

第1章 GitHub

Guava 24.1 已发布，Guava 是 Google 的一个开源项目，包含许多 Google 核心的 Java 常用库，如：集合 [collections]、缓存 [caching]、原生类型支持 [primitives support]、并发库 [concurrency libraries]、通用注解 [common annotations]、字符串处理 [string processing]、I/O 等等。（下载地址：<https://github.com/google/guava/archive/v24.1.zip>）

参考文献

- [1] GitHub 入门之一：使用 github 下载项目
http://blog.csdn.net/icyfox_bupt/article/details/22966629
- [2] 怎样使用 GitHub? <https://www.zhihu.com/question/20070065>
- [3] Github 简明教程: <http://www.runoob.com/w3cnote/git-guide.html>
- [4] 如何高效利用 GitHub: <http://www.yangzhiping.com/tech/github.html>
- [5]

第2章 并发编程

Tips:

并发具有可论证的正确性，实际上具有不可确定性。
Java 是一门多线程语言。

1 并发的多面性

并发主要解决两大类问题：

- 速度
- 设计可管理性，通常是为了某些软件的并发需求。

提升速度的两个方面：

- 多处理器上的并发；
- 单个处理器上的并发（基于的前提是任务可能会发生阻塞，否则并不能提升性能），典型应用是界面交互的事件驱动编程；

Tips:

并发最直接的方式是基于操作系统的多进程机制，而Java是基于线程的实现。不使用进程的原因有两点：一是需要依赖操作系统，二进程间是隔离的。

Java 并发的实现方式，即线程机制，是在由执行程序表示的单一进程中创建任务。

对于设计，通常需要改进代码的设计，需要使用多线程的协作。

Tips:

Java 的线程机制是抢占式的，线程的调度机制会周期性的中断线程，将上下文切换到另外一个线程，从而为每个线程都提供时间片，使每个线程都有时间去执行它的任务。

虽然是抢占式，但是程序猿同样可以主动的放弃控制，这样效率会更高。

2 基本的线程机制

Tips:

线程是进程中的一个单一的顺序控制流。线程驱动的对象是任务。

2.1 任务的定义

定义任务的方法：实现 Runnable 接口，并编写 run 方法。

Tips:

这仅仅是一个任务，需要加到一个线程里才能够得到执行。

基本概念

操作系统中 heap 和 stack 的区别

什么是基于注解的切面实现

什么是 对象/关系 映射集成模块

什么是 Java 的反射机制

什么是 ACID

BS 与 CS 的联系与区别

Cookie 和 Session 的区别
fail-fast 与 fail-safe 机制有什么区别
get 和 post 请求的区别
Interface 与 abstract 类的区别
IOC 的优点是什么
IO 和 NIO 的区别, NIO 优点
Java 8 / Java 7 为我们提供了什么新功能
什么是竞态条件? 举个例子说明。
JRE、JDK、JVM 及 JIT 之间有什么不同
MVC 的各个部分都有那些技术来实现?如何实现?
RPC 通信和 RMI 区别
什么是 Web Service (Web 服务)
JSWDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI,WSDL 解释。
WEB 容器主要有哪些功能? 并请列出一些常见的 WEB 容器名字。
一个 ".java" 源文件中是否可以包含多个类 (不是内部类)? 有什么限制
简单说说你了解的类加载器。是否实现过类加载器
解释一下什么叫 AOP (面向切面编程)
请简述 Servlet 的生命周期及其相关的方法
请简述一下 Ajax 的原理及实现步骤
简单描述 Struts 的主要功能
什么是 N 层架构
什么是 CORBA? 用途是什么
什么是 Java 虚拟机? 为什么 Java 被称作是 "平台无关的编程语言"
什么是正则表达式? 用途是什么? 哪个包使用正则表达式来实现模式匹配
什么是懒加载 (Lazy Loading)
什么是尾递归, 为什么需要尾递归
什么是控制反转 (Inversion of Control) 与依赖注入 (Dependency Injection)

String

StringBuffer

ByteBuffer 与 StringBuffer 有什么区别

Java 的容器接口

问:

ArrayList、Vector、LinkedList 的存储性能和特性?

答:

ArrayList、Vector 是通过数组存储对象的, LinkedList 是基于链接的。

问:

HashMap 的工作原理是什么? 内部的数据结构是什么?

答：

拉链式存储。

问：

HashMap 的 table 的容量如何确定？loadFactor 是什么？该容量如何变化？这种变化会带来什么问题？

答：

初始值容量是 16，负载因子 0.75，容量上限为 12。容量增长是指数增长的，容量的最大值为 $1 < 30$ 。

首先有一个每个元素都是链表（可能表述不准确）的数组，当添加一个元素（key-value）时，就首先计算元素 key 的 hash 值，以此确定插入数组中的位置，但是可能存在同一 hash 值的元素已经被放在数组同一位置了，这时就添加到同一 hash 值的元素的后面，他们在数组的同一位置，但是形成了链表，同一各链表上的 Hash 值是相同的，所以说数组存放的是链表。在 JDK1.8 后，当链表长度太长时，链表就转换为红黑树，这样大大提高了查找的效率。

当链表数组的容量超过初始容量的 0.75 时，再散列将链表数组扩大 2 倍，把原链表数组的搬移到新的数组中。为什么需要使用加载因子，为什么需要扩容呢？因为如果填充比很大，说明利用的空间很多，如果一直不进行扩容的话，链表就会越来越长，这样查找的效率很低，因为链表的长度很大（当然最新版本使用了红黑树后会改进很多），扩容之后，将原来链表数组的每一个链表分成奇偶两个子链表分别挂在新链表数组的散列位置，这样就减少了每个链表的长度，增加查找效率。

HashMap 本来是以空间换时间，所以填充比没必要太大。但是填充比太小又会导致空间浪费。如果关注内存，填充比可以稍大，如果主要关注查找性能，填充比可以稍小。

问：

HashMap 和 Hashtable、ConcurrentHashMap 的区别？

答：

除了非同步和允许使用 null 之外，HashMap 类与 Hashtable 大致相同。

问：

HashMap 的遍历方式及效率？

答：

根据 Hashcode 找位置，再比较键值。效率和桶的高度有关。

HashSet 和 TreeSet 有什么区别

HashSet 内部是如何工作的

WeakHashMap 是怎么工作的？

Set

Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 == 还是 equals()？它们有何区别？

TreeMap：TreeMap 是采用什么树实现的？TreeMap、HashMap、LindedHashMap 的区别。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？

TreeSet：一个已经构建好的 TreeSet，怎么完成倒排序。

EnumSet 是什么

Hash 算法

HashCode 的作用

简述一致性 Hash 算法

有没有可能 两个不相等的对象有相同的 hashCode？当两个对象 hashCode 相同怎么办？如何获取值对象

为什么在重写 equals 方法的时候需要重写 hashCode 方法？equals 与 hashCode 的异同点在哪里

a.hashCode() 有什么用？与 a.equals(b) 有什么关系

hashCode() 和 equals() 方法的重要性体现在什么地方

Object：Object 有哪些公用方法？Object 类 hashCode,equals 设计原则？sun 为什么这么设计？Object 类的概述

如何在父类中为子类自动完成所有的 hashCode 和 equals 实现？这么做有何优劣。

可以在 hashCode() 中使用随机数字吗？

LinkedHashMap

LinkedHashMap 和 PriorityQueue 的区别是什么

List

List, Set, Map 三个接口，存取元素时各有什么特点

List, Set, Map 是否继承自 Collection 接口

遍历一个 List 有哪些不同的方式

LinkedList

LinkedList 是单向链表还是双向链表

LinkedList 与 ArrayList 有什么区别

描述下 Java 中集合(Collections)，接口(Interfaces)，实现(Implementations)的概念。LinkedList 与 ArrayList 的区别是什么？

