抽象方法需要子类重写，而静态的方法是无法被重写的

1. ****阐述静态变量和实例变量的区别。****  
   答：静态变量是被static修饰符修饰的变量，也称为类变量，它属于类，不属于类的任何一个对象，一个类不管创建多少个对象，静态变量在内存中有且仅有一个拷贝；实例变量必须依存于某一实例，需要先创建对象然后通过对象才能访问到它。静态变量可以实现让多个对象共享内存。
2. 是否可以从一个静态（static）方法内部发出对非静态（non-static）方法的调用？  
   答：不可以，静态方法只能访问静态成员，因为非静态方法的调用要先创建对象，在调用静态方法时可能对象并没有被初始化
3. 内部类必须在外部类实例化后才能实例化，一个内部类对象可以访问创建它的外部类对象的成员，包括私有成员
4. Java 中的final关键字有哪些用法？  
   答：(1)修饰类：表示该类不能被继承；(2)修饰方法：表示方法不能被重写；(3)修饰变量：表示变量只能一次赋值以后值不能被修改（常量）。
5. 创建对象时构造器的调用顺序是：先初始化静态成员，然后调用父类构造器，再初始化非静态成员，最后调用自身构造器。相关代码见mavenTest工程com.yx.mavenTest包下A,B,C类；

32、怎样将GB2312编码的字符串转换为ISO-8859-1编码的字符串？  
答：代码如下所示：

|  |  |
| --- | --- |
| 1  2 | String s1 = "你好";  String s2 = new String(s1.getBytes("GB2312"), "ISO-8859-1"); |
|  |  |

