

ZENQ

Microsoft.Net

常见问题集锦

Dong

2009-3-16

目录

一、	Microsoft.Net 平台基础.....	6
1、	GC 是什么? 为什么要有 GC?	6
二、	编程语言	6
1、	String 和 StringBuilder 的区别?	6
2、	类的 static 型构造方法是干什么用的?	6
3、	String s = new String("xyz");创建了几个 String Object?	7
4、	abstract class 和 interface 有什么区别?	7
5、	. 启动一个线程是用 run() 还是 start()?	7
6、	. 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)?	7
7、	. 构造器 Constructor 是否可被 override?	8
8、	. 是否可以继承 String 类?	8
9、	. 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?	8
10、	. 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?	8
11、	. switch 是否能作用在 byte 上, 是否能作用在 long 上, 是否能作用在 String 上?	8
12、	. List, Set, Map 是否继承自 Collection 接口?	8
13、	. Set 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用 == 还是 equals()? 它们有何区别?	8
14、	. 数组有没有 length() 这个方法? String 有没有 length() 这个方法?	8
15、	. sleep() 和 wait() 有什么区别?	9
16、	. short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?	9
17、	. 谈谈 final, finally, finalize 的区别。	9
18、	. 如何处理几十万条并发数据?	9
19、	. 进程和线程的区别?	10
20、	. 堆和栈的区别?	10
21、	. 成员变量和成员函数前加 static 的作用?	10

22、	. 简述 private、protected、public、internal 修饰符的访问权限。	10
三、	APS.NET 开发	10
23、	. ASP.NET 与 ASP 相比, 主要有哪些进步?	10
24、	. Session 有什么重大 BUG, 微软提出了什么方法加以解决?	10
25、	. 产生一个 int 数组, 长度为 100, 并向其中随机插入 1-100, 并且不能重复。	10
26、	. 列举 ASP.NET 页面之间传递值的几种方式。	11
27、	. 一列数的规则如下: 1、1、2、3、5、8、13、21、34..... 求第 30 位数是多少, 用递归算法实现。	11
28、	. C#中的委托是什么? 事件是不是一种委托?	12
29、	. override 与重载的区别	12
30、	. 如果在一个 B/S 结构的系统中需要传递变量值, 但是又不能使用 Session、Cookie、Application, 您有几种方法进行处理?	12
31、	. 请编程遍历页面上所有 TextBox 控件并给它赋值为 string.Empty?	12
32、	8. 请编程实现一个冒泡排序算法?	13
33、	. 描述一下 C#中索引器的实现过程, 是否只能根据数字进行索引?	13
34、	. 求以下表达式的值, 写出您想到的一种或几种实现方法: $1-2+3-4+\cdots+m$	13
35、	. 用 .net 做 B/S 结构的系统, 您是用几层结构来开发, 每一层之间的关系以及为什么要这样分层?	14
36、	. 什么叫应用程序域?	14
37、	. CTS、CLS、CLR 分别作何解释?	14
38、	. 什么是装箱和拆箱?	14
39、	. 什么是受管制的代码?	14
40、	. 什么是强类型系统?	15
41、	. net 中读写数据库需要用到那些类? 他们的作用?	15
42、	. ASP.net 的身份验证方式有哪些? 分别是什么原理?	15
43、	. 什么是 Code-Behind 技术?	15
44、	. 在 .net 中, 配件的意思是?	15
45、	. 常用的调用 Webservice 的方法有哪些?	15
46、	. .net Remoting 的工作原理是什么?	15

47、	. 在 C# 中, string str = null 与 string str = “” 请尽量使用文字或图象说明其中的区别。	15
48、	. 请详述在 dotnet 中类(class)与结构(struct)的异同?	16
49、	. SQLSERVER 服务器中, 给定表 table1 中有两个字段 ID、LastUpdateDate, ID 表示更新的事务号, LastUpdateDate 表示更新时的服务器时间, 请使用一句 SQL 语句获得最后更新的事务号	16
50、	. 根据线程安全的相关知识, 分析以下代码, 当调用 test 方法时 i>10 时是否会引起死锁?并简要说明理由。	16
51、	. 简要谈一下您对微软.NET 构架下 remoting 和 webservice 两项技术的理解以及实际中的应用。	16
四、	数据库	16
1、	. 写出一条 Sql 语句: 取出表 A 中第 31 到第 40 记录 (SQLServer, 以自动增长的 ID 作为主键, 注意: ID 可能不是连续的。	17
答: 解 1:	select top 10 * from A where id not in (select top 30 id from A)	17
解 2:	select top 10 * from A where id > (select max(id) from (select top 30 id from A)as A)	17
五、	问题集锦	17
1、	. 什么是中间语言 (IL)? 它的作用?	17
2、	. 什么是装箱和拆箱?	17
3、	出 ADO.NET 中读写数据库的主要的几个类? 它们的作用?	17
4、	C#中接口和类有什么异同?	17
5、	解释 ASP.NET 中的 code-behind?	18
6、	什么是 ASP.NET 中的用户控件?	18
7、	在 C# 中, string str = null 与 string str = “” 请尽量说明其中的区别。	18
8、	. 什么是受管制代码?	18
9、	C#中常用的访问形式? 它们各自的作用域?	18
10、	. SQL SERVER 中的存储过程的理解?	18
11、	. CTS, CLS, CLR 分别作何解释?	18
12、	. ASP.NET 和 ASP 的主要优势?	18
13、	. NET 的错误处理机制是什么?	18
14、	. 解释 Page.IsPostBack 属性?	19
15、	. .net framework 包含哪几种技术?	19

