1. abstract class和interface有什么区别?

相同点：都不能被直接实例化，都可以通过继承实现其抽象方法。

不同点：

* 接口支持多继承；抽象类不能实现多继承。
* 接口只能定义行为；抽象类既可以定义行为，还可能提供实现。
* 接口只包含方法(Method)、属性(Property)、索引器(Index)、事件(Event)的签名，但不能定义字段和包含实现的方法；
* 抽象类可以定义字段、属性、包含有实现的方法。

1. 是否可以继承String类?

不可以，String类是由sealed修饰，不能被继承

1. 什么是泛型

泛型。即通过参数化类型来实现在同一份代码上操作多种数据类型

1. StringBuilder 和 String 的区别？

String是不可变类型，每次变更都会生成一个新对象，如果过多的String类型进行拼接，很多中间过程的对象会占用很多内存空间，然后导致大量的空间需要被垃圾回收

StringBuilder是会预留足够的空间来存储string，拼接时把新的string存储在预留空间里，空间不足时会进行扩容，就不会有String的缺点。

StringBuilder的初始容量是16，当需要拼接大量的字符串的时候，可以把StringBuilder的初始容量设置大点，这样就可以减少扩容的发生。

1. 抽象方法、虚方法和接口方法区别，什么情况下会用到虚方法？

接口方法是没有实体的，其子类必须要实现接口方法，只需要其继承的子类实现

抽象方法是没有方法体的，类型必须为抽象类，所以其子类(任何深度的子类)必须要实现抽象方法

虚方法是可以有方法体的，所以其所有子类(任何深度的子类)可以重写(override)也可不重写,使用override关键字来重写，由于任何深度的子类都能重写，所以override关键字前面就不能在定义virtual了。Override前面可以有abstract修饰(dotnet core中)

抽象方法、虚方法和接口都能实现多态，而虚方法是可以有实体的，所以当父类有特定的方法实现的时候使用虚方法，子类继承之后可以重写方法。(ToString、GetHashCode和Equals)

1. ReferenceEquals和Equals两个静态方法
2. string、String；int、Int32；Boolean、bool的区别

别名，int定位到实现时就是会跳转到Int32

1. private、 protected、 public、 internal 修饰符的访问权限。

private : 私有成员, 在类的内部才可以访问。(注意：在类内部创建实例，用实例调用 也不能调用)

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。

1. C#中的委托是什么？事件是不是一种委托？事件和委托的关系。

委托是一个方法的包装器，可以把方法包装起来，进行传递和使用。

委托里会有方法的信息和该方法的对象(target)，该方法是静态方法时，target为null

事件也是使用的委托，xxxHandle就是使用的delegate定义的，even编译器会最终转换为委托链的形式来使用委托，

使用委托的时候可以不用自己定义委托，已经有现成的委托类型能满足大部分的需求了。

Action<>和Func<>泛型类

1. override与重载（overload）的区别

override是重写父类虚方法，overload是方法重载，一个类中可以存在多个相同名称的方法，只要方法签名不一样就行。

写远程过程调用(wcf、grpc)的时候，最好不要使用重载，因为很多其他语言是不支持重载的。

1. 什么是装箱(boxing)和拆箱(unboxing)

装箱和拆箱只有值类型会发生。值类型是存储在栈中的，当把值类型转换为object类型时，需要有类型对象指针，所以需要把值类型复制到堆中进行存储，然后在栈中进行引用，这个就是装箱过程，需要在堆中分配内存，还涉及到数据的复制传输，所以对性能有影响。拆箱就是相反的流程。

1. 列举ASP.NET 页面之间传递值的几种方式。(这个是ASP.NET的)

使用QueryString, 如....?id=1; response. Redirect()....

使用Session变量

使用Server.Transfer

使用Application

使用Cache

使用HttpContext的Item属性

使用文件

使用数据库

使用Cookie

1. Windows的Controls(winform)

获取控件

1. 索引器
2. Mvc三层架构优缺点
3. 父类和子类初始化顺序

子类全局变量 -> 父类全局变量 -> 父类构造函数 ->子类构造函数

方法是在类声明里，不用初始化

1. 什么叫应用程序域？、

应用程序域可以理解为一种轻量级进程。起到安全的作用。占用资源小。

1. CTS、CLS、CLR分别作何解释？

答：CTS：通用语言系统。CLS：通用语言规范。CLR：公共语言运行库。

1. 什么是受管制(托管)的代码？
2. net中读写数据库需要用到那些类？他们的作用？

DataSet:数据集。

DataCommand:执行语句命令。

DataAdapter:数据的集合，用语填充。

DataReader:数据只读器

1. string str = null 与 string str = “”

null没有分配空间，””分配了空间

1. class和struct异同
2. string strTmp = "abcdefg某某某";
3. 编码问题

int i= System.Text.Encoding.Default.GetBytes(strTmp).Length;

int j= strTmp.Length;

以上代码执行完后，i= j=

答：i=13,j=10

1. 面向对象的语言三特性

封装、继承、多态

1. 能用foreach遍历访问的对象需要实现 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_接口或声明\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_方法的类型。

答：IEnumerable 、 GetEnumerator。

1. GC是什么? 为什么要有GC?GC代？

GC是垃圾收集器。程序员不用担心内存管理，因为垃圾收集器会自动进行管理

1. 线程的run()和start()方法
2. 接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?

