1、简述 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。  
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: 当前程序集内可以访问。

2、ADO.NET中的五个主要对象  
Connection：主要是开启程序和数据库之间的连接。没有利用连接对象将数据库打开，是无法从数据库中取得数据的。Close和Dispose的区别，Close以后还可以Open，Dispose以后则不能再用。  
Command：主要可以用来对数据库发出一些指令，例如可以对数据库下达查询、新增、修改、删除数据等指令，以及调用存在数据库中的存储过程等。这个对象是架构在Connection 对象上，也就是Command 对象是透过连接到数据源。  
DataAdapter：主要是在数据源以及DataSet 之间执行数据传输的工作，它可以透过Command 对象下达命令后，并将取得的数据放入DataSet 对象中。这个对象是架构在Command对象上，并提供了许多配合DataSet 使用的功能。  
DataSet：这个对象可以视为一个暂存区（Cache），可以把从数据库中所查询到的数据保留起来，甚至可以将整个数据库显示出来，DataSet是放在内存中的。DataSet 的能力不只是可以储存多个Table 而已，还可以透过DataAdapter对象取得一些例如主键等的数据表结构，并可以记录数据表间的关联。DataSet 对象可以说是ADO.NET 中重量级的对象，这个对象架构在DataAdapter对象上，本身不具备和数据源沟通的能力；也就是说我们是将DataAdapter对象当做DataSet 对象以及数据源间传输数据的桥梁。DataSet包含若干DataTable、DataTableTable包含若干DataRow。  
DataReader：当我们只需要循序的读取数据而不需要其它操作时，可以使用DataReader 对象。DataReader对象只是一次一笔向下循序的读取数据源中的数据，这些数据是存在数据库服务器中的，而不是一次性加载到程序的内存中的，只能（通过游标）读取当前行的数据，而且这些数据是只读的，并不允许作其它的操作。因为DataReader 在读取数据的时候限制了每次只读取一笔，而且只能只读，所以使用起来不但节省资源而且效率很好。使用DataReader 对象除了效率较好之外，因为不用把数据全部传回，故可以降低网络的负载。  
ADO.NET 使用Connection 对象来连接数据库，使用Command 或DataAdapter对象来执行SQL语句，并将执行的结果返回给DataReader 或 DataAdapter ,然后再使用取得的DataReader 或DataAdapter 对象操作数据结果。

3、列举ASP.NET 页面之间传递值的几种方式。  
1.使用QueryString, 如….?id=1; response. Redirect()….  
2.使用Session变量  
3.使用Server.Transfer  
4.Cookie传值  
5.Application传值  
6.PreviosPage

Server.Transfer和Response.Redirect的区别：Server.Transfer是服务器内部的转接，浏览器不知晓；Response.Redirect是有浏览器参与的，所以在地址栏中可以看到地址的变化。

4、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指针。  
委托和事件没有可比性，因为委托是类型，事件是对象，下面说的是委托的对象（用委托方式实现的事件）和（标准的event方式实现）事件的区别。事件的内部是用委托实现的。因为对于事件来讲，外部只能“注册自己+=、注销自己-=”，外界不可以注销其他的注册者，外界不可以主动触发事件，因此如果用Delegate就没法进行上面的控制，因此诞生了事件这种语法。事件是用来阉割委托实例的，类比用一个自定义类阉割List。事件只能add、remove自己，不能赋值。事件只能+=、-=，不能= 。加分的补充回答：事件内部就是一个private的委托和add、remove两个方法。

面试聊：用Reflector查看.Net的类的内部实现，解决问题。

5、override与重载（overload）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。重载（overload）是面向过程的概念。  
Override 是进行基类中函数的重写。Override是面向对象的概念

6、C#中索引器是否只能根据数字进行索引？是否允许多个索引器参数？  
参数的个数和类型都是任意的。加分的补充回答：用reflector反编译可以看出，索引器的内部本质上就是set\_item、get\_item方法。加分的补充回答：回答.net培训中讲解设计模式中开发的SettingsProvider就是用的string类型的参数名做索引器参数。

7、属性和public字段的区别是什么？调用set方法为一个属性设值，然后用get方法读取出来的值一定是set进去的值吗？  
属性可以对设值、取值的过程进行非法值控制，比如年龄禁止设值负数，而字段则不能进行这样的设置。虽然一般情况下get读取的值就是set设置的值，但是可以让get读取的值不是set设置的值的，极端的例子。Public Age{get{return 100;}set{}}。加分的补充回答：用reflector反编译可以看出，属性内部本质上就是set\_\*\*\*、get\_\*\*\*方法。  
class Person  
{  
public int Age  
{  
get  
{  
return 3;  
}  
set  
{

}  
}  
}  
Person p1 = new Person();  
p1.Age = 30;  
p1.Age++;  
Console.Write(p1.Age);//输出3

8、三层架构  
通常意义上的三层架构就是将整个业务应用划分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。  
区分层次的目的即为了“高内聚，低耦合”的思想。  
表现层（UI）：通俗讲就是展现给用户的界面，即用户在使用一个系统的时候的所见所得。  
业务逻辑层（BLL）：针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。  
数据访问层（DAL）：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等每层之间是一种垂直的关系。  
三层结构是N层结构的一种，一般来说，层次之间是向下依赖的，下层代码未确定其接口（契约）前，上层代码是无法开发的，下层代码接口（契约）的变化将使上层的代码一起变化。  
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。

9、MVC模式 （\*）  
MVC(Model View Controller)模型－视图－控制器  
aspx就是View，视图；Model：DataSet、Reader、对象；Controller：cs代码。

MVC是典型的平行关系，没有说谁在上谁在下的关系，模型负责业务领域的事情，视图负责显示的事情，控制器把数据读取出来填充模型后把模型交给视图去处理。而各种验证什么的应该是在模型里处理了。它强制性的使应用程序的输入、处理和输出分开。MVC最大的好处是将逻辑和页面分离。

10、什么是装箱(boxing)和拆箱(unboxing)？ （\*）  
装箱：从值类型接口转换到引用类型。  
拆箱：从引用类型转换到值类型。  
object obj = null;//引用类型  
obj = 1;//装箱，boxing。把值类型包装为引用类型。  
int i1 = (int)obj;//拆箱。unboxing