插入数据时，ArrayList, LinkedList, Vector 谁速度较快？

ArrayList

ArrayList 和 HashMap 的默认大小是多数

ArrayList 和 LinkedList 的区别，什么时候用 ArrayList？

ArrayList 和 Set 的区别？

ArrayList, LinkedList, Vector 的区别

ArrayList 是如何实现的，ArrayList 和 LinkedList 的区别

ArrayList 如何实现扩容

Array 和 ArrayList 有何区别？什么时候更适合用 Array

说出 ArraList, Vector, LinkedList 的存储性能和特性

Map

Map, Set, List, Queue, Stack

Map 接口提供了哪些不同的集合视图

为什么 Map 接口不继承 Collection 接口

Collections

介绍 Java 中的 Collection Framework。集合类框架的基本接口有哪些

Collections 类是什么？Collection 和 Collections 的区别？Collection、Map 的实现

集合类框架的最佳实践有哪些

为什么 Collection 不从 Cloneable 和 Serializable 接口继承

说出几点 Java 中使用 Collections 的最佳实践？

Collections 中 遗留类 (HashTable、Vector) 和 现有类的区别

什么是 B+树，B-树，列出实际的使用场景。

接口

Comparator 与 Comparable 接口是干什么的？列出它们的区别

对象

拷贝(clone)

如何实现对象克隆

深拷贝和浅拷贝区别

深拷贝和浅拷贝如何实现激活机制

写 clone() 方法时，通常都有一行代码，是什么

比较

在比较对象时,"==" 运算符和 equals 运算有何区别

如果要重写一个对象的 equals 方法, 还要考虑什么

两个对象值相同(x.equals(y) == true), 但却可有不同的 hash code, 这句话对不对

问:

队列和栈是什么, 列出它们的区别?

答:

略。

问:

BlockingQueue 是什么?

答:

线程安全的队列, 可以以阻塞和非阻塞模式工作。常见的实现类有:

- ArrayBlockingQueue
- LinkedBlockingQueue
- DelayQueue
- PriorityBlockingQueue
- SynchronousQueue

问:

简述 **ConcurrentLinkedQueue**、**LinkedBlockingQueue** 的用处和不同之处。

答:

构造器

构造器链是什么

创建对象时构造器的调用顺序

不可变对象

什么是不可变象 (immutable object)

为什么 Java 中的 String 是不可变的 (Immutable)

如何构建不可变的类结构? 关键点在哪里

能创建一个包含可变对象的不可变对象吗

如何对一组对象进行排序

方法

构造器 (constructor) 是否可被重写 (override)

方法可以同时即是 static 又是 synchronized 的吗

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

Java 支持哪种参数传递类型

一个对象被当作参数传递到一个方法, 是值传递还是引用传递

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

我们能否重载 main() 方法

如果 main 方法被声明为 private 会怎样

GC

概念

GC 是什么? 为什么要有 GC

什么时候会导致垃圾回收

GC 是怎么样运行的

新老以及永久区是什么

GC 有几种方式? 怎么配置

什么时候一个对象会被 GC? 如何判断一个对象是否存活

System.gc() Runtime.gc() 会做什么事情? 能保证 GC 执行吗

垃圾回收器可以马上回收内存吗? 有什么办法主动通知虚拟机进行垃圾回收?

Minor GC、Major GC、Young GC 与 Full GC 分别在什么时候发生

垃圾回收算法的实现原理

如果对象的引用被置为 null, 垃圾收集器是否会立即释放对象占用的内存?

垃圾回收的最佳做法是什么

GC 收集器有哪些

垃圾回收器的基本原理是什么?

串行(serial)收集器和吞吐量(throughput)收集器的区别是什么

Serial 与 Parallel GC 之间的不同之处

CMS 收集器 与 G1 收集器的特点与区别

CMS 垃圾回收器的工作过程

JVM 中一次完整的 GC 流程是怎样的? 对象如何晋升到老年代

吞吐量优先和响应优先的垃圾收集器选择

GC 策略

举个实际的场景，选择一个 GC 策略

JVM 的永久代中会发生垃圾回收吗

收集方法

标记清除、标记整理、复制算法的原理与特点？分别用在什么地方

如果让你优化收集方法，有什么思路

JVM

参数

说说你知道的几种主要的 jvm 参数

-XX:+UseCompressedOops 有什么作用

类加载器(ClassLoader)

Java 类加载器都有哪些

JVM 如何加载字节码文件

内存管理

JVM 内存分哪几个区，每个区的作用是什么

一个对象从创建到销毁都是怎么在这些部分里存活和转移的

解释内存中的栈(stack)、堆(heap)和方法区(method area)的用法

JVM 中哪个参数是用来控制线程的栈堆栈小

简述内存分配与回收策略

简述重排序，内存屏障，happen-before，主内存，工作内存

Java 中存在内存泄漏问题吗？请举例说明

简述 Java 中软引用（SoftReferenc）、弱引用（WeakReference）和虚引用

内存映射缓存区是什么

jstack, jstat, jmap, jconsole 怎么用

32 位 JVM 和 64 位 JVM 的最大堆内存分别是多数？32 位和 64 位的 JVM，int 类型变量的长度是多数？

怎样通过 Java 程序来判断 JVM 是 32 位 还是 64 位

JVM 自身会维护缓存吗？是不是在堆中进行对象分配，操作系统的堆还是 JVM 自己管理堆

什么情况下会发生栈内存溢出

双亲委派模型是什么

多线程

基本概念

什么是线程

多线程的优点

多线程的几种实现方式

用 Runnable 还是 Thread

什么是线程安全

Vector, SimpleDateFormat 是线程安全类吗

什么 Java 原型不是线程安全的

哪些集合类是线程安全的

多线程中的忙循环是什么

如何创建一个线程

编写多线程程序有几种实现方式

什么是线程局部变量

线程和进程有什么区别？进程间如何通讯，线程间如何通讯

什么是多线程环境下的伪共享（false sharing）

同步和异步有何异同，在什么情况下分别使用他们？举例说明

Current

ConcurrentHashMap 和 Hashtable 的区别

ArrayBlockingQueue, CountdownLatch 的用法

ConcurrentHashMap 的并发度是什么

CyclicBarrier 和 CountdownLatch 有什么不同？各自的内部原理和用法是什么

Semaphore 的用法

Thread

启动一个线程是调用 run() 还是 start() 方法？start() 和 run() 方法有什么区别

调用 start() 方法时会执行 run() 方法，为什么不能直接调用 run() 方法

sleep() 方法和对象的 wait() 方法都可以让线程暂停执行，它们有什么区别

yield 方法有什么作用？sleep() 方法和 yield() 方法有什么区别

Java 中如何停止一个线程

stop() 和 suspend() 方法为何不推荐使用

如何在两个线程间共享数据

如何强制启动一个线程

如何让正在运行的线程暂停一段时间

什么是线程组，为什么在 Java 中不推荐使用

你是如何调用 wait（方法的）？使用 if 块还是循环？为什么

生命周期

有哪些不同的线程生命周期

线程状态，BLOCKED 和 WAITING 有什么区别

画一个线程的生命周期状态图

ThreadLocal 用途是什么，原理是什么，用的时候要注意什么

ThreadPool

线程池是什么？为什么要使用它

如何创建一个 Java 线程池

ThreadPool 用法与优势

提交任务时，线程池队列已满时会发生什么

newCachedThreadPool 和 newFixedThreadPool 有什么区别？简述原理。构造函数的各个参数的含义是什么，比如 coreSize, maxSize 等

线程池的实现策略

线程池的关闭方式有几种，各自的区别是什么

线程池中 submit() 和 execute() 方法有什么区别？

线程调度

Java 中用到的线程调度算法是什么

什么是多线程中的上下文切换

你对线程优先级的理解是什么

什么是线程调度器 (Thread Scheduler) 和时间分片 (Time Slicing)

