00java题总结

通用

- 职业规划，带团队
- 设计模式：简单工厂和抽象工厂区别
- 面向对象的三大特性：封装、继承、多态,从一定角度来看，封装和继承几乎都是为多态而准备的。这是我们最后一个概念，也是最重要的知识点。多态的定义：指允许不同类的对象对同一消息做出响应。即同一消息可以根据发送对象的不同而采用多种不同的行为方式。（发送消息就是函数调用）实现多态的技术称为：动态绑定（dynamic binding），是指在执行期间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。多态的作用：消除类型之间的耦合关系。
- Java Web服务器：Tomcat、GlassFish、WebLogic、JBoss、WebSphere、Jetty、JRun

web协议

- jsonp,单点登录,cookie、session存储原理
- HTTP请求4个部分组成：请求行（request line）、请求头部（header）、空行和请求数据四个部分组成。
- HTTP响应4个部分组成：状态行、消息报头、空行和响应正文。
- get和post区别：get是把参数URL中(浏览器地址可以看到)，post是把提交的内容放到请求body里，浏览器中不能看到，两者的提交长度http协议没有限制，只是浏览器和web服务器的限制
- 响应的类别共分4种类别:

  1xx：指示信息--表示请求已接收，继续处理
  2xx：成功--表示请求已被成功接收、理解、接受
  3xx：重定向--要完成请求必须进行更进一步的操作
  4xx：客户端错误--请求有语法错误或请求无法实现
  5xx：服务器端错误--服务器未能实现合法的请求

- HTTP最常用的状态码有：

  200 (OK): 找到了该资源，并且一切正常。
  304 (NOT MODIFIED): 该资源在上次请求之后没有任何修改。这通常用于浏览器的缓存机制。
  401 (UNAUTHORIZED): 客户端无权访问该资源。这通常会使得浏览器要求用户输入用户名和密码，以登录到服务器。
  403 (FORBIDDEN): 客户端未能获得授权。这通常是在401之后输入了不正确的用户名或密码。
  404 (NOT FOUND): 在指定的位置不存在所申请的资源。

java基础

- int 和 Integer 区别：引用类型和原始类型（或内置类型），引用类型缺省值为null。
- short s1 = 1; s1 = s1 + 1;有什么错?

  short s1 = 1; s1 = s1 + 1;（s1+1运算结果是int型，需要强制转换类型,运算时会自动提升表达式的类型，所以结果是int型，编译器将报告需要强制转换类型的错误）
  short s1 = 1; s1 += 1;（可以正确编译,+= 是java语言规定的运算符，java编译器会对它进行特殊处理）

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

  Math.round(11.5)==12
  Math.round(-11.5)==-11
  round方法返回与参数最接近的长整数，参数加1/2后求其floor（向下取整，向下舍入，即取不大于x的最大整数）。

- 关键字：final类、final变量
- java注入方式：set和构造方法；spring注入方式：Set注入、构造方法注入、静态工厂的方法注入、实例工厂的方法注入。
- 阻塞队列
- jsp
- tomcat并发参数 和 jvm调优
- 去除List集合中的重复值（4种）
- String str = new String（“aa”）创建了几个对象：

  一个是“aa”二个new String（）
  这个跟常量池没有关系，只要是new，都是重新分配堆空间，如果不区分栈和堆，这里创建了1个String Object。如果是从jvm角度来说的话，它是创建了两个对象，String s是在栈里创建了一个变量，new String("xyz")是在堆里创建了一个对象并被s

引用到。

如果是String s = "xyz"，那就要看常量池里有没有"xyz"，如果有直接引用，如果没有则创建再引用

这里"xyz"本身就是pool中的一个对象，而在运行时执行new String()时，将pool中的对象复制一份放到heap中，并且把heap中的这个对象的引用交给s持有。ok，这条语句就创建了2个String对象。

