目　录

1.简历模板

1.1 软件工程师简历

个人资料			
姓　　名	*****	性　　别	男
年　　龄	21	籍　　贯	湖北武汉
现 居 地	上海	政治面貌	党员
联系电话	1234567890	E - MAIL	12345@163.com
求职意向			
目标职能	J2EE 软件工程师	工作性质	全职
薪资待遇	面议	到岗时间	随时
教育背景			
毕业院校	**职业技术学院		
专业	计算机应用技术		
工作经验			
工作时间	2016/07-----2018/07		
公司名称	***信息科技有限公司		
工作职责	java 软件工程师		
专业技能			
Java 技术			
	熟练掌握 java 面向对象编程的思想。 熟练使用 Hibernate、Struts、及 Spring 框架进行 MVC 模式应用的开发。 掌握 JSP、Servlet、JDBC、Java Bean 等 J2EE 核心技术。 熟练运用 MyEclipse 开发工具，掌握 Tomcat 6.0 等 Web 容器的配置以及部署。		
数据库			
	熟悉 Oracle、Mysql 等数据库操作。 能使用视图、事物、存储过程增加数据的安全性，提高数据存取的效率。 熟悉数据库设计规范，有一定的数据库设计经验。		
其他			
	1. 熟悉 HTML、DIV+CSS、JavaScript、XML 等 Web 开发技术。		

	2. 具有一定的软件设计及文档编写能力，代码编写规范，具有较强的学习和沟通能力。

项目经验

武汉运捷酒店管理系统	
开发周期	2017 年 3 月—2017 年 6 月
软件环境	Windows XP　Oracle 11g　JDK1.6
开发工具	MyEclipse，Dreamweaver
开发人数	6 人
项目描述	武汉运捷酒店是一家集住宿、点餐、娱乐于一体的时尚精品商务酒店，为更好的对酒店进行管理，同时能够大大节省酒店的人力资源，特委托本公司为其量身打造一款软件。 项目主要包括： 房间预订：信息录入、资料调出、取消预订 宾客入住：入住登记、房态图 业务管理：加床退床、宾客转房、用户留言 客房收费：退房结算、住客赔偿 员工管理：员工信息、添加功能 统计报表：收银数据、客房数据、历史数据 系统管理：用户管理、更改密码、系统设置
运用技术	该项目基于 J2EE 平台，B/S 模式开发。采用 Struts, Hibernate, Spring, Ajax, xml 等技术,采用分层模式开发，降低了程序间的耦合性。 系统采用 MVC 设计模式,并把 spring 的 AOP 思想合理的使用在日志处理和 系统异常处理,使用 Dom4j 进行 xml 解析，使用 Log4j 实现日志管理。 表示层使用 jsp+javascript 实现页面特效，DIV+CSS 样式表使页面布局更加规范。 Web 层使用 Struts 框架，结合 EL 表达式，使用 AJAX 技术实现无刷新提交。 控制层扩展 Struts 框架的 Action 设计理念，同时使用 Spring 整合来管理所有的 Action。 DAO 层使用抽象工厂模式和外观模式，同时使用 Spring 提供的 Hibernate 模板和 Hibernate API 访问持久层，这样可以进一步解决，在系统查询模块处大量使用正规表达式进行查询操作。 持久层使用轻量级框架技术 Hibernate 来实现 ORM 处理，同时使用 Spring 容器来管理持久层，通过 Spring 提供的 HibernateTemplate 和 Hiberante API 可以很好的解决代码的重复冗余问题。
责任描述	负责宾客入住部分的代码编写
武汉花店联盟销售网	
开发周期	2016 年 9 月—2016 年 11 月