线程同步

请说出你所知的线程同步的方法

synchronized 的原理是什么

synchronized 和 ReentrantLock 有什么不同

什么场景下可以使用 volatile 替换 synchronized

有 T1, T2, T3 三个线程，怎么确保它们按顺序执行？怎样保证 T2 在 T1 执行完后执行，T3 在 T2 执行完后执行

同步块内的线程抛出异常会发生什么

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

使用 `synchronized` 修饰静态方法和非静态方法有什么区别

如何从给定集合那里创建一个 `synchronized` 的集合

锁

Java Concurrency API 中的 `Lock` 接口是什么？对比同步它有什么优势

`Lock` 与 `Synchronized` 的区别？`Lock` 接口比 `synchronized` 块的优势是什么

`ReadWriteLock` 是什么？

锁机制有什么用

什么是乐观锁 (Optimistic Locking)？如何实现乐观锁？如何避免 ABA 问题

解释以下名词：重排序，自旋锁，偏向锁，轻量级锁，可重入锁，公平锁，非公平锁，乐观锁，悲观锁

什么时候应该使用可重入锁

简述锁的等级方法锁、对象锁、类锁

Java 中活锁和死锁有什么区别？

什么是死锁(Deadlock)？导致线程死锁的原因？如何确保 N 个线程可以访问 N 个资源同时又不导致死锁

死锁与活锁的区别，死锁与饥饿的区别

怎么检测一个线程是否拥有锁

如何实现分布式锁

有哪些无锁数据结构，他们实现的原理是什么

读写锁可以用于什么应用场景

`Executors` 类是什么？`Executor` 和 `Executors` 的区别

什么是 Java 线程转储(Thread Dump)，如何得到它

如何在 Java 中获取线程堆栈

说出 3 条在 Java 中使用线程的最佳实践

在线程中你怎么处理不可捕捉异常

实际项目中使用多线程举例。你在多线程环境中遇到的常见的问题是什么？你是怎么解决它的

请说出与线程同步以及线程调度相关的方法

程序中有 3 个 `socket`，需要多少个线程来处理

假如有一个第三方接口，有很多个线程去调用获取数据，现在规定每秒钟最多有 10 个线程同时调用它，如何做到

如何在 Windows 和 Linux 上查找哪个线程使用的 CPU 时间最长

如何确保 `main()` 方法所在的线程是 Java 程序最后结束的线程

非常多个线程（可能是不同机器），相互之间需要等待协调才能完成某种工作，问怎么设计这种协调方案

你需要实现一个高效的缓存，它允许多个用户读，但只允许一个用户写，以此来保持它的完整性，你会怎样去实现它

异常

基本概念

Error 和 Exception 有什么区别

UnsupportedOperationException 是什么

NullPointerException 和 ArrayIndexOutOfBoundsException 之间有什么相同之处

什么是受检查的异常，什么是运行时异常

运行时异常与一般异常有何异同

简述一个你最常见到的 runtime exception(运行时异常)

finally

finally 关键词在异常处理中如何使用

如果执行 finally 代码块之前方法返回了结果，或者 JVM 退出了，finally 块中的代码还会执行吗

try 里有 return, finally 还执行么？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 前还是后

在什么情况下，finally 语句不会执行

throw 和 throws 有什么区别？

OOM 你遇到过哪些情况？你是怎么搞定的？

SOF 你遇到过哪些情况？

既然我们可以用 RuntimeException 来处理错误，那么你认为为什么 Java 中还存在检查型异常

当自己创建异常类的时候应该注意什么

导致空指针异常的原因

异常处理 handle or declare 原则应该如何理解

怎么利用 JUnit 来测试一个方法的异常

catch 块里别不写代码有什么问题

你曾经自定义实现过异常吗？怎么写的

什么是 异常链

在 try 块中可以抛出异常吗

JDBC

通过 JDBC 连接数据库有哪几种方式
阐述 JDBC 操作数据库的基本步骤
JDBC 中如何进行事务处理
什么是 JdbcTemplate
什么是 DAO 模块
使用 JDBC 操作数据库时，如何提升读取数据的性能？如何提升更新数据的性能
列出 5 个应该遵循的 JDBC 最佳实践

IO

File

File 类型中定义了什么方法来创建一级目录
File 类型中定义了什么方法来判断一个文件是否存在

流

为了提高读写性能，可以采用什么流
Java 中有几种类型的流
JDK 为每种类型的流提供了一些抽象类以供继承，分别是哪些类
对文本文件操作用什么 I/O 流
对各种基本数据类型和 String 类型的读写，采用什么流
能指定字符编码的 I/O 流类型是什么

序列化

什么是序列化？如何实现 Java 序列化及注意事项
Serializable 与 Externalizable 的区别

Socket

socket 选项 TCP NO DELAY 是指什么
Socket 工作在 TCP/IP 协议栈是哪一层
TCP、UDP 区别及 Java 实现方式

说几点 IO 的最佳实践
直接缓冲区与非直接缓冲器有什么区别？
怎么读写 ByteBuffer？ByteBuffer 中的字节序是什么
当用 `System.in.read(buffer)` 从键盘输入一行 n 个字符后，存储在缓冲区 buffer 中的字节数是多少

如何使用扫描器类 (Scanner Class) 令牌化

面向对象编程 (OOP)

解释下多态性 (polymorphism), 封装性 (encapsulation), 内聚 (cohesion) 以及耦合 (coupling)

多态的实现原理

封装、继承和多态是什么

对象封装的原则是什么?

类

获得一个类的类对象有哪些方式

重载 (Overload) 和重写 (Override) 的区别。重载的方法能否根据返回类型进行区分?

说出几条 Java 中方法重载的最佳实践

抽象类

抽象类和接口的区别

抽象类中是否可以有静态的 main 方法

抽象类是否可实现(implements)接口

抽象类是否可继承具体类(concrete class)

匿名类 (Anonymous Inner Class)

匿名内部类是否可以继承其它类? 是否可以实现接口

内部类

内部类分为几种

内部类可以引用它的包含类 (外部类) 的成员吗

请说一下 Java 中为什么要引入内部类? 还有匿名内部类

继承

继承 (Inheritance) 与聚合 (Aggregation) 的区别在哪里

继承和组合之间有什么不同

为什么类只能单继承, 接口可以多继承

存在两个类, B 继承 A, C 继承 B, 能将 B 转换为 C 么? 如 C = (C) B

如果类 a 继承类 b, 实现接口 c, 而类 b 和接口 c 中定义了同名变量, 请问会出现什么问题

接口

接口是什么

接口是否可继承接口

为什么要使用接口而不是直接使用具体类? 接口有什么优点

泛型

泛型的存在是用来解决什么问题

泛型的常用特点

List 能否转为 List

工具类

日历

Calendar Class 的用途

如何在 Java 中获取日历类的实例

解释一些日历类中的重要方法

GregorianCalendar 类是什么

SimpleTimeZone 类是什么

Locale 类是什么

如何格式化日期对象

如何添加小时(hour)到一个日期对象(Date Objects)

如何将字符串 YYYYMMDD 转换为日期

Math

Math.round()什么作用? Math.round(11.5) 等于多少? Math.round(-11.5)等于多少?

XML

XML 文档定义有几种形式? 它们之间有何本质区别? 解析 XML 文档有哪几种方式? DOM 和 SAX 解析器有什么不同?

Java 解析 XML 的方式

用 jdom 解析 xml 文件时如何解决中文问题? 如何解析

你在项目中用到了 XML 技术的哪些方面? 如何实现

动态代理

描述动态代理的几种实现方式, 分别说出相应的优缺点

设计模式

什么是设计模式 (Design Patterns)? 你用过哪种设计模式? 用在什么场合
你知道哪些商业级设计模式?