16、	. 触发器分为事前触发和事后触发，有何区别？	19
17、	. 解释 SQL SERVER2000 中 HAVING 的用法？]	19
18、	. 解释 SQL SERVER2000 中 GROUP BY 的用法？	19
19、	. 解释下 C#中的方法的重载？	19
20、	. 事务是什么？	19
21、	. 描述下 ASP.NET 中使用 DataReader 对象 和 DataSet 对象的区别？ .	19
22、	. 分析类和结构的异同？	19
23、	. 解释 ADO.NET 中的使用 Connection 类的 CLOSE（）和 DISPOSE（）的 区别？	19
24、	. 叙述 ADO.NET 中的 DataAdapte 类的作用？	19
25、	. 什么是 WEBSERVICE ？	20
26、	. 行下面代码后：	20
	String strTemp ="abcdefg 某某某";	20
27、	. . 叙述 ASP.NET 中的几种验证控件(equiredFieldValidato, CompareValidator, RangeValidator, RegularExpressionValitor, CustomValidator, ValidationSummary) 及作用？	20
28、	. 在.net (C# or vb.net) 中, Appplication.Exit 和 Form.Close 有什么 不同？	20
29、	. C#中的委托是什么？ 事件是不是一种委托？	20
30、	. 解释下 C#中的程序集？	20
31、	. 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?	20
32、	. SQL SERVER 的两种索引是何形式？ 索引的作用？ 索引的优缺点？	20
33、	. 谈谈 C/S 和 B/S 的特点	20
34、	. 分析类和结构的异同？	21
六、	英语面试题	21

一、Microsoft.Net 平台基础

1、GC 是什么？为什么要有 GC？

答：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。要请求垃圾收集，可以调用下面的方法之一：

```
System.gc()

Runtime.getRuntime().gc()
```

二、编程语言

1、String 和 StringBuilder 的区别？

System.String 类是一种传统的修改字符串的方式，它确实可以完成把一个字符串添加到另一个字符串上的工作没错。但是在 .NET 框架下，这个操作实在是划不来。因为系统先是将两个字符串写入内存，接着删除原来的 String 对象，然后创建一个 String 对象，并读取内存中的数据赋给该对象。这一来二去的，耗了不少时间。

而使用 System.Text 命名空间下面的 **StringBuilder** 类就不是这样了，它提供的 Append 方法，能够在已有对象的原地进行字符串的修改，简单而且直接。如果想要编写将连续操作依次连接起来的单个语句，这将很方便。

如果要操作一个不断增长的字符串，请你不要再用 String 类了。为什么涕淌要这样说呢？我们来看看两个类的工作原理。

String 类是一种传统的修改字符串的方式，它确实可以完成把一个字符串添加到另一个字符串上的工作没错。但是在 .NET 框架下，这个操作实在是划不来。因为系统先是将两个字符串写入内存，接着删除原来的 String 对象，然后创建一个 String 对象，并读取内存中的数据赋给该对象。这一来二去的，耗了不少时间。

而使用 System.Text 命名空间下面的 StringBuilder 类就不是这样了，它提供的 Append 方法，能够在已有对象的原地进行字符串的修改，简单而且直接。

当然，一般情况下你是觉察不出这二者效率的差异的，但如果你要对某个字符串进行大量的添加操作，那么用 StringBuilder 类所耗费的时间和 String 类简直不是一个数量级的。下面是一个例子，你自己试一试：

2、类的 static 型构造方法是干什么用的？

静态构造函数用于初始化任何静态数据，或用于执行仅需执行一次的特定操作。在创建第一个实例或引用任何静态成员之前，将自动调用静态构造函数

3、 `String s = new String("xyz");`创建了几个 `String` Object?

答：两个对象，一个是“xyz”，一个是指向“xyz”的引用对象 s。

4、 `abstract class` 和 `interface` 有什么区别?

答：声明方法的存在而不去实现它的类被叫做抽象类（`abstract class`），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该方法的情况。不能创建 `abstract` 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。`Abstract` 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（`interface`）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 `static final` 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，`instanceof` 运算符可以用来决定某对象的类是否实现了接口。

5、 `.启动一个线程是用 run() 还是 start()?`

答：启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

6、 `.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)?`

答：接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

7、. 构造器 Constructor 是否可被 override?

答: 构造器 Constructor 不能被继承, 因此不能重写 Overriding, 但可以被重载 Overloading。

8、. 是否可以继承 String 类?

答: String 类是 final 类故不可以继承。

9、. 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?

答: 会执行, 在 return 前执行。

10、. 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?

答: 不对, 有相同的 hash code。

11、. switch 是否能作用在 byte 上, 是否能作用在 long 上, 是否能作用在 String 上?

答: switch (expr) 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。long, string 都不能作用于 switch。

12、. List, Set, Map 是否继承自 Collection 接口?

答: List, Set 是 Map 不是

13、. Set 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用 == 还是 equals()? 它们

有何区别?

答: Set 里的元素是不能重复的, 那么用 iterator() 方法来区分重复与否。equals() 是判读两个 Set 是否相等。

equals() 和 == 方法决定引用值是否指向同一对象 equals() 在类中被覆盖, 为的是当两个分离的对象的内容和类型相配的话, 返回真值。

14、. 数组有没有 length() 这个方法? String 有没有 length() 这个方法?

