

详解笔试面试题集

(1)下列各题 ABCD 四个选项中，只有一个选项是正确的，请将正确选项填写在答题纸上。

(1)下列关于栈的描述中错误的是()

- A. 栈是先进后出的线性表
- B. 栈只能顺序存储
- C. 栈具有记忆作用
- D. 对栈的插入与删除操作中，不需要改变栈底指针

正确答案： B

分析： 栈是链式存储的

(2)对于长度为 n 的线性表，在最坏情况下，下列各排序法所对应的比较次数中正确的是()

- A. 冒泡排序为 $n/2$
- B. 冒泡排序为 n
- C. 快速排序为 n
- D. 快速排序为 $n(n-1)/2$

正确答案： D

分析：

想想快速排序，每次都会选取线性表的轴值，随后以此轴值划分为两个子线性表再分别进行快排，在最坏情况下，也就是说每次选出的线性表轴值完全不能将这个线性表划分为两个子线性表。那么此时快速排序退化为冒泡排序了。

那么第一趟排序时，轴值（线性表的中间位置）被选出，这个值绝对是这个线性表中最大的（不然也不能是最坏情况），其他值都比他小，那么线性表现在分为完全不对等的两段（一段是 0,另一段是 $n-1$ ），一段是这个值，一段是其他值。同样第二趟排序在刚才剩下的值中选中间值（剩余值中最大的那个），又分为不对等两段，依次递推。也就是说每次都比较了 $N-1$ 个元素（轴值选出后都与它比较大小），那么肯定是比较了 $n-1$ 次（如第一次先挑了个轴值，然后剩下 $n-1$ 比较）， n 代表当前子线性表中元素个数由此最白痴的数列问题出现了，如下

$$1 + 2 + 3 + \dots + n - 2 + n - 1 = n(n-1)/2$$

还有一种投机取巧的方法，在最垃圾情况下既然快排变为冒泡，那由时间复杂度知其必为 $O(n^2)$ 的复杂度，答案中 ABC 都是线性时间复杂，显然错误

(3)下列对于线性链表的描述中正确的是()

- A. 存储空间不一定是连续，且各元素的存储顺序是任意的

- B 存储空间不一定是连续，且前件元素一定存储在后件元素的前面
- C 存储定间必须连续，且前件元素一定存储在后件元素的前面
- D 存储空间必须连续，且各元素的存储顺序是任意的

答案是 A, 分析 链接存储并不要求存储空间连续,存储位置也不用有序

(4)为了使模块爆可能独立，要求()

- A 模块的内聚程度要尽量高，且各模块间的耦合程度要尽量强
- B 模块的内聚程度要尽量高，且各模块间的耦合程度要尽量弱
- C 模块的内聚程度要尽量低，且各模块间的耦合程度要尽量弱
- D 模块的内聚程度要尽量低，且各模块间的耦合程度要尽量强

答案: B

分析: 高内聚，低耦合。

5)下列叙述中，正确的是()

- A Java 语言的标识符是区分大小写的
- B 源文件名与 public 类名可以不相同
- C 源文件扩展名为.jar
- D 源文件中 public 类的数目不限

答案: A

6)下列属于合法的 Java 标识符是()

- A _cat
- B 5books
- C +static
- D -3.14159

答案: A

7) 在 Java 中，表示换行符的转义字符是()

- A \n
- B \f
- C 'n'
- D \dd

答案: A

(8)扯 Java 中，所有类的根类是()

- A. java.lang.Object
- B. java.lang.Class
- C. java.applet.Applet
- D. java.awt.Frame

答案: A

(9). 在 Java+中，用 Package 语句说明一个包时，该包的层次结构必须是()

- A 与文件的结构相同
- B 与文件目录的层次相同

- C 与文件类型相同
- D 与文件大小相同

答案: A

(10)在读字符文件 Employee.dat 时, 使用该文件作为参数的类是()。

- A BufferedReader
- B DataInputStream
- C DataOutoutStream
- D FileInputStream

答案: D

(11)在 Java 中, 能实现多重继承效果的方式是()。

- A 内部类
- B 适配器
- C 接口
- D 同步

答案: C

(12) char 类型的取值范围是()。

- A $2^7 \sim 2^7 - 1$
- B $0 \sim 2^{16} - 1$
- C $-2^{15} \sim 2^{15} - 1$
- D $0 \sim 2^8 - 1$

答案: A

(13)下列方法中可以用来创建一个新线程的是()。

- A 实现 java.lang.Runnable 接口并重写 start0 方法
- B 实现 java.iang.Runnable 接口并重写 run0 方法
- C 继承 java.lang.Thread 类并重写 run0 方法
- D 继承 java.lang.Thread 类并重写 start0 方法

答案: C

(14)下列关于线程优先级的说法中, 正确的是()。

- A. 线程的优先级是不能改变的
- B. 线程的优先级是在创建线程时设置的
- C. 在创建线程后的任何时候都可以设置
- D. B 和 C

答案: C

(15)下列代码中, 将引起一个编译错误的行是()。

- 1) public class Test{
- 2) int m, n;
- 3) public Test(){}

```

4)    public Test(int a){ m=a;}
5)    public static void main(String args0){
6)        Test t1, t2;
7)        int j, k;
8)        j=0; k=0;
9)        t1 = new Test();
10)       t2 = new Test(j, k);
11)    }
12)}

```

- A. 第 3 行
- B. 第 5 行
- C. 第 6 行
- D. 第 10 行

答案: D

(16)阅读下列代码后

```

public class Person{
    int arr[]=new int[10];
    public static void main(String args[ ]){
        System.out.println(arr[1]);
    }
}

```

正确的说法是()。

- A 编译时将产生错误
- B 编译时正确, 运行时将产生错误
- C 输出零
- D 输出空

答案: A

(17) AIDL 支持以下哪种数据类型? 1)string 2)list 3)map 4)All native java datatype() {,

- A. 1, 2 和 3
- B. 4
- C. 以上所有
- D. 都不支持

答案: C

分析:

AIDL 服务只支持有限的数据类型, 如果用 AIDL 服务传递一些复杂的数据就需要做更进一步处理,AIDL 服务支持的数据类型如下:

1. Java 的原生类型
2. String 和 CharSequence
3. List 和 Map ,List 和 Map 对象的元素必须是 AIDL 支持的数据类型;
以上三种类型都不需要导入(import)

4. AIDL 自动生成的接口

需要导入(import)

5. 实现 android.os.Parcelable 接口的类. 需要导入(import)。

二、多项选择题

下列各题 A.、B.、C.、D. 四个选项中，有一个或多个选项是正确的，

(1)下列说法正确的有()。

- A. 环境变量可在编译 source code 时指定.
- B. 在编译程序时，所能指定的环境变盛不包括 class path
- C. javac -次可同时编译数个 Java 源文件
- D. javac.exe 能指定编译结果要置于哪个目录 (directory)

答案: BCD

(2)不能用来修饰 interface 的有()

- A. private B.public C.protected D.static

答案: ACD

(3)下列说法错误的有()

- A.在类方法中可用 this 来调用本类的类方法
- B.在类方法中调用本类的类方法时可直接调用
- C.在类方法中只能调用本类中的类方法
- D.在类方法中绝对不能调用实例方法

答案: ACD

请将正确选项填写在答题纸上

(4)下列说法错误的有()

- A. Java 面向对象语言容许单独的过程与函数存在
- B. Java 面向对象语言容许单独的方法存在
- C. Java 语言中的方法属于类中的成员(member)
- D. Java 语言中的方法必定隶属于某...类 (对象)，调用方法与过程或函数相同

答案: ABC

(5)下列说法错误的有()。

- A.能被 java.exe 成功运行的 java class 文件必须有 main()方法
- B. J2SDK 就是 Java API
- C.Appletviewer.exe 可利用 jar 选项运行 Jar 文件
- D.能被 Appletviewer 成功运行的 java class 文件必须有 main()方法

答案: BCD

三、判断题

请将每空的正确答案写在答题纸上，答在试卷上不得分。

- (1) Java 程序中的起始类名称必须与存放该类的文件名相同。(对)
- (2) Unicode 是用 16 位来表示一个字的。(对)
- (3) 原生类中的数据类型均可任意转换。(对)

三、填空题

请将每空的正确答案写在答题纸上【1】至【25】序号的横线上，答在试卷上不得分。

- (1) 某二叉树中度为 2 的结点有 18 个，则该二叉树中有【1】 19 个叶子结点。

二叉树的终端结点（叶子结点）数等于双分支结点数加 1。

- (2) 在面向对象方法中，类的实例称为【2】 对象

- (3) 诊断和改正程序中错误的工作通常称为【3】 Debug

- (4) 在关系数据库中，把数据表示成二维表，每一个二维表称为【4】 关系

- (5) Java 源文件中最多只能有一个【5】 public 类，其他类的个数不限。

- (6) 线程在生命周期中要经历 5 中状态，分别是新建状态、可运行状态、运行状态、【6】 暂停 状态和终止状态。

- (7) FileInputStream 是字节流；BufferedWriter 是字符流， ObjectOutputStream 是【7】 对象序列化流

- (8) 当使用 Thread t = new Thread(r) 创建一个线程时，表达式 r instanceof Thread 的值为【8】 false 。 r instanceof Runnable

1. 抽象类和接口的区别？

抽象类：

- 1) 抽象方法，只有行为的概念，没有具体的行为实现。使用 abstract 关键字修饰，没有方法体。子类必须重写这些抽象方法。
- 2) 包含抽象方法的类，一定是抽象类。
- 3) 抽象类只能被继承，一个类只能继承一个抽象类。

接口：

- 1) 全部的方法都是抽象方法，属性都是常量

- 2) 不能实例化，可以定义变量。
- 3) 接口变量可以引用具体实现类的实例
- 4) 接口只能被实现，一个具体类实现接口，必须实现全部的抽象方法
- 5) 接口之间可以多实现
- 6) 一个具体类可以实现多个接口，实现多继承现象

2. 列举你所知道的线程同步的方法：

- 1) `wait()`: 让线程等待。将线程存储到一个线程池中。
- 2) `notify()`: 唤醒被等待的线程。通常都唤醒线程池中的第一个。让被唤醒的线程处于临时阻塞状态。
- 3) `notifyAll()`: 唤醒所有的等待线程。将线程池中的所有线程都唤醒。

3. `equals()`方法和“==”操作符的区别是什么？重写 `equals()`方法的时候，我们还需要重写哪些方法？为什么？

“==”比较的是一个对象在内存中的地址值，栈中的内容是否相同

`equals()`是针对引用的，比较的是引用类型的内容一样不一样，堆中的内容是否相同

重写 `equals()`方法的时候，需重写 `hashCode` 方法，这样比较的时候，比较的是两个对象的 `hash` 值，不同对象的 `hash` 值不同。

7. 写一个函数计算但参数为 **n** (**n** 很大) 时的值 **1-2+3-4+5-6+7……+n**。(提示：考虑程序执行效率)

```
private static void jisuan(int n) {  
    int sum=0;  
    if(n%2==0){  
        sum=-(n/2);  
    }
```

```

        System.out.println(""+sum);
    }else{
        sum=- (n-1)/2+n;
        System.out.println(""+sum);
    }
}
}

```

(11) 简述 **String** 和 **StringBuffer**、**ArrayList** 和 **LinkedList**、**HashMap** 和 **HashTable** 的特点及区别。

答:String 对象的长度一旦定义就固定就不可以改变,对于已经存在的 String 对象的修改都是创建一个新的对象,然后把新的值存进去,String 类不能被继承。StringBuffer 是一个可变对象,当对它进行修改的时候不会像 String 那样重新建立对象。它只能通过构造函数来建立对象。另外 StringBuffer 还是一个线程安全的类。