软件环境	Windows XP　Mysql　JDK1.6
开发工具	MyEclipse，Dreamweaver，Tomcate 6.0
开发人数	4 人
项目描述	武汉市花店联盟电子商务有限公司以经营鲜花礼品电子商务网站为主,承诺本城 2 小时送货,全国 24 小时到货,为增加业务量,特委托我公司对其旗下鲜花礼品网重新开发.项目具体如下: 一．店面管理 1.前台管理：网站主页商品显示、用户注册/登陆/找回密码、查询/修改个人信息、商品搜索、订单管理、买家留言 2.商品浏览：对各种不同种类的商品信息的分类浏览 3.商品检索：对网店内所售商品的查询；可通过商品名、商品类型、花语等进行查询 4.购物车的实现：增加、修改、删除购物车中的数据以及提交购物车的数据生成订单号 二．后台管理 1.客户信息管理：通过管理员身份可以对系统的所有用户进行修改、删除、查看等操作，同时可以根据不同的信息查看用户的详细信息 2.鲜花信息管理：通过管理员的身份可以对网店的所有鲜花进行增删改查的操作 3.订单信息管理：通过管理员的身份可以对订单的信息进行处理，如修改状态等
运用技术	该项目基于 J2EE 平台，B/S 模式开发。采用 Struts, Hibernate, Spring, Ajax, xml 等技术，采用分层模式开发，降低了程序间的耦合性 表示层使用 jsp+javascript 实现页面特效，DIV+CSS 样式表使页面布局更加规范。 控制层扩展 Struts 框架的 Action 设计理念，同时使用 Spring 整合来管理所有的 Action 数据层使用 Spring 提供的 Hibernate 模板和 Hibernate API 访问持久层。 使用正则表达式验证用户、校验业务数据。
责任描述	负责前台管理、商品检索部分的代码编写
麦德利航空公司武汉锻件厂工资结算与考勤系统	
开发周期	2017 年 1 月—2017 年 4 月
软件环境	Windows XP　Tomcat6.0　JDK1.6
开发工具	Visual Studio 2005、SQLServer 2008
开发人数	4 人
项目描述	登录。分角色登录。（员工和管理员）。新进员工进行注册 并在登录的过程中可以找回密码和密码重置的功能。而密码重置主要是根据邮箱的绑定来动态找回。 考勤管理。（员工登录）员工照相打卡、员工信息的录入、员工照片的管理和照片的对

	比。 产品信息管理。（管理员登录）对产品进行增删改查。 用户信息管理。（管理员和员工登录）管理员登录可以查看所有用户的信息。并可以添加新进的员工。员工登录可以清楚的了解员工的自己的详细信息。但是没有修改的权限。 工资结算。根据已统计的每个月的每个员工的考勤以及每个员工每天的工作量对每个员工进行工资结算。
运用技术	1.本系统前台页面使用 DIV+CSS 设计页面布局，使用 JavaScript 制作页面特效、AJAX 技术实现无刷新提交，提高用户体验。 2.使用 JS 实现了弹出对话框、滚动字幕、按钮渐变、自动换背景等页面特效。 3.使用 EXT 制作进度条，下拉菜单、树形结构、数据全选、批量删除、文件上传功 能。 4.使用了 PS\FLASH 提升了页面显示效果。 5.使用 T-SQL 语句进行数据查询、通过视图、事务、多表联合查询等方式实现数据的操作。
责任描述	主要负责产品信息管理的模块。（管理员登录）对商品的进行增删改查的库存管理。商品进入时对商品数量进行添加。以及自动录入商品的入库时间。商品的库存不够时、及时的提醒管理员应该进货。当商品已用完或是已经不再需要的时候则对商品进行删除。以及每个月对商品的出库和入库的情况生产一个报表。清楚的了解每件商品的出入库和了解每个商品的成本价。以便于后面的工资结算。

1.2 网页设计师简历

简　历			
个人信息			
姓　名：	****	性　别：	男
出生日期：	1997 年 10 月 1 日	居 住 地：	上海-徐汇区
工作年限：	一年以上	电子邮件：	1234567@163.com
手　机：	1234567890		
自我评价			
本人具有良好的编程技巧和编码风格，熟悉网站制作；工作认真负责，积极主动，能吃苦耐劳，抗压力能力极强，能够适应加班和出差；有较强的组织能力和团队协作精神，具有较强的适应能力；纪律性强，			

工作积极配合；有较强的学习能力和接受新事物的能力；具有一定的审美观。

求职意向

到岗时间：　随时

工作性质：　全职

希望行业：　计计算机软件；互联网/电子商务

目标地点：　上海

期望薪水：　面议/月

目标职能：　网页设计/制作/美工

工作经验

2016/11--2017/7：武汉市**网络有限公司（少于 50 人）

所属行业：　互联网/电子商务

开发部　前台 WEB 工程师

为了提升对所学专业的技能，学校特安我们排进入武汉市新领域网络公司进行实习。加强我们的专业技能的工作经验。

项目经验

2017/3 -- 2017/6 ：长沙市古玩竞价拍卖系统

软件环境：　WindowsXP + Dreamweaver 8+ photoshop + MyEclipse8.5

硬件环境：　电脑

开发工具：　Dreamweaver 8 + MyEclipse8.5

项目描述：　竞价拍卖系统是一个 C2C 模式下的电子商务系统，为买方和卖方提供了网上拍卖市场，拍卖的方式出售自己的产品。

详细功能如下：

前台：

1)首页：用户通过首页可以执行发布商品、对商品出价、登录、注册、浏览个人历史竞拍等操作。

2)用户登录：在发布新商品或对商品竞价前，必须先登录。如果用户忘记密码，可以点击忘记密码超链接，系统将弹出取回密码对话框。用户输入会员名和邮箱地址，密码即会自动发送到指定邮箱中。

