



不要慌，问题不大

# 数据采集48份Java试卷简答题

数据采集48份Java试卷简答题

面向对象的特征有哪些方面？

抽象，封装，继承，多态。

Java的OOP中的动态绑定指的是什么？有什么优点？

动态绑定是指在执行期间（非编译期）判断所引用对象的实际类型，根据其实际的类型调用其相应的方法。程序运行过程中，把方法的调用与响应调用所需要的代码相结合的过程称为动态绑定。

优点是不需要对现存的代码进行修改，就可以对程序进行扩展。

简述垃圾回收的优点和原理。

优点: Java语言中一个显著的特点就是引入了垃圾回收机制，它使得Java程序员在编写程序的时候不用考虑内存管理，垃圾回收可以有效的防止内存泄漏。

原理: 垃圾回收器通常是作为一个单独的低级别的线程运行，对于那些已经死亡的或者很久没有使用的对象进行清除和回收，但是我们程序员不能实时的调用垃圾回收器对某个对象进行垃圾回收。

编写Web应用程序时，客户端向服务器端发出服务请求，服务器每次根据客户的请求创建线程，他们之间的通信结束后销毁线程，采用何种技术避免反复创建和销毁线程而带来的系统资源开销过大问题？

对于一个web程序，每当一个用户请求服务器连接时，服务器就可以启动一个新的线程来为用户服务。但是每次创建和销毁线程会有一定的开销，所以线程池就是避免为了频繁创建和销毁线程而存在的。就是说每当服务器接收到一个新的请求之后，服务器就从线程池中挑选一个等待的线程并执行请求处理。处理完毕后，线程并不结束，而是转为阻塞状态再次放到线程池中，这样就避免了频繁创建和销毁线程。

为什么wait()/notify()/notifyall()必须放在synchronized代码块中？为什么 wait()/ notify()  
()/notifyall()必须成对使用？

当有一个线程进入synchronized代码块的时候，这个线程就拥有了对象锁，其他的线程就不能进入这个代码块。而wait()/notify()/notifyall()只能由拥有对象锁的线程来调用，所以这三个方法要放在synchronized代码块中。

如果一个线程只调用wait()方法，那么其他的线程将会因为申请不到资源而进入阻塞状态，这时候Java

程序将会被挂起，程序也会不能再继续前进，就会出现死锁现象，因此我们需要使用`notify()`或者`notifyall()`方法通知其他线程继续申请资源。所以这三个方法必须成对使用。

列出Java中的访问控制符，并说明各个访问控制符的控制权限。

- ☐ `public`: 公共的，一个类如果被声明为公共类，它可以在当前类，子孙类，同一个包类，不同的包内都能被访问到。
- ☐ `protected`: 被保护的，刚刚写了段代码，如果不是内部类，是不能声明为`protected`的，外部类只能声明为`public`或者默认的`friendly`，`protected`修饰的成员变量可以被三种类所引用: 该类自身，与他在同一个包类的其他类，在其他包中的该类的子类。使用`protected`修饰符的主要作用是允许其它包内中的他的子类来访问父类的某些特定的属性。
- ☐ `friendly`: 在当前类和同一个包中能被访问到，继承父类的子类无法访问到父类中`friendly`修饰的成员变量。
- ☐ `private`: 只能在当前类中被访问到。子孙类可以被访问。 > 简述`abstract class`与`Interface`的区别？以及它们的作用？

`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类的情况。`interface`是抽象类的实体。在接口中，所有的方法都是抽象的。多继承性可以通过实现这样的接口而获得。

不同点：

1. 抽象类中可以有构造方法，但是接口中不能有；
2. 抽象类中可以有普通的成员变量，接口中不能有普通的成员变量。但是我试过，可以定义`public int a = 1;`这点要特别注意的是，就算你定义的是变量，但是`a`还是个常量。默认是`public final int a = 1;`
3. 抽象类中包含抽象的普通方法，接口中所有的方法必须是抽象的，不能有非抽象的方法。
4. 抽象类中可以包含静态方法，但是接口中不可以。在1.8之前的版本好像接口是不能有静态方法的，但是1.8的版本我测试是可以写静态方法的。而且可以写方法体。现在的学习慢慢在打破以前学过的旧理论了。
5. 抽象类中的抽象方法的访问类型可以是`public`，`protected`。但是接口中的访问类型默认是`public`。

Java中为什么要引入布局管理器？

什么是异常？Java为什么要引入异常处理机制？

异常就是程序执行过程中的不正常，它会打断程序的执行流程。在程序运行的过程中发生的异常事件，比如除0溢出，数组越界，文件找不到等，都会打断指令的正常流程。  
为了加强程序的健壮性，必须考虑到可能发生的异常事件并做出相应的处理。