答: 数组没有 length() 这个方法, 有 length 的属性。String 有 length() 这个方法。

15、.sleep() 和 wait() 有什么区别？

答：sleep() 方法是使线程停止一段时间的方法。在 sleep 时间间隔期满后，线程不一定立即恢复执行。这是因为在那个时刻，其它线程可能正在运行而且没有被调度为放弃执行，除非(a)“醒来”的线程具有更高的优先级

(b)正在运行的线程因为其它原因而阻塞。

wait() 是线程交互时，如果线程对一个同步对象 x 发出一个 wait() 调用，该线程会暂停执行，被调对象进入等待状态，直到被唤醒或等待时间到。

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

答：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。

17、.谈谈 final, finally, finalize 的区别。

答：

final—修饰符（关键字）如果一个类被声明为 final，意味着它不能再派生出新的子类，不能作为父类被继承。因此 一个类不能既被声明为 abstract 的，又被声明为 final 的。将变量或方法声明为 final，可以保证它们在使用中 不被改变。被声明为 final 的变量必须在声明时给定初值，而在以后的引用中只能读取，不可修改。被声明为 final 的方法也同样只能使用，不能重载

finally—再异常处理时提供 finally 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 catch 子句就会 执行，然后控制就会进入 finally 块（如果有的话）。

finalize—方法名。Java 技术允许使用 finalize() 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理 工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 Object 类中定义的 ，因此所有的类都继承了它。子类覆盖 finalize() 方法以整理系统资源或者执行其他清理工作。finalize() 方法是在垃圾收集器删除对象之前对这个对象调用的。

18、.如何处理几十万条并发数据？

答：用存储过程或事务。取得最大标识的时候同时更新..注意主键不是自增量方式这种方法并发的时候是不会有重复主键的..取得最大标识要有一个存储过程来获取。

19、. 进程和线程的区别？

答：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。

20、. 堆和栈的区别？

答：

栈：由编译器自动分配、释放。在函数体中定义的变量通常在栈上。

堆：一般由程序员分配释放。用 new、malloc 等分配内存函数分配得到的就是在堆上。

21、. 成员变量和成员函数前加 static 的作用？

答：它们被称为常成员变量和常成员函数，又称为类成员变量和类成员函数。分别用来反映类的状态。比如类成员变量可以用来统计类实例的数量，类成员函数负责这种统计的动作。

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

答 . 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。

三、APS.NET 开发

23、.ASP.NET 与 ASP 相比，主要有哪些进步？

答：asp 解释形，aspx 编译型，性能提高，有利于保护源码。

24、.Session 有什么重大 BUG，微软提出了什么方法加以解决？

答：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。

25、. 产生一个 int 数组，长度为 100，并向其中随机插入 1-100，并且不能重复。

```

int[] intArr=new int[100];

ArrayList myList=new ArrayList();

Random rnd=new Random();

while(myList.Count<100)

{

    int num=rnd.Next(1,101);

    if(!myList.Contains(num))

        myList.Add(num);

}

for(int i=0;i<100;i++)

    intArr[i]=(int)myList[i];

```

26、. 列举 ASP.NET 页面之间传递值的几种方式。

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

2. 使用 Session 变量

3. 使用 Server.Transfer

27、. 一列数的规则如下: 1、1、2、3、5、8、13、21、34..... 求第 30 位数是多少, 用递归算法实现。

答: public class MainClass

```

{

    public static void Main()

    {

        Console.WriteLine(Foo(30));

    }

    public static int Foo(int i)

    {

        if (i <= 0)

            return 0;

```

```

        else if(i > 0 && i <= 2)

            return 1;

        else return Foo(i -1) + Foo(i - 2);

    }

}

```

28、.C#中的委托是什么？事件是不是一种委托？

答：

委托可以把一个方法作为参数代入另一个方法。

委托可以理解为指向一个函数的引用。

是，是一种特殊的委托

29、.override 与重载的区别

答：

override 与重载的区别。重载是方法的名称相同。参数或参数类型不同，进行多次重载以适应不同的需要

Override 是进行基类中函数的重写。为了适应需要。

30、. 如果在一个 B/S 结构的系统中需要传递变量值,但是又不能使用 Session、Cookie、Application,

您有几种方法进行处理？

答：

this.Server.Transfer

31、. 请编程遍历页面上所有 TextBox 控件并给它赋值为 string.Empty?

答:

```

        foreach (System.Windows.Forms.Control control in this.Controls)

        {

            if (control is System.Windows.Forms.TextBox)

            {

                System.Windows.Forms.TextBox tb = (System.Windows.Forms.TextBox)control ;

                tb.Text = String.Empty ;
            }
        }

```

```
}
    }
```

32、8. 请编程实现一个冒泡排序算法？

答：

```
int [] array = new int [*] ;

int temp = 0 ;

for (int i = 0 ; i < array.Length - 1 ; i++)

{

for (int j = i + 1 ; j < array.Length ; j++)

{

if (array[j] < array[i])

{

temp = array[i] ;

array[i] = array[j] ;

array[j] = temp ;

}

}

}
```

33、. 描述一下 C# 中索引器的实现过程，是否只能根据数字进行索引？

答：不是。可以用任意类型。

34、. 求以下表达式的值，写出您想到的一种或几种实现方法： $1-2+3-4+\dots+m$

答：

```
int Num = this.TextBox1.Text.ToString() ;

int Sum = 0 ;

for (int i = 0 ; i < Num + 1 ; i++)

{

if((i%2) == 1)
```

```

{
    Sum += i ;
}

else
{
    Sum = Sum - I ;
}
}

System.Console.WriteLine(Sum.ToString());

System.Console.ReadLine() ;

```

35、. 用 .net 做 B/S 结构的系统,您是用几层结构来开发,每一层之间的关系以及为什么要这样分层?

答: 一般为 3 层

数据访问层, 业务层, 表示层。

数据访问层对数据库进行增删查改。

业务层一般分为二层, 业务表现层实现与表示层的沟通, 业务规则层实现用户密码的安全等。

表示层为了与用户交互例如用户添加表单。

优点: 分工明确, 条理清晰, 易于调试, 而且具有可扩展性。

缺点: 增加成本。

36、. 什么叫应用程序域?

