

1 在进行顺序查找过程中，如果线性表中被查的元素是线性表中的最后一个，或者被查元素根本不在线性表中，则为了查找这个元素需要与线性表中所有元素进行比较，这是顺序查找最坏的情况。信息隐蔽是指在一个模块内包含的信息（过程或数据），对于不需要这些信息的其他模块来说是不能访问的。2 模块独立性是指每个模块只完成系统要求的独立的子功能，并且与其他模块的联系最少且接口简单。衡量软件的模块独立性的度量标准是耦合性和内聚性。一个模块的内聚性越强，则该模块的模块独立性越强。而内聚性是信息隐蔽和局部化概念的自然扩展。面向对象的设计方法与传统的面向过程的方法有本质不同，它的基本原理是，使用现实世界的概念抽象地思考问题从而自然地解决问题。它强调模拟现实世界中的概念而不强调算法，它鼓励开发者在软件开发的绝大部分中都用应用领域的概念去思考。总体设计过程通常由两个主要阶段组成：系统设计，确定系统的具体实现方案；结构设计，确定软件结构。为确定软件结构，首先需要从实现角度把复杂的功能进一步分解。分析员结合算法描述仔细分析数据流图中的每个处理，如果一个处理的功能过分复杂，必须把它的功能适当地分解成一系列比较简单的功能。3 数据库概念设计的过程中，视图设计一般有三种设计次序，它们是：1、自顶向下。这种方法是先从抽象级别高且普遍性强的对象开始逐步细化、具体化与特殊化。2、由底向上。这种设计

方法是先从具体的对象开始，逐步抽象，普遍化与一般化，最后形成一个完整的视图设计。3、由内向外。这种设计方法是先从最基本与最明显的对象着手逐步扩充至非基本、不明显的其它对象。4 在方法中定义的局部变量在方法退出时被自动撤消，只有静态变量不会被撤消。如果未对变量指定修释符，在JAVA中将自动将其设置为局部变量，局部变量也叫自动变量，对于方法外面的变量，将在对象被创建时一起被创建，其初值可通过构造函数对其进行初始化。如果方法中是一个简单类型的形参，由于其是按值传递，在编译时会占用存储空间，直到该对象释放时为止。如果形参为一个类类型，其参数是按地址传递，编译时不会为其分配空间，因此该参变量不会存在于方法中。5 类加载器（class loader）加载程序运行所需要的所有类，它通过区分本机文件系统的类和网络系统导入的类增加安全性，这可以限制任何的特洛伊木马程序，因为本机类总是先被加载，一旦所有的类被加载完，执行文件的内存划分就固定了，在这个时候特定的内存地址被分配给对应的符号引用，查找表（lookup table）也被建立，由于内存划分发生在运行时，解释器在受限制的代码区增加保护防止未授权的访问；然后字节码校验器（byte code verifier）进行校验，主要执行下面的检查：类符合JVM规范的文件格式，没有违反访问限制，代码没有造成堆栈的上溢或者下溢，所有操作代码的参数类型都是正确的，没有非法的数据类型转换（例如将整型数转换成对象类型）发生；校验通过的字节码被解释器（interpreter）执行，解释器在必要时通过运行时系统执行对底层硬件的合适调用。7

```
public class Parent { public int addValue( int a, int b) { int s. s = a b.
```

return s. } } class Child extends Parent { } 选项中\_\_B\_\_方法可以正确加入类Child中且父类的方法不会被覆盖。 A、 int addValue( int a, int b ){// do something...} B、 public void addValue (){// do something...} C、 public int addValue( int a, int b )throws MyException {//do something...} D、 public float addValue ( int a ,int b, float b=1.0 ){// do something...} 解析：此题涉及方法重载（overload），方法重写（override）以及类派生时方法重写的规则。方法重载的规则是：一、参数列表必须不同，个数的不同完全可以，如果个数相同，则参数类型的不同不会引起歧意，例如int和long，float和double就不能作为唯一的类型不同；二、返回值可以不同，但是不能是重载时唯一的不同点（这点和c中不同，c中返回类型必须一致）。方法重写发生在类继承时，子类可以重写一个父类中已有的方法，必须在返回类型和参数列表一样时才能说是重写，否则就是重载，java中方法重写的一个重要而且容易被忽略的规则是重写的方法的访问权限不能比被重写的方法的访问权限低！重写的另一个规则是重写的方法不能比被重写的方法抛弃(throws)更多种类的异常，其抛弃的异常只能少，或者是其子类，不能以抛弃异常的个数来判断种类，而应该是异常类层次结果上的种类。此题中选取项A的错误就是重写的访问权限比被重写的方法的低，而B属于重载，选项C的错误在于比被重写的方法抛弃了更多种类的异常。选项D中，当只传递两个变量的值的时候，那系统就不知道到底调用哪个方法了，因为b为常量，可以不指定，所以只有选项B是符合题意的。8 假设有以下代码 String s= "hello". String t = "hello". char c[] = { ' h ', ' e ', ' l ', ' l ', ' o ' }. 下列选项中返回false的语句是 B。 A

、 s.equals(t). B、 t.equals(c). C、 s==t. D、 t.equals(new String("hello")). 解析：==操作符比较的是操作符两端的操作数是否是同一个对象，而String的equals()方法比较的是两个String对象的内容是否一样，其参数是一个String对象时才有可能返回true，其它对象都返回假。需要指出的是由于s和t并非使用new创建的，他们指向内存池中的同一个字符串常量，因此其地址实际上是相同的（这个可以从反编译一个简单的测试程序的结果得到，限于篇幅不列出测试代码和反编译的分析），因此答案C也是正确的。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)