）：作用：分割字符串。返回字符串类型的数组

使用：字符串变量，方法名（用于分割的字符数组）；

例如：+\_abcd,.cdf 清楚中间的无用字符

Substring（）：作用：截取字符串。返回截取后的字串

使用：字符串变量，方法名（开始截取的位置）；

字符串变量.方法名（开始截取的位置，截取多长）；

今天中午吃什么?包含自己往后截取。

Replace（）作用：将字符串中的某个字符串替换成一个新的字符串。返回新的字符串。、

使用：字符串变量.方法名（旧字符串，新字符串）；

Contains（）作用：判断某个字符串中是否包含指定的字符串。如果包含返回真。否则返回假。

使用：字符串变量,方法名（子字符串）

IndexOf（）：作用：查找某个字符串在字符串中第一次出现的位置。

返回所在的索引位置值。如果没有找到，返回-1.

使用：字符串变量,方法名（子字符串）；

LastIndexOf（）：作用：查找某个字符串在字符串中的最后一次出现的位置。

返回所在的索引位置值。如果没有找到，返回-1.

使用：字符串变量。方法名（子字符串）；

StartsWith(): 作用 ：判断是否以..字符串开始。如果是，返回真；如果不是，返回假。

使用：字符串变量.方法名（子字符串）

EndsWith(): 作用 ：判断是否以..字符串结束。如果是，返回真；如果不是，返回假。

使用：字符串变量.方法名（子字符串）

Trim（）作用：去掉字符串中前后空格。返回处理后的字符串。

使用：字符串变量.方法名（）；

TrimEnd（）作用：去掉字符串中结束后空格。返回处理后的字符串。

使用：字符串变量.方法名（）；

TrimStart（）作用：去掉字符串中开始前空格。返回处理后的字符串。

使用：字符串变量.方法名（）

IsNullOrEmpty（）作用：判断一个字符串是否为Null或者为空。如果为null或者空，返回真；否则返回假。Null是不占内存空间的，而空字符串（““）是占内存空间的。

使用：string.方法名（字符串变量）；

# 13

堆好比内存：引用类型的值存储string 数组

栈好比运行内存： 值类型的值存储int double bool char struct enum

**Stopwatch:计时器**

Stopwatch sw =new Stopwatch（）；

Sw.Start();//计时器开始。

Sw.Stop（）；//计时器结束。

Sw.Elapsed//开始到结束之间的时长。

**StringBuilder**

String,字符串；Builder，构建器：连起来是“字符串构建器“

StringBuilder是一个类。SB类型的变量是引用类型。

StringBuilde类型的“字符串变量“，一直操作同一块内存空间，不会产生拉丝机数据，且执行效率远远高于string类型的字符串变量。

StringBuilder使用方法：

1. 创建StringBuilde类型的变量

StringBuilde sb=new StringBuilde（）；//创建一个对象

1. 往sb中追加数据

sb.Append(i);//追加数据。

Sb.Tostring（）；//将sb转成字符串形式。

1. 清空sb中的数据

Sb.Clear();//将sb清空。

# 14

父类私有字段不能继承，只能通过接口去访问。

子类并没有继承父类的构造函数，但是，子类会默认的调用父类无参的构造函数。

创建父类对象，让子类可以使用父类中的成员。

所以，如果在父类中重写了一个有参数的构造函数之后，那个无参数的就没有了，子类就调用不到了，所以子类会报错。

子类在实例化时系统会默认创建父类对象，需要调用父类的默认构造函数

解决：在子类中显示的调用父类的构造函数，使用关键字：base（）//关键字base代表父类

子类有构造函数时，实例化先跑到子类构造函数，之后在到父类的构造函数，

# 15

Object类是所有类型的父类（基类）

里氏转换？

1. 子类对象可以直接赋给父类；
2. 子类对象可以调用父类中的成员，但是父类对象永远只能调用自己的成员
3. 如果父类对象中装的是子类对象，可以将这个父类对象强转为子类对象