答: 应用程序域可以理解成一种轻量级进程。起到安全的作用。占用资源小。

37、. 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。

38、. 什么是装箱和拆箱?

答: 从值类型接口转换到引用类型装箱。从引用类型转换到值类型拆箱。

39、. 什么是受管制的代码?

答: unsafe: 非托管代码。不经过 CLR 运行。

40、. 什么是强类型系统?

答: RTTI: 类型识别系统。

41、.net 中读写数据库需要用到那些类? 他们的作用?

答: DataSet: 数据存储器。

DataCommand: 执行语句命令。

DataAdapter: 数据的集合, 用语填充。

42、.ASP.net 的身份验证方式有哪些? 分别是什么原理?

答: 10。Windows(默认)用 IIS...From(窗体)用帐户...Passport(密钥)

43、. 什么是 Code-Behind 技术?

答: 代码后植。

44、. 在.net 中, 配件的意思是?

答: 程序集。(中间语言, 源数据, 资源, 装配清单)

45、. 常用的调用 Webservice 的方法有哪些?

答: 1. 使用 WSDL.exe 命令行工具。

2. 使用 VS.NET 中的 Add Web Reference 菜单选项

46、.net Remoting 的工作原理是什么?

答: 服务器端向客户端发送一个进程编号, 一个程序域编号, 以确定对象的位置。

47、. 在 C# 中, string str = null 与 string str = "" 请尽量使用文字或图象说明其中的区别。

答: string str = null 是不给他分配内存空间, 而 string str = "" 给它分配长度为空字符串的内存空间。

48、. 请详述在 dotnet 中类(class)与结构(struct)的异同?

答: Class 可以被实例化, 属于引用类型, 是分配在内存的堆上的, Struct 属于值类型, 是分配在内存的栈上的.

49、. SQLSERVER 服务器中, 给定表 table1 中有两个字段 ID、LastUpdateDate, ID 表示更新的事务号, LastUpdateDate 表示更新时的服务器时间, 请使用一句 SQL 语句获得最后更新的事务号

答: Select ID FROM table1 Where LastUpdateDate = (Select MAX(LastUpdateDate) FROM table1)

50、. 根据线程安全的相关知识, 分析以下代码, 当调用 test 方法时 i>10 时是否会引起死锁?并简要说明理由。

```
public void test(int i)
{
    lock(this)
    {
        if (i>10)
        {
            i--;
            test(i);
        }
    }
}
```

答: 不会发生死锁, (但有一点 int 是按值传递的, 所以每次改变的都只是一个副本, 因此不会出现死锁。但如果把 int 换做一个 object, 那么死锁会发生)

51、. 简要谈一下您对微软.NET 构架下 remoting 和 webservice 两项技术的理解以及实际中的应用。

答: WS 主要是可利用 HTTP, 穿透防火墙。而 Remoting 可以利用 TCP/IP, 二进制传送提高效率。

四、数据库

- 1、. 写出一条 Sql 语句：取出表 A 中第 31 到第 40 记录（SQLServer, 以自动增长的 ID 作为主键, 注意：ID 可能不是连续的。

答：解 1: select top 10 * from A where id not in (select top 30 id from A)

解 2: select top 10 * from A where id > (select max(id) from (select top 30 id from A)as A)

五、问题集锦

一. 填空题：

1. 类的三大特性是（封装）（继承）（多态）。
2. 在 SELECT 语句的 FROM 子句中最多可以指定（256）个表或视图，相互之间要用（逗号）分隔，当所查询的表不在当前数据库时，可用（数据库名.所有者名.表名）格式来指出表或视图对象。
3. 创建存储过程的命令是（create proc 存储过程名）删除表的命令是（drop 表名）。
4. 计算字段的累加和的函数是：（sum），统计项目数的函数是：（count）。
5. 关系数据库中，主键是（外键）。
6. 声明抽象类的关键字是（abstract），抽象类不能（sealed）。
7. .NET 运行库支持被称为（委托）的引用类型，其作用类似于 C++中函数指针的用途。
8. XML 的全称是（Extensible Markup Language）。
9. C#中所有的类型实质上都是从（Object）类派生而来的。
10. 声明静态变量的关键字是（static），声明常量的关键字是（const）。
11. C#中处理内存管理功能的是（托管代码）。
12. 在 C#中用于显式地将一个值或引用转换为另一种不同的类型称为（拆箱）。
13. 在 ADO.NET 中，command 对象是用（ExecuteNonQuery）（ExecuteScalar）（ExecuteDataReader）方法执行命令。
14. B/S 通常使用的结构设计模型分为（模型）（视图）（控制）。
15. SQL SERVER 中索引类型包括的三种类型分别是（主键索引）（唯一索引）（聚集索引）。

二. 简答题：

- 1、. 什么是中间语言（IL）？它的作用？

答：代码在执行前都要被编译为托管代码，与 CPU 无关的指令集。

其作用：支持平台无关性，还支持语言的互操作性。

- 2、. 什么是装箱和拆箱？

答：值类型转换成引用类型是装箱，引用类型转换成值类型是拆箱。

- 3、 出 ADO.NET 中读写数据库的主要的几个类？它们的作用？

答：Connection 数据库连接；Command 执行 SQL 语句或存储过程；DataAdapter 用于填充 DataSet 和更新数据库的一组数据命令和数据库连接，断开模式；DataReader，读取数据库信息，是只读向前的；DataSet 数据在内存中的缓存。