在一个类中，如果局部变量、实例变量和其父类的变量同名，如何加以区分？

在一个类中，使用`this.`变量名代表本类中的实例变量，用前缀`super.`变量名代表父类的变量。

在Java的网络编程中，怎样使服务器为多个客户服务？

使用多线程技术，即创建多个线程处理多个连接，每个线程对应一个客户。这样既可以满足多用户的同时连接请求，也能够使这些线程资源共享，有效的提高了效率。

简要说明Java通过哪两种方式实现多线程操作，他们有什么主要区别？

Java有两种方式实现多线程操作：实现`Runnable`接口和继承`Thread`类。如果只对`run()`方法进行编程，使用两种方法都可以；如果还要对其他的方法（`start()`）进行编程需要继承`Thread`类。

简述Java通过JDBC进行数据库操作的步骤。

1. 打开一个数据库连接
2. 创建一个`statement`对象
3. 执行一个查询并且返回一个结果集
4. 处理结果集
5. 关闭结果集和`statement`对象
6. 关闭连接

怎样用数据输入流和输出流读写数据？

数据流`DataInputStream`和`DataOutputStream`用机器独立形式分别读和写Java的原始数据类型，允许使用者在一机器上写数据文件，以及从其他装有不同操作系统或者文件结构的机器上读取文件。他们常用于输入输出数据。

简述Java面向对象开发中重载和覆盖的区别。

在一个类中，如果多个方法有相同的名字，但是有不同的参数，那么便构成了重载。由于方法的返回值不属于方法的签名部分，所以返回值不同不属于重载。

覆盖是指子类继承父类，并且重写父类中的方法，方法必须要有相同的参数，相同的返回值。

举例说明什么是上转型对象，上转型对象的操作原则是什么？

上转型对象，如Father类是Son的父类，`Father person = new Son()`；则person是Son对象的上转型对象，上转型对象可以强制转化为子类的对象，如`Son son = (Son)person`；上转型对象不能操作子类新增的成员变量和方法，上转型对象可以操作子类继承或者重写的成员变量和方法，如果子类重写了父类方法，则上转型对象调用的是子类重写的方法。上转型对象可再强制转换到一个子类对象，该子类对象又具备了子类所有的属性。

举例说明事件处理机制的原理和使用方法。

首先需要在事件源(文本框，按钮，下拉列表)中登记事件监听器，即注册事件监听器。当有事件发生时，Java虚拟机会产生一个事件对象，事件对象记录处理该事件所需要的各种信息，当事件源接收到事件对象时，就会启动在该事件源中注册的事件监听器，并将事件对象传递给相应的事件监听器，这时事件监听器接收到事件对象并对事件进行处理。

说明List和Set的区别。

1. List主要用来处理序列，Set用来处理集；
2. List中的内容可以重复，但是Set中不可以。 > 什么是多态？如何实现多态？

多态是指在基类中定义的属性或方法被派生类继承后，可以具有不同的数据类型，或者表现出不同的行为，从而使得同一个属性或方法在基类及其派生类中具有不同的语义。

Java的多态性表现在两个方面：

1. 方法重载实现的静态多态性，又称为编译时多态；
2. 方法覆盖/重写实现的动态多态性，又称为运行时多态。运行时多态就是动态绑定，程序在执行时判断所引用对象的实际类型，根据实际类型调用相应的属性和方法。

在变量前面加上static 和不加static 有什么区别？

被static修饰的属性成为静态属性，这类属性的一个最基本的特点就是：它们是类的属性，而不属于任何一个类的具体对象。换句话说对于该类的任何一个具体对象而言，静态属性是一个公共的存储单元，任何一个类的对象访问它时，取到的都是相同的数值，同样对于一个类的对象去修改它时，也都是在对同一个内存单元做操作，它可以在类不被实例化时使用。

感觉没答道点子上，我再补充一点：

加上static之后：可以直接通过类名. 静态变量得到这个值。

不加static之后：需要实例化这个类，再通过对象. 变量访问这个值。

char型变量中能不能存储一个中文汉字？为什么？

Java中的字符不是指占一个字节的ASCII字符，而是占2个字节的Unicode字符，因为Unicode字符被设计用来处理现在世界上所有书面语言的字符，所以一个汉字也是被当作一个字符来处理的。

在编写Java的循环结构程序时，“break”和“continue”各自的作用是什么？

在Java语言中，break和continue都可以用来控制循环的流程。其中，break用户强行退出循环，不执行循环中剩下的语句，而执行循环后面的语句。continue则是停止执行当前的循环，开始新的循环。

Java在进行异常处理时，throw和throws语句的作用有何不同？

throw的作用为手动抛出异常，在程序中若是抛出异常则采用如下形式：throw e;throws的作用为声明方法抛出可能出现可能异常并将这些异常抛给这个方法的调用者那里，有点像下属处理不了的问题就交到上司手里一样，这种情况称为回避异常。

在编写异常处理程序时，使用的多重catch语句有明确的父子关系时，应该如何安排各catch的顺序，为什么？

应该先将子catch语句放在前面，父catch块放在后面，否则放在后面的子catch块永远得不到执行。

Java事件编程包括哪几个步骤？

1. 创建事件监听器类，在相应的方法中编写事件处理代码。
2. 声明一个事件监听器对象。

3. 将事件监听器对象添加到事件源的监听器列表。