3)用户注册：只有会员注册成功后，才能发布新商品并对商品出价。

4)发布新商品：用户在首页上单击卖东西按钮，系统将弹出选择发布方式的对话框。

5)商品出价：商品出价是指买方对卖方参与竞拍的商品出价，以促使该商品成交。

6)我的竞拍：用户可以查询自己卖出、买进、竞拍、结束竞拍商品的信息。

后台：

1)管理用户：可以对用户进行管理，如锁定、解锁和删除等。

2)管理短消息：用户可以管理系统所有短消息。

3)商品类别管理：用户可以管理商品类别，如修改分类、删除分类、添加分类和分类级别设定等。

项目技术描述：

使用 div+css 进行网页布局；

使用 JavaScript 制作动态效果；

使用正则表达式进行表单验证；

使用 jsp+servlet+javabean 搭建三层架构模式；

使用 SQL Server 数据库操作数据

责任描述：　本人主要负责首页、用户登录、用户注册页面的设计与实现，以及特效的设计与实现。

2016/11 -- 2017/2 ：新领域网络公司游戏点卡销售充值系统

软件环境：　WindowsXP + Dreamweaver 8+ photoshop + MyEclipse8.5

硬件环境：　电脑

开发工具：　Dreamweaver 8 + MyEclipse8.5

项目描述：　游戏点卡销售系统是一个 C2C 模式下的电子商务系统，为买方和卖方提供了网上购买点卡平台，玩家可以以拍卖的方式出售自己的奖品。

详细功能如下：

1).游戏点卡销售网站首页：网站 logo 要求定位在页面其他元素之上，根据浏览器窗口大小自适应自身位置，页面中间实现广告图片轮换效果。

2).游戏点卡分类查询页面 ：卡片介绍 DIV 大小根据内容自适应，设定最小高度 。

3).用户登录页面：用户名和密码用 Javacript 验证是否为空，实现提交按钮的动态效果。

4).购物车页面：计算购物车中商品的总金额。

5).个人信息管理页面：实现点击右侧链接时在右侧框架中显示链接内容。

6).我们钱包页面：显示账户余额以及相关提示信息。

7).用户购物记录页面：显示订单信息列表，每次只显示一个订单的详细信息。

8).新用户注册页面：验证表单数据的合法性，实现即时错误提示特效；用户输入数据时，出 输入提示。

9).修改个人资料页面：输入元素只定义下边框样式，实现提交按钮动态效果。

10).后台管理页面：潜入框架用户显示链接页面的内容，右侧导航菜单高度自适应，背景 y 轴 方向平铺。实现点击左侧链接时在右侧框架中显示链接内容。

11).账户充值页面：显示账户信息列表，点击“充值”链接时弹出一个窗口供管理员

	输入数据。实现提交按钮的动态效果。 项目技术描述： 使用 div+css 进行网页布局； 使用 JavaScript 制作动态效果； 使用正则表达式进行表单验证； 使用 SQL Server 数据库操作数据
责任描述：	负责前端首页、点卡查询页面、注册页面、购物记录页面的设计与实现，以及特效的设计与实现。

教育经历			
2014/9 -- 2016/6	**职业技术学院	计算机应用技术	大

个人技能
1．熟练掌握了 DIV+CSS 布局的 HTML 代码编写，兼容主流浏览器，对 W3C 网页标准有较深理解。 2．熟练掌握了 JavaScript 特效以及表单验证的制作。 3．熟练使用 photoshop、dreamweaver 等设计软件。 4．熟悉在 MVC 设计模式进行 JSP 应用程序页面的开发。 5．熟悉 jquery 框架和 SSH 框架，ajax 提高用户体验。 6．对 Struts、Hibernate、Spring 框架有了一定了解。 7．对 SQL Server、MySQL、Oracle 数据库的设计和操作有了一定了解。