- 4、 C#中接口和类有什么异同？

答：**类**是引用类型，可以继承类、接口和被继承，有默认的构造函数，有析构函数，可以使用 abstract 和 sealed，有 protected 修饰符，必须使用 new 初始化。

结构是值类型，只能继承接口，不能被继承，没有默认的构造函数，可以创建，没有析构函数，不可以用 abstract 和 sealed，没有 protected 修饰符，可以不用 new 初始化。

如何选择使用结构还是类：

1. 堆栈的空间有限，对于大量的逻辑的对象，创建类要比创建结构好一些

2. 结构表示如点、矩形和颜色这样的轻量对象，例如，如果声明一个含有 1000 个点对象的数组，则将为引用每个对象分配附加的内存。在此情况下，结构的成本较低。

3. 在表现抽象和多级别的对象层次时，类是最好的选择

4. 大多数情况下该类型只是一些数据时，结构是最佳的选择

5、解释 ASP.NET 中的 code-behind?

答：代码隐藏，显示逻辑的代码和处理逻辑的代码分离

6、什么是 ASP.NET 中的用户控件？

答 是 asp.net 代码创建的控件，就像在标准的 ASP.NET Web 页面中创建控件一样，可以在多个 ASP.NET 页面中重用它们。

7、在 C# 中，string str = null 与 string str = "" 请尽量说明其中的区别。

答：string str=null 没有分配空间，string str="" 分配了空间

8、什么是受管制代码？

答：由公共语言运行环境进行管理的代码，它主要负责内存管理。在 .net 平台上运行的代码都是受管制代码

9、C# 中常用的访问形式？它们各自的作用域？

答：public 不受限制的访问。

private 只能在本类中访问

protected 本类和继承他的类中访问

internal 表示在同一个应用程序（application）或类库（library）中都能使用

10、SQL SERVER 中的存储过程的理解？

答：存储过程是 SQL 语句和可选控制流语句的预编译集合，以一个名称存储并作为一个单元处理。

存储过程存储在数据库内，可由应用程序通过一个调用执行，而且允许用户声明变量、有条件执行以及其它强大的编程功能。

11、CTS, CLS, CLR 分别作何解释？

答：CTS:公共类型系统. 主要是为了语言之间的进行标准化处理.

CLS:公共语言规范. 主要是确保代码可以在任何语言中访问的最小标准集体

CLR:公共语言运行时. 主要是管理代码.. 处理, 加载代码, 以及所有服务的代码

12、ASP.NET 和 ASP 的主要优势？

答：ASP.Net 的优势：

ASP.Net 摆脱了以前 ASP 使用脚本语言来编程的缺点，理论上可以使用任何编程语言包括 C++，VB，JS 等等，当然，最合适的编程语言还是 MS 为 .Net Framework 专门推出的 C(读 c sharp)，它可以看作是 VC 和 Java 的混合体吧，尽管 MS 自己讲 C# 内核中更多的象 VC，但实际上我还是认为它和 Java 更象一些吧。首先它是面向对象的编程语言，而不是一种脚本，所以它具有面向对象编程语言的一切特性，比如封装性、继承性、多态性等等，这就解决了刚才谈到的 ASP 的那些弱点。封装性使得代码逻辑清晰，易于管理，并且应用到 ASP.Net 上就可以使业务逻辑和 Html 页面分离，这样无论页面原型如何改变，业务逻辑代码都不必做任何改动；继承性和多态性使得代码的可重用性大大提高，你可以通过继承已有的对象最大限度保护你以前的投资。并且 C# 和 C++、Java 一样提供了完善的调试/纠错体系。

13、.NET 的错误处理机制是什么？

答：try{可能要出错的代码}

```
catch{ 扑捉到错误后的处理}
finally{ 不论代码是否出错都要执行}
```

14、. 解释 Page. IsPostBack 属性？

答：使用者第一次浏览这个网页时 Page. IsPostBack 会传回 False, 不是第一次浏览这个网页时就传回 True

15、. .net framework 包含哪几种技术？

答：Application Framework, Domain Framework

16、. 触发器分为事前触发和事后触发，有何区别？

答：事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。通常事前触发器可以获取事件之前和新的字段值

17、. 解释 SQL SERVER2000 中 HAVING 的用法？]

答：在分组时筛选数据

18、. 解释 SQL SERVER2000 中 GROUP BY 的用法？

答：分组

19、. 解释下 C#中的方法的重载？

答：重载方法：是指使用相同的方法名，但指定不同的参数（个数，类型）。

20、. 事务是什么？

答：在许多大型、关键的应用程序中，计算机每秒钟都在执行大量的任务。更为经常的不是这些任务本身，而是将这些任务结合在一起完成一个业务要求，称为事务。当某一个任务失败时，就恢复到指定的恢复点，这个叫回滚。

21、. 描述下 ASP. NET 中使用 DataReader 对象 和 DataSet 对象的区别？

答：dataset 是保存数据的数据结构，断开模式，而 DataReader 不承担保存数据的责任，需要自己手动关闭连接数据库，它只负责从数据源读取数据到本地而已，是只读向前的，它不是数据结构，而是网络通讯组件的高层封装。

22、. 分析类和结构的异同？

答：类是引用类型，可以继承类、接口和被继承，有默认的构造函数，有析构函数，可以使用 abstract 和 sealed，有 protected 修饰符，必须使用 new 初始化。

结构是值类型，只能继承接口，不能被继承，没有默认的构造函数，可以创建，没有析构函数，不可以用 abstract 和 sealed，没有 protected 修饰符，可以不用 new 初始化。

如何选择使用结构还是类：

1. 堆栈的空间有限，对于大量的逻辑的对象，创建类要比创建结构好一些
2. 结构表示如点、矩形和颜色这样的轻量对象，例如，如果声明一个含有 1000 个点对象的数组，则将为引用每个对象分配附加的内存。在此情况下，结构的成本较低。
3. 在表现抽象和多级别的对象层次时，类是最好的选择
4. 大多数情况下该类型只是一些数据时，结构是最佳的选择

23、. 解释 ADO. NET 中的使用 Connection 类的 CLOSE（）和 DISPOSE（）的区别？

答：close 只是关闭，dispose 是释放资源。

24、. 叙述 ADO. NET 中的 DataAdapte 类的作用？

答：断开模式连接数据库，并且填充 DataSet。

25、. 什么是 WEBSERVICE ?

