1. 简单

Java基础

1. 什么是多态？

多态是指一个名字多种实现。多态使得一个实体通过一个通用的方式来实现不同的操作。具体的操作是由实际的实现来决定的。

多态在Java里有三种表现方式：a)方法重载, b)通过继承实现方法重写, c)通过Java接口进行方法重写。

1. Java中的访问修饰符有哪些？它们有什么区别？

访问权限修饰符是表明类成员的访问权限类型的关键字。使用这些关键字来限定程序的方法或者变量的访问权限。它们包含：

* public: 所有类都可以访问
* protected: 同一个包内以及所有子类都可以访问
* private: 只有归属的类才能访问
* 默认: 归属类及相同包下的子类可以访问

1. Java中int是多少位？Integer和int有什么异同？Integer为什么要避免使用＝＝比较相等？

32bit。

Integer是int的封装类。Int的默认初始值为0，而Integer为null。

Integer比较相等，仅能使用equals方法。原因：Integer为－127～128初始化了整数对象，保存在IntegerCache中。若使用Integer i=100方式，则i使用的IntegerCache中的对象。因此，对于－127～128的Integer对象用＝＝比较，则为true，其他的为false。

1. 静态变量的特点？
2. 声明时使用static修饰符
3. 被类的所有实例共享
4. 位于方法区
5. &和&&区别

&是位运算符，对于boolean类型，运算结果与逻辑与相同。

&&是逻辑与运算符。

&&具有短路功能，当左操作数为false时，不执行右操作数。因此逻辑运算中包含赋值运算时，需要考虑短路功能带来的影响。

1. transient关键字的含义？

Transient 表示一个域不是该对象串行化的一部分。当一个对象被串行化的时候，transient型变量的值不包括在串行化的表示中，然而非transient型的变量是被包括进去的。

1. Java中如何判断两个对象内容相同？如何判断是否是同一个对象？

判断对象内容：equals方法，对于自定义的class，需要重写equals方法

判断是否同一对象：＝＝

1. Finalize作用？final关键字作用？

finalize()方法在一个对象被销毁和回收前会被调用。

Final：组织继承。声明类时使用final，类不可继承；方法声明为final时，子类不能覆盖方法；域声明为final，构造对象后不允许改变域的值。

1. 抽象类和接口区别？

语法角度：

1. 抽象类可以实现方法，接口中的方法均为abstract
2. 抽象类单继承，接口多继承
3. 抽象类中的方法不能为private；接口中的方法均为public，不可修改

语义角度：

1. 抽象类定义的是对象公共属性及行为；接口约定的是对象具有的行为
2. Java8中引入了默认方法，二者可以具备相同语义。但是接口的多继承会导致在实现类中需要指定使用的默认方法。
3. 什么是值传递？什么是引用传递？Java中参数传递类型是哪种？

对象被值传递，意味着传递了对象的一个副本。因此，就算是改变了对象副本，也不会影响源对象的值。

对象被引用传递，意味着传递的并不是实际的对象，而是对象的引用。因此，外部对引用对象所做的改变会反映到所有的对象上。

Java中的参数传递均为值传递。

1. 什么是运行时异常和一般异常？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机操作中可能遇到的异常，是一种常见运行错误。Java编译器要求方法声明必须抛出可能发生的非运行时异常，但不要求必须声明抛出而未被捕获的运行时异常。

常见的运行时异常：NullPointerException, IndexOutOfBoundsExcepiton, IllegaArgumentException

1. Java中线程如何实现线程？实现方式区别

有三种方式可以用来创建线程：

* 继承Thread类
* 实现Runnable接口
* 实现Callable接口

实现Runnable接口这种方式更受欢迎，因为这不需要继承Thread类。在应用设计中已经继承了别的对象的情况下，这需要多继承（而Java不支持多继承），只能实现接口。同时，线程池也是非常高效的，很容易实现和使用。

1. 什么是Lambda表达式？什么场景下适用？（JDK8）

lamda表达式： 匿名函数

Lamda表达式可以使代码简洁，便于阅读，适用于代码复用比较少的情景，例如：做为事件处理函数，比较函数等。

1. 什么是泛型？列举熟悉的范型类？

泛型：参数化类型。在定义Class或Interface时，可以通过<T>方式指定处理的对象类型。例如：ArrayList<Integer>。

其他常用泛型类型:HashMap<K,V>

1. String、StringBuffer和StringBuilder的异同

三者均表示字符串。

String为final类型，不可继承，不能修改字符串的字符。

StringBuffer和StringBuilder类似于字符串数组，用来动态构造字符串数据。StringBuffer是线程安全的，StringBuilder非线程安全。通常使用StringBuilder。