11、什么叫应用程序域(AppDomain) （\*）  
一种边界，它由公共语言运行库围绕同一应用程序范围内创建的对象建立（即，从应用程序入口点开始，沿着对象激活的序列的任何位置）。  
应用程序域有助于将在一个应用程序中创建的对象与在其他应用程序中创建的对象隔离，以使运行时行为可以预知。  
在一个单独的进程中可以存在多个应用程序域。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。

12、CTS、CLS、CLR分别作何解释（\*）  
CTS：Common Type System 通用系统类型。Int32、Int16→int、String→string、Boolean→bool  
CLS：Common Language Specification 通用语言规范。不同语言语法的不同。  
CLR：Common Language Runtime 公共语言运行时，就是.Net提供的那些 类。

13、在dotnet中类(class)与结构(struct)的异同？  
Class可以被实例化,属于引用类型,是分配在内存的堆上的。类是引用传递的。  
Struct属于值类型,是分配在内存的栈上的。结构体是复制传递的。加分的回答：Int32、Boolean等都属于结构体。

14、堆和栈的区别？  
栈是编译期间就分配好的内存空间，因此你的代码中必须就栈的大小有明确的定义；局部值类型变量、值类型参数等都在栈内存中。  
堆是程序运行期间动态分配的内存空间，你可以根据程序的运行情况确定要分配的堆内存的大小。

15、能用foreach遍历访问的对象的要求  
需要实现IEnumerable接口或声明GetEnumerator方法的类型。

16、GC是什么? 为什么要有GC?  
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。GC只能处理托管内存资源的释放，对于非托管资源则不能使用GC进行回收，必须由程序员手工回收，一个例子就是FileStream或者SqlConnection需要程序员调用Dispose进行资源的回收。  
要请求垃圾收集，可以调用下面的方法：GC.Collection()一般不需要手动调用GC.Collection()。加分的回答：有一次用OLEAutomation的方式操作Excel的时候，Excel经常不能正常退出，在MSDN网站上找到官方解答说要调用一下GC.Collection()，至于为什么这样就可以也没找到答案。

17、String s = new String(“xyz”);创建了几个String Object?  
两个对象，一个是“xyx”,一个是指向“xyx”的引用对象。

18、值类型和引用类型的区别？  
1.将一个值类型变量赋给另一个值类型变量时，将复制包含的值。引用类型变量的赋值只复制对对象的引用，而不复制对象本身。  
2.值类型不可能派生出新的类型：所有的值类型均隐式派生自 System.ValueType。但与引用类型相同的是，结构也可以实现接口。  
3.值类型不可能包含 null 值：然而，可空类型功能允许将 null 赋给值类型。  
4.每种值类型均有一个隐式的默认构造函数来初始化该类型的默认值。

19、C#中的接口和类有什么异同。

不同点：  
不能直接实例化接口。  
接口不包含方法的实现。  
接口可以多继承，类只能单继承。  
类定义可在不同的源文件之间进行拆分。  
相同点：  
接口、类和结构都可以从多个接口继承。  
接口类似于抽象基类：继承接口的任何非抽象类型都必须实现接口的所有成员。  
接口和类都可以包含事件、索引器、方法和属性。

20、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。  
接口可以用于支持回调（CallBack）；抽象类不能实现回调，因为继承不支持。  
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。  
接口可以作用于值类型(Struct)和引用类型(Class)；抽象类只能作用于引用类型。例如，Struct就可以继承接口，而不能继承类。  
加分的补充回答：讲设计模式的时候SettingsProvider的例子。

21、是否可以继承String类?  
String类是sealed类故不可以继承。

22、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?  
会执行，在return前执行。  
加分的补充回答（也助记）：读取数据库中数据的条数的程序  
public int QueryCount()  
{  
…..  
try  
{  
return cmd.ExecuteScalar();  
}  
finally  
{  
cmd.Dispose();  
}  
}  
如果C#设计的是先执行cmd.Dispose()再执行return就会出现return执行失败了，因为cmd已经Dispose了。

23、new 关键字用法（\*）  
new 运算符 用于创建对象和调用构造函数。  
new 修饰符 用于向基类成员隐藏继承成员。  
new 约束 用于在泛型声明中约束可能用作类型参数的参数的类型。

24、如何把一个Array复制到ArrayList里（\*）

实现1 string[] s ={ “111″, “22222″ }; ArrayList list = new ArrayList(); list.AddRange(s);  
实现2 string[] s ={ “111″, “22222″ }; ArrayList list = new ArrayList(s);

25、描述线程与进程的区别？（\*）  
1.线程(Thread)与进程（Process）二者都定义了某种边界，不同的是进程定义的是应用程序与应用程序之间的边界，不同的进程之间不能共享代码和数据空间，而线程定义的是代码执行堆栈和执行上下文的边界。  
2.一个进程可以包括若干个线程，同时创建多个线程来完成某项任务，便是多线程。而同一进程中的不同线程共享代码和数据空间。用一个比喻来说，如果一个家庭代表一个进程，在家庭内部，各个成员就是线程，家庭中的每个成员都有义务对家庭的财富进行积累，同时也有权利对家庭财富进行消费，当面对一个任务的时候，家庭也可以派出几个成员来协同完成，而家庭之外的人则没有办法直接消费不属于自己家庭的财产。

26、什么是强类型，什么是弱类型？哪种更好些？为什么?  
C#中  
int i=3;  
i=”a”;  
不可以  
JavaScript中  
var i=3;  
i=”a”;  
可以  
强类型是在编译的时候就确定类型的数据，在执行时类型不能更改，而弱类型在执行的时候才会确定类型。  
没有好不好，二者各有好处，强类型安全，因为它事先已经确定好了，而且效率高。弱类型更灵活，但是效率低，而且出错概率高  
一般用于编译型编程语言，如c++,java,c#,pascal等,弱类型相比而言不安全，在运行的时候容易出现错误，但它灵活，多用于解释型编程语言，如javascript,vb等