这时用==判断就可知，虽然两个对象的"内容"相同(equals()判断)，但两个引用变量所持有的引用不同

- 补充一个面试题：String str = "aaa" + new String("bbb")创建了几个String对象？

  "aa"一个对象　new Sring()一个对象 "bbb"一个对象 "aa" + new String("bbb");一个对象

- ==和equals区别：==比较引用的同一个对象，equals比较对象的“值”是否相等
- String、StringBuilder和StringBuffer的区别：

速度比较：StringBuilder > StringBuffer > String，String是字符串常量， StringBuilder非线程安全，StringBuffer线程安全。

- final, finally, finalize的区别

  - final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。
  - finally是异常处理语句结构的一部分，表示总是执行。
  - finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

- abstract class和interface有什么区别?

  相同点

  A. 两者都是抽象类，都不能实例化。

  B. interface实现类及abstrct class的子类都必须要实现已经声明的抽象方法。

  不同点

  A. interface需要实现，要用implements，而abstract class需要继承，要用extends。

  B. 一个类可以实现多个interface，但一个类只能继承一个abstract class。

  C. interface强调特定功能的实现，而abstract class强调所属关系。

  D. 尽管interface实现类及abstrctclass的子类都必须要实现相应的抽象方法，但实现的形式不同。interface中的每一个方法都是抽象方法，都只是声明的(declaration, 没有方法体)，实现类必须要实现。而abstract class的子类可以有选择地实现。

  这个选择有两点含义：

  一是Abastract class中并非所有的方法都是抽象的，只有那些冠有abstract的方法才是抽象的，子类必须实现。那些没有abstract的方法，在Abstrct class中必须定义方法体。

  二是abstract class的子类在继承它时，对非抽象方法既可以直接继承，也可以覆盖；而对抽象方法，可以选择实现，也可以通过再次声明其方法为抽象的方式，无需实现，留给其子类来实现，但此类必须也声明为抽象类。既是抽象类，当然也不能实例化。

  E. abstract class是interface与Class的中介。

  interface是完全抽象的，只能声明方法，而且只能声明pulic的方法，不能声明private及protected的方法，不能定义方法体，也不能声明实例变量。然而，interface却可以声明常量变量，并且在JDK中不难找出这种例子。但将常量变量放在interface中违背了其作为接口的作用而存在的宗旨，也混淆了interface与类的不同价值。如果的确需要，可以将其放在相应的abstract class或Class中。

  abstract class在interface及Class中起到了承上启下的作用。一方面，abstractclass是抽象的，可以声明抽象方法，以规范子类必须实现的功能；另一方面，它又可以定义缺省的方法体，供子类直接使用或覆盖。另外，它还可以定义自己的实例变量，以供子类通过继承来使用。

  Java8中Oracle已经开始尝试向接口中引入默认方法和静态方法

- Java语言的特点：跨平台性（字节码）、面向对象、安全性（语言级安全性、编译时安全性、运 行时安全性、可执行代码安全性）、多线程、简单易用（可以用记事本、文本编辑器）
- J2EE是Java的子集，是“企业版”，比如Jsp、XML、RMI、EJB等都属于J2EE的范畴。(整个 web开发都属j2ee) Java = J2SE + J2ME + J2EE.
- Java中多态的实现方式：接口实现，继承父类进行方法重写，同一个类中进行方法重载
- Overload和Override的区别。

  Overloaded的方法是否可以改变返回值的类型?

  方法的重写Overriding和重载Overloading是Java多态性的不同表现。

  重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被**重写** (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被"屏蔽"了。

  如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的**重载(Overloading)**。Overloaded的方法是可以改变返回值的类型。

- 抽象类与接口的区别？

简单来说，接口是公开的，里面不能有私有的方法或变量，是用于让别人使用的，而抽象类是可以有私有方法或私有变量的，

另外，实现接口的一定要实现接口里定义的所有方法，而实现抽象类可以有选择地重写需要用到的方法，一般的应用里，最顶级的是接口，然后是抽象类实现接口，最后才到具体类实现。还有，接口可以实现多重继承，而一个类只能继承一个超类，但可以通过继承多个接口实现多重继承，接口还有标识（里面没有任何方法，如Remote接口）和数据共享（里面的变量全是常量）的作用

- 接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类?