1. 一般来说，断言用于保证程序最基本、关键的正确性。断言检查通常在开发和测试时开启。为了保证程序的执行效率，在软件发布后断言检查通常是关闭的。断言不应该以任何方式改变程序的状态。简单的说，如果希望在不满足某些条件时阻止代码的执行，就可以考虑用断言来阻止它。
2. Error表示系统级的错误和程序不必处理的异常，是恢复不是不可能但很困难的情况下的一种严重问题；比如内存溢出，不可能指望程序能处理这样的情况；Exception表示需要捕捉或者需要程序进行处理的异常，是一种设计或实现问题；也就是说，它表示如果程序运行正常，从不会发生的情况。
3. 用递归编写程序时一定要牢记两点：1. 递归公式；2. 收敛条件（什么时候就不再继续递归）。
4. try{}里有一个return语句，那么紧跟在这个try后的finally{}里的代码会不会被执行，什么时候被执行，在return前还是后?  
   答：会执行，在方法返回调用者前执行。在finally中改变返回值的做法是不好的，因为如果存在finally代码块，try中的return语句不会立马返回调用者，而是记录下返回值待finally代码块执行完毕之后再向调用者返回其值，然后如果在finally中修改了返回值，就会返回修改后的值。显然，在finally中返回或者修改返回值会对程序造成很大的困扰*，*
5. finally为确保一段代码不管发生什么异常状况都要被执行；try语句可以嵌套，每当遇到一个try语句，异常的结构就会被放入异常栈中，直到所有的try语句都完成。如果下一级的try语句没有对某种异常进行处理，异常栈就会执行出栈操作，直到遇到有处理这种异常的try语句或者最终将异常抛给JVM。
6. 列出一些你常见的运行时异常？  
   答：  
   - ArithmeticException（算术异常）  
   - ClassCastException （类转换异常）  
   - IllegalArgumentException （非法参数异常）  
   - IndexOutOfBoundsException （下标越界异常）  
   - NullPointerException （空指针异常）  
   - SecurityException （安全异常）
7. ArrayList 和Vector都是使用数组方式存储数据，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中的方法由于添加了synchronized修饰，因此Vector是线程安全的容器，但性能上较ArrayList差，因此已经是Java中的遗留容器；LinkedList使用双向链表实现存储，形成一个可以按序号索引的线性结构，与数组的连续存储方式相比，内存的利用率更高，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，由于ArrayList和LinkedListed都是非线程安全的，如果遇到多个线程操作同一个容器的场景，则可以通过工具类Collections中的synchronizedList方法将其转换成线程安全的容器后再使用。
8. Collection和Collections的区别？  
   答：Collection是一个接口，它是Set、List等容器的父接口；Collections是个一个工具类，提供了一系列的静态方法来辅助容器操作，这些方法包括对容器的搜索、排序、线程安全化等等。
9. TreeMap和TreeSet在排序时如何比较元素？Collections工具类中的sort()方法如何比较元素？  
   答：TreeSet要求存放的对象所属的类必须实现Comparable接口，该接口提供了比较元素的compareTo()方法，当插入元素时会回调该方法比较元素的大小。TreeMap要求存放的键值对映射的键必须实现Comparable接口从而根据键对元素进行排序。Collections工具类的sort方法有两种重载的形式，第一种要求传入的待排序容器中存放的对象比较实现Comparable接口以实现元素的比较；第二种不强制性的要求容器中的元素必须可比较，但是要求传入第二个参数，参数是Comparator接口的子类型（需要重写compare方法实现元素的比较），相当于一个临时定义的排序规则，其实就是通过接口注入比较元素大小的算法，也是对回调模式的应用。前者在定义对象属性时就重写compareTo方法，short方法则需要传入要比较的元素容器和要比较的对象Comparator<>,之后重写compare方法。
10. sleep()方法（休眠）是线程类（Thread）的静态方法，调用此方法会让当前线程暂停执行指定的时间，将执行机会（CPU）让给其他线程，但是对象的锁依然保持，因此休眠时间结束后会自动恢复，线程回到就绪状态；wait()是Object类的方法，调用对象的wait()方法导致当前线程放弃对象的锁（线程暂停执行），进入对象的等待池（wait pool），只有调用对象的notify()方法（或notifyAll()方法）时才能唤醒等待池中的线程进入等锁池（lock pool），如果线程重新获得对象的锁就可以进入就绪状态。线程是进程的一个实体，是CPU调度和分派的基本单位，
11. 线程的sleep()方法和yield()方法有什么区别？  
    答：① sleep()方法给其他线程运行机会时不考虑线程的优先级，因此会给低优先级的线程以运行的机会；yield()方法只会给相同优先级或更高优先级的线程以运行的机会；  
    ② 线程执行sleep()方法后转入阻塞（blocked）状态，而执行yield()方法后转入就绪（ready）状态；  
    ③ sleep()方法声明抛出InterruptedException，而yield()方法没有声明任何异常；  
    ④ sleep()方法比yield()方法（跟操作系统CPU调度相关）具有更好的可移植性。
12. 当一个线程进入一个对象的synchronized方法A之后，其它线程是否可进入此对象的synchronized方法B？synchronized关键字可以将对象或者方法标记为同步，  
    答：不能。其它线程只能访问该对象的非同步方法，同步方法则不能进入。因为非静态方法上的synchronized修饰符要求执行方法时要获得对象的锁，如果已经进入A方法说明对象锁已经被取走，那么试图进入B方法的线程就只能在等锁池（注意不是等待池哦）中等待对象的锁。
13. 请说出与线程同步以及线程调度相关的方法。  
    答：- wait()：使一个线程处于等待（阻塞）状态，并且释放所持有的对象的锁；  
    - sleep()：使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要处理InterruptedException异常；  
    - notify()：唤醒一个处于等待状态的线程，当然在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由JVM确定唤醒哪个线程，而且与优先级无关；  
    - notityAll()：唤醒所有处于等待状态的线程，该方法并不是将对象的锁给所有线程，而是让它们竞争，只有获得锁的线程才能进入就绪状态；
14. 编写多线程程序有几种实现方式？一种是继承Thread类；另一种是实现Runnable接口。两种方式都要通过重写run()方法来定义线程的行为，推荐使用后者，因为Java中的继承是单继承，实现Callable接口，该接口中的call方法可以在线程执行结束时产生一个返回值。
15. 所谓的同步就是指阻塞式操作，而异步就是非阻塞式操作.
16. 启动一个线程是调用run()还是start()方法？  
    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()方法是线程启动后要进行回调（callback）的方法.