27、什么是反射?  
程序集包含模块，而模块又包括类型，类型下有成员，反射就是管理程序集，模块，类型的对象，它能够动态的创建类型的实例，设置现有对象的类型或者获取现有对象的类型，能调用类型的方法和访问类型的字段属性。它是在运行时创建和使用类型实例  
加分的补充回答：聊用反射实现编辑器动态插件和如鹏网项目中插件体系实现第三方支付和定时任务的案例。

28、int、DateTime、string是否可以为null?  
int、DateTime不能，因为其为Struct类型，而结构属于值类型，值类型不能为null,只有引用类型才能被赋值null。string可以为null。

29、using关键字有什么用？什么是IDisposable？  
using可以声明namespace的引入，还可以实现非托管资源的释放，实现了IDisposiable的类在using中创建，using结束后会自动调用该对象的Dispose方法，释放资源。加分的补充回答：using其实等价于try……finally，用起来更方便。  
int?→Nullable

30、Assembly.Load(“foo.dll”); 这句话是否正确？（\*）  
错误，正确的应该是Assembly.Load(“foo”); 或者Assembly.LoadFrom(“foo.dll”);

31、XML 与 HTML 的主要区别  
1. XML是区分大小写字母的，HTML不区分。  
2. 在HTML中，如果上下文清楚地显示出段落或者列表键在何处结尾，那么你可以省略

或者

之类的结束 标记。在XML中，绝对不能省略掉结束标记。  
HTML：

XML：  
  
  
3. 在XML中，拥有单个标记而没有匹配的结束标记的元素必须用一个 / 字符作为结尾。这样分析器就知道不用 查找结束标记了。  
4. 在XML中，属性值必须分装在引号中。在HTML中，引号是可用可不用的。  
5. 在HTML中，可以拥有不带值的属性名。在XML中，所有的属性都必须带有相应的值。

XML是用来存储和传输数据的  
HTML是用来显示数据的

32、string str = null 与 string str = “”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。 string str = null没有string对象，string str = “”有一个字符串对象。

33.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)  
解答3：ROW\_NUMBER（推荐）

34.面向对象的语言具有\_\_\_\_\_\_\_\_性、\_\_\_\_\_\_\_\_\_性、\_\_\_\_\_\_\_\_性  
答：封装、继承、多态。  
不要背，脑子中要有联想。

35.在.Net中所有可序列化的类都被标记为\_\_\_\_\_?  
答：[serializable]  
36.在.Net托管代码中我们不用担心内存漏洞，这是因为有了\_\_\_\_\_\_?  
答：GC。

37、什么叫应用程序域？什么是受管制的代码？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？  
应用程序域为安全性、可靠性、版本控制以及卸载程序集提供了隔离边界。应用程序域通常由运行库宿主创建，运行库宿主负责在运行应用程序之前引导公共语言运行库。应用程序域提供了一个更安全、用途更广的处理单元，公共语言运行库可使用该单元提供应用程序之间的隔离。  
受管制的代码：在.Net环境中运行的任何代码都是受管制的代码（managed code），.Net外部的代码也运行在windows上，这些代码称为未受管制的代码（unmanaged code）。  
使用基于公共语言运行库的语言编译器开发的代码称为托管代码；托管代码具有许多优点，例如：跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。  
强类型语言是能够禁止任何违反类型系统的代码的语言，或者说是能够捕获所有违反类型系统的错误的语言。我们说C++相对于C是强类型的，是因为C++禁止了一些C中的隐式转换，比如将void\*转换为任意的指针类型。  
装箱和拆箱使值类型能够被视为对象。对值类型装箱将把该值类型打包到 Object 引用类型的一个实例中。这使得值类型可以存储于垃圾回收堆中。拆箱将从对象中提取值类型。  
每个类型成员都有一个唯一的签名。方法签名由方法名称和一个参数列表（方法的参数的顺序和类型）组成。只要签名不同，就可以在一种类型内定义具有相同名称的多种方法。当定义两种或多种具有相同名称的方法时，就称作重载。  
CTS通用类型系统 (common type system)  
一种确定公共语言运行库如何定义、使用和管理类型的规范。  
CLR公共语言运行库  
.NET Framework 提供了一个称为公共语言运行库的运行时环境，它运行代码并提供使开发过程更轻松的服务。  
CLS公共语言规范  
要和其他对象完全交互，而不管这些对象是以何种语言实现的，对象必须只向调用方公开那些它们必须与之互用的所有语言的通用功能。为此定义了公共语言规范 (CLS)，它是许多应用程序所需的一套基本语言功能。

38、什么是code-Behind技术。  
就是代码隐藏，在ASP.NET中通过ASPX页面指向CS文件的方法实现显示逻辑和处理逻辑的分离，这样有助于web应用程序的创建。比如分工，美工和编程的可以个干各的，不用再像以前asp那样都代码和html代码混在一起，难以维护。

39、接口是一种引用类型，在接口中可以声明（ a），但不可以声明公有的域或私有的成员变量。  
a) 方法、属性、索引器和事件；

c) 索引器和字段；  
d) 事件和字段；

40. 在ADO.NET中，对于Command对象的ExecuteNonQuery()方法和ExecuteReader()方法，下面叙述错误的是（c）。  
a) insert、update、delete等操作的Sql语句主要用ExecuteNonQuery()方法来执行；  
b) ExecuteNonQuery()方法返回执行Sql语句所影响的行数。  
c) Select操作的Sql语句只能由ExecuteReader()方法来执行；  
d) ExecuteReader()方法返回一个DataReder对象；

解答：ExecuteScalar→select getdate()。

41. 下列关于C#中索引器理解正确的是(c )  
a) 索引器的参数必须是两个或两个以上  
b) 索引器的参数类型必须是整数型  
c) 索引器没有名字  
d) 以上皆非

42. 要创建多文档应用程序，需要将窗体的(d )属性设为true。  
a) DrawGrid；  
b) ShowInTaskbar；  
c) Enabled；  
d) IsMdiContainer；

43. 如果设treeView1=new TreeView()，则treeView1.Nodes.Add(“根节点”)返回的是一个 (a)类型的值。  
a) TreeNode；  
b) int；  
c) string；  
d) TreeView；