  接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数（可以继承，但是和实体类的继承一样，也要求父类可继承，并且拥有子类可访问到的构造器)。抽象类可以继承实体类吗.net

- Static Nested Class 和 Inner Class的不同

  如果你不需要内部类对象与其外围类对象之间有联系，那你可以将内部类声明为static。这通常称为嵌套类（nested class）。Static Nested Class是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。想要理解static应用于内部类时的含义，你就必须记住，普通的内部类对象隐含地保存了一个引用，指向创建它的外围类对象。然而，当内部类是static的时，就不是这样了。嵌套类意味着：
  1. 嵌套类的对象，并不需要其外围类的对象。
  2. 不能从嵌套类的对象中访问非静态的外围类对象。

- 什么时候用assert

  assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制，如C，C++和Eiffel等，但是支持的形式不尽相同，有的是通过语言本身、有的是通过库函数等。另外，从理论上来说，通过assertion方式可以证明程序的正确性，但是这是一项相当复杂的工作，目前还没有太多的实践意义。
  在实现中，assertion就是在程序中的一条语句，它对一个boolean表达式进行检查，一个正确程序必须保证这个boolean表达式的值为true；如果该值为false，说明程序已经处于不正确的状态下，系统将给出警告并且退出。一般来说，assertion用于保证程序最基本、关键的正确性。assertion检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion检查通常是关闭的。

- IO流的知识点？
- 有哪些常用的函数？
- 你们是怎么使用随机函数的？

JAVA集合

- HashTable和HashMap区别，

  HashMap可以接受null键值和值，而HashTable则不能；HashMap是非synchronized;HashMap很快；以及HashMap储存的是键值对。ConcurrentHashMap当然可以代替HashTable，但是HashTable提供更强的线程安全性。

  hashmap 线程不安全 允许有null的键和值 效率高一点、 方法不是Synchronize的要提供外同步 有containsvalue和containsKey方法

  hashtable 线程安全 不允许有null的键和值 效率稍低、 方法是是Synchronize的 有contains方法方法

- List, Set, Map是否继承自Collection接口?List,Set,Map继承.note

  List，Set是，Map不是

  Collection
  ├List
  │ ├LinkedList
  │ ├ArrayList
  │ └Vector
  │ └Stack
  └Set

  Map
  ├Hashtable
  ├HashMap
  └WeakHashMap

- Collection 和 Collections的区别

  Collection是集合类的上级接口，继承与他的接口主要有Set 和List.

  Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

- ArrayList,Vector, LinkedList存储性能和特性

  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。

java原理

- jvm载入类过程：加载，连接(验证、准备、解释)，初始化，使用，卸载
- Servlet定义：Servlet（Server Applet）全称Java Servlet，是用Java编写的服务器端程序
- Jsp与servlet的区别

  jsp经编译后就变成了Servlet，JSP的本质就是Servlet，JVM只能识别java的类，不能识别JSP的代码，Web容器将JSP的代码编译成JVM能够识别的java类

- Servlet的生命周期，并说出Servlet和CGI的区别：

  Servlet 生命周期：Servlet加载--->初始化 调用init()方法--->响应客户请求　调用service()方法--->终止 调用destroy()方法。
  Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法 自动派遣运行与请求对应的doXXX方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其destroy方法。
  与cgi的区别在于servlet处于服务器进程中，它通过多线程方式运行其service方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而CGI对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于servlet。

- forward 和redirect的区别

  forward方式：request.getRequestDispatcher("/somePage.jsp").forward(request, response);
  redirect方式：response.sendRedirect("/somePage.jsp");
  　forward是服务器内部重定向，程序收到请求后重新定向到另一个程序，客户机并不知道；redirect则是服务器收到请求后发送一个状态头给客户，客户将再请求一次，这里多了两次网络通信的来往。当然forward也有缺点，就是forward的页面的路径如果是相对路径就会有些问题了。
  　forward会将request state , bean 等等信息带往下一个jsp ,redirect是送到client端后再一次request , 所以资料不被保留.
  　　使用forward你就可以用getAttribute()来取的前一个jsp所放入的bean,forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。
  　　redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以session,request参数都可以获取。