1.3 软件测试工程师简历

个人信息

姓　　名：	XXXXX	性　　别：	男	
年　　龄：	22	学　　历：	本科	
联系电话：		电子邮箱：		
毕业院校：	XXXX	专　　业：	计算机信息管理	

自我评价

在工作能力方面。学生工作在大学生活中是一项重要内容，在能力锻炼方面，我积极参加学生工作和社会实践活动，并取得了一定的成绩。我曾担任班长，在工作过程中我认真负责，积极工作，极大地锻炼了我的工作能力和交际能力，积累了许多宝贵的实践经验。获得“学生工作积极分子”、“优秀学生干部”称号

求职意向

工作性质：	全职
目标地点：	北京
到岗时间：	
期望薪水：	面议
目标职位：	软件测试工程师

个人技能

熟悉软件测试基本理论、熟悉测试流程。

熟练掌握软件测试用例的设计方法，例如：等价划分、边界值、错误推断、因果法等

熟练使用测试用例管理工具 Testlink。

熟悉 Bugfree 缺陷管理工具。

熟悉自动化测试工具 selenium、QTP，以及 LoadRunner 性能测试工具。

熟悉 Visio、Project、office 等项目管理软件，具备一定的文档编写能力。

熟悉操作 SQL Server2008 数据库，以及 SQL 语句。

了解 Java 基础，有一定的项目开发经验（这句话最好加上）。

项目经验

锐气实业（公司主页管理平台）

测试工具：TestLink，BugFree

软件环境：windows7、SQLSERVER2008、IIS7.0

测试周期：2014-05-12——2014-06-10

项目描述：

随着互联网技术的发展，利用强大的互联网技术所实施的应用也越来越多。各类中小型公司，希望在网络上推广自己的公司创建自己公司的个性化主页，从而达到网络营销的效果，但实际公司主页的开发成本偏高。因此本公司开发出一套成熟的公司主页创建及维护的成熟解决方案。

主要功能：

网站管理平台

平台登录：输入账号密码及验证码，通过验证登录系统对主页网站进行维护。

系统管理：

管理员管理

对于系统管理员的信息进行管理。

管理员录入

添加新的管理员信息

角色管理

设置管理员的权限，规范管理员对于管理平台的操作。

网站信息设置

关键字设置

设置主页的中文关键字，利于百度、google 等搜索引擎的抓取。

底部版权信息

设置主页的中文底部版权信息。

关于我们

设置中文关于我们的页面。

图片列表

Banner 列表

滚动横幅图片列表显示、修改及删除。

Banner 录入

滚动横幅图片的添加。

新闻中心

新闻列表

对于新闻的列表显示、修改及删除。

新闻录入

新闻的添加功能。

新闻类别

对于新闻的分类进行管理。

产品管理

产品列表

对于公司产品的信息进行列表显示、修改及删除。 产品添加 添加公司产品信息。 产品类型 对于产品分类进行管理。 案例管理 案例列表 对于成功案例信息进行列表显示、修改及删除。 案例添加 对于成功案例的添加。 案例类型 对于成功案例的类型进行管理。 公司主页 前台页面，根据管理平台的设置，将设置的信息生成完整的公司主页并发布展示。
项目职责： 组织测试组成员对于该项目的需求进行分析，对于需求进行测试并对需求中不合理的地方提出问题。 根据项目需求模块，分配组员编写对应模块的测试用例（本人负责，登录及系统管理模块） 组织测试组成员对于编写好的案例进行统一评审，使测试案例达到可实施性。 分人员按照测试案例进行测试，并将测的案例登录 TestLink 进行管理。 测试期间出现的 bug 登录 Bugfree 分配给相应的开发人员进行修改，并跟踪 bug 的修改情况。 测试结束时，对于测试的文档进行收集，并对测试工作进行总结

备注：

1） 以上的三类简历只能作为参考简历模板，可以参考内容，大家去制作自己模板的时候不要一模一样，要有自己的风格，会给面试官留下较好的初印象；

2） 除了企业里的工作经验，其在学校里的主要经历、社会经历也是可以加到简历中的。

2.面试前准备

2.1 面试前准备了解企业

面试前首先通过询问、网上查找等途径了解企业信息，包括：公司背景、主要产品、组织构架、主要业务、主要领导人、企业文化和信条等；接着要详细了解职位要求，特别是技术要求；最后要确认面试流程及面试官。