44. 下面关于XML的描述错误的是（d）。  
a) XML提供一种描述结构化数据的方法；  
b) XML 是一种简单、与平台无关并被广泛采用的标准；  
c) XML文档可承载各种信息；  
d) XML只是为了生成结构化文档；

45、以下的C#代码，试图用来定义一个接口：  
public interface IFile  
{  
int A;  
int delFile()  
{  
A = 3;  
}  
void disFile();  
}  
关于以上的代码，以下描述错误的是(d )。  
a) 以上的代码中存在的错误包括：不能在接口中定义变量，所以int A代码行将出现错误；  
b) 以上的代码中存在的错误包括：接口方法delFile是不允许实现的，所以不能编写具体的实现函数；  
c) 代码void disFile();声明无错误，接口可以没有返回值；  
d) 代码void disFile();应该编写为void disFile(){}；

空和null不一样。

46 在ASP.NET中有Button控件myButton，要是单击控件时，导航到其他页面http://www.abc.com, 正确的代码为( c)。  
a) private void myButton\_Click(object sender, System.EventArgs e){Redirect(“http://www.abc.com”);}  
b) private void myButton\_Click(object sender, System.EventArgs e){Request.Redirect(“http://www.abc.com”);}  
c) private void myButton\_Click(object sender, System.EventArgs e){Reponse.Redirect(“http://www.abc.com”);}  
d) private void myButton\_Click(object sender, System.EventArgs e){Request.Redirect(“http://www.abc.com”);return true;}

47. 声明一个委托public delegate int myCallBack(int x); 则用该委托产生的回调方法的原型应该是（b ）。  
a) void myCallBack(int x) ；  
b) int receive(int num) ；  
c) string receive(int x) ；  
d) 不确定的；

48.StringBuilder 和 String 的区别？  
答：String 在进行运算时（如赋值、拼接等）会产生一个新的实例，而 StringBuilder 则不会。所以在大量字符串拼接或频繁对某一字符串进行操作时最好使用 StringBuilder，不要使用 String  
如果要操作一个不断增长的字符串，尽量不用String类,改用StringBuilder类。两个类的工作原理不同: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。因为系统先是把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，一般情况下觉察不到这二者效率的差异，但如果你要对某个字符串进行大量的添加操作，那么StringBuilder类所耗费的时间和String类简直不是一个数量级的。

49、.请叙述属性与索引器的区别。 （\*）没啥意义  
属性 索引器  
通过名称标识。 通过签名标识。  
通过简单名称或成员访问来访问。 通过元素访问来访问。  
可以为静态成员或实例成员。 必须为实例成员。  
属性的 get 访问器没有参数。 索引器的 get 访问器具有与索引器相同的形参表。  
属性的 set 访问器包含隐式 value 参数。 除了 value 参数外，索引器的 set 访问器还具有与索引器相同的形参表。

50.请叙述const与readonly的区别。（\*）  
const 关键字用于修改字段或局部变量的声明。它指定字段或局部变量的值不能被修改。常数声明引入给定类型的一个或多个常数。  
const数据成员的声明式必须包含初值，且初值必须是一个常量表达式。因为它是在编译时就需要完全评估。  
const成员可以使用另一个const成员来初始化，前提是两者之间没有循环依赖。  
readonly在运行期评估赋值，使我们得以在确保“只读访问”的前提下，把object的初始化动作推迟到运行期进行。  
readonly 关键字与 const 关键字不同：　const 字段只能在该字段的声明中初始化。readonly 字段可以在声明或构造函数中初始化。因此，根据所使用的构造函数，readonly 字段可能具有不同的值。另外，const 字段是编译时常数，而 readonly 字段可用于运行时常数。  
　　readonly 只能在声明时或者构造函数里面初始化，并且不能在 static 修饰的构造函数里面。  
参考资料：

http://www.cnblogs.com/royenhome/archive/2010/05/22/1741592.html

http://www.cnblogs.com/jams742003/archive/2010/03/23/1692913.html

http://www.cnblogs.com/wayfarer/archive/2006/04/27/386658.html

51． 请解释ASP。NET中的web页面与其隐藏类之间的关系？  
一个ASP.NET页面一般都对应一个隐藏类,一般都在ASP.NET页面的声明中指定了隐藏类例如一个页面Tst1.aspx的页面声明如下  
<%@ Page language="c#" Codebehind="Tst1.aspx.cs" AutoEventWireup="false" Inherits="T1.Tst1" %>  
Codebehind=”Tst1.aspx.cs” 表明经编译此页面时使用哪一个代码文件  
Inherits=”T1.Tst1″ 表用运行时使用哪一个隐藏类

52． 什么是viewstate，能否禁用？是否所用控件都可以禁用?  
Viewstate是保存状态的一种机制，EnableViewState属性设置为false即可禁用。最有力的举例子：Label、TextBox在禁用ViewState后不同的表现来说明。

53、你对web service的体会？  
答：如果自己写的一些程序也希望别人可以通过Web服务来使用，也可以把自己编写的方法贴上标签[WebMethed]来实现Web 服务。[是当编写程序时，希望实现一些别的网站已经实现过的，也用Web服务可视成XML语言的编码。可以使用别人的编码生成的XML找到自己需要的信息，来实现自己编写的程序的一些功能。  
SOAP、WSDL。

54您在什么情况下会用到虚方法？它与接口有什么不同？  
答案：子类重新定义父类的某一个方法时，必须把父类的方法定义为virtual  
在定义接口中不能有方法体，虚方法可以。  
实现时,子类可以不重新定义虚方法，但如果一个类继承接口，那必须实现这个接口。