- heap和stack有什么区别Stack和Heap的区别.net

  heap 堆(dui) stack 栈(zhan) h比s先 d比z先 所以 堆栈的记忆按照他们的首字母的顺序即可。堆栈 heap stack
  1.heap是堆，stack是栈。
  2.stack的空间由操作系统自动分配和释放，heap的空间是手动申请和释放的，heap常用new关键字来分配。
  3.stack空间有限，heap的空间是很大的自由区。
  在Java中，
  若只是声明一个对象，则先在栈内存中为其分配地址空间，
  若再new一下，实例化它，则在堆内存中为其分配地址。
  4.举例：
  数据类型 变量名；这样定义的东西在栈区。
  如：Object a =null; 只在栈内存中分配空间
  new 数据类型();或者malloc(长度);　这样定义的东西就在堆区
  如：Object b =new Object(); 则在堆内存中分配空间

- ASM框架 Bytecode
- java web有那些常用的对象：

垃圾回收

- GC是什么? 为什么要有GC?

  Java GC（Garbage Collection，垃圾收集，垃圾回收）机制，是Java与C++/C的主要区别之一，在使用JAVA的时候，一般不需要专门编写内存回收和垃圾清理代 码。这是因为在Java虚拟机中，存在自动内存管理和垃圾清扫机制。

- 垃圾回收原理?垃圾回收（GC）原理.note

  基于对对象生命周期分析后得出的垃圾回收算法。把对象分为年青代、年老代、持久代，对不同生命周期的对象使用不同的算法（上述方式中的一个）进行回收。现在的垃圾回收器（从J2SE1.2开始）都是使用此算法的。

1. Young（年轻代）
年轻代分三个区。一个Eden区，两个 Survivor区。大部分对象在Eden区中生成。当Eden区满时，还存活的对象将被复制到Survivor区（两个中的一个），当这个 Survivor区满时，此区的存活对象将被复制到另外一个Survivor区，当这个Survivor去也满了的时候，从第一个Survivor区复制过来的并且此时还存活的对象，将被复制“年老区(Tenured)”。需要注意，Survivor的两个区是对称的，没先后关系，所以同一个区中可能同时存在从Eden复制过来对象，和从前一个Survivor复制过来的对象，而复制到年老区的只有从第一个Survivor去过来的对象。而且，Survivor区总有一个是空的。
2. Tenured（年老代）
年老代存放从年轻代存活的对象。一般来说年老代存放的都是生命期较长的对象。
3. Perm（持久代）
用于存放静态文件，如今Java类、方法等。持久代对垃圾回收没有显著影响，但是有些应用可能动态生成或者调用一些class，例如Hibernate等，在这种时候需要设置一个比较大的持久代空间来存放这些运行过程中新增的类。持久代大小通过-XX:MaxPermSize=<N>进行设置。

- 软引用，弱引用，虚引用

多线程

- 什么是线程安全：线程安全就是多线程访问时，采用了加锁机制，当一个线程访问该类的某个数据时，进行保护，其它线程不能进行访问直到该线程读取完。不会出现数据不一致或污染。线程不安全就是不提供数据访问保护，有可能出现多个线程先后更改数据造成所得到的数据是脏数据。
- 怎么实现线程安全、线程有那些辅助类、线程池的理解
- Java多线程实现方式主要有4种：继承Thread类、实现Runnable接口、实现Callable接口通过FutureTask包装器来创建Thread线程、使用ExecutorService、Callable、Future实现有返回结果的多线程。
- 多线程同步有几种实现方式：synchronized同步方法或代码块，特殊域变量(volatile？)，lock重入锁（java.util.concurrent包下，ReentrantLock类需要在finally代码释放锁，是可重入、互斥、实现了Lock接口的锁,wait与notify）,ThreadLocal管理变量，CAS(Compare-And-Swap since Java6)
- Lock接口与synchronized关键字的区别：Lock接口具有锁的**可操作性**，可**中断获取**以及**超时获取锁**，还有两个强大的实现类**重入锁**和**读写锁**：

