**1 请你说说 .NET 中类和结构的区别？**

答：结构和类具有大体的语法，但是结构受到的限制比类要多。结构不能申明有默认的构造函数，为结构的副本是又编译器创建和销毁的，所以不需要默认的构造函数和析构函数。结构是值类型，所以对结构变量所做的改变不会影响其的原值，而类是应用类型，改变其变量的值会改变其原值。申明结构用Struck 关键字，申明类用 class 关键字，向方法传递结构是时是通过值传递的，而不是通过引用。与类不同，结构的实例化可以不使用New关键字。类可以实现接口。

类：

类是引用类型在堆上分配，类的实例进行赋值只是复制了引用，都指向同一段实际对象分配的内存

类有构造和析构函数

类可以继承和被继承

结构：

结构是值类型在栈上分配（虽然栈的访问速度比较堆要快，但栈的资源有限放），结构的赋值将分配产生一个新的对象。

结构没有构造函数，但可以添加。结构没有析构函数

结构不可以继承自另一个结构或被继承，但和类一样可以继承自接口

一些轻量级的对象最好使用结构，但数据量大或有复杂处理逻辑对象最好使用类。

**2，死锁的必要条件？怎么克服？**

答：系统的资源不足，进程的推进的顺序不合适，资源分配不当，一个资源每次只能被一个进程使用，一个资源请求资源时，而此时这个资源已阻塞，对已获得资源不放，进程获得资源时，未使用完前，不能强行剥夺。

**3，接口是否可以继承接口？抽象类是否可以实现接口？抽象类是否可以继承实体类？**

答：接口是可以继承接口的，抽象类是可以实现接口的，抽象类可以继承实体类，但是有个条件，条件是，实体类必须要有明确的构造函数。

**4，构造器Constructor是否可以被继承？是否可以被Override？**

答：Constructor不可以被继承，因此不能被重写(Overriding)，但可以被重载(Overloading).

**5，当一个线程进入一个对象的方法后，其它线程是否可以进入该对象的方法？**

答：不可以，一个对象的方法只能由一个线程访问。

**6，用最有效的方法算出等已8对于几？**

答：2<<3.

**7,C#是否可以对内存直接进行操作？**

答：这个问题比较难回答，也是个很大的问题。但是可以这样问答。C#是可以对内存进行直接操作的，虽然很少用到指针，但是C#是可以使用指针的，在用的时候需要在前边加unsafe,，在.net中使用了垃圾回收机制(GC)功能，它替代了程序员，不过在C#中不可以直接使用finalize方法，是在析构函数中调用基类的finalize()方法。

**8，Error和Exception有是区别？**

答：error表示恢复不是不可能，但是很困难，exception表示一种实际或实现问题，它表示程序运行正常不可以发生的。

**9，谈谈final，finally，finallize的区别？**

答：final 用于申明属性，方法和类，表示属性不可变，方法不可以被覆盖，类不可以被继承。Finally 是异常处理语句结构中，表示总是执行的部分。Finallize 表示是object类一个方法，在垃圾回收机制中执行的时候会被调用被回收对象的方法。

**10,HashMap和Hashtable区别？**

答：HashMap是Hashtable的轻量级实现，非线程安全的实现他们都实现了map接口，主要区别是HashMap键值可以为空null,效率可以高于Hashtable。

**11，Collection和Collections的区别？**

答：Collection 是集合类的上级接口，Collections 是针对集合类的一个帮助类，它提供一系列静态方法来实现对各种集合的搜索，排序，线程安全化操作。

**12，C#中委托是什么?事件是不是一种委托？**

答：委托是一种安全的类似与函数指针，但是她比指针要安全的多，它可以把方法作为一个参数传递给另一个方法，可以理解为指向函数的引用。事件是一种消息机制，它是一种委托，委托不带方法体。

**13，Override,Overload，的区别？**

答：Override是重写的意思，它表示重写基类的方法，而且方法的名称，返回类型，参数类型，参数个数要与基类相同。Overload 是重载是意思，它也表示重写基类的方法，但是只要方法名相同，别的可以不同。

**14，在一个BS结构中需要传递变量值时，不能使用session,coolke,application，你有几中方法？**

答：this.server.Transfer,Querystring.