讲到这里了  
55. 不定项选择：  
(1) 以下叙述正确的是： B C  
A. 接口中可以有虚方法。B. 一个类可以实现多个接口。 C. 接口不能被实例化。 D. 接口中可以包含已实现的方法。  
(2) 从数据库读取记录，你可能用到的方法有：B C D  
A. ExecuteNonQuery B. ExecuteScalar C. Fill D. ExecuteReader  
（3）.对于一个实现了IDisposable接口的类，以下哪些项可以执行与释放或重置非托管资源相关的应用程序定义的任务？(多选) ( ABC )  
A.Close B.Dispose C.Finalize  
D.using E.Quit  
（4）以下关于ref和out的描述哪些项是正确的？(多选) ( ACD )  
A.使用ref参数，传递到ref参数的参数必须最先初始化。  
B.使用out参数，传递到out参数的参数必须最先初始化。  
C.使用ref参数，必须将参数作为ref参数显式传递到方法。  
D.使用out参数，必须将参数作为out参数显式传递到方法。  
讲到这里  
56. 单项选择：  
（1）下列选项中，（C）是引用类型。  
a)enum类型 b)struct类型 c)string类型 d)int类型

（2）.关于ASP.NET中的代码隐藏文件的描述正确的是（C）  
a)Web窗体页的程序的逻辑由代码组成，这些代码的创建用于与窗体交互。编程逻辑唯一与用户界面不同的文件中。该文件称作为“代码隐藏”文件，如果用C＃创建，该文件将具有“.ascx.cs”扩展名。  
b)项目中所有Web窗体页的代码隐藏文件都被编译成.EXE文件。  
c)项目中所有的Web窗体页的代码隐藏文件都被编译成项目动态链接库（.dll）文件。  
d)以上都不正确。

（3）.下列描述错误的是（D）  
a)类不可以多重继承而接口可以；  
b)抽象类自身可以定义成员而接口不可以；  
c)抽象类和接口都不能被实例化；  
d)一个类可以有多个基类和多个基接口；

57、DataReader和DataSet的异同？  
DataReader使用时始终占用SqlConnection,在线操作数据库  
每次只在内存中加载一条数据,所以占用的内存是很小的  
是只进的、 只读的  
DataSet则是将数据一次性加载在内存中.抛弃数据库连接..读取完毕即放弃数据库连接(非连接模式)  
DataSet将数据全部加载在内存中.所以比较消耗内存…但是确比DataReader要灵活..可以动态的添加行,列,数据.对数据库进行 回传更新操作(动态操作读入到内存的数据)

58、public static const; int A = 1;这段代码有错误么？  
错误：const不能被修饰为static ；因为定义为常量 （const ）后就是静态的（static ）。

59传入某个属性的set方法的隐含参数的名称是什么？  
value，它的类型和属性所声名的类型相同。

60、C#支持多重继承么？  
类之间不支持，接口之间支持。类对接口叫做实现，不叫继承。

61、C#中所有对象共同的基类是什么？  
System.Object

62、通过超链接怎样传递中文参数？  
答：用URL编码，通过QueryString传递，用urlencode编码 用urldecode解码

63、string、String；int、Int32；Boolean、bool的区别  
String、Int32、Boolean等都属于.Net中定义的类，而string、int、bool相当于C#中对这些类定义的别名。CTS。

64、Server.Transfer和Response.Redirect的区别是什么？（常考）  
答: Server.Transfer仅是服务器中控制权的转向，在客户端浏览器地址栏中不会显示出转向后的地址；Response.Redirect则是完全的跳转，浏览器将会得到跳转的地址，并重新发送请求链接。这样，从浏览器的地址栏中可以看到跳转后的链接地址。  
Server.Transfer是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。 这个过程中浏览器和Web服务器之间经过了一次交互。  
Response.Redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求。这个过程中浏览器和Web服务器之间经过了两次交互。

65、不是说字符串是不可变的吗？string s=”abc”;s=”123″不就是变了吗？  
String是不可变的在这段代码中，s原先指向一个String对象，内容是 “abc”，然后我们将s指向”123″，那么s所指向的那个对象是否发生了改变呢？答案是没有。这时，s不指向原来那个对象了，而指向了另一个 String对象，内容为”123″，原来那个对象还存在于内存之中，只是s这个引用变量不再指向它了。

66、是否可以从一个static方法内部发出对非static方法的调用？  
不可以。因为非static方法是要与对象关联在一起的，必须创建一个对象后，才可以在该对象上进行方法调用，而static方法调用时不需要创建对象，可以直接调用。也就是说，当一个static方法被调用时，可能还没有创建任何实例对象，如果从一个static方法中发出对非static方法的调用，那个非static方法是关联到哪个对象上的呢？这个逻辑无法成立，所以，一个static方法内部不能发出对非static方法的调用。

67、.Net中会存在内存泄漏吗，请简单描述。 (\*)  
所谓内存泄露就是指一个不再被程序使用的对象或变量一直被占据在内存中。.Net中有垃圾回收机制，它可以保证一对象不再被引用的时候，即对象编程了孤儿的时候，对象将自动被垃圾回收器从内存中清除掉。虽然.Net可以回收无用的对象，但是.Net仍然存在由于使用不当导致的内存泄露问题。.Net中的内存泄露的情况：长生命周期的对象持有短生命周期对象的引用就很可能发生内存泄露，尽管短生命周期对象已经不再需要，但是因为长生命周期对象持有它的引用而导致不能被回收，这就是.Net中内存泄露的发生场景，通俗地说，就是程序员可能创建了一个对象，以后一直不再使用这个对象，这个对象却一直被引用，即这个对象无用但是却无法被垃圾回收器回收的，这就是.Net中可能出现内存泄露的情况，例如，缓存系统，我们加载了一个对象放在缓存中(例如放在一个全局Dictionary对象中)，然后一直不再使用它，这个对象一直被缓存引用，但却不再被使用。  
formManager  
{  
public static IList

窗体顶端

forms = new …..;  
}  
扩展：使用弱引用那么及时被引用也可以被回收。

窗体底端

68说出一些常用的类、接口，请各举5个  
要让人家感觉你对.Net开发很熟，所以，不能仅仅只列谁都能想到的那些东西，要多列你在做项目中涉及的那些东西。就写你最近写的那些程序中涉及的那些类。  
常用的类：StreamReader、WebClient、Dictionary、StringBuilder、SqlConnection、FileStream、File、Regex、List  
常用的接口：IDisposable、IEnumerable、IDbConnection、IComparable、ICollection、IList、IDictionary  
要出乎意料！不要仅仅完成任务！