String在拼接时，会生成新的字符串，效率较低。当有大量字符串拼接或修改操作时，应使用StringBuffer或StringBuilder。

1. 什么是Mark Interface？作用？

Mark Interface（标记接口）中未声明任何方法需要自类实现，仅表示子类具有某类特征。例如：Serializable中无方法声明，仅用来表明实现该接口的类为可序列化的。标记接口可用于基础框架中，用于对具有该接口的类进行处理。应用程序开发中，避免使用此类无功能的接口。

1. ArrayList和LinkedList异同？如果要排序，如何实现？

二者均为数据结构中线性表实现。ArrayList为动态数组，LinkedList为链表。

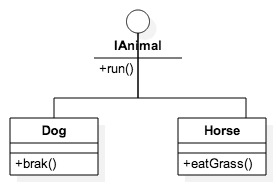
Java中实现对二者中元素排序，可通过Collections.sort实现排序。排序规则可继承Comparator来实现，或元素实现Comparable接口，重写compareTo方法实现排序规则。

1. 如果调用Collections.sort进行排序，需要对象实现什么接口？

实现Comparable接口，重写compareTo方法

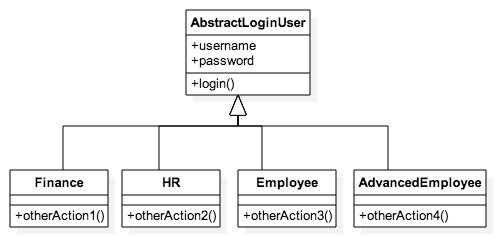
面相对象分析与设计基础

1. 请用UML描述下列场景：狗会叫会跑，马会吃草会跑。



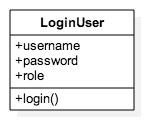
要点：

1. 抽取公共方法，放入接口
2. 也可定义dog和horse的接口，继承自IAnimal
3. 请用UML描述下列场景：一公司有财务、HR、普通员工、高级雇员四种角色，请设计账户体系，实现公司内部系统的统一账户登录。



示例中的设计，适用的场景为每个角色需要有不同行为。

若仅区分角色，也可使用下列设计。



编程

1. 请用Java中的栈模拟队列操作？或队列模拟栈操作？
2. 中等
3. 解释什么是反射？请举例反射的应用场景，优势和缺点

反射：运行时分析类的能力。

场景：

a）运行中分析类

b) 运行中查看对象

c）AOP

d）框架实现

优势：

1. 实现灵活，通用性好

缺点：

1. 存在性能问题
2. 调试不便
3. 什么是BIO ，NIO和AIO？

BIO： 同步阻塞式IO，Blocking I/O

在Jdk1.4之前，用Java编写网络请求都是建立在一个ServerSocket上，然后客户端简历Socket时会询问是否有线程可以处理：如果没有，要么拒绝，要么等到

一个连接，要求Server对应一个处理线程

NIO: 同步非阻塞式IO, Non-Blocking I/O

Jdk1.4之后新加入的网络编程jar包

* 在java.nio包及其子包内
* 基于事件驱动思想来完成的，其主要想解决的是BIO大并发问题
* NIO基于Reactor设计模式，当socket有流可读或可写入socket时，操作系统会相应的通知应用程序进行处理，应用程序再将流读取到缓冲区/写入操作系统

AIO： 异步非阻塞式IO, Asynchronous I/O

* 对于读操作而言，当有流可读取时，操作系统会将可读的流传入read方法的缓冲区，并通知应用程序
* 对于写操作而言，当操作系统将write方法传递的流写入完毕时，操作系统主动通知 应用程序
* 在Jdk1.7中被引入AIO，被称作NIO.2，主要在java.nio.channels中

1. 什么是注解？注解作用？

插入到源代码中用于某种工具处理的标签。

Java编译器对于包含注解和不包含注解的代码生成相同的虚拟机指令。因此，注解需要有对应的工具，才有效。

适用场景：

* 附属文件自动生成
* 测试、日志、事务语义等代码自动生成

1. 使用Java写文件，常使用什么类？

二进制文件：FileOutputStream, BufferedOutputStream等

文本文件：FileWriter，OutputStreamWriter等

1. HashMap和Hashtable区别？

Key Answer:

Hashtable线程安全，HashMap线程不安全

HashMap可将null作为key或者value;Hashtable 不可以

高性能并发访问时，推荐使用ConcurrentHashMap

1. CocurrentHashMap的实现原理？

锁分段

Reference:

https://www.ibm.com/developerworks/cn/java/java-lo-concurrenthashmap/

http://www.infoq.com/cn/articles/ConcurrentHashMap

1. Java实现生产者和消费者，使用何种队列？

BlockingQueue<E>的实现类，常用：

LinkedBlockingQueue<E>

ArrayBlockingQueue<E>

Hibernate

1. Hibernate二级缓存的应用

Hibernate 二级缓存独立于Hibnerate，通常使用Ehcache，需在SessionFactory中关联二级缓存。

二级缓存的好处：1. 提升性能

二级缓存不适用场景：1. 多应用写入数据库，会造成缓存与数据库不一致

Spring

1. 请简述对Spring的理解？

Spring框架早期主要提供IOC和AOP，用于Bean的托管，可以通过配置实现对象的创建和注入。

现在的Spring包含Spring-webmvc,Spring-Data, 原先的spring框架为spring framework项目中的一部分。

1. 简述Spring中AOP的实现原理。

AOP的基本原理是代理模式，调用方调用代理类，代理类与目标类具有相同的方法签名。实现有两种方式：1）JDK代理，基于接口生成实现类；2）CGLib代理，生成目标类的子类，实现代理。

AOP基于java的反射机制。

J2EE

1. 什么是EJB？EJB的基本Bean有哪些？

EJB，企业级JavaBean，是Java EE的规范之一。在该规范中，EJB分为Session Bean，Entity Bean和Message Driven Bean。Session Bean为非持久对象，用于实现服务器端业务逻辑组件。EntityBean为持久对象。Message Driven Bean通常用于消息通信。

1. 什么是Servlet？Servlet生命周期？

Java 服务器端小程序。

生命周期：加载、实例化、初始化、处理请求和服务结束。

Servlet被服务器实例化，容器运行期init方法，请求到达时运行service方法，service方法自动派遣运行与请求对应的doGet或者doPost方法，当服务器决定将实例销毁时，调用其destroy方法。

1. Java如何解析XML？

SAX: Simple API for XML Parsing。基于事件的XML解析方式。主要通过实现ContentHandler，当满足条件时调用对应的函数。

DOM：Document Object Model。DocumentBuilder的parse方法，返回Document对象。

第三方类库：dom4j，jdom

1. 什么是XPath？

XML的查询检索语言。

基本语法：

|  |  |
| --- | --- |
| Expression | Description |
| Nodename | 选择所有nodename的节点 |
| / | 从根节点选择 |
| // | 从当前文档选择所有匹配节点 |
| . | 当前节点 |
| .. | 父节点 |
| @ | 属性 |

1. 什么是DTD和XML Schema？作用？

DTD：Document Type Definition，文档类型定义。定义文档的元素，属性和结构。类型仅定义了PCDATA（parsed character data，会被解析器解析的文本）和CDATA（character data，不会被解析器解析）

XML Schema： 使用XML语法描述XML文档元素，属性，子元素次序，子元素数目和数据类型等。

二者都是XML文档元素和结构的定义。不同点：DTD使用自己的语法，XML Schema使用XML语法，提供不包含和数据类型。比DTD更完善。

1. 数字签名原理？

非对称加密算法和摘要算法。

发送方根据文件内容生成消息摘要，使用私钥加密消息摘要。

接收方使用发送方公钥解密，并验证生成的文件摘要和接收到的文件摘要是否相等。如果相等，则认为接收到的文件，未被修改过。

1. Forward与redirect区别？

二者均实现了页面的跳转。

不同点：

forward，为服务器端跳转，将请求转发到服务器端的另一个资源，并将结果返回到客户端，浏览器端的url显示地址不变。

redirect，为服务器端返回response，告知客户端重定向的地址，客户端向重定向地址发起请求，浏览器端的url显示发生改变。

1. 请解释RMI？原理？

Remote method invocation(远程方法调用)，客户端对象调用服务器端某一对象的方法。

一个正常工作的RMI系统由下面几个部分组成：

* 远程服务的接口定义
* 远程服务接口的具体实现
* 桩（Stub）和框架（Skeleton）文件
* 一个运行远程服务的服务器
* 一个RMI命名服务，它允许客户端去发现这个远程服务
* 类文件的提供者（一个HTTP或者FTP服务器）
* 一个需要这个远程服务的客户端程序



RMI的主要优点如下：

**面向对象**

**可移动属性：**RMI可将属性(类实现程序)从客户机移动到服务器，或者从服务器移到客户机。这样就能具备最大的灵活性，因为政策改变时只需要您编写一个新的Java类，并将其在服务器主机上安装一次即可。

设计方式：对象传递功能使您可以在分布式计算中充分利用面向对象技术的强大功能，如二层和三层结构系统。如果您能够传递属性，那么您就可以在您的解决方案中使用面向对象的设计方式。所有面向对象的设计方式无不依靠不同的属性来发挥功能，如果不能传递完整的对象--包括实现和类型--就会失去设计方式上所提供的优点。