哪些设计模式可以增加系统的可扩展性

单例模式

除了单例模式，你在生产环境中还用过什么设计模式？

写 Singleton 单例模式

单例模式的双检锁是什么

如何创建线程安全的 Singleton

什么是类的单例模式

写出三种单例模式实现

适配器模式

适配器模式是什么？什么时候使用

适配器模式和代理模式之前有什么不同

适配器模式和装饰器模式有什么区别

什么时候使用享元模式

什么时候使用组合模式

什么时候使用访问者模式

什么是模板方法模式

请给出 1 个符合开闭原则的设计模式的例子

开放问题

用一句话概括 Web 编程的特点

Google 是如何在一秒内把搜索结果返回给用户

哪种依赖注入方式你建议使用，构造器注入，还是 Setter 方法注入

树（二叉或其他）形成许多普通数据结构的基础。请描述一些这样的数据结构以及何时可以使用它们

某一项功能如何设计

线上系统突然变得异常缓慢，你如何查找问题

什么样的项目不适合用框架

新浪微博是如何实现把微博推给订阅者

简要介绍下从浏览器输入 URL 开始到获取到请求界面之后 Java Web 应用中发生了什么

请你谈谈 SSH 整合

高并发下，如何做到安全的修改同一行数据

12306 网站的订票系统如何实现，如何保证不会票不被超卖

网站性能优化如何优化的

聊了下曾经参与设计的服务器架构

请思考一个方案，实现分布式环境下的 countDownLatch

请思考一个方案，设计一个可以控制缓存总体大小的自动适应的本地缓存
在你的职业生涯中，算得上最困难的技术挑战是什么

如何写一篇设计文档，目录是什么
大写的 O 是什么？举几个例子
编程中自己都怎么考虑一些设计原则的，比如开闭原则，以及在工作中的应用

解释一下网络应用的模式及其特点
设计一个在线文档系统，文档可以被编辑，如何防止多人同时对同一份文档进行编辑更新

说出数据连接池的工作机制是什么
怎么获取一个文件中单词出现的最高频率
描述一下你最常用的编程风格
如果有机会重新设计你们的产品，你会怎么做
如何搭建一个高可用系统
如何启动时不需输入用户名与密码
如何在基于 Java 的 Web 项目中实现文件上传和下载
如何实现一个秒杀系统，保证只有几位用户能买到某件商品。
如何实现负载均衡，有哪些算法可以实现
如何设计一个购物车？想想淘宝的购物车如何实现的
如何设计一套高并发支付方案，架构如何设计
如何设计建立和保持 100w 的长连接
如何避免浏览器缓存。
如何防止缓存雪崩
如果 AB 两个系统互相依赖，如何解除依
如果有人恶意创建非法连接，怎么解决
如果有几十亿白名单，每天白天需要高并发查询，晚上需要更新一次，如何设计这个功能

如果系统要使用超大整数（超过 long 长度范围），请你设计一个数据结构来存储这种超大型数字以及设计一种算法来实现超大整数加法运算）
如果要设计一个图形系统，请你设计基本的图形元件 (Point, Line, Rectangle, Triangle) 的简单实现

如果让你实现一个并发安全的链表，你会怎么做
应用服务器与 WEB 服务器的区别？应用服务器怎么监控性能，各种方式的
区别？你使用过的应用服务器优化技术有哪些
大型网站在架构上应当考虑哪些问题
有没有处理过线上问题？出现内存泄露，CPU 利用率标高，应用无响应时如何处理的

最近看什么书，印象最深刻的是什么
描述下常用的重构技巧
你使用什么版本管理工具？分支（Branch）与标签（Tag）之间的区别在哪里

你有了解过存在哪些反模式（Anti-Patterns）吗
你用过的网站前端优化的技术有哪些
如何分析 Thread dump

你如何理解 AOP 中的连接点 (Joinpoint)、切点 (Pointcut)、增强 (Advice)、引入 (Introduction)、织入 (Weaving)、切面 (Aspect) 这些概念

你是如何处理内存泄露或者栈溢出问题的

你们线上应用的 JVM 参数有哪些

怎么提升系统的 QPS 和吞吐量

知识面

解释什么是 MESI 协议(缓存一致性)

谈谈 reactor 模型

Java 9 带来了怎样的新功能

Java 与 C++ 对比, C++ 或 Java 中的异常处理机制的简单原理和应用

简单讲讲 Tomcat 结构, 以及其类加载器流程

虚拟内存是什么

阐述下 SOLID 原则

请简要讲一下你对测试驱动开发 (TDD) 的认识

CDN 实现原理

Maven 和 ANT 有什么区别

UML 中有哪些常用的图

Linux

Linux 下 IO 模型有几种, 各自的含义是什么。

Linux 系统下你关注过哪些内核参数, 说说你知道的

Linux 下用一行命令查看文件的最后五行

平时用到哪些 Linux 命令

用一行命令输出正在运行的 Java 进程

使用什么命令来确定是否有 Tomcat 实例运行在机器上

什么是 N+1 难题

什么是 paxos 算法

什么是 restful, 讲讲你理解的 restful

什么是 zab 协议

什么是领域模型(domain model)? 贫血模型(anaemic domain model) 和充血模型(rich domain model)有什么区别

什么是领域驱动开发 (Domain Driven Development)

介绍一下了解的 Java 领域的 Web Service 框架

Web Server、Web Container 与 Application Server 的区别是什么

微服务 (MicroServices) 与巨石型应用 (Monolithic Applications) 之间的区别在哪里

描述 Cookie 和 Session 的作用, 区别和各自的应用范围, Session 工作原理

你常用的持续集成 (Continuous Integration)、静态代码分析 (Static Code

Analysis) 工具有哪些

简述下数据库正则化 (Normalizations)

KISS, DRY, YAGNI 等原则是什么含义

分布式事务的原理, 优缺点, 如何使用分布式事务?

布式集群下如何做到唯一序列号

网络

HTTPS 的加密方式是什么, 讲讲整个加密解密流程

HTTPS 和 HTTP 的区别

HTTP 连接池实现原理

HTTP 集群方案

Nginx、lighttpd、Apache 三大主流 Web 服务器的区别

是否看过框架的一些代码

持久层设计要考虑的问题有哪些? 你用过的持久层框架有哪些

数值提升是什么

你能解释一下里氏替换原则吗

你是如何测试一个应用的? 知道哪些测试框架

传输层常见编程协议有哪些? 并说出各自的特点

第3章 Java 编程题

1 试题 A 部分

1.1 计算加班费

问题:

加班 10 小时以下加班费是时薪的 1.5 倍。加班 10 小时或以上, 按 4 元/时算。提示: (一个月工作 26 天, 一天正常工作 8 小时)

计算 1000 月薪, 加班 9 小时的加班费

计算 2500 月薪, 加班 11 小时的加班费

计算 1000 月薪, 加班 15 小时的加班费

1.2 卖东西

一家商场有红苹果和青苹果出售。(红苹果 5 元/个, 青苹果 4 元/个)。

模拟一个进货。红苹果跟青苹果各进 200 个。

模拟一个出售。红苹果跟青苹果各卖出 10 个。每卖出一个苹果需要进行统计。

提示：一个苹果是一个单独的实体。

日期提取

有这样一个时间字符串：2008-8-8 20:08:08，请编写能够匹配它的正则表达式，并编写 Java 代码将日期后面的时分秒提取出来，即：20:08:08

线程

8 设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。

用 Java 写一个多线程程序，如写四个线程，二个加 1，二个对一个变量减一，输出

wait-notify 写一段代码来解决生产者-消费者问题

数字

判断 101-200 之间有多少个素数，并输出所有素数

用最有效率的方法算出 2 乘以 17 等于多少

有 1 亿个数字，其中有 2 个是重复的，快速找到它，时间和空间要最优

2 亿个随机生成的无序整数，找出中间大小的值

10 亿个数字里里面找最小的 10 个

1 到 1 亿的自然数，求所有数的拆分后的数字之和，如 286 拆分成 2、8、6，如 1 到 11 拆分后的数字之和 $\Rightarrow 1 + \dots + 9 + 1 + 0 + 1 + 1$

一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数

一个数组中所有的元素都出现了三次，只有一个元素出现了一次找到这个元素

一球从 100 米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第 10 次落地时，共经过多少米？第 10 次反弹多高？

求 100—1000 内质数的和

求 1 到 100 的数的平均数

求 $s=a+a+aaa+aaaa+aa\dots a$ 的值，其中 a 是一个数字。例如 $2+22+222+2222+22222$ (此时共有 5 个数相加)，几个数相加有键盘控制。求出 1 到 100 的和

算出 1 到 40 的质数，放进数组里

显示数组里的数

找出第[5]个数

删除第[9]个数，再显示删除后的第[9]个

有 $3n+1$ 个数字，其中 $3n$ 个中是重复的，只有 1 个是不重复的，怎么找出来。

有一组数 1.1.2.3.5.8.13.21.34。写出程序随便输入一个数就能给出和前一组数字同规律的头 5 个数

计算指定数字的阶乘

开发 Fizz Buzz

给定一个包含 N 个整数的数组，找出丢失的整数

一个排好序的数组，找出两数之和为 m 的所有组合

将一个正整数分解质因数。例如：输入 90,打印出 $90=2*3*3*5$ 。

打印出所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如：153 是一个“水仙花数”，因为 $153=1$ 的三次方+5 的三次方+3 的三次方