69、BS与CS的联系与区别。  
C/S是Client/Server的缩写。客户端需要安装专用的客户端软件。  
B/Ｓ是Brower/Server的缩写，客户机上只要安装一个浏览器。在这种结构下，用户界面完全通过WWW浏览器实现，一部分事务逻辑在前端实现，但是主要逻辑在服务器端实现。浏览器通过Ｗeb Server 同数据库进行数据交互。  
C/S 与 B/S 区别：  
１）．硬件环境不同:  
　　C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务.  
　　B/S 建立在广域网之上的, 不必是专门的网络硬件环境,例与电话上网, 租用设备. 信息自己管理. 有比C/S更强的适应范围, 一般只要有操作系统和浏览器就行  
２)．对安全要求不同  
　　C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用C/S 结构适宜. 可以通过B/S发布部分可公开信息.  
　　B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。  
3)．处理问题不同  
　　C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统。C/S的客户端由于是本地程序，因此和本地硬件、程序的交互性很强，比如可以控制本机的其他程序、可以读写本地磁盘文件、可以与硬件交互。  
　　B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S无法作到的. 与操作系统平台关系最小，正应为如此B/S很难和本地硬件、程序、文件进行交互，比如很难控制本机的其他程序、很难读写本地磁盘文件、很难与硬件交互，当然可以用ActiveX技术解决，比如网银等就是这么解决的，这样做的问题就是可能会被客户拒绝，而且只局限于Windows操作系统。

C/S也分为两层架构和三层架构。两层架构：客户端程序程序直连数据库；三层架构：客户端访问服务端的服务，主要逻辑代码写在服务端的服务中，然后服务端服务再去访问数据库，Oracle版分布式呼叫中心，WCF。

70、编写一个单例（Singleton）类。  
参考复习ppt。  
把构造函数设置为private，设置一个public、static的对象实例  
public FileManager  
{  
private FileManager(){}  
public static FileManager Instance = new FileManager();  
}

扩展：搜“C# Singleton”

71、什么是sql注入？如何避免sql注入？  
用户根据系统的程序构造非法的参数从而导致程序执行不是程序员期望的恶意SQL语句。使用参数化的SQL就可以避免SQL注入。  
详细参考复习ppt。举例子，摆事实！  
1′ or 1=1

72、数据库三范式是什么?  
第二范式：表必须有主键。  
第三范式：表引用其他的表必须通过主键引用。

73、说出一些数据库优化方面的经验?  
索引内部原理：想象成Dictionary，插入、删除、更新的速度慢了，加上索引也多占用了空间，查询的速度快了。加上索引以后速度提升非常明显。  
（1）在经常检索的字段上（select \* from Person where Name=@Name）使用索引提高查询速度。（2）select中只列出必要的字段，而不是\*。（3）避免隐式类型转换造成的全表扫描，在索引上使用函数也会造成全表扫描（因为索引只是为字段建立的，一旦使用表达式或者函数，那么索引就是失效了，当然也可以使用“函数索引”、“表达式索引”解决这个问题），使用索引不一定能提高查询速度。（4）避免在索引列上使用计算（where Name+’A'=@MyName）  
加分的回答：不进行无意义优化，根据性能查看器的报表，对最耗时的SQL进行优化。

74、什么是数据库事务？  
事务三个特性：一致性、原子性、隔离性。  
几个SQL语句，要么全部执行成功，要么全部执行失败。举例就用最经典的银行转账的例子就行，然后再举一个项目中用的例子：批量导入客户数据的时候要么全部导入成功，要么全部导入失败。事务的三个操作：Begin、Commit、RollBack。  
取刚插入数据的id值，就要用事务来隔离：Insert 和select top 1 id from ….

75、谈谈你对设计模式的理解  
世界上本没有模式，用的人多了，也就有了模式。不用刻意记每种模式的名字，手中无模式，心中有模式。  
大部分设计模式就是接口、抽象类的一个组合。ADO.Net基于接口编程，SafeDbCommand、SafeConnection。SettingsProvider。没必要记模式的名字。  
举例：SettingsProvider、第三方支付的例子。

76、谈谈ViewState。谈谈对ASP.Net原理的理解。谈谈ASP.Net生命周期。  
Label、Textbox在禁用ViewState的时候自增的不同表现  
Http是无状态的，WebForm为了实现基于http的“有状态”从而简化开发，ViewState用来在两次请求间保持状态。页面返回前将界面的状态序列化到ViewState中，这样就将状态保存在了浏览器中，下次页面提交的时候会顺便再将ViewState提交到服务器，这样服务器就根据ViewState还原了状态，这样WebForm开发就像有状态一样了。谈谈自定义AutoInc控件。

77、post、get的区别  
get的参数会显示在浏览器地址栏中，而post的参数不会显示在浏览器地址栏中；  
使用post提交的页面在点击【刷新】按钮的时候浏览器一般会提示“是否重新提交”，而get则不会；  
用get的页面可以被搜索引擎抓取，而用post的则不可以；  
用post可以提交的数据量非常大，而用get可以提交的数据量则非常小(2k)，受限于网页地址的长度。  
用post可以进行文件的提交，而用get则不可以。  
参考阅读：http://www.cnblogs.com/skynet/archive/2010/05/18/1738301.html

78、下面的程序执行结果是什么？  
class Person  
{  
public int Age { get; set; }  
}  
int i1 = 20;  
int i2 = i1;  
i1++;  
Console.WriteLine(i2);

Person p1 = new Person();  
p1.Age = 20;  
Person p2 = p1;  
p1.Age++;  
Console.WriteLine(p2.Age);  
答案：  
20  
21

解释：引用类型和值类型的区别。

79、Http默认的端口号是（80）、Ftp默认的端口号是（21）、SQLServer的默认端口号是（1433）  
80、运算符++a表示（先将a加1，然后表达式的值为a加1后的值），i++表示（表达式的值为a，然后a加1）。  
81、下面程序的执行结果是什么？  
int i = 10;  
Console.WriteLine(i++);  
Console.WriteLine(++i);  
Console.WriteLine(i=20);  
答案：  
10  
12  
20  
解答：前两个参考第80题，i++表达式的值为i自增之前的值，所以打印10，执行完成后i变成11，执行++i的时候表达式的值为自增以后的值12。C#中赋值表达式的值为变量被赋值的值，因此最后一个就输出i赋值后的值20  
while((line=reader.ReadLine())!=null)  
{  
}