ArrayList 是实现了基于动态数组的数据结构用于,LinkedList 基于连表的数据结构,二者都可以存储对象的引用。对于随机访问 get 和 set, ArrayList 优于 LinkedList,因为 LinkedList 要移动指针。对于新增和删除操作 add 和 remove, LinkedList 比较占优势,因为 ArrayList 要移动数据。

HashMap 是 Hashtable 的轻量级实现(非线程安全的实现),他们都完成了 Map 接口,主要区别在于 HashMap 允许空(null)键值(key),由于非线程安全,在只有一个线程访问的情况下,效率要高于 Hashtable。

HashMap 允许将 null 作为一个 entry 的 key 或者 value,而 Hashtable 不允许。HashMap 把 Hashtable 的 contains 方法去掉了,改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类,而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

最大的不同是,Hashtable 的方法是 Synchronize 的,而 HashMap 不是,在多个线程访问 Hashtable 时,不需要自己为它的方法实现同步,而 HashMap 就必须

为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。就 HashMap 与 Hashtable 主要从三方面来说。

一. 历史原因:Hashtable 是基于陈旧的 Dictionary 类的，HashMap 是 Java 1.2 引进的 Map 接口的一个实现

二. 同步性:Hashtable 是线程安全的，也就是说是同步的，而 HashMap 是线程序不安全的，不是同步的

三. 值：只有 HashMap 可以让你将空值作为一个表的条目的 key 或 value

求质数的多种优化方式

答：一般我们求质数时都会去计算小于某一个数 N 的质数而不会不加以限定，现在我们以求小于 N 的质数来说一说优化方式。在我们求质数的过程中一般会用到两种方法试除法和筛选法两种，现在来说一下试除法：

①. 判断小于 N 的数 X 是否是质数，就是从 2 一直尝试到 X-1，这种做法效率最差，并不可取——>②如果 X 是质数，那么它如果不能被小于 X/2 的数整除即可，这样算法效率提高一些——>③除了 2 以外，所有的质数都只能是奇数，所以我们可以将数 X 先试除 2，然后尝试从 3 一直到 X/2 的所有奇数——>④其实判断一个数是否是质数，只需判断一个数能不能被除了 1 之外小于 \sqrt{x} 的数整除即可，——>⑤最后，我们可以利用前面求出来的质数来判断，我们只需判断 X 能不能被小于 \sqrt{x} 的质数整除即可，这样效率是不是更高。

再说筛选法：对于使用筛选法的优化主要是从空间上考虑

①定义一个容器，将数据放入容器中，然后遍历其中的数据，将合数的数据删除，最后剩余的就是质数了——>②我们可以定义一个布尔类型的数组容器，将其中的值都赋值为 true，在筛选的过程中将不是质数的数作为数组的下标将对应元素的值改为 false，最后取出值为 true 的数组的下标即可——>③构造定长的 byte 数组，数组的每个 byte 存储 8 个布尔值，这样性能是不是有了提

高呢。

(14) 简述几种排序方式（至少四种，可以从以下各个方面来比较这几种算法，例如从时间复杂度和空间复杂度）

排序法	最差时间分析	平均时间复杂度	稳定度	空间复杂度
冒泡排序	$O(n^2)$	$O(n^2)$	稳定	$O(1)$
快速排序	$O(n^2)$	$O(n \cdot \log_2 n)$	不稳定	$O(\log_2 n) \sim O(n)$
选择排序	$O(n^2)$	$O(n^2)$	稳定	$O(1)$
二叉树排序	$O(n^2)$	$O(n \cdot \log_2 n)$	不一定	$O(n)$
插入排序	$O(n^2)$	$O(n^2)$	稳定	$O(1)$
堆排序	$O(n \cdot \log_2 n)$	$O(n \cdot \log_2 n)$	不稳定	$O(1)$
希尔排序		$O(n^{1.25})$	不稳定	$O(1)$

冒泡排序：（Bubble Sort）冒泡排序方法是最简单的排序方法。这种方法的基本思想是，将待排序的元素看作是竖着排列的“气泡”，较小的元素比较轻，从而要往上浮。在冒泡排序算法中我们要对这个“气泡”序列处理若干遍。所谓一遍处理，就是自底向上检查一遍这个序列，并时刻注意两个相邻的元素的顺序是否正确。如果发现两个相邻元素的顺序不对，即“轻”的元素在下面，就交换它们的位置。显然，处理一遍之后，“最轻”的元素就浮到了最高位置；处理二遍之后，“次轻”的元素就浮到了次高位置。在作第二遍处理时，由于最高位置上的元素已是“最轻”元素，所以不必检查。一般地，第 i 遍处理时，不必检查第 i 高位置以上的元素，因为经过前面 $i-1$ 遍的处理，它们已正确地排好序。

插入排序（Insertion Sort）插入排序的基本思想是，经过 $i-1$ 遍处理后， $L[1..i-1]$ 已排好序。第 i 遍处理仅将 L 插入 $L[1..i-1]$ 的适当位置，使得 $L[1..i]$ 又是排好序的序列。要达到这个目的，我们可以用顺序比较的方法。首

先比较 L 和 $L[i-1]$ ，如果 $L[i-1] \leq L$ ，则 $L[1..i]$ 已排好序，第 i 遍处理就结束了；否则交换 L 与 $L[i-1]$ 的位置，继续比较 $L[i-1]$ 和 $L[i-2]$ ，直到找到某一个位置 $j(1 \leq j \leq i-1)$ ，使得 $L[j] \leq L[j+1]$ 时为止。

选择排序 (Selection Sort) 选择排序的基本思想是待排序的记录序列进行 $n-1$ 遍的处理，第 i 遍处理是将 $[i..n]$ 中最小者与位置 i 交换位置。这样，经过 i 遍处理之后，前 i 个记录的位置已经是正确的了。

快速排序 (Quick Sort) 快速排序是对冒泡排序的一种本质改进。它的基本思想是通过一趟扫描后，使得排序序列的长度能大幅度地减少。在冒泡排序中，一次扫描只能确保最大数值的数移到正确位置，而待排序序列的长度可能只减少 1。快速排序通过一趟扫描，就能确保某个数（以它为准点吧）的左边各数都比它小，右边各数都比它大。然后又用同样的方法处理它左右两边的数，直到基准点的左右只有一个元素为止。

一、专业知识

1、 $a_0=1$ 、 $a_1=1$ 、 $a_2=a_1+a_0$ 、 $a_3=a_2+a_1$ ，以此类推，请写代码用递归算出 a_{30} ?