原地交换两个变量的值

找出 4 字节整数的中位数

找到整数的平方根

实现斐波那契

网络

用 Java Socket 编程，读服务器几个字符，再写入本地显示

反射

反射机制提供了什么功能？

反射是如何实现的

哪里用到反射机制

反射中 `Class.forName` 和 `ClassLoader` 区别

反射创建类实例的三种方式是什么

如何通过反射调用对象的方法

如何通过反射获取和设置对象私有字段的值

反射机制的优缺点

数据库

写一段 JDBC 连 Oracle 的程序,并实现数据查询

算法

50 个人围坐一圈，当数到三或者三的倍数出圈，问剩下的人是谁，原来的位置是多少

实现一个电梯模拟器用

写一个冒泡排序

写一个折半查找

随机产生 20 个不能重复的字符并排序

写一个函数，传入 2 个有序的整数数组，返回一个有序的整数数组

写一段代码在遍历 ArrayList 时移除一个元素

古典问题：有一对兔子，从出生后第 3 个月起每个月都生一对兔子，小兔子长到第四个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少

约瑟芬环游戏

正则

请编写一段匹配 IP 地址的正则表达式

写出一个正则表达式来判断一个字符串是否是一个数字

字符串

写一个方法，入一个文件名和一个字符串，统计这个字符串在这个文件中出现的次数。

写一个程序找出所有字符串的组合，并检查它们是否是回文串

写一个字符串反转函数，输入 abcde 转换成 edcba 代码

小游戏，倒转句子中的单词

将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串

请写一段代码来计算给定文本内字符“A”的个数。分别用迭代和递归两种方式

编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。但是要保证汉字不被截半个，如“我 ABC”4，应该截为“我 AB”，输入“我 ABC 汉 DEF”，6，应该输出为“我 ABC”而不是“我 ABC+汉的半个”

给定 2 个包含单词列表（每行一个）的文件，编程列出交集

打印出一个字符串的所有排列

将一个键盘输入的数字转化成中文输出(例如：输入 1234567，输出:一百二十拾三万四千五百六拾七)

在 Web 应用开发过程中经常遇到输出某种编码的字符，如从 GBK 到 ISO8859-1 等，如何输出一个某种编码的字符串

日期

计算两个日期之间的差距

如何成为一个牛掰的 Java 大神?

链接: <http://www.hollischuang.com/archives/489>

一、基础篇

1.1 JVM

1.1.1. Java 内存模型, Java 内存管理, Java 堆和栈, 垃圾回收

<http://www.jcp.org/en/jsr/detail?id=133>

<http://ifeve.com/jmm-faq/>

1.1.2. 了解 JVM 各种参数及调优

1.1.3. 学习使用 Java 工具

jps, jstack, jmap, jconsole, jinfo, jhat, javap, ...

<http://kenai.com/projects/btrace>

<http://www.crashub.org/>

<https://github.com/taobao/TProfiler>

<https://github.com/CSUG/HouseMD>

<http://wiki.cyclopsgroup.org/jmxterm>

<https://github.com/jlusdy/TBJMap>

1.1.4. 学习 Java 诊断工具

<http://www.eclipse.org/mat/>

<http://visualvm.java.net/oqlhelp.html>

1.1.5. 自己编写各种 outofmemory, stackoverflow 程序

HeapOutOfMemory

Young OutOfMemory

MethodArea OutOfMemory

ConstantPool OutOfMemory

DirectMemory OutOfMemory

Stack OutOfMemory

Stack OverFlow

1.1.6. 使用工具尝试解决以下问题, 并写下总结

当一个 Java 程序响应很慢时如何查找问题

当一个 Java 程序频繁 FullGC 时如何解决问题, 如何查看垃圾回收日志

当一个 Java 应用发生 OutOfMemory 时该如何解决, 年轻代、年老代、永久

代解决办法不同，导致原因也不同

1.1.7. 参考资料

<http://docs.oracle.com/javase/specs/jvms/se7/html/>

<http://www.cs.umd.edu/~pugh/java/memoryModel/>

<http://gee.cs.oswego.edu/dl/jmm/cookbook.html>

1.2. Java 基础知识

1.2.1. 阅读源代码

java.lang.String

java.lang.Integer

java.lang.Long

java.lang.Enum

java.math.BigDecimal

java.lang.ThreadLocal

java.lang.ClassLoader & java.net.URLClassLoader

java.util.ArrayList & java.util.LinkedList

java.util.HashMap & java.util.LinkedHashMap & java.util.TreeMap

java.util.HashSet & java.util.LinkedHashSet & java.util.TreeSet

1.2.2. 熟悉 Java 中各种变量类型

1.2.3. 熟悉 Java String 的使用，熟悉 String 的各种函数

1.2.4. 熟悉 Java 中各种关键字

1.2.5. 学会使用 List, Map, Stack, Queue, Set

上述数据结构的遍历

上述数据结构的使用场景

Java 实现对 Array/List 排序

java.util.Arrays.sort()

java.util.Collections.sort()

Java 实现对 List 去重

Java 实现对 List 去重，并且需要保留数据原始的出现顺序

Java 实现最近最少使用 cache，用 LinkedHashMap

1.2.6. Java IO&Java NIO，并学会使用

java.io.*

java.nio.*

nio 和 reactor 设计模式

文件编码，字符集

1.2.7. Java 反射与 javassist

反射与工厂模式

java.lang.reflect.*

1.2.8. Java 序列化

java.io.Serializable

什么是序列化，为什么序列化

序列化与单例模式

google 序列化 protobuf

1.2.9. 虚引用，弱引用，软引用

java.lang.ref.*

实验这些引用的回收

1.2.10. 熟悉 Java 系统属性

java.util.Properties

1.2.11. 熟悉 Annotation 用法

java.lang.annotation.*

1.2.12. JMS

javax.jms.*

1.2.13. JMX

java.lang.management.*

javax.management.*

1.2.14. 泛型和继承，泛型和擦除

1.2.15. 自动拆箱装箱与字节码

1.2.16. 实现 Callback

1.2.17. java.lang.Void 类使用

1.2.18. Java Agent，premain 函数

java.lang.instrument

1.2.19. 单元测试

Junit, <http://junit.org/>

Jmockit, <https://code.google.com/p/jmockit/>

djUnit, <http://works.dgic.co.jp/djunit/>

1.2.20. Java 实现通过正则表达式提取一段文本中的电子邮件，并将@替换为

#输出

java.lang.util.regex.*

1.2.21. 学习使用常用的 Java 工具库

commons.lang, commons.*...

guava-libraries

netty

1.2.22. 什么是 API&SPI

http://en.wikipedia.org/wiki/Application_programming_interface

http://en.wikipedia.org/wiki/Service_provider_interface

1.2.23. 参考资料

JDK src.zip 源代码

<http://openjdk.java.net/>

<http://commons.apache.org/>

<https://code.google.com/p/guava-libraries/>

<http://netty.io/>

<http://stackoverflow.com/questions/2954372/difference-between-spi-and-api>

<http://stackoverflow.com/questions/11404230/how-to-implement-the-api-spi-patt>

ern-in-java

1.3. Java 并发编程

1.3.1. 阅读源代码，并学会使用

java.lang.Thread
java.lang.Runnable
java.util.concurrent.Callable
java.util.concurrent.locks.ReentrantLock
java.util.concurrent.locks.ReentrantReadWriteLock
java.util.concurrent.atomic.Atomic*
java.util.concurrent.Semaphore
java.util.concurrent.CountDownLatch
java.util.concurrent.CyclicBarrier
java.util.concurrent.ConcurrentHashMap
java.util.concurrent.Executors