82、如果不用VisualStudio，用哪个命令行编译C#程序？（\*）  
答：csc.exe。

83、下列哪个名词和WebService不直接相关？（B）  
A、UDDI  
B、GUID  
C、WSDL  
D、SOAP

SOAP和HTTP的关系：SOAP基于Http协议的，和普通网页不同的是网页返回HTML，SOAP则是符合SOAP协议的XML数据。

84、.Net Framework中的类是专门供C#调用的是吗？  
答：错。VB.Net等语言都可以调用.Net Framework中的类。CTS、CLS。

85、开放性问题：说出尽可能多的基于.Net Framework的语言。（\*）  
参考解答：C#、VB.Net、F#、PowerShell、IronPython、J#、Ruby.Net

86、.Net、ASP.Net、C#、VisualStudio之间的关系是什么？  
答：.Net一般指的是.Net Framework，提供了基础的.Net类，这些类可以被任何一种.Net编程语言调用，.Net Framework还提供了CLR、JIT、GC等基础功能。  
ASP.Net是.Net中用来进行Web开发的一种技术，ASP.Net的页面部分写在aspx 文件中，逻辑代码通常通过Code-behind的方式用C#、VB.Net等支持.Net的语言编写。  
C#是使用最广泛的支持.Net的编程语言。除了C#还有VB.Net、IronPython等。  
VisualStudio是微软提供的用来进行.Net开发的集成开发环境（IDE），使用VisualStudio可以简化很多工作，不用程序员直接调用csc.exe等命令行进行程序的编译，而且VisualStudio提供了代码自动完成、代码高亮等功能方便开发。除了VisualStudio，还有SharpDevelop、MonoDevelop等免费、开源的IDE，VisualStudio Express版这个免费版本。

87、AJAX解决什么问题？如何使用AJAX？AJAX有什么问题需要注意？项目中哪里用到了AJAX？  
答：AJAX解决的问题就是“无刷新更新页面”，用传统的HTML表单方式进行页面的更新时，每次都要将请求提交到服务器，服务器返回后再重绘界面，这样界面就会经历：提交→变白→重新显示这样一个过程，用户体验非常差，使用AJAX则不会导致页面重新提交、刷新。  
AJAX最本质的实现是在Javascript中使用XMLHttpRequest进行Http的请求，开发中通常使用UpdatePanel、JQuery等方式简化AJAX的开发，UpdatePanel的方式实现AJAX最简单，但是数据通讯量比较大，因为要来回传整个ViewState，而且不灵活，对于复杂的需求则可以使用JQuery提供的ajax功能。  
UpdatePanel的内部原理。  
AJAX最重要的问题是无法跨域请求(www.rupeng.com →so.rupeng.com)，也就是无法在页面中向和当前域名不同的页面发送请求，可以使用在当前页面所在的域的服务端做代理页面的方式解决。  
在如鹏网项目中发帖的时候显示相关帖的功能、站内搜索项目中显示搜索Suggestion、数据采集项目中都用到了AJAX。  
88、NET通过对各语言先编译成（ IL），然后再执行时用（ JIT）将其编译成本地平台代码，来实现异构平台下对象的互操作。（\*）  
89、编写正则表达式验证一个字符串是否为Email。复习正则表达式部分的知识。  
90、Application 、Cookie和 Session 两种会话有什么不同？  
答：Application是用来存取整个网站全局的信息，而Session是用来存取与具体某个访问者关联的信息。

91、开放式问题：你经常访问的技术类的网站是什么？  
博客园（www.cnblogs.com）、csdn、codeplex、codeproject、msdn文档、msdn论坛（遇到问题先到网上搜解决方案，还不行就问同事，同事也解决不了就去MSDN论坛提问，一定能得到解决）

92、你对.net的GC的理解  
GC是.Net的垃圾收集器，可以进行内存资源的回收，程序员无需关心资源的回收，当一个对象没有任何引用的时候就可以被回收了。一个对象可以被回收并不意味着一定会被立即回收，GC会选择时机进行回收。可以调用GC.Collect()让GC立即回收。GC不能回收非托管资源，对于非托管资源一般都实现了IDisposable接口，然后使用using关键字进行资源的回收。

93、请写一个SQL语句：从user表中取出name列中的起始字符是“北京”的全部记录  
select \* from [user] wherer name like’北京%’

94、请你简单的说明数据库建立索引的优缺点  
使用索引可以加快数据的查询速度，不过由于数据插入过程中会建索引，所以会降低数据的插入、更新速度，索引还会占磁盘空间。

95、什么是WEB服务控件？使用WEB服务控件有那些优势、劣势？  
web服务控件就是可以在服务器执行的控件，优势在于可以回传数据（ViewState），带有事件驱动（doPostBack），简化了Html标签的使用，使得开发ASP.net程序可以像开发Winform程序一样简单。Web服务控件最终还是被渲染（Render）为Html标签。使用WEB服务控件的劣势是会生成一些无用的属性、ViewState等，不利于SEO，一般在内网系统或者互联网系统的后台中使用WEB服务控件。如果没有复杂的服务器交互动作，就不用Web服务器控件。  
runat=server的HTML控件有什么特点：路径不用程序员解析，可以使用“~”来虚拟路径。

96、下面程序的执行结果是什么？  
int i = 10;  
if (i <=0) ;  
Console.WriteLine(“i必须大于0″);  
Console.WriteLine(“ok”);  
答案：  
i必须大于0  
ok

解答：注意if (i <=0)后面的“;”，由于有了“;”，所以Console.WriteLine(“i必须大于0″)不再是和if一起的一部分，而是独立的语句。  
for循环也是如此。

97、下面程序的执行结果是什么？ （\*）  
static void Main(string[] args)  
{  
Console.WriteLine(Calc());

Console.ReadKey();  
}

static int Calc()  
{  
int i = 0;  
try  
{  
return i;  
}  
finally  
{  
Console.WriteLine(“finally”);  
i++;  
}  
}  
答案：  
finally  
0