3、简述值类型和引用类型的区别

*值类型*包括简单类型、结构体类型和枚举类型，*引用类型*包括自定义类、数组、接口、委托等

他们两在内存中存储的方式不同，值类型以栈的方式存储的，引用类型以堆的方式；前者是真实的存储空间，后者只是存储地址的引用！

4、简述类中的静态成员和非静态成员的区别

静态变量使用 `static` 修饰符进行声明，在类被实例化时创建，通过类进行访问。不带有 `static` 修饰符声明的变量称做非静态变量，在对象被实例化时创建，通过对象进行访问。一个类的所有实例的同一静态变量都是同一个值，同一个类的不同实例的同一非静态变量可以是不同的值。静态函数的实现里不能使用非静态成员，如非静态变量、非静态函数等。

5、什么是单例？

一个类中只有一个实例并且自行化向整个系统提供这个实例叫单例

下面程序段的输出结果是：

```
Void complicatedex() {  
    int x=20,y=30;  
    boolean b; b=x>50&&y>60 || x>50&&y<-6 || x<-50&&y>60 || x<-50&&y<-60;  
    system.out.println(b);  
}
```

结果为：false

super()和 this()的区别

`Super()` 指父类的无参构造方法，`this()` 指当前类的无参构造方法，两者都必须写在构造方法的第一句

2、Java 中 public,private,protected,和默认的区别

- a、private 修饰词，表示成员是私有的，只有自身可以访问；
- b、protected，表示受保护权限，体现在继承，即子类可以访问父类受保护成员，同时相同包内的其他类也可以访问 protected 成员。
- c、无修饰词（默认），表示包访问权限（friendly，java 语言中是没有 friendly 这个修饰符的，这样称呼应该是来源于 c++ ），同一个包内可以访问，访问权限是包级访问权限；
- d、public 修饰词，表示成员是公开的，所有其他类都可以访问；

3、描述一下 java 的事件委托机制和垃圾回收机制

java 事件委托机制：一个源产生一个事件时，把他送到一个或多个监听器那里，在这种机制种，监听器只是等待，一旦收到事件，处理事件并返回；

Java 垃圾回收：是一种动态存储管理技术，它自动地释放不再被程序引用的对象，按照特定的垃圾收集算法来实现资源自动回收的功能。当一个对象不再被引用的时候，内存回收它占领的空间，以便空间被后来的新对象使用，以免造成内存泄露。

什么是 java 序列化，如何事件 java 序列化

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

```
public class Cat implements Serializable {}
```

.....

```
Try{  
    FileOutputStream fos = new FileOutputStream("catDemo.out");  
    ObjectOutputStream oos = new ObjectOutputStream(fos);  
    System.out.println(" 1> " + cat.getName());  
    cat.setName("My Cat");  
    oos.writeObject(cat);  
    oos.close();  
}
```

```
} catch (Exception e) {  
.....  
}  
.....
```

overload 和 override 的区别。Overloaded 的方法和是否可以改变返回值类型

override (重写, 覆盖)

- 1、方法名、参数、返回值相同。
- 2、子类方法不能缩小父类方法的访问权限。
- 3、子类方法不能抛出比父类方法更多的异常(但子类方法可以不抛出异常)。
- 4、存在于父类和子类之间。
- 5、方法被定义为 final 不能被重写。

overload (重载, 过载)

- 1、参数类型、个数、顺序至少有一个不相同。
- 2、不能重载只有返回值不同的方法名。
- 3、存在于父类和子类、同类中。

Overloaded 的方法不能改变返回值类型

final 类有什么特点

该类不能被继承

写出下面代码的输出结果

```
Package test;
```

```
Public class OuterClass{
```

```

Private class InterClass{
    Public InterClass{
        System.out.println( "interClass create" );
    }
}

Public OuterClass() {
    InterClass ic= new InterClass();
    System.out.println( "outerclass create" );
}

Public static void main(String[] args){
    OuterClass oc=new OuterClass();
}

}

```

输出：
interClass create
Outerclass create

如何格式化日期

java 日期的格式话主要用的是 SimpleDateFormat df = new SimpleDateFormat
() ;

ArrayList Vector LinkedList 的存储性能好特性, HashMap 和 Hashtable 的区别

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法(线程安全)，通常性能上较 ArrayList 差，而 LinkedList

使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

1. Hashtable 的方法是同步的，HashMap 未经同步，所以在多线程场合要手动同步 HashMap 这个区别就像 Vector 和 ArrayList 一样。

2. Hashtable 不允许 null 值 (key 和 value 都不可以), HashMap 允许 null 值 (key 和 value 都可以)。

3. Hashtable 有一个 contains(Object value)，功能和 containsValue(Object value) 功能一样。

4. Hashtable 使用 Enumeration，HashMap 使用 Iterator。

String 是基本数据类型吗？

不是

多线程有几种实现方式，同步有几种实现方式，**stop()**和**suspend()**方法为什么不推荐使用

多线程有两种实现方法，分别是继承 Thread 类与实现 Runnable 接口
同步的实现方面有两种，分别是 synchronized, wait 与 notify”

反对使用 stop()，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在。suspend() 方法容易发生死锁。调用 suspend() 的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成