答: web 服务, 是利用 SOAP (简单对象访问协议, Simple Object Access Protocol) 在 HTTP 上执行远程方法调用的一种方法, 也可以使用 WSDL (Web Service Description Language, Web 服务描述语言) 来完整的描述 Web 服务, 基于 XML 标准。

26、. 行下面代码后:

```
String strTemp = "abcdefg 某某某";  
int i = System.Text.Encoding.Default.GetBytes(strTemp).Length;  
int j = strTemp.Length;  
问: i=( 14) ; j=( 11)
```

原因是什么?

答: 汉字占 2 个字节, 所以 i=14

27、..叙述 ASP.NET 中的几种验证控件(RequiredFieldValidator, CompareValidator, RangeValidator, RegularExpressionValidator, CustomValidator, ValidationSummary)及作用?

答:

- a) RequiredFieldValidator (必须字段验证);
- b) CompareValidator (比较验证);
- c) RangeValidator (范围验证);
- d) RegularExpresionValidator (正则表达式);
- e) ValidationSummary (验证总结);
- f) CustomValidator (自定义验证)
- g)

28、. 在.net (C# or vb.net) 中, Application.Exit 和 Form.Close 有什么不同?

答: Application.Exit 是关闭进程, Form.Close 是关闭当前的窗口。

29、. C#中的委托是什么? 事件是不是一种委托?

答: 委托有点类似于指针, 但他是类型安全的, 通过对于方法特征和返回类型的声明, 封装了具有相同特征和返回类型的方法。不关心该方法是什么类型的对象, 甚至是静态的还是实例的。委托可以连接在一起, 可以对一个事件调用多个方法。不是, 事件是一种消息。

30、. 解释下 C#中的程序集?

答: 程序集是自我描述的安装单元, 由一个或多个文件组成, 一个程序集可以是一个包含元数据的 DLL 或 EXE, 也可以由多个文件组成, 例如资源文件、元数据、DLL 和 EXE。程序集的安装仅是复制所有的文件而已, 使用 xcopy 命令即可完成安装

31、. 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?

答: finally 里的代码是不论是否成功运行都会被执行的, 所以是在 return 之前执行。

32、. SQL SERVER 的两种索引是何形式? 索引的作用? 索引的优缺点?

答: 集聚索引, 非集聚索引。提高查询速度。但是会过多的占用磁盘空间。

33、. 谈谈 C/S 和 B/S 的特点

答: C/S 是客户端/服务器端, C/S 的程序通常也叫胖客户端, 也就是一个程序的大部分功能, 都在客户端实现, 而服务器端只实现一小部分功能。通过这点不难看出, C/S 的程序大部分在客户端实现, 对于服务器端的压力相对小一些, 服务器端可以节省一些。而且 C/S 的程序用窗口来做, 个人认为开发效率上快一点。但 C/S 的程序一大弊端就是, 必须要在客户端安装并部署程序才能运行, 也就是用户必须得到客户端程序才可以运行。所以得考虑不同系统之间的安装与配置。

B/S 是浏览器/服务器端, B/S 的程序通常也叫瘦客户端, 与 C/S 相反。B/S 的程序大部分功能都要在服务器端实现, 客户端只用来做辅助的控制功能。因为 B/S 的程序直接部署安装在服务器上。用户只需要有浏览器, 并知道网址就可以使用程序。客户端不必做任何配置和安装, 即可使用应用程序。而 B/S 也是有缺点的, 首先 B/S 大部分功能都要在服务器端实现, 对于服务器的要求就很高。服务器同时要应付用户访问与程序处理, 所以服务器的配置必须能够应付。另外就是 B/S 的程序是浏览器中执行, 所以 B/S 的程序操作客户端电脑的硬件, 一般就得安装 ActiveX 等插件在客户端才可以操作。因此 B/S 的程序对于操作硬件, 优势没有 C/S 的程序要好。

说出下面几个函数的区别:

`private void test(string str){...}` : 通过值传送变量

`private void test(ref string str){...}` : 值参数通过引用传递给方法, 必须先初始化 str

`private void test(out string str){...}`: str 通过引用被传送, 从 test 中返回时, 保留该方法对其的改变

34、. 分析类和结构的异同?

答: **类**是引用类型, 可以继承类、接口和被继承, 有默认的构造函数, 有析构函数, 可以使用 `abstract` 和 `sealed`, 有 `protected` 修饰符, 必须使用 `new` 初始化。

结构是值类型, 只能继承接口, 不能被继承, 没有默认的构造函数, 可以创建, 没有析构函数, 不可以用 `abstract` 和 `sealed`, 没有 `protected` 修饰符, 可以不用 `new` 初始化。

如何选择使用结构还是类:

1. 堆栈的空间有限, 对于大量的逻辑的对象, 创建类要比创建结构好一些
2. 结构表示如点、矩形和颜色这样的轻量对象, 例如, 如果声明一个含有 1000 个点对象的数组, 则将为引用每个对象分配附加的内存。在此情况下, 结构的成本较低。
3. 在表现抽象和多级别的对象层次时, 类是最好的选择
4. 大多数情况下该类型只是一些数据时, 结构是最佳的选择

27. 解释下在 ASP.NET 中 Request 的作用?