1.3.2. 学习使用线程池，自己设计线程池需要注意什么

1.3.3. 锁

什么是锁，锁的种类有哪些，每种锁有什么特点，适用场景是什么
在并发编程中锁的意义是什么

1.3.4. synchronized 的作用是什么，synchronized 和 lock

1.3.5. sleep 和 wait

1.3.6. wait 和 notify

1.3.7. 写一个死锁的程序

1.3.8. 什么是守护线程，守护线程和非守护线程的区别以及用法

1.3.9. volatile 关键字的理解

C++ volatile 关键字和 Java volatile 关键字

happens-before 语义

编译器指令重排和 CPU 指令重排

http://en.wikipedia.org/wiki/Memory_ordering

http://en.wikipedia.org/wiki/Volatile_variable

<http://preshing.com/20130702/the-happens-before-relation/>

1.3.10. 以下代码是不是线程安全？为什么？如果为 count 加上 volatile 修饰是否能够做到线程安全？你觉得该怎么做是线程安全的？

```
public class Sample {  
    private static int count = 0;  
    public static void increment() {  
        count++;  
    }  
}
```

1.3.11. 解释一下下面两段代码的差别

// 代码 1

```

public class Sample {
    private static int count = 0;
    synchronized public static void increment() {
        count++;
    }
}

```

// 代码 2

```

public class Sample {
    private static AtomicInteger count = new AtomicInteger(0);
    public static void increment() {
        count.getAndIncrement();
    }
}

```

1.3.12. 参考资料

<http://book.douban.com/subject/10484692/>

<http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>

二、进阶篇

2.1. Java 底层知识

2.1.1. 学习了解字节码、class 文件格式

http://en.wikipedia.org/wiki/Java_class_file

http://en.wikipedia.org/wiki/Java_bytecode

http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings

<http://www.csg.ci.i.u-tokyo.ac.jp/~chiba/javassist/>

<http://asm.ow2.org/>

2.1.2. 写一个程序要求实现 javap 的功能（手工完成，不借助 ASM 等工具）

如 Java 源代码：

```

public static void main(String[] args) {
    int i = 0;
    i += 1;
    i *= 1;
    System.out.println(i);
}

```

编译后读取 class 文件输出以下代码：

```
public static void main(java.lang.String[]);
```

Code:

Stack=2, Locals=2, Args_size=1

0: iconst_0

1: istore_1

```

2:   iinc      1, 1
5:   iload_1
6:   iconst_1
7:   imul
8:   istore_1
9:   getstatic      #2; //Field java/lang/System.out:Ljava/io/PrintStream;
12:  iload_1
13:  invokevirtual   #3; //Method java/io/PrintStream.println:(I)V
16:  return

```

LineNumberTable:

line 4: 0

line 5: 2

line 6: 5

line 7: 9

line 8: 16

2.1.3. CPU 缓存, L1, L2, L3 和伪共享

<http://duartes.org/gustavo/blog/post/intel-cpu-caches/>

<http://mechanical-sympathy.blogspot.com/2011/07/false-sharing.html>

2.1.4. 什么是尾递归

2.1.5. 熟悉位运算

用位运算实现加、减、乘、除、取余

2.1.6. 参考资料

<http://book.douban.com/subject/1138768/>

<http://book.douban.com/subject/6522893/>

http://en.wikipedia.org/wiki/Java_class_file

http://en.wikipedia.org/wiki/Java_bytecode

http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings

2.2. 设计模式

2.2.1. 实现 AOP

CGLIB 和 InvocationHandler 的区别

<http://cglib.sourceforge.net/>

动态代理模式

Javassist 实现 AOP

<http://www.csg.ci.i.u-tokyo.ac.jp/~chiba/javassist/>

ASM 实现 AOP

<http://asm.ow2.org/>

2.2.2. 使用模板方法设计模式和策略设计模式实现 IOC

2.2.3. 不用 synchronized 和 lock, 实现线程安全的单例模式

2.2.4. nio 和 reactor 设计模式

2.2.5. 参考资料

<http://asm.ow2.org/>

<http://cglib.sourceforge.net/>

<http://www.javassist.org/>

2.3. 网络编程知识

2.3.1. Java RMI, Socket, HttpClient

2.3.2. 用 Java 写一个简单的静态文件的 HTTP 服务器

实现客户端缓存功能, 支持返回 304

实现可并发下载一个文件

使用线程池处理客户端请求

使用 nio 处理客户端请求

支持简单的 rewrite 规则

上述功能在实现的时候需要满足“开闭原则”

2.3.3. 了解 nginx 和 apache 服务器的特性并搭建一个对应的服务器

<http://nginx.org/>

<http://httpd.apache.org/>

2.3.4. 用 Java 实现 FTP、SMTP 协议

2.3.5. 什么是 CDN? 如果实现? DNS 起到什么作用?

搭建一个 DNS 服务器

搭建一个 Squid 或 Apache Traffic Server 服务器

<http://www.squid-cache.org/>

<http://trafficserver.apache.org/>

http://en.wikipedia.org/wiki/Domain_Name_System

2.3.6. 参考资料

<http://www.ietf.org/rfc/rfc2616.txt>

<http://tools.ietf.org/rfc/rfc5321.txt>

http://en.wikipedia.org/wiki/Open/closed_principle

2.4. 框架知识

spring, spring mvc, 阅读主要源码

ibatis, 阅读主要源码

用 spring 和 ibatis 搭建 java server

2.5. 应用服务器知识

熟悉使用 jboss, <https://www.jboss.org/overview/>

熟悉使用 tomcat, <http://tomcat.apache.org/>

熟悉使用 jetty, <http://www.eclipse.org/jetty/>

三、高级篇

3.1. 编译原理知识

3.1.1. 用 Java 实现以下表达式解析并返回结果（语法和 Oracle 中的 select sysdate-1 from dual 类似）

sysdate

sysdate - 1

sysdate - 1/24

sysdate - 1/(12*2)

3.1.2. 实现对一个 List 通过 DSL 筛选

```
QList<Map<String, Object>> mapList = new QList<Map<String, Object>>;
mapList.add({"name": "hatter test"});
mapList.add({"id": -1, "name": "hatter test"});
mapList.add({"id": 0, "name": "hatter test"});
mapList.add({"id": 1, "name": "test test"});
mapList.add({"id": 2, "name": "hatter test"});
mapList.add({"id": 3, "name": "test hatter"});
mapList.query("id is not null and id > 0 and name like '%hatter%'");
```

要求返回列表中匹配的对象，即最后两个对象；

3.1.3. 用 Java 实现以下程序(语法和变量作用域处理都和 JavaScript 类似):

代码:

```
var a = 1;
var b = 2;
var c = function() {
var a = 3;
println(a);
println(b);
};
c();
println(a);
println(b);
```

输出:

```
3
2
1
2
```

3.1.4. 参考资料

http://en.wikipedia.org/wiki/Abstract_syntax_tree

<https://javacc.java.net/>

[http://www.antlr.org/](http://wwwantlr.org/)

3.2. 操作系统知识

Ubuntu

Centos

使用 linux, 熟悉 shell 脚本

3.3. 数据存储知识

3.3.1. 关系型数据库

MySQL

如何看执行计划

如何搭建 MySQL 主备

binlog 是什么

Derby, H2, PostgreSQL

SQLite

3.3.2. NoSQL

Cache

Redis

Memcached

Leveldb

Bigtable

HBase

Cassandra

Mongodb

图数据库

neo4j

3.3.3. 参考资料

<http://db-engines.com/en/ranking>

<http://redis.io/>

<https://code.google.com/p/leveldb/>

<http://hbase.apache.org/>

<http://cassandra.apache.org/>

<http://www.mongodb.org/>

<http://www.neo4j.org/>

3.4. 大数据知识

3.4.1. Zookeeper, 在 linux 上部署 zk

3.4.2. Solr, Lucene, ElasticSearch

在 linux 上部署 solr, solrcloud, , 新增、删除、查询索引

3.4.3. Storm, 流式计算, 了解 Spark, S4

在 linux 上部署 storm, 用 zookeeper 做协调, 运行 storm hello world, local 和 remote 模式运行调试 storm topology。