死锁。所以不应该使用 `suspend()`，而应在自己的 `Thread` 类中置入一个标志，

1.请阐述一下你对 **Java** 多线程中同步的几种使用方式，区别，以及其重要性。

同步有两种实现方式

1. 同步方法
2. 同步代码块

多线程的同步时为了保证能够保证关键数据在单位时间只能有一个线程操作，保证数据的同步性，安全性

2.**final** ,**finally** , **finalize** 的区别。

Final 有最终的不可改变的意思，可以用来修饰基本数据使之成为常量，该常量只能在编译期进行修改。

当修饰对象时，则该对象引用不能改变，但该对象本身可以做修改。

当修饰方法时，表示该方法在被继承时不能被修改，或覆盖

当修饰类时，表示该类不会别的类继承

Finally 用于在 `Try-catch-finally` 语句中 `finally` 中的代码块时必须执行的

finalize 当某个对象要被当做垃圾回收时，会调用 `finalize()` 方法，该方法用于检查对象不被运行状态的对象引用或间接地引用

sleep() 和 **wait()**有什么区别？

`Sleep()` 是线程中的一个方法，该方法用于控制自身线程的流程，当执行 `sleep()` 方法睡眠时保持对象锁。

`Wait()` 方法是 `object` 类中的一个方法，该方法用于让一个线程处于等待状态并释放对象锁，当该线程想要再次执行时需要调用 `notity` 方法唤醒此线程

4. abstract 的 method 是否可同时是 static, 是否可同时是 native, 是否可同时是 synchronized ?

不可以

5 当一个线程进入一个对象的 Synchronized 方法后, 其他线程是否可进入此对象的其他方法

可以去访问非 Synchronized 方法。

6. 当一个对象被当做参数传递到一个方法后, 此方法可改变这个对象的属性, 并可返回变化后的结果, 那么这里到底是值传递还是引用传递?

值传递

7. 创建一个 class 对象有几种方法? 分别是什么?

有三种:

1. `Class c=Class.forName(“java.lang.String”);`
2. `String str=new String();`
`Class c=String.getClass();`

```
3. Class c=String. Class;
```

9.如何获取一个目录下有多少个文件?

```
File f=new File( "D://" );  
Int count=f.list().length;
```

10.请写出下列代码的输出结果:

```
public class FatherClass{  
    public FatherClass(){  
        System.out.println( "FatherClass Create" );  
    }  
    public static class ChildClass extends FatherClass{  
        public ChildClass(){  
            System.out.println( "ChildClass Create" );  
        }  
    }  
  
    public static void main(){  
        FatherClass fc=new FatherClass();  
        ChildClass cc=new ChildClass();  
    }  
}
```

```
FatherClass Create  
FatherClass Create  
ChildClass Create
```

13. 当一个对象不再被使用时, 如何才能从内存中消失?

将该对象引用值为空

14. 请写一个整数组成的矩阵转置程序, 及行变成列, 列变成行?

```
int[][] transpose(int[][] t);

public int[][] transpose(int[][] t) {
    int [][] f=new int[t[0].length][t.length];
    for(int i=0;i<t.length;i++){
        int index=0;
        index++;
        for(int x=0;x<t[i].length;x++){
            int m=0;
            m++;
            f[m][index]=t[i][x];
        }
    }
    Return f;
}
```

1. 下列哪些语句关于内存回收的说明是正确的?(b)

- A, 程序员必须创建一个线程来释放内存
- B, 内存回收程序负责释放无用内存
- C, 内存回收程序允许程序员直接释放内存
- D, 内存回收程序可以在指定的时间释放内存对象

2.下面异常是属于 Runtime Exception 的是(abcd)(多选)

A, ArithmeticException

B, IllegalArgumentException

C, NullPointerException

D, BufferUnderflowException

3.Math.round(11.5)等于多小().Math.round(-11.5)等于多小().(c)

A 11,-11 B 11,-12 C 12,-11 D 12,-12

4.下列程序段的输出结果是:(b)

```
Void complicatedexpression_r(){
```

```
int x=20,y=30;
```

```
boolean b;
```

```
b=x>50&& y>60||x>50&&y<-60||x<-50&&y>60||x<-50&&y<-60;
```

```
System.out.println(b);
```

A,ture B,false C,1 D,011.activity

5.以下程序的运行结果 (B) //[-128~127]从对象池取，超了后新建对象

```
Integer a = 34556;
```

```
Integer b =34556;
```

```
If(a==b){
```

```
    System.out.println( "Equal" );
```

```
}else{
```

```
        System.out.println( "Not equal" );  
    }  
}
```

A.Equal B.Not equal C.无输出 D.编译错误

6.以下程序的运行结果是 (B)

```
public class test{  
    public static void main(String args[]){  
        System.out.print(100%3);  
        System.out.print( " " );  
        System.out.print(100%3.0);  
    }  
}
```

A.1,1 B.1,1.0 C.1.0,1 D.1.0,1.0

7.Java 语言中字符串 “学 Java” 所占的内存空间是 (B)

A.5 个字节 B.6 个字节 C.7 个字节 D.10 个字节

Math.round (11.5) 等于多少 ()。 Math.round (-11.5) 等于多少 ? (C)

A . 11 , -11 B.11 , -12 C. 12.-11 D.12 , -12

8.下列程序段的输出结果是 : (B)

```

void complicatedexpression_r ( ) {

    int x=20 , y=30 ;

    boolean b;

    b=x>50&&y>50&&y<-60||x<-50&&y>60||x<-50&&y<-60;

    System.out.println(b);

}

```

A. true B.false C.1 D.0

代码段如下，选项中描述正确的是 A

```

static int ternary ( int s ) {

    int i=s;

    return i < 10 ? i * 100 : i * 10 ;

}

```

A 这是个静态方法，不需要实例化就可以被调用

B 返回值是方法临时变量的值

C 返回值等于 i * 10

D 返回值等于 i * 100

9.如下描述正确的是 AD

A 一个 java 中只能有一个 class 是 public 的

B 如果一个类的构造方法是私有的，那么实例化此类就需要有一个非私有的方法

C protected 的方法无法被子类访问

D abstract 类不可以直接实例化

10.关于 final 关键字，正确的是 ABCDE

- A. 使用 final 关键字，是出于设计与效率的考量
- B 使用 final 关键字修饰属性，句柄指向的基础类型是一个常量
- C final 关键字修饰属性，句柄指向一个对象，则对象不可以改变
- D final 关键字修饰方法，则此方法不允许覆盖
- E final 关键字修饰方法，则此方法不允许重载