**安　　全：**RMI使用Java内置的安全机制保证下载执行程序时用户系统的安全。RMI使用专门为保护系统免遭恶意小应用程序侵害而设计的安全管理程序，可保护您的系统和网络免遭潜在的恶意下载程序的破坏。在情况严重时，服务器可拒绝下载任何执行程序。

**便于编写和使用：**RMI使得Java远程服务程序和访问这些服务程序的Java客户程序的编写工作变得轻松、简单。远程接口实际上就是Java接口。服务程序大约用三行指令宣布本身是服务程序，其它方面则与任何其它Java对象类似。这种简单方法便于快速编写完整的分布式对象系统的服务程序，并快速地制做软件的原型和早期版本，以便于进行测试和评估。因为RMI程序编写简单，所以维护也简单。

**可连接现有/原有的系统：**RMI可通过Java的本机方法接口JNI与现有系统进行进行交互。利用RMI和JNI，您就能用Java语言编写客户端程序，还能使用现有的服务器端程序。在使用RMI/JNI与现有服务器连接时，您可以有选择地用Java重新编写服务程序的任何部分，并使新的程序充分发挥Java的功能。类似地，RMI可利用JDBC、在不修改使用数据库的现有非Java源代码的前提下与现有关系数据库进行交互。

**编写一次，到处运行：**RMI是Java“编写一次，到处运行 ”方法的一部分。任何基于RMI的系统均可100%地移植到任何Java虚拟机上，RMI/JDBC系统也不例外。如果使用RMI/JNI与现有系统进行交互工作,则采用JNI编写的代码可与任何Java虚拟机进行编译、运行。

**分布式垃圾收集：**RMI采用其分布式垃圾收集功能收集不再被网络中任何客户程序所引用的远程服务对象。与Java 虚拟机内部的垃圾收集类似，分布式垃圾收集功能允许用户根据自己的需要定义服务器对象，并且明确这些对象在不再被客户机引用时会被删除。

1. 什么是序列化？请列举所知的序列化实现方式？

将对象状态转为可存储或传输形式的过程。

Java在实现序列化时，需要实现Serializable接口，默认使用二进制文件存储对象信息。

其他实现：XML，JSON，SOAP。序列化结果包含语义，便于人阅读。SOAP用于Web Service，JSON多见于Restful Web Service。

设计模式

1. 装饰模式，请用UML描述



装饰模式与适配器模式区别：

1）关于新职责：适配器也可以在转换时增加新的职责，但主要目的不在此。装饰者模式主要是给被装饰者增加新职责的。

2）关于原接口：适配器模式是用新接口来调用原接口，原接口对新系统是不可见或者说不可用的。装饰者模式原封不动的使用原接口，系统对装饰的对象也通过原接口来完成使用。（增加新接口的装饰者模式可以认为是其变种--“半透明”装饰者）

3）关于其包裹的对象：适配器是知道被适配者的详细情况的（就是那个类或那个接口）。装饰者只知道其接口是什么，至于其具体类型（是基类还是其他派生类）只有在运行期间才知道。

Java中例子：IO Stream

缺点：

1）这种比继承更加灵活机动的特性，也同时意味着更加多的复杂性。

2） 装饰模式会导致设计中出现许多小类，如果过度使用，会使程序变得很复杂。

3） 装饰模式是针对抽象组件（Component）类型编程。但是，如果你要针对具体组件编程时，就应该重新思考你的应用架构，以及装饰者是否合适。当然也可以改变Component接口，增加新的公开的行为，实现“半透明”的装饰者模式。在实际项目中要做出最佳选择。