接口可以继承接口，抽象类可以继承实体类

1. 构造器(构造函数)Constructor是否可被override?

不能，构造函数只能overload

1. try {}里有一个return语句，那么紧跟在这个try后的finally {}里的code会不会被执行，什么时候被执行，在return前还是后?

会执行，在return后执行，return后方法还没结束，会把结果添加到栈中，在finally之后方法才结束

1. 两个对象值相同(x.equals(y) == true)，但却可有不同的hash code，这句话对不对?

不对，equals就是对比的hash code

1. swtich是否能作用在byte上，是否能作用在long上，是否能作用在String上?

switch（expr1）中，expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、 short、 char 或者 byte。

1. short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?

short s1 = 1; s1 = s1 + 1;有错，s1是short型，s1+1是int型,不能显式转化为short型。可修改为s1 =(short)(s1 + 1) 。short s1 = 1; s1 += 1正确。

1. Session有什么重大BUG，微软提出了什么方法加以解决？

是iis中由于有进程回收机制，系统繁忙的话Session会丢失，可以用Sate server或SQL Server数据库的方式存储Session不过这种方式比较慢，而且无法捕获Session的END事件。

1. 进程和线程的区别？

进程是系统进行资源分配和调度的单位；线程是CPU调度和分派的单位，一个进程可以有多个线程，这些线程共享这个进程的资源。

1. 堆和栈的区别？
2. DataReader与Dataset有什么区别？
3. 软件开发过程一般有几个阶段？每个阶段的作用？

可行性分析(风险控制)，需求分析，架构设计，代码编写，测试，部署，维护

1. implicit和explicit
2. 什么是反射？

通过类型元数据动态获取类型的信息，包括方法、属性、特性等。

1. 写单例模式

恶汉模式、双锁、线程安全类、依赖注入

1. 线程池优点

减少线程重复创建的消耗

提高线程的可管理性

1. 什么是XML？

XML即可扩展标记语言

1. c#中的三元运算符是？

？：

1. public static const int A=1;这段代码有错误么？是什么？

答：const不能用static修饰。

1. 委托声明的关键字是\_\_\_\_\_\_?

Delegate

1. throw和throw ex区别
2. 对于这样的一个枚举类型：

enum Color:byte

{

Red,

Green,

Blue,

Orange

}

答：string[] ss=Enum.GetNames(typeof(Color));

byte[] bb=Enum.GetValues(typeof(Color));

1. &和&&的区别。

&和&&都可以用作逻辑与的运算符，表示逻辑

&&还具有短路的功能，即如果第一个表达式为false，则不再计算第二个表达式。

&还可以用作位运算符

1. HashTable、Dictionary和HashSet

Hashtable和Dictionary都是key-value数据集合，Hashtable不是泛型，Dictionary是泛型，Hashtable可以通过Synchronized静态类型方法来创建线程安全类，也可手动lock SyncRoot

var myCollection = new Hashtable();

lock (myCollection.SyncRoot)

{

foreach (object item in myCollection)

{

// Insert your code here.

}

}

Hashtable mySyncdHT = Hashtable.Synchronized(myHT);

单线程推荐使用Dictionary，泛型使用起来更方便

HashSet<T>类主要是设计用来做高性能集运算的，

例如对两个集合求交集、并集、差集等。

集合中包含一组不重复出现且无特性顺序的元素，HashSet拒绝接受重复的对象

1. 索引器

class Sample {

public string this [int index] {

get {return "You passed " + index; }

}

}

1. 129.下面的例子中

using System;

class A

{

public static int X;

static A(){

X=B.Y+1;

}

}

class B

{

public static int Y=A.X+1;

static B(){}

static void Main(){

Console.WriteLine("X={0},Y={1}",A.X,B.Y);

}

}

产生的输出结果是什么？

答：x=1,y=2

1. 值类型和引用类型的区别

值类型都是继承自ValueType,存储在栈中，引用类型是存储在堆中，值类型不能为null，可空类型Nullable<>其实只是对值类型进行包装，是个引用类型。值类型在使用时必须所有字段都初始化，就是确保在栈中分配内存。引用类型可以为null，当引用类型为null时，是还未在堆中分配内存的。