11.如下描述正确的是 ABCD

- A HashMap 使用键值对描述数据
- B List 和 Set 都是接口
- C List 的实现类存储数据允许重复
- D Set 的实现类存储数据是无序的
- E List 的实现类存储数据是无序的

12.如下描述正确的是 AD

- A 继承 Thread 类，可以创建一个线程
- B run()方法中定义循环部分，并指出循环终结条件
- C 每个对象默认都包含了一把锁
- D 多线程竞争临界资源，可能会导致死锁

13 下面描述正确的是 ABCD

- A. MVC 涉及模式中，servlet 实现 controller 功能

B applet 浏览器运行的 java 小程序

C javabean 用于对应数据库中的数据表

D SSH(Struts, Spring, Hibernate) Spring 进行流程控制, Struts 进行业务流转, Hibernate 进行数据库操作的封装

14. 下面说法正确的是 ABD

A 用 new 关键字实例化接口, 必须实现接口中的所有抽象方法

B 接口中不可以定义私有方法

C 接口中可以定义私有属性

D 类可以实现多个接口

15. 下面说法正确的是 ABC

A 调用 equal() 方法所实现的功能, 取决于当前类及其继承树中对此方法的定义。

B java 不允许使用者重载操作符

C java 中, 操作符+ 针对字符串操作时把字符串相连接的意思。

D java 允许操作符重载

16. UML 中哪些图表示系统行为 AE

A. 状态图 B. 用例图 C. 协作图 D. 类图 E. 活动图 F. 时序图

17. 如下程序段, 正确的是 BD

```
A abstract class Something {  
  
    private abstract String doSomething();  
  
}
```

```
B class Something {
```

```
    Int i;
```

```
    public void doSomething(){
```

```
        System.out.println( "i+" +i)
```

```
    }
```

```
}
```

```
C . public class Something{
```

```
    public static void main (String [] args){
```

```
        Something s = new Something();
```

```
        System.out.println( "s.doSomething() returns :"+doSomething());
```

```
    }
```

```
    public String doSomething(){return "Do something....." };
```

```
}
```

```
D. abstract class Name {
```

```
    Private String name ;
```

```
    Public abstract int test();
```

```
}
```

18.下面的java 程序编译运行结果是 C

```
interface A {int x = 0;}
```

```
Class B {int x = 1;}
```

```
Class C extends B implements A {
```

```
    Public void pX () {system.out.println(super.x);}
```

```
    Public static void main (String [] args ){new C ().Px();}
```

```
}
```

- A 产生运行期错误
- B 产生编译期错误
- C 程序运行，输出结果为 1
- D 存续运行，输出结果为 0

19.下列程序段的描述正确的是 AD

假设 int i = 10;

```
int a (int i){
```

```
    return ++ i;
```

```
}
```

```
int b (final int i){
```

```
    return ++ i;
```

```
}
```

```
int c (integer i){
```

```
    return ++ i;
```

```
}
```

- A 调用方法 a , 返回 11
- B 调用方法 a , 返回 10
- C 调用方法 c , 返回 10
- D 调用方法 c , 返回 11

20.In linux, the command (c) can change the user and / or group ownership of each given file_____ A _____

- A chmod B touch C chown D cat

21.java.io 包中定义了多个流类型来实现输入和输出功能，可以从不同的角度对其进行分类，按功能分为：(C) ,如果为读取的内容进行后台处理后在输出，需要使用下列哪种流？

(A)

- A、 输入流和输出流
- B、 字节流和字符流
- C、 节点流和处理流
- D、 File stream
- E、 Pipe stream
- F、 Random stream
- G、 Filter stream

22 . 下列代码的执行结果是：(B)

```
public class Test3{
```

```

public static void main(String args[]){

System.out.print(100%3);

System.out.print( " " );

System.out.print(100%3.0);

}

}

```

- A、 .1 , 1
- B、 1 , 1.0
- C、 1.0 , 1
- D、 1.0 ,1.0

23、在继承中，关于构造方法的说明，下列说法错误的是（ D ）

- A、子类无条件的继承父类中的无参构造方法
- B、子类可以引用父类中的有参构造方法，使用 super 关键字
- C、如果子类没有构造方法，则父类无参构造方法作为自己的构造方法
- D、如果子类有无参的构造方法，而父类的无参构造方法则被覆盖

24、以下的程序的运行结果为（ D ）

```

Public class IfTest{

Public static void main(String args[]){

int x=3;

int y=1;

```

```
if(x==y)

System.out.println( "Not equal" );

Else

System.out.println( "Equal" );

}

}
```

- A、 Not equal
- B、 Equal
- C、 无输出
- D、 编译错误

25、Java 语言中字符串 “学 java” 所占的内存空间是 (C)

- A、 6 个字节
- B、 7 个字节
- C、 10 个字节
- D、 11 个字节

26.下列哪些语句关于内存回收的说明是正确的？ (b)

- A . 程序员必须创建一个线程来释放内存
- B. 内存回收程序负责释放无用内存
- C. 内存回收程序允许程序员直接释放内存
- D. 内存回收程序可以在指定的时间释放内存对象

27、关于 ContentValues 类说法正确的是 (a)。

A. 他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是 String 类型，而值都是基本类型。

B. 他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是任意类型，而值都是基本类型。

C. 他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名，可以为空，而值都是 String 类型。

D. 他和 Hashtable 比较类似，也是负责存储一些名值对，但是他存储的名值对当中的名是 String 类型，而值也是 String 类型。

28、JAVA 中，对于 ServletRequest 接口的获取请求参数的方法的描述，正确的是 (d)。

A. getParameter 方法只用于接收 POST 请求参数，接收 GET 请求参数需要使用 getQueryString 方法。

B. 如果一个参数 key 有多个值，那么 getParameter(key)方法会返回空字符串。

C. 如果一个参数 key 有多个值，那么 getParameterValues(key)方法会返回一个包含所有值的字符串数组。

D. getParameter 方法返回 Object 对象，使用前要强制类型转换：如 String str=(String)request.getParameter(key)。

29、在 Spring 中使用 Hibernate 实现 DAO 层时，一般需要从 (d) 类继承，以简化 DAO 代码的书写。

A. HibernateTemplate

B. HibernateDao

C. HibernateSupport

D. HibernateDaoSupport

30、关于下列程序段的输出结果，说法正确的是（ A ）

```
public class MyClass{  
  
    static int i;  
  
    public static void main(String args[]){  
  
        System.out.println(i);  
  
    }  
  
}
```