答: Request 对象的主要作用就是: 在服务器端接受并得到从客户端浏览器提交或上传的信息。

包括从 HTML 表格用 POST 方法或 GET 方法传递的参数、cookie 和用户认证。

28. C#中能够使用指针吗?如果不能使用说明理由,如果能使用说出使用方法?

答: 可以使用。只能在标记为 `unsafe` 关键字的地方使用, 任何方法、类、结构、成员都能标记为 `unsafe`, 也可以把方法中的一个代码块标记为 `unsafe`, 但要注意, 不能把局部变量本身标记为 `unsafe`, 除非在 `unsafe` 的方法或代码块中声明。

六、英语面试题

This is a list of questions I have gathered from other sources and created myself over a period of time from my experience, many of which I felt were incomplete or simply wrong. I have finally taken the time to go through each question and correct them to the best of my ability. However, please feel free to post feedback to challenge, improv

e, or suggest new questions. I want to thank those of you that have contributed quality questions and corrections thus far.

There are some question in this list that I do not consider to be good questions for an interview. However, they do exist on other lists available on the Internet so I felt compelled to keep them easy access.

General Questions

1. Does C# support multiple-inheritance?

No.

2. Who is a protected class-level variable available to?

It is available to any sub-class (a class inheriting this class).

3. Are private class-level variables inherited?

Yes, but they are not accessible. Although they are not visible or accessible via the class interface, they are inherited.

4. Describe the accessibility modifier "protected internal".

It is available to classes that are within the same assembly and derived from the specified base class.

5. What's the top .NET class that everything is derived from?

System.Object.

6. What does the term immutable mean?

The data value may not be changed. Note: The variable value may be changed, but the original immutable data value was discarded and a new data value was created in memory.

7. What's the difference between System.String and System.Text.StringBuilder classes?

System.String is immutable. System.StringBuilder was designed with the purpose of having a mutable string where a variety of operations can be performed.

8. What's the advantage of using System.Text.StringBuilder over System.String?

StringBuilder is more efficient in cases where there is a large amount of string manipulation. Strings are immutable, so each time a string is changed, a new instance in memory is created.

9. Can you store multiple data types in System.Array?

No.

10. What's the difference between the `System.Array.CopyTo()` and `System.Array.Clone()`?

The `Clone()` method returns a new array (a shallow copy) object containing all the elements in the original array. The `CopyTo()` method copies the elements into another existing array. Both perform a shallow copy. A shallow copy means the contents (each array element) contains references to the same object as the elements in the original array. A deep copy (which neither of these methods performs) would create a new instance of each element's object, resulting in a different, yet identical object.

11. How can you sort the elements of the array in descending order?

By calling `Sort()` and then `Reverse()` methods.

12. What's the .NET collection class that allows an element to be accessed using a unique key?

`HashTable`.

13. What class is underneath the `SortedList` class?

A sorted `HashTable`.

14. Will the finally block get executed if an exception has not occurred?

Yes.

15. What's the C# syntax to catch any possible exception?

A catch block that catches the exception of type `System.Exception`. You can also omit the parameter data type in this case and just write `catch {}`.

16. Can multiple catch blocks be executed for a single try statement?

No. Once the proper catch block processed, control is transferred to the finally block (if there are any).

17. Explain the three services model commonly know as a three-tier application.

Presentation (UI), Business (logic and underlying code) and Data (from storage or other sources).

Class Questions

1. What is the syntax to inherit from a class in C#?

Place a colon and then the name of the base class.

Example: `class MyNewClass : MyBaseClass`

2. Can you prevent your class from being inherited by another class?

Yes. The keyword "sealed" will prevent the class from being inherited.

3. Can you allow a class to be inherited, but prevent the method from being over-ridden?

Yes. Just leave the class public and make the method sealed.

4. What's an abstract class?

A class that cannot be instantiated. An abstract class is a class that must be inherited and have the methods overridden. An abstract class is essentially a blueprint for a class without any implementation.

5. When do you absolutely have to declare a class as abstract?

- . When the class itself is inherited from an abstract class, but not all base abstract methods have been overridden.
- . When at least one of the methods in the class is abstract.

6. What is an interface class?

Interfaces, like classes, define a set of properties, methods, and events. But unlike classes, interfaces do not provide implementation. They are implemented by classes, and defined as separate entities from classes.

7. Why can't you specify the accessibility modifier for methods inside the interface?

They all must be public, and are therefore public by default.

8. Can you inherit multiple interfaces?

Yes. .NET does support multiple interfaces.

9. What happens if you inherit multiple interfaces and they have conflicting method names?

It's up to you to implement the method inside your own class, so implementation is left entirely up to you. This might cause a problem on a higher-level scale if similarly named methods from different interfaces expect different data, but as far as the compiler cares you're okay.

To Do: Investigate

10. What's the difference between an interface and abstract class?

In an interface class, all methods are abstract - there is no implementation. In an abstract class some methods can be concrete. In an interface class, no accessibility modifiers are allowed. An abstract class may have accessibility modifiers.

11. What is the difference between a Struct and a Class?

Structs are value-type variables and are thus saved on the stack, additional overhead but faster retrieval. Another difference is that structs cannot inherit.

Method and Property Questions

1. What's the implicit name of the parameter that gets passed into the set method/property of a class?

Value. The data type of the value parameter is defined by whatever data type the property is declared as.

2. What does the keyword "virtual" declare for a method or property?

The method or property can be overridden.