举例：小明可以把自己对象的灵魂注入到他父亲的体内。那么我们是不是可以认为雄安命的父亲是他的对象呢。

# 16

Is和as

Is和as两个关键字都可以进行类型转换

Is：如果转换成功，返回true，失败返回false。F is Ming

As：如果转换成功，返回对应的对象，失败返回null。Hong h=f as Hong

# 17

ArrayList 集合

List.Count长度//list是对象

Add添加单个元素，

AddRenge（new int【】{1，2，3}）添加集合类

Count：表示这个集合中实际包含的元素的个数

Capcity：表示这个集合中可以包含元素个数

每次集合中实际包含的元素个数（count）超过了可以包含的元素的个数（capcity）的时候，集合就会向内存中申请多开辟一倍的空间，来保证集合的长度一直够用

Random随机数（next）

List<int> list=new List<int>()//list的泛型集合

三种其实都是集合，因为他们都继承并实现了ICollection  
集合里所有的元素都是Object，如果元素是值类型，会自动装箱。  
泛型集合可以定义元素类型，相对于集合，泛型集合可以避免装箱拆箱，提高性能，同时程序具有更好的可读性。  
数组本身可以认为是一种泛型集合结构体

# 18

装箱：就是将值类型转换为引用类型

拆箱：将引用类型转换为值类型

看两种类型是否发生了装箱或者拆箱，要看。这两种类型是否存在继承关系。如果存在继承关系则可能发生装箱或者拆箱，如果没有继承关系则不可能发生装箱或者拆箱

封箱是把值类型转换为System.Object类型，或者转换为由值类型实现的接口类型：

# 19

虚方法：

在父类中使用Virtual关键字修饰的方法，就是虚方法。在子类中可以使用override关键字对该虚方法进行重写。

（使父类对象能够调用子类的方法）

父类：

Public virtual 返回值类型 方法名（）

子类：

Public override 返回值类型 方法名（）

# 20

抽象方法的返回值类型用关键字abstract修饰，且无方法体

抽象方法必须存在于抽象类中

。

抽象类语法：在定义类的关键字class前面加abstract修饰的类就是抽象类。

子类继承抽象类，使用override关键字重写父类中所有的抽象方法。

注意：

1.抽象类中不一定有抽象方法，但是抽象方法必须存在于抽象类中。

2.抽象类不能被实例化，应为抽象类中有抽象方法（无方法体），如果真能实例化抽象类的话，调用这些无方法体的方法是没有任何意义的，所以无法实例化。

使用场景：

当父类中的方法不知道如何去实现的时候，可以考虑将父类写成抽象类，将方法写成抽象类。

# 21

访问修饰符

Public：公开的公共的

Private：私有的，只能在当前类的内部访问

Protected：受保护的，只能在当前类的内部以及该类的子类中访问

Internal：只能在当前项目中访问。在同一个项目总，internal和public的权限是一样的。

Protected internal ：protected+internal

# 22

interface

当一个类继承了一个接口必须实现该接口的全部成员

接口方法可以有返回值

接口中的成员不允许添加访问修饰符，默认就是public。

接口中不允许写具体的方法体函数

接口中不允许写字段，因为接口不是存数据的。存数据可以用类去存。

接口不可以写属性，但是可以自动属性。程序会给自动属性自动添加字段。自动属性本质上和属性没有什么区别。

接口方法：

接口使用interface关键字定义，没有class关键字，接口名一般以I开头这种方式进行书写，在一堆脚本中通过名字判断，I开头的都是接口

接口注意事项：

1. 接口中所有的方法都是抽象方法，所以接口不能被实例化；
2. 一个类可以实现多个接口，被实现的多个接口之间用逗号分隔
3. 接口需要注意的问题
4. 接口能不能被实例化？

为了多态。接口不能被实例化。也就是说，接口不能new（不能创建对象）。

1. 接口可以继承接口吗？

可以，接口与接口之间可以继承，并且可以多继承。

1. 接口可以继承类吗？

不可以，接口并不能去继承一个类，而类可以继承接口（接口只能继承于接口，而类既可以继承接口，也可以继承类）

1. 一个类可以同时继承一个类并实现多个接口，如果一个子类同时继承了父类H，并实现了接口IH，那么语法上H必须写在IH的前 面。

什么时候用虚方法来实现多态

如果这个父类中的方法你知道怎么写并且还需要创建这个父类的对象那就用虚方法来实现。

什么时候用抽象来实现多态？

如果几个类当中，你能提取出来一个父类并且父类当中必须写上几个子类共有的方法然后你还不知道怎么写这个方法。那就是哦那个抽象类来实现

什么时候用接口来实现多态

如果这几个类当中根本就提取不出来父类，但是这几个类都有一个共同的行为或者能力。那就用接口来是实现。