1. 桥接模式和适配器模式区别
2. 策略模式和状态模式
3. 困难

Java高级应用

1. 什么是垃圾回收？Java发生垃圾回收的内存区域有哪些？

回收不再使用的对象空间。垃圾回收会发生在：堆、方法去和直接内存。

垃圾回收分为：Full GC，Major GC和Minor GC。

Full GC回收所有区域，包括堆内存和直接内存

Major GC：回收老年代，通常伴随一次Minor GC

Minor GC：回收新生代，速度快，频繁

1. Java内存模型

Java内存分为方法区、栈（Java栈和本地方法栈）、堆和程序计数器



Java栈：存储栈帧，包括局部变量表，操作数栈，动态链接，方法出口等

本地方法栈：JNI执行线程的内存区域

堆：共享数据区，存储所有类实例和数组实例；GC管理主要区域

方法区：共享数据区，逻辑上是堆的一部分，存储类、常量、静态变量、JIT编译代码等。



新生代、老年代和持久代

新生代：一个Eden区，两个Survior区，存储新建对象

老年代：存储需要占用足够大空间的对象，或者age达到指定阈值。

持久代：存放class信息、常量和静态属性。

1. Java多线程如何实现同步/互斥？
2. Synchnorize代码块或方法区
3. 伪原子操作：java.util.concurrent.atomic下，如AtomicInteger、AtomicBoolean等
4. java.util.concurrent.Semaphore，信号量，acquire获得一个许可，若无则阻塞；release释放许可
5. java.util.concurrent下其他工具
6. Volatile的作用？

用volatile修饰的变量，线程在每次使用变量的时候，都会读取变量修改后的最的值。volatile保证从主内存加载到线程工作内存的值是最新的，无法保证互斥访问。

1. 什么是JPA？JPA实现有哪些？

JPA全称Java Persistence API.JPA通过JDK 5.0注解或XML描述对象－关系表的映射关系，并将运行期的实体对象持久化到数据库中。

JPA实现：Hibernate，Spring Data，OPEN JPA。

Refer：http://www.cnblogs.com/holbrook/archive/2012/12/30/2839842.html

1. 什么是两阶段提交？

分布式事务。

第一阶段：

　　首先，协调者在自身节点的日志中写入一条的日志记录，然后所有参与者发送消息prepare T，询问这些参与者（包括自身），是否能够提交这个事务；

　　参与者在接受到这个prepare T 消息以后，会根据自身的情况，进行事务的预处理，如果参与者能够提交该事务，则会将日志写入磁盘，并返回给协调者一个ready T信息，同时自身进入可提交状态；如果不能提交该事务，则记录日志，并返回一个not commit T信息给协调者，同时撤销在自身上所做的数据库改；

第二阶段：

　　协调者会收集所有参与者的意见。（1）如果收到参与者发来的not commit T信息，则标识着该事务不能提交，协调者会将Abort T 记录到日志中，并向所有参与者发送一个Abort T 信息，让所有参与者撤销在自身上所有的预操作；（2）如果协调者收到所有参与者发来prepare T信息，那么协调者会将Commit T日志写入磁盘，并向所有参与者发送一个Commit T信息，提交该事务。（3）若协调者迟迟未收到某个参与者发来的信息，则认为该参与者发送了一个VOTE\_ABORT信息，从而取消该事务的执行。

　　参与者接收到协调者发来的Abort T信息以后，参与者会终止提交，并将Abort T 记录到日志中；如果参与者收到的是Commit T信息，则会将事务进行提交，并写入记录。

1. 什么是Web Service和Restful Web Service？

Web service是一个平台独立的，低耦合的，自包含的、基于可编程的web的应用程序，可使用开放的XML（标准通用标记语言下的一个子集）标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。包含的标准：SOAP，WSDL，UUDI。

Restful Web Service：Representational state transfer Web Service，是一种架构风格。特点：无状态，公开目录结构的URL，HTTP方式访问。

1. 如何修改java 字节码？

Javaassist，ASM等工具。

1. 请介绍你所熟悉的Java框架？使用Java的优势？
2. Java 性能调优的步骤？核心是为了减少什么？
3. 打印垃圾回收日志，使用下列参数

-XX:+PrintGCDetails -Xloggc:<filename>

1. 根据日志调整参数

通常调整－Xmx和－Xms，调整堆的初始值和最大值。根据日志，还可能需要调整老年代、方法区的大小等。

1. Java中存在内存泄漏么？

内存泄漏：内存空间不再被程序使用，却一直占据内存，无法被释放，即Java中的垃圾。Java中有可能存在内存泄漏，例如：直接内存区域，只有在Full GC时，才会触发回收，若一直无Full GC，则直接内存区域中的数据就会发生内存泄漏。

1. Hibernate，Spring Data，iBatis都有OR－Mapping，如何从中做选择？

iBatis仅做OR Mapping，需要手写sql，便于优化SQL。

Spring Data使用方便，仅需声明接口，也可指定SQL语句，便于开发。

Hibernate功能强大，配置项较为繁琐，适合大型项目。

1. Union和Union all有什么区别？

Union会删除重复记录返回结果

Union all 简单返回合并结果。

效率：union all更快。

1. 能否自己实现class，起名java.lang.String?

可以。但是需要自行实现类加载器，否则由于Java的安全机制，系统的类加载器永远只加载jre/jdk下的java.lang.String.