How is method overriding different from method overloading?

When overriding a method, you change the behavior of the method for the derived class. Overloading a method simply involves having another method with the same name within the class.

3. Can you declare an override method to be static if the original method is not static?

No. The signature of the virtual method must remain the same. (Note: Only the keyword virtual is changed to keyword override)

4. What are the different ways a method can be overloaded?

Different parameter data types, different number of parameters, different order of parameters.

5. If a base class has a number of overloaded constructors, and an inheriting class has a number of overloaded constructors; can you enforce a call from an inherited constructor to a specific base constructor?

Yes, just place a colon, and then keyword base (parameter list to invoke the appropriate constructor) in the overloaded constructor definition inside the inherited class.

Events and Delegates

1. What's a delegate?

A delegate object encapsulates a reference to a method.

2. What's a multicast delegate?

A delegate that has multiple handlers assigned to it. Each assigned handler (method) is called.

XML Documentation Questions

1. Is XML case-sensitive?

Yes.

2. What's the difference between `//` comments, `/* */` comments and `///` comments?

Single-line comments, multi-line comments, and XML documentation comments.

3. How do you generate documentation from the C# file commented properly with a command-line compiler?

Compile it with the `/doc` switch.

Debugging and Testing Questions

1. What debugging tools come with the .NET SDK?

. CorDBG – command-line debugger. To use CorDbg, you must compile the original C# file using the `/debug` switch.

. DbgCLR – graphic debugger. Visual Studio .NET uses the DbgCLR.

2. What does `assert()` method do?

In debug compilation, `assert` takes in a Boolean condition as a parameter, and shows the error dialog if the condition is false. The program proceeds without any interruption if the condition is true.

3. What's the difference between the `Debug` class and `Trace` class?

Documentation looks the same. Use `Debug` class for debug builds, use `Trace` class for both debug and release builds.

4. Why are there five tracing levels in `System.Diagnostics.TraceSwitcher`?

The tracing dumps can be quite verbose. For applications that are constantly running you run the risk of overloading the machine and the hard drive. Five levels range from None to Verbose, allowing you to fine-tune the tracing activities.

5. Where is the output of `TextWriterTraceListener` redirected?

To the Console or a text file depending on the parameter passed to the constructor.

6. How do you debug an ASP.NET Web application?

Attach the `aspnet_wp.exe` process to the `DbgClr` debugger.

7. What are three test cases you should go through in unit testing?

- . Positive test cases (correct data, correct output).
- . Negative test cases (broken or missing data, proper handling).
- . Exception test cases (exceptions are thrown and caught properly).

8. Can you change the value of a variable while debugging a C# application?

Yes. If you are debugging via Visual Studio.NET, just go to Immediate window.

ADO.NET and Database Questions

1. What is the role of the DataReader class in ADO.NET connections?

It returns a read-only, forward-only rowset from the data source. A DataReader provides fast access when a forward-only sequential read is needed.

2. What are advantages and disadvantages of Microsoft-provided data providers in ADO.NET?

SQLServer.NET data provider is high-speed and robust, but requires SQL Server license purchased from Microsoft. OLE-DB.NET is universal for accessing other sources, like Oracle, DB2, Microsoft Access and Informix. OLE-DB.NET is a .NET layer on top of the OLE layer, so it's not as fastest and efficient as SqlServer.NET.

3. What is the wildcard character in SQL?

Let's say you want to query database with LIKE for all employees whose name starts with La. The wildcard character is %, the proper query with LIKE would involve 'La%'.

4. Explain ACID rule of thumb for transactions.

A transaction must be:

1. Atomic - it is one unit of work and does not depend on previous and following transactions.
2. Consistent - data is either committed or roll back, no "in-between" case where something has been updated and something hasn't.
3. Isolated - no transaction sees the intermediate results of the current transaction).
4. Durable - the values persist if the data had been committed even if the system crashes right after.
- 5.

6. What connections does Microsoft SQL Server support?

Windows Authentication (via Active Directory) and SQL Server authentication (via Microsoft SQL Server username and password).

7. Between Windows Authentication and SQL Server Authentication, which one is trusted and which one is untrusted?

Windows Authentication is trusted because the username and password are checked with the Active Directory, the SQL Server authentication is untrusted, since SQL Server is the only verifier participating in the transaction.

8. What does the Initial Catalog parameter define in the connection string?

The database name to connect to.

9. What does the Dispose method do with the connection object?

Deletes it from the memory.

To Do: answer better. The current answer is not entirely correct.

10. What is a pre-requisite for connection pooling?

Multiple processes must agree that they will share the same connection, where every parameter is the same, including the security settings. The connection string must be identical.

11.

Assembly Questions

1. How is the DLL Hell problem solved in .NET?

Assembly versioning allows the application to specify not only the library it needs to run (which was available under Win32), but also the version of the assembly.

2. What are the ways to deploy an assembly?

An MSI installer, a CAB archive, and XCOPY command.

3. What is a satellite assembly?

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

4. What namespaces are necessary to create a localized application?

System.Globalization and System.Resources.

5. What is the smallest unit of execution in .NET?

an Assembly.

6. When should you call the garbage collector in .NET?

As a good rule, you should not call the garbage collector. However, you could call the garbage collector when you are done using a large object (or set of objects) to force the garbage collector to dispose of those very large objects from memory. However, this is usually not a good practice.

7. How do you convert a value-type to a reference-type?

Use `boxing`.

8. What happens in memory when you `Box` and `Unbox` a value-type?

`Boxing` converts a value-type to a reference-type, thus storing the object on the heap. `Unboxing` converts a reference-type to a value-type, thus storing the value on the stack.