解释：参考第22题的值，return先执行，finally后执行，所以return的值是没有i++之前的0。注意并不是return的时候函数真的就“返回、执行结束”了，return只是标记函数的返回值是0，标记完了还会执行finally中的代码，只有finally中的代码执行完成后函数才真正的返回。

98、开放式问题：工作中遇到不懂的问题是怎样去解决的？  
参考解答：首先到百度上搜索中文的资料，一般都可以找到解决方案，如果还找不到解决方案就到google上搜索英文的资料，如果还没有解决方案就请教同事，如果同事也没有解决方案，就将问题提交到MSDN论坛。

99、开放性问题：你能说出哪些数据库？  
参考解答：MSSQLServer、Oracle、Access、DB2、Sybase

100、开放性问题：你能说出尽可能多的你听说过的编程语言？ （\*）  
参考解答：C#、VB.Net、Ada、Cobol、C、C++、Delphi、Erlang、F#、Fortran、Java、JavaScript、Pascal、Python、PowerShell、Ruby、Smalltalk。

101、请翻译下面的文章（\*）  
The BaseDataBoundControl is the root of all data-bound control classes. It defines the DataSource and DataSourceID properties and validates their assigned content. DataSource accepts enumerable objects obtained and assigned the ASP.NET 1.x way.

Mycontrol1.DataSource = dataSet;

Mycontrol1.DataBind();

DataSourceID is a string and refers to the ID of a bound data source component. Once a control is bound to a data source, any further interaction between the two (in both reading and writing) is handled out of your control and hidden from view. This is both good and bad news at the same time. It is good (rather, great) news because you can eliminate a large quantity of code. The ASP.NET framework guarantees that correct code executes and is written according to recognized best practices. You’re more productive because you author pages faster with the inherent certainty of having no subtle bugs in the middle. If you don’t like this situation—look, the same situation that many ASP.NET 1.x developers complained about—you can stick to the old-style programming that passes through the DataSource property and DataBind method. Also in this case, the base class saves you from common practices even though the saving on the code is less remarkable.

参考翻译（自我感觉翻译的很烂，仅供参考。）：  
BaseDataBoundControl是所有的数据绑定控件类的根类。它定义了DataSource、DataSourceID两个属性，并且对为它们赋值的内容进行验证。DataSource属性接受读取、设置ASP.Net 1.x方式的IEnumerable 对象。  
Mycontrol1.DataSource = dataSet;  
Mycontrol1.DataBind();  
DataSourceID 是一个字符串，引用绑定的数据源控件的ID。一旦一个控件绑定到了数据源，这两者之间的任何交互（读和写）都会在你的控件之外被处理，并且被隐藏起来。这既是好消息也是坏消息。ASP.Net框架保证代码按照最佳实践进行代码的执行、编写。你会更加高效，因为你编写了更快的、没有任何错误的分页代码。如果你不喜欢这样的行为（看，就和ASP.Net 1.x开发人员抱怨的一样），你仍然可以使用老的编程风格，也就是将对象传递给DataSource属性，然后调用DataBind方法。另外在这种情况下，基类可以帮你避免常规的做法，即使看起来改进不明显。

102、在下述选项时,没有构成死循环的程序是 (C)  
A.int i=100;while (1) { i=i%100+1; if (i>100) break; }  
B.for (;;);  
C.int k=1000; do { ++k; }while(k>=10000);  
D.int s=36; while (s);–s;

103、下列哪个是和数据库访问技术关系不大的技术（C）  
A、SQLHelper  
B、EnterPrise Library  
C、AppDomain  
D、Linq

104、下列哪些说法是错误的  
A、将bmp格式的文件修改为jpg是修改图片格式最简单的方法  
B、将QQ动态表情显示在网页上用gif格式比较好  
C、将bmp格式的图片放到网页中可以优化网站速度  
D、png格式用在网页中可以实现透明的效果。

答案：A、C。A错误是因为修改文件的后缀并没有真的修改文件的格式，要使用Photoshop、mspaint等图片处理工具进行转换才可以。C错误是因为在网页中不能使用bmp图片。

105、从以下简写中任选一个简单描述  
OA\MIS\HRM\CRM\KM

参考答案：  
OA(Office Automation)：办公自动化  
MIS(Managment Information system)：管理信息系统  
HRM(Human Resource Managment)：人力资源管理，包括员工管理、薪资管理、入职管理、离职管理、报销管理、请假管理等  
CRM(Customer Relation Managment)：客户关系管理系统，包括客户管理、客户关怀、客户回访、投诉管理、咨询管理、报修管理等  
KM(Knowledge Managment)：知识管理

106、请在数据库中设计表来保存一个树状结构的公司组织结构  
参考呼叫中心项目的代码和板书。

107、下面程序的执行结果是什么？  
public struct Point  
{  
public int x;  
public int y;  
public Point(int x, int y)  
{  
this.x = x;  
this.y = y;  
}  
}  
Point p1 = new Point(100, 100);  
Point p2 = p1;  
p1.x = 200;  
Console.WriteLine(“{0},{1}”, p1.x, p2.x);  
答案：  
200,100

解答：结构体是复制传递的。

108、在Windows命令行中敲入下面的指令各有什么反应 ：  
cd ..

notepad  
cmd  
ipconfig  
dir

答案：  
cd ..：进入上一级目录  
notepad：打开记事本  
cmd：启动另外一个命令行  
ipconfig：打印网络设置信息  
dir：显示当前目录下的文件

109、下列这些软件用过哪些？他们是用来做什么的？  
TypeAndRun、FoxMail、Reflector、UltraEdit、VSS、VirtualPC、VMWare  
参考答案：  
TypeAndRun：快速通过敲命令的方式启动其他程序的工具。  
FoxMail：邮件收发客户端  
Reflector：.net程序反编译工具  
UltraEdit：高级文本编辑器  
VSS：源代码管理工具  
VirtualPC：在一台计算机中虚拟出另外一个计算机的软件，是微软的产品  
VMWare：和VirtualPC一样，也是一种虚拟机软件