**15,C#中索引器实现过程，是否只能根据数字索引？**

答：不是的，可以是任意类型。

**16，New有几种用法？**

答：有3种，第一种是，实例化如：NewClass()

第二种是，public new 隐藏基类的方法

第三种是，在泛型类申明中的任何类型参数都必须有公共的无参构造函数。

**17，任何把一个Array复制到Arraylist中？**

答：Foreach(object o in Array),ArrayList.Add(0)等有好多中种方法。自己想。

**18，概述反射和序列化？**

答：反射：要给发射下一个定义还是比较难的，这里先说说我的理解。反射提供了封装程序集，模块和类型对象，可以用反射动态地创建类型的实例，将类型绑定到现有对象，或者从现有对象类型里获取类型，然后调用类型的方法或访问字段和属性。

序列化：将对象转换为另一种媒介传输的格式过程。如，序列化一个对象，用Http通过internet在客户端和服务器之间传递该对象，在另一端用反序列化从该流中重新得到对象。

**19，Const和ReadOnly？**

答：Const用来申明编程时申明常量，ReadOnly用来申明运行时常量。

**20，UDP和TCP连接有和异同？**

答：TCP是传输控制协议，提供的是面向连接的，是可靠的，字节流服务，当用户和服务器彼此进行数据交互的时候，必须在他们数据交互前要进行TCP连接之后才能传输数据。TCP提供超时重拨，检验数据功能。

UDP是用户数据报协议，是一个简单的面向数据报的传输协议，是不可靠的连接。

**21，进程和线程分别该怎么理解？**

答：进程是比线程大的程序运行单元，都是由操作系统所体会的系统运行单元，一个程序中至少要有一个进程，有一个进程中，至少要有一个线程，线程的划分尺度要比进程要小，进程拥有独立的内存单元，线程是共享内存，从而极大的提高了程序的运行效率同一个进程中的多个线程可以并发执行。

**22，ASP.NET页面之间传递值的几种方式？**

答：QueryString,session,cookies,application,server.Transfer,respose.redictor.

**23.什么叫应用程序域？什么是托管代码？什么是强类型系统？什么是装箱和拆箱？什么是重载？CTS、CLS和CLR分别作何解释？**

答：应用程序域：就是为安全性，可靠性，隔离性，和版本控制，及卸载程序提供的隔离边界。它通常由运行库宿主创建，应用程序域提供了一个更安全，用途更广的处理单元。托管代码：使用CLR编译语言编辑器开发编写的代码就叫托管代码。装箱和拆箱：是把值类型转换为引用类型的过程，是隐式的，相反的过程就是拆箱，是显式的。CTS是公共类型系统，CLS是公共语言规范，CLR公共语言运行库。强类型系统：每个变量和对象都必须具有申明类型。

**24.值类型和引用类型的区别？**

答：值类型的值是存放在堆栈中的，改变其值，不改变变量原有的值，而引用类型的值是存放在栈中的，其引用的地址是存放在堆栈中的，改变其值也就改变了变量原有的值。值类型不允许包含null值，然而可空类型可以将null赋值给值类型l。

**25.ASP.net的身份验证方式有哪些？**

答：windows,forms,passport

**26.解释一下UDDI、WSDL的意义及其作用？**

答：UDDI是统一描述集成协议，是一套基于Web的，分布式的，为WEB服务提供的信息注册的实现标准规范，同时为也是为企业本身提供的Web服务注册以让别的企业能够发现并访问的协议标准。提供了基于标准的规范，用于描述和发现服务，还提供了一组基于因特网的实现。WSDL这是一个基于XML的描述WEB服务的接口。

**27.什么是SOAP？**

答：是简单访问协议。是在分布式环境中，交换信息并实现远程调用的协议。是一个基于XML的协议。使用SOAP，可以不考虑任何传输协议，但通常还是HTTP协议，可以允许任何类型的对象或代码，在任何平台上，以任一种语言相互通信。它是一种轻量级协议。

**28.如何部署一个ASP.net页面？**

答：vs2003,vs2005里边都有发表机制，vs2003可以发布然后在复制部署。Vs2005可以直接部署到对应的位置。

**29.如何理解.net中的垃圾回收机制？**

答：.NET中的垃圾回收机制是引用程序对内存的回收和释放。当每次用new关键字创建一个对象时，运行库都要从托管堆中为其分配内存，因为空间是有限的，最终垃圾回收机制是要回收不用的内存的。已释放内存，重新使用。

**30.面向对象的三大基本原则？**

答：封装，继承，多态。

**31.在.NET中所有类的基类是？**

答：object。

**32.能用foreach遍历访问的对象需要实现？**

答：需要实现IEnumerable接口和GetEnumerator()方法。

**33.Heap与stack的差别？**

答：Heap是堆，空间是由手动操作分配和释放的，它的存储区很大的自由存储区。Stack是栈，是由是操作系统自动分配和释放的，栈上的空间是有限的。程序在编译期间变量和函数分配内存都是在栈上进行的，且在运行时函数调用时的参数的传递也是在栈上进行的。