  Lock接口可以尝试非阻塞地获取锁 当前线程尝试获取锁。如果这一时刻锁没有被其他线程获取到，则成功获取并持有锁。
  Lock接口能被中断地获取锁 与synchronized不同，获取到锁的线程能够响应中断，当获取到的锁的线程被中断时，中断异常将会被抛出，同时锁会被释放。
  Lock接口在指定的截止时间之前获取锁，如果截止时间到了依旧无法获取锁，则返回。

- sleep() 和 wait() 有什么区别?

sleep是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

- 线程池的理解:

异常处理

- 谈谈异常处理机制的原理？

Throwable： 有两个重要的子类：Exception（异常）和 Error（错误），二者都是 Java 异常处理的重要子类，各自都包含大量子类。

运行时异常与**一般异常**（runtime exception和checked exception）：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的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

- error和exception有什么区别?

  error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。exception 表示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。

  - 最常见到的runtime exception?

  算术异常类：ArithmeticExecption
  空指针异常类：NullPointerException
  类型强制转换异常：ClassCastException
  数组负下标异常：NegativeArrayException
  数组下标越界异常：ArrayIndexOutOfBoundsException
  向数组中存放与声明类型不兼容对象异常：ArrayStoreException
  安全异常：SecturityException
  文件已结束异常：EOFException
  文件未找到异常：FileNotFoundException
  字符串转换为数字异常：NumberFormatException
  操作数据库异常：SQLException
  输入输出异常：IOException
  方法未找到异常：NoSuchMethodException
  抽象方法错误。当应用试图调用抽象方法时抛出。Java.lang.AbstractMethodError
  传递非法参数异常：IllegalArgumentException
  创建一个大小为负数的数组错误异常 NegativeArraySizeException

不支持的操作异常：UnsupportedOperationException

java web/jsp

- javabean是什么?JAVABEAN是什么.net

  Javabean 就是一个类，这个类就定义一系列 get<Name> 和 set<Name> 方法。就是为了和 jsp 页面传数据化简交互过程而产生的。

- jsp中使用javabean的类的两种方法：

  法一： jsp 中使用 <jsp:useBean> 标记符访问 javabean ：
  <jsp:useBean id="test" class="test.TestBean" />
  法二： jsp 中嵌入 java 代码方式访问 javabean ：
  首行导入 javabean ：
  <%@ page import="com.javaBean.TestBean" %>
  下边就可以像在 java 语言中那样用了：
  <% TestBean testBean=new TestBean(); %>

- JSP中动态INCLUDE与静态INCLUDE的区别？

  动态include在使用的时候，会先解析所要包含的页面，解析后在和主页面放到一起显示；
  静态include在使用的时候，不会解析所要包含的页面，也就是说，不管你的included.htm中有什么，我的任务就是把你包含并显示，
  jsp:include是先编译一下included.jsp文件，然后再包含 先编译，后包含
  @ include是先把文件包含就来，然后统一编译 先包含，后编译

Spring

- Spring MVC流程、好处
  - 过滤器和拦截器的区别：

  ①拦截器是基于Java的反射机制的? 而过滤器是基于函数回调?
  ②拦截器不依赖与servlet容器，过滤器依赖与servlet容器。
  ③拦截器只能对action请求起作用，而过滤器则可以对几乎所有的请求起作用。
  ④拦截器可以访问action上下文、值栈里的对象，而过滤器不能访问。
  ⑤在action的生命周期中，拦截器可以多次被调用，而过滤器只能在容器初始化时被调用一次。
  ⑥拦截器可以获取IOC容器中的各个bean，而过滤器就不行，这点很重要，在拦截器里注入一个service，可以调用业务逻辑。

- Spring原理和作用：Spring工作原理及其作用 - 【设计改变世界】 果然如此的专栏 - 博客频道 - CSDN.NET

jdbc数据库

- mybatis hibernate 比较
- 数据连接池的工作机制是什么?