A、 有错误，变量 i 没有初始化

B、 Null

C、 1

D、 0

31、下列那些语句关于内存回收的说明是正确的（ B ）

A、 程序员必须写一个内存来释放内存

B、 内存回收程序负责释放无用内存

C、 内存回收程序允许程序员直接释放内存

D、 内存回收程序可以在指定的时间释放内存对象

32、下面异常是属于 Runtime Exception 的是（ B C ）？（多选）

A、 ArithmeticException

B、 IllegalArgumentException

C、NullPointerException

D、BufferUnderflowException

33、Math.round(11.5)等于多少？Math.round(-11.5)等于多少（C）

A、11，-11

B、11，-12

C、12，-11

D、12，-12

34、下列程序段输出的结果是：（B）

```
Void complicatedexpression_r(){
```

```
int x=20,y=30;
```

```
boolean b;
```

```
b=x>50&&y>60||x>50&&y<-60||x<-50&&y>60||x<-50&&y<-60;
```

```
System.out.println(b)
```

```
}
```

A、 true

B、 false

C、 1

D、 0

二、简答题

1 简述你对 java 反射机制的理解 JAVA 反射机制是在运行状态中，对于任意一个类，都

能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性

2 简述数据库事务的 ACID 属性指在 DBMS 中事务的四个特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）

3 以下 linux 命令的作用：chmod pwd ls mkdir 改变权限，查看路径，创建文件夹

4 面向对象与面向过程的区别面向过程是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现；面向对象是把构成问题事务分解成各个对象，建立对象描述某个事物在整个解决问题的步骤中的行为。

5 写出 java 异常处理的语句并描述各部分的作用

try{：开始捕获异常

}catch{：捕获到异常的处理代码块

}finally{：运行时必须执行的代码

throw：抛出异常

throws：在类定义时将类中产生的一场抛给上一级调用方法

6 程序运行的结果是：__good and gbc_____。

```

Public class Example{

    String str=new String("good");  char[]ch={'a','b','c'};

    public static void main(String args[]){

        Example ex=new Example();

        ex.change(ex.str,ex.ch);

        System.out.print(ex.str+ " and ");

        Sytem.out.print(ex.ch);

    }

    Public void change(String str,char ch[]){

        str="test ok";

        ch[0]='g';

    }

}

```

7 常用的两个 List 的非同步实现类是_____ArrayList、LinkedList _____。

8 数据库正常启动所经历的几种为__STARTUP NOMOUNT 数据库实例启动、
 _____STARTUP MOUNT - 数据库装载____、_____、STARTUP OPEN 数据库打
 开。

9 Java 中线程的四种状态分别是_运行____、____就绪____、挂起、结束；让线程从运行状态转换为暂停状态的有____睡眠__、等待。

10 struts2 框架中为了处理中文乱码问题，需在 JSP 页面中设置__pageEncoding=utf-8____，同时也应该在 struts.xml 中设置__struts.i18n.encoding=utf-8____。

11 Spring 简化了 Bean 的配置，提供了自动装配(autowire)机制，根据指定的原则(通过<Bean>的 autowire 属性指定)进行 Bean 的自动装配，Spring 本身为 autowire 属性提供了__byName____、__byType____、__constructor____、autodetect 以及 no 五个选项。

12 java.io 包中的____ObjectInputStream____和__ObjectOutputStream____类主要用于对对象(Object)的读写。

13 谈谈hashmap和hashtable的区别，如何去遍历一个map

答：

Hashtable继承了Dictionary，是线程安全的，键值对中不能出现null。

HashMap实现了Map接口，是Hashtable的轻量级实现（非线程安全的实现），即多个线程访问HashMap时需要为其准备外同步，且HashMap的键、值均可以为null。

遍历map有两种方式：keyset（效率低）和entryset（效率高），两种方式都需要创建Iterator迭代器对象，不同的是使用keyset需要从迭代器中获取map的key，并且通过key获取value（key=iterator.next(); value=map.get(key)）；使用entryset则需要从迭代器中获取Map.Entry对象，再从中获取map的key和value（Map.Entry entry=(Map.Entry)iterator.next(); key=entry.getKey(); value=entry.getValue()）。

```
Map map=new HashMap();
```

```
Iterator iterator=map.entrySet().iterator();
```



```

while(iterator.hasNext()){
    Map.Entry entry=(Map.Entry)iterator.next();
    Object key=entry.getKey();
    Object value=entry.getValue();
}
Map map=new HashMap();
Iterator iterator=map.keySet().iterator();
while(iterator.hasNext){
    Object key=iterator.next();
    Object value=map.get(key);
}