3.4.4. Hadoop, 离线计算

Hdfs: 部署 NameNode, SecondaryNameNode, DataNode, 上传文件、打开文件、更改文件、删除文件

MapReduce: 部署 JobTracker, TaskTracker, 编写 mr job

Hive: 部署 hive, 书写 hive sql, 得到结果

Presto: 类 hive, 不过比 hive 快, 非常值得学习

3.4.5. 分布式日志收集 flume, kafka, logstash

3.4.6. 数据挖掘, mahout

3.4.7. 参考资料

<http://zookeeper.apache.org/>

<https://lucene.apache.org/solr/>

<https://github.com/nathanmarz/storm/wiki>
<http://hadoop.apache.org/>
<http://prestodb.io/>
<http://flume.apache.org/>, <http://logstash.net/>, <http://kafka.apache.org/>
<http://mahout.apache.org/>
3.5. 网络安全知识

- 3.5.1. 什么是 DES、AES
- 3.5.2. 什么是 RSA、DSA
- 3.5.3. 什么是 MD5, SHA1
- 3.5.4. 什么是 SSL、TLS, 为什么 HTTPS 相对比较安全
- 3.5.5. 什么是中间人攻击、如果避免中间人攻击
- 3.5.6. 什么是 DOS、DDOS、CC 攻击
- 3.5.7. 什么是 CSRF 攻击
- 3.5.8. 什么是 CSS 攻击
- 3.5.9. 什么是 SQL 注入攻击
- 3.5.10. 什么是 Hash 碰撞拒绝服务攻击
- 3.5.11. 了解并学习下面几种增强安全的技术

<http://www.openauthentication.org/>
HOTP <http://www.ietf.org/rfc/rfc4226.txt>
TOTP <http://tools.ietf.org/rfc/rfc6238.txt>
OCRA <http://tools.ietf.org/rfc/rfc6287.txt>
[http://en.wikipedia.org/wiki/Salt_\(cryptography\)](http://en.wikipedia.org/wiki/Salt_(cryptography))
3.5.12. 用 openssl 签一个证书部署到 apache 或 nginx
3.5.13. 参考资料
http://en.wikipedia.org/wiki/Cryptographic_hash_function
http://en.wikipedia.org/wiki/Block_cipher
http://en.wikipedia.org/wiki/Public-key_cryptography
http://en.wikipedia.org/wiki/Transport_Layer_Security
<http://www.openssl.org/>
<https://code.google.com/p/google-authenticator/>
四、 扩展篇

4.1. 相关知识

- 4.1.1. 云计算, 分布式, 高可用, 可扩展
- 4.1.2. 虚拟化
<https://linuxcontainers.org/>
http://www.linux-kvm.org/page/Main_Page
<http://www.xenproject.org/>
<https://www.docker.io/>

4.1.3. 监控

<http://www.nagios.org/>

<http://ganglia.info/>

4.1.4. 负载均衡

<http://www.linuxvirtualserver.org/>

4.1.5. 学习使用 git

<https://github.com/>

<https://git.oschina.net/>

4.1.6. 学习使用 maven

<http://maven.apache.org/>

4.1.7. 学习使用 gradle

<http://www.gradle.org/>

4.1.8. 学习一个小语种语言

Groovy

Scala

LISP, Common LISP, Schema, Clojure

R

Julia

Lua

Ruby

4.1.9. 尝试了解编码的本质

了解以下概念

ASCII, ISO-8859-1

GB2312, GBK, GB18030

Unicode, UTF-8

不使用 `String.getBytes()` 等其他工具类/函数完成下面功能

```
public static void main(String[] args) throws IOException {
    String str = "Hello, 我们中国人。";
    byte[] utf8Bytes = toUTF8Bytes(str);
    FileOutputStream fos = new FileOutputStream("f.txt");
    fos.write(utf8Bytes);
    fos.close();
}

public static byte[] toUTF8Bytes(String str) {
    return null; // TODO
}
```

想一下上面的程序能不能写一个转 GBK 的？

写个程序自动判断一个文件是哪种编码

4.1.10. 尝试了解时间的本质

时区 & 冬令时、夏令时

http://en.wikipedia.org/wiki/Time_zone

<ftp://ftp.iana.org/tz/data/asia>

<http://zh.wikipedia.org/wiki/%E4%B8%AD%E5%9C%8B%E6%99%82%E5%8D%80>

闰年

http://en.wikipedia.org/wiki/Leap_year

闰秒

<ftp://ftp.iana.org/tz/data/leapseconds>

System.currentTimeMillis() 返回的时间是什么

4.1.11. 参考资料

<http://git-scm.com/>

<http://en.wikipedia.org/wiki/UTF-8>

<http://www.iana.org/time-zones>

4.2. 扩展学习

4.2.1. JavaScript 知识

4.2.1.1. 什么是 prototype

修改代码，使程序输出“1 3 5”：

<http://jsfiddle.net/Ts7Fk/>

4.2.1.2. 什么是闭包

看一下这段代码，并解释一下为什么按 Button1 时没有 alert 出“This is button: 1”，如何修改：

<http://jsfiddle.net/FDPj3/1/>

4.2.1.3. 了解并学习一个 JS 框架

jQuery

ExtJS

AngularJS

4.2.1.4. 写一个 Greasemonkey 插件

<http://en.wikipedia.org/wiki/Greasemonkey>

4.2.1.5. 学习 node.js

<http://nodejs.org/>

4.2.2. 学习 html5

AngularJS, <https://docs.angularjs.org/api>

4.2.3. 参考资料

<http://www.ecmascript.org/>

<http://jsfiddle.net/>

<http://jsbin.com/>

<http://runjs.cn/>

<http://userscripts.org/>

五、 推荐书籍

《深入 Java 虚拟机》
《深入理解 Java 虚拟机》
《Effective Java》
《七周七语言》
《七周七数据》
《Hadoop 技术内幕》
《Hbase In Action》
《Mahout In Action》
《这就是搜索引擎》
《Solr In Action》
《深入分析 Java Web 技术内幕》
《大型网站技术架构》
《高性能 MySQL》
《算法导论》
《计算机程序设计艺术》
《代码大全》
《JavaScript 权威指南》

百度“Java 面试题”前 200 页：

github.com/tangyouhua/program-resource/blob/master/program-interview/java-baidu-200.md

数据库相关

FileMaker

FileMaker 是一个关于数据库的应用软件，它因易于使用且有不要使用另外的第三方应用程序就能动态地为网页服务的能力而著称。

FileMaker 是一个关于数据库的应用软件，它因易于使用且有不要使用另外的第三方应用程序就能动态地为网页服务的能力而著称。有了它的安装式网络服务（被称为“网络同伴”）和 out-of-the-box 的商业模板，FileMaker 由于为网站管理员提供了理想的应用软件而获得很高的赞赏，这些网站管理员希望快速地增加网络数据库将信息发布到他们的网站或企业内部互联网的能力。

FileMaker 最初是作为 Macintosh 计算机应用的个人数据库应用程序而被开发的，由 Claris Corporation 以及后来归于 Windows 旗下来开发的。在为建造 FileMaker 对用户友善的信誉而努力的过程中，FileMaker（现在它的最新版本 FMP5）被刻意重新设计，以使它的用户界面更加简易，就像 Microsoft 的办公室软件的界面一样。FMP5 有两种 Web 发布模式：Instant Web Publishing（这种模式使用预先设计的模板并且不要懂得超文本标示语言(HTML)，和 Custom Web

Publishing（这种模式要求懂得超文本标示语言，并要懂得被称为 Claris Dynamic Markup Language 的所有权归 Web Companion 的设计语言）。

FMP5 可以与广泛多样的中间设备应用程序、公共网关接口（CGI）脚本以及提供附加网络性能的应用服务器一起被使用。例如，为了提供加密套接字协议层（SSL）支持，FMP5 能够使用 WebSta 来连接到 Mac OS 服务器，或者使用 Microsoft's Web Server 来连接到 Windows NT 服务器。为了增加可量测性，它可能建立一个普通计算机的阵列（RAID）来提供负荷平衡，并用 FMP5 的多重拷贝来确保一致的页面服务。