  J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其表记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。

- **MySQL**

  1. MySQL存储引擎：MyISAM与InnoDB，InnoDB支持事务适合频繁修改以及涉及到安全性较高的应用，MyISAM不支持事务适合查询以及插入为主的应用。
  2. 是否可以分区？
  3. MySql优化：优化sql和索引；加缓存，memcached,redis；主从复制或主主复制，读写分离；分区表；垂直拆分；水平切分；
  4. mysql支持的复制类型：基于语句的复制,基于行的复制,混合类型的复制

- **SqlServer**

  1. 事务的特性：ACID，原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）
  2. 事务隔离性：SQL Server利用加锁和阻塞来保证事务之间不同等级的隔离性
     a. 脏读：一个事务读取了另一个事务未提交的数据,而这个数据是有可能回滚；
     b. 不可重复读(Unrepeatable Read)：一个事务范围内两个相同的查询却返回了不同数据，原因是查询时其他事务修改的提交而引起的；
     c. 幻读(phantom read)：执行批量更新后发现有未更新的数据，原因是更新过程中，另一个事务插入了新数据。
  3. 数据库系统提供了四种事务隔离级别：

a. Serializable（串行化）：一个事务在执行过程中完全看不到其他事务对数据库所做的更新（事务执行的时候不允许别的事务并发执行。事务串行化执行，事务只能一个接着一个地执行，而不能并发执行。）。
b. Repeatable Read（可重复读）：一个事务在执行过程中可以看到其他事务已经提交的新插入的记录，但是不能看到其他其他事务对已有记录的更新。
c. Read Commited（读已提交数据）：一个事务在执行过程中可以看到其他事务已经提交的新插入的记录，而且能看到其他事务已经提交的对已有记录的更新。
d. Read Uncommitted（读未提交数据）：一个事务在执行过程中可以看到其他事务没有提交的新插入的记录，而且能看到其他事务没有提交的对已有记录的更新。

大数据

- 如何解决双十一大访问量：缓存+队列异步化；不是传统意义上的Redis/Memcache等系统Cache组件， 而是从用户端到最后数据层所有环节的Cache，为用户行为所有环节加上合理的Cache。电商网站应对双十一大流量的常见策略 - 码厦（微信号） - 博客频道 - CSDN.NET
- 内存索引
- Redis
  - Redis的5种数据类型：String、Hash、List、Set、Sorted set。

    String：set,get,decr,incr,mget
    Hash：hget,hset,hgetall 等
    List：lpush,rpush,lpop,rpop,lrange等vcxsf
    Set：sadd,spop,smembers,sunion 等
    Sorted set：zadd,zrange,zrem,zcard等
    Pub/Sub：发布（Publish）与订阅（Subscribe）
    Transactions：key进行Watch再执行Transactions，Watched的值进行了修改，Transactions会发现并拒绝执

- memcached
- mongodb

- 搜索
  - 倒排索引：倒排索引是实现“单词-文档矩阵”的一种具体存储形式，通过倒排索引，可以根据单词快速获取包含这个单词的文档列表。倒排索引主要由两个部分组成：“单词词典”和“倒排文件”。正排索引是从文档到关键字的映射（已知文档求关键字），倒排索引是从关键字到文档的映射（已知关键字求文档）。
  - 索引算法：B+、B- Tree(mysql,oracle,mongodb)，Hash表+桶(redis)，Booleam Filter（HBase）
  - 分词算法：基于字典、词库匹配的分词方法；基于词频度统计的分词方法和基于知识理解的分词方法。中文分词算法总结.note
  - 中文分词：IK、盘古词库，
  - Lucene