```

14 .说出线程同步的几个方法

答：

synchronized同步代码块：synchronized(同步监视器){...}

synchronized修饰的同步方法：public synchronized 返回值 方法名(){...}

wait();：wait方法必须在用有synchronized修饰的地方才能使用

notify();：唤醒了有相同同步监视器的wait()，从wait()的下一行代码开始执行

notifyAll();：唤醒了所有调用wait()的方法

15 java中char占几个字节，为什么

答：2个字节。java是采用unicode编码，使用16位的编码空间，即每个字符都占两个字节

16 .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之前还是之后

答：会被执行，在return之后执行，finally{}中的语句只有在虚拟机停止的情况下才不执行。

17 从操作系统角度请阐述一下线程与进程的区别

答：进程是系统进行资源分配和调度的一个独立单位，线程是CPU调度和分派的基本单位

进程和线程的关系：

(1) 一个线程只能属于一个进程，而一个进程可以有多个线程，但至少有一个线程。

(2) 资源分配给进程，同一进程的所有线程共享该进程的所有资源。

(3) 线程在执行过程中，需要协作同步。不同进程的线程间要利用消息通信的办法实现同步。

(4) 处理机分给线程，即真正在处理机上运行的是线程。

(5) 线程是指进程内的一个执行单元，也是进程内的可调度实体。

线程与进程的区别：

(1) 调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位。

(2) 并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可以并发执行。

(3) 拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源。

(4) 系统开销：在创建或撤销进程的时候，由于系统都要为之分配和回收资源，导致系统的明显大于创建或撤销线程时的开销。但进程有独立的地址空间，进程崩溃后，在保护模式下不会对其他的进程产生影响，而线程只是一个进程中的不同的执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但是在进程切换时，耗费的资源较大，效率要差些。

18 请阐述一下你对 java 多线程中同步的几种方式，区别，及其重要性。

(1) wait 方法：

该方法属于 Object 的方法，wait 方法的作用是使得当前调用 wait 方法所在部分（代码块）的线程停止执行，并释放当前获得的调用 wait 所在的代码块的锁，并在其他线程调用 notify 或者 notifyAll 方法时恢复到竞争锁状态（一旦获得锁就恢复执行）。

调用 wait 方法需要注意几点：

第一点 :wait 被调用的时候必须在拥有锁(即 synchronized 修饰的)的代码块中。

第二点 :恢复执行后,从 wait 的下一条语句开始执行,因而 wait 方法总是应当在 while 循环中调用,以免出现恢复执行后继续执行的条件不满足却继续执行的情况。

第三点 :若 wait 方法参数中带时间,则除了 notify 和 notifyAll 被调用能激活处于 wait 状态 (等待状态) 的线程进入锁竞争外,在其他线程中 interrupt 它或者参数时间到了之后,该线程也将被激活到竞争状态。

第四点 :wait 方法被调用的线程必须获得之前执行到 wait 时释放掉的锁重新获得才能够恢复执行。

(2) notify 方法和 notifyAll 方法 :

notify 方法通知调用了 wait 方法 ,但是尚未激活的一个线程进入线程调度队列(即进入锁竞争), 注意不是立即执行。并且具体是哪一个线程不能保证。另外一点就是被唤醒的这个线程一定是在等待 wait 所释放的锁。

notifyAll 方法则唤醒所有调用了 wait 方法, 尚未激活的进程进入竞争队列。

(3) synchronized 关键字 :

第一点 :synchronized 用来标识一个普通方法时, 表示一个线程要执行该方法, 必须取得该方法所在的对象的锁。

第二点 :synchronized 用来标识一个静态方法时, 表示一个线程要执行该方法, 必须获得该方法所在的类的类锁。

第三点：synchronized 修饰一个代码块。类似这样：synchronized(obj)

{ //code..... }。表示一个线程要执行该代码块，必须获得 obj 的锁。这样做的目的是减小锁的粒度，保证当不同块所需的锁不冲突时不用对整个对象加锁。利用零长度的 byte 数组对象做 obj 非常经济。

(4) atomic action (原子操作):

在 JAVA 中，以下两点操作是原子操作。但是 c 和 c++ 中并不如此。

第一点：对引用变量和除了 long 和 double 之外的原始数据类型变量进行读写。

第二点：对所有声明为 volatile 的变量 (包括 long 和 double) 的读写。

另外：在 java.util.concurrent 和 java.util.concurrent.atomic 包中提供了一些不依赖于同步机制的线程安全的类和方法。

19 、final,finally,finalize 的区别。

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。

20 、`sleep()`和 `wait()`有什么区别？

`sleep` 是线程被调用时,占着 `cpu` 去睡觉,其他线程不能占用 `cpu`,`os` 认为该线程正在工作,不会让出系统资源,`wait` 是进入等待池等待,让出系统资源,其他线程可以占用 `cpu`

21、`abstract` 的 `method` 是否可同时是 `static`,是否可同时是 `native`,是否可同时是 `synchronized`?

不可以

22、当一个线程进入一个对象的一个 `synchronized` 方法后,其他线程是否可以进入此对象的其他方法？

可以去访问非 `Synchronized` 方法。

23、当一个对象被当作参数传递到一个方法后,此方法可改变这个对象的属性,并可返回变化后的结果,那么这里到底是值传递还是引用传递？

值传递

24、创建一个 `class` 对象有几种方法？分别是什么？

1. `Class c=Class.forName("java.lang.String");`
2. `String str=new String();`

```
Class c=Str.getClass();
```

```
3.Class c=String.Class;
```

25、如何获取一个目录下有多少个文件？

```
File f=new File( "D:/" );
```

```
Int count=f.list().length;
```

26、请写出

下列代码

的输出结果：

FatherClass

ChildClass

FatherClass

ChildClass

ChildClass

Create

```
public class FatherClass
{
    Public FatherClass()
    {
        System.out.println( "FatherClass Create" );
    }
}
```

```
{
    public ChildClass()
    {
        System.out.println( "ChildClass Create" );
    }
}
public static void main(String[] args)
{
    FatherClass fc = new FatherClass();
    ChildClass cc = new ChildClass();
}
}
```

27、请用 Socket 写一个客户端，并从中读出一行打印出来。

```
Socket socket=null;
```

```
try
```

```

{
    socket=new Socket("127.0.0.1",8888);

    BufferedReader      pw      =      new      BufferedReader(new
InputStreamReader(socket.getInputStream()));

    System.out.println(pw.readLine());

    pw.close();
}

catch(Exception e)

{

    e.printStackTrace();

}

Finally

{

    Try

    {

        socket.close();

    }

    catch(IOException e

    {

        e.printStackTrace();

```



```
}
```

```
}
```

28、当一个对象不再被使用时，如何才能从内存中消失？

将该对象引用值为空