FMP 也可以与 Palm 组合来提供数据库管理程序，它使用 Palm 的无线互联网服务把信息加入到 FMP 数据库中。

云计算

据知名市场调研机构 IDC 公司声称，在今后四年，公共云计算市场的产值有望翻番，会从 2015 年的约 700 亿美元增加到 2019 年的逾 1410 亿美元。这个巨大市场为已在云计算领域确立了领头羊地位的那些公司提供了大好机会。

整理多家公司的财务报告和市场调研结果后列出了这份十大云计算公司榜单。虽然许多科技公司提供与云相关的产品和服务，但本文还是侧重于基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）厂商。

Tips:

这个排名是 2016-08-08 的结果（来自 IT 圈子）。

每家公司按估计的云计算收入排序。然而，许多厂商没有将云计算的销售数字与其他销售数字分开来，所以这份排名只是个大概，不是很精确。

值得关注的是，所有这些公司在最近一个季度的报告中发现云计算销售出现了显著增长。事实上，的几家公司报告云计算收入出现了两位数、甚至三位数的增长。

1. 亚马逊网络服务

亚马逊网络服务（AWS）是云计算领域无可争议的市场领头羊。据亚马逊的最新季度财务报告显示，在截至 2016 年 6 月 30 的那个季度中，AWS 创造收入达 29 亿美元，比去年同一季度的 18 亿美元有了大幅增长。

该公司提供一系列全面的 IaaS 和 PaaS 服务。其中最有名的服务包括：弹性计算云（EC2）、Elastic Beanstalk、简单存储服务（S3）、弹性块存储（EBS）、Glacier 存储、关系型数据库服务（RDS）和 DynamoDB NoSQL 数据库。它还提供与这几个方面有关的云服务：网络、数据分析、机器学习、物联网、移动服务、开发、云管理、云安全及更多方面务。

相关链接：<https://aws.amazon.com/cn/?tag=vig-20>

2. 微软 Azure

想搞清楚微软从云计算创造了多少收入要困难一点。在 2016 年第四季度财报中，公司表示，“智能云”（Intelligent Cloud）的收入增长 5%，达到 67 亿美元。这样一来，微软的云业务似乎比亚马逊还庞大，只不过微软把非常可观的服

务器收入连同 Azure 云计算服务一并算入到“智能云”这个类别。公司之前表示，Azure 业务的年化收入为 100 亿美元，即每个季度收入约 25 亿美元。大多数市场分析师把微软 Azure 列在第二位，仅次于亚马逊。

除了 Azure IaaS 和 PaaS 产品外，微软还有几款 SaaS 产品，包括 Office 365 产品、在线版的 Dynamics 系列企业软件和在线开发工具。

相关链接：<https://www.microsoft.com/en-us/>

3. IBM

虽然 IBM 并不始终被认为是“三大”云计算公司之一，但它的云计算业务一直势头强劲。在最近一个季度的报告中，IBM 表示，“云计算即服务的收入猛增 50%”，年化收入为 67 亿美元。

IBM 最知名的云服务就是 Bluemix PaaS，这项服务主要面向企业开发团队。该公司还有许多基于 SaaS 模式的企业软件，还销售云基础设施、云管理工具和云托管服务。

相关链接：<http://www.ibm.com/us-en/>

4. 谷歌云平台

谷歌没有将云计算财报数字单独列出来，所以很难知道其云平台到底创造了多少收入。在最近的报告中，Synergy 估计，谷歌在 IaaS 和 PaaS 市场名列第四位，在这个市场占有 4% 的份额。报告还特别指出，谷歌的云计算收入在迅速攀升，2015 年同比激增 108%。

与亚马逊和微软一样，谷歌也提供阵容非常齐全的 IaaS 和 PaaS 服务，涵盖计算、存储、网络、大数据、机器学习、开发工具和安全。它的一些最知名的云服务包括：计算引擎（Compute Engine）、应用程序引擎（App Engine）、容器引擎（Container Engine）、云存储（Cloud Storage）和 BigQuery。

相关链接：<https://www.google.com>

5. Salesforce.com

不像这份榜单上迄今为止介绍的其他公司，Salesforce.com 所赚的钱几乎全部来自云计算。就 2017 年财年第一季度而言，公司报告公认会计准则（GAAP）收入达 19.2 亿美元，同比增长 27%。

Salesforce 的收入大部分来自其 SaaS 产品，这分为三大类：销售云（Sales Cloud）、服务云（Service Cloud）和营销云（Marketing Cloud）。其中销售云最大，占 Salesforce 的云计算收入的 41%。该公司还有面向开发人员的三种 PaaS 产品：App Cloud、Force.com 和 Heroku。

相关链接：<https://www.salesforce.com>

6. Adobe

如果你提及各大云计算公司，Adobe 可能不是第一个出现在脑海的公司，可是与 Salesforce.com 一样，Adobe 的大笔收入来自 SaaS 产品。就 2016 年第二季度而言，公司报告，创意云（Creative Cloud）产品的收入达 7.55 亿美元，营销

云 (Marketing Cloud) 的收入为 3.85 亿美元, 文档云 (Document Cloud) 的收入为 1.88 亿美元。云计算创造的季度收入合计 13 亿美元。

过去, Adobe 主要销售面向创意和营销专业人员的桌面版软件, 而在 2013 年, 它改用了完全基于云计算的模式。其创意云产品中最知名的部分包括: Photoshop 图片编辑、Illustrator 矢量图形、InDesign 页面布局和 Premiere Pro 视频编辑等软件。

相关链接: <http://www.adobe.com/?sdid=Y2KRH14K>

7. 甲骨文云

甲骨文要赶上市场领头羊还有很长的一段路要走, 不过它的云计算产品看到了强劲增长的势头。就 2016 年财年第四季度而言, 甲骨文报告, 总的云收入为 8.59 亿美元, 猛增 49%。SaaS 和 PaaS 的销售表现尤为抢眼, 猛增 66%, 达到了 6.9 亿美元。

甲骨文宣称其公共云是“业界覆盖面最广、集成度最高的”。它的 SaaS 应用系统包括: 人力资源云 (HR Cloud)、客户体验云 (CX Cloud)、企业资源规划云 (ERP Cloud)、企业绩效管理云 (EPM Cloud)、供应链管理云 (SCM Cloud) 和分析云 (Analytics Cloud)。

相关链接: <https://www.oracle.com/index.html>

8. SAP

企业软件行业巨头 SAP 的云计算产品也迎来强劲增长的势态。在最近的财务报表中, 该公司报告云订阅和支持收入合计 7.2 亿欧元, 合计 8.05 亿美元。据该公司声称, 这意味着增长了 30%。

除了 HANA 云平台 PaaS 和 HANA 企业云 IaaS 外, SAP 还提供一系列广泛的 SaaS 解决方案。据公司官方网站声称, 它拥有 1.1 亿云用户。

相关链接: <http://go.sap.com/index.html>

9. Rackspace

Rackspace 的收入几乎全部来自云计算, 不过其商业模式与这份榜单上的其他云计算公司有一点不同。虽然它确实在其数据中心提供专门托管服务, 但也为在其他知名公共云上运行工作负载的公司提供支持和服务。它号称是世界上最大的托管云服务提供商。

在最近一个季度的报告中, Rackspace 宣布总收入达 5.18 亿美元, 比去年同一季度增长 7.9%。该公司预计第二个季度收入为 5.19 亿美元到 5.24 亿美元之间。

相关链接: <https://www.rackspace.com/en-us>

10. Workday

Workday 是这份名单上另一家让人稍微有点惊讶的公司, 它提供基于云计算的商业软件。在截至 2016 年 4 月 30 日的一个季度, 该公司报告收入达 2.8 亿美元, 同比猛增 39%。总收入达到 3.454 亿美元。

Workday 的最畅销产品是面向财务团队和人力资源团队的 SaaS 解决方案。

它还提供服务自动化、企业规划和高等教育解决方案。

相关链接: <http://www.workday.com/ap/>

参考文献:

2016 年十大云计算公司新鲜出炉

<http://mp.weixin.qq.com/s/yqfzSsnu-eLLgAd6oRqmCQ>