2.2 面试前准备确定路线

1．输入：登陆 http://map.baidu.com/，点击公交

2. 输入起点和终点，可适当参考下拉菜单。

3. 得出相关路线指引后，点击发送到手机，收取短信。

4. 确定路线之后预算乘车时间，做好时间安排，保证可以准时到达，最好能够提前 20 分钟到达。

2.3 面试前准备：用品、着装

面试前整理用品，保证资料证件齐全，包括：身份证（及身份证复印件）、1 寸免冠照片（4 张以上）、简历、笔记本（或便签纸）、黑色水性笔、文凭（及文凭复印件，学历及技能证书，视具体情况而定） 共 6 件物品。

面试前整理好着装，正装、整洁大方，具体可以参考以下标准：

面试着装 （男性）

1.短发，清洁、整齐，不要太新潮；

2.精神饱满，面带微笑；

3.每天刮胡须，饭后洁牙；

4.白色或单色衬衫，领口、袖口无污迹；

5.领带紧贴领口，系得美观大方（颜色、长短、领带夹）

6.西装平整，有裤线；

7.西装口袋不放物品（笔）；

8.短指甲，保持清洁；

9.皮鞋光亮，深色袜子；

10.全身三种颜色以内；

面试着装（女性）：

1. 发型文雅、庄重，梳理整齐，长发要用发夹夹好，不扎马尾巴；

2. 化淡妆，面带微笑；

3. 着正装套装，大方、得体；

4. 指甲不宜过长，并保持清洁。涂指甲油时须自然色；

5. 裙子长度适宜；

6. 肤色丝袜，无破洞（备用袜）；

7. 写字光亮、清洁；

8. 全身三种颜色以内。

备注：以上着装根据不同公司进行不同服装定位！**凡事预则立，不预则废！**

3.面试中注意事项

3.1 自我介绍

- 介绍内容要与个人简历一致
- 表达方式上尽量口语化
- 要切中要害，不谈无关、无用的内容，清楚简单地说出自己的背景和所申请的工作之间的联系是什么
- **［参考回答 1］**我叫×××，来自湖南长沙。我一直都比较喜欢计算机，所以在 2016 年开始学习**计算机应用技术专业**。我为这样的职位已经准备两年了。这两年的学习过程中曾在公司有工作经验，在项目组做过 3 个重要项目。在做项目的过程中，我发现我自己特别喜欢开发工作，使我自己很满足。我相信在这里工作我的能力会得到发挥，我也相信可以为贵公司贡献自己的力量。
- **［参考回答 2］**我叫×××，来自湖北武汉。今天来应聘贵公司的软件工程师这一职位，该职位需要的技术正是我所具备的。我个人比较喜欢在软件开发方向发展（懂得的技术特长），我也曾参加过公司的真实项目开发，做起来比较顺手。申请贵公司主要考虑到公司从事的业务方向和我以前工作过的公司比较相似，给我一种很熟悉的感觉，我应该能很好的融入这样的氛围。
- 注：以上回答均根据简历中 IT 技能一栏于（懂得的技术特长）处增加自己技术特长的阐述

3.2 对加班的看法

- IT 企业中，技术人员的工作很多时候是跟着项目走的，因此阶段性的加班是非常正常的事，因此企业要考察求职者是否能够承受加班。
- 说明自己工作中会是高效的，不会把应在工作时间内完成的任务拖到休息时间。
- 如果是工作需要，特别是项目进度的要求，是非常乐意加班的。
- 最好还能说明自己可以加班的条件，比如目前尚无家室或家庭比较支持，可以抽出较多时间来工作。

［参考回答］从我个人的角度讲，我是很愿意加班的，第一说明我们公司的业务需求大，发展前景好，才会加班。而且我现在单身，能够在加班的时间不断提升自己技能还能和同事增加相处的时间培养感情，这对于一个年轻人来说，是更宝贵的财富。所以，我愿意加班。

3.3 介绍一下你做过的项目

列举做过的项目数量，以 2-3 个为宜，其中详细介绍 1-2 个，然后举例说明。

3.4 问题不会回答时

- **缓兵之计**

您的意思是……?(进一步确认)

不知道您看出来没有，我太重视这次面试了，所以有些紧张，您能给我一分钟时间让我重新考虑一下这个问题好吗？

- **转移话题**

➢坦白地讲，我对这个问题不太了解。但我对××问题倒是有一些研究，我可以讲讲这方面的内容吗？或者说，我对这个技术不太了解，但是我在做项目的过程中，经常运用类似的技术，我可以讲讲这方面的知识吗？

给出解决问题的方法的所涉及的知识点（虽然不知道怎么解决，你会按照什么思路解决）

➢**放慢语速**

放慢语速的同时，思考对方提出的问题，进一步思考。

4.常规人力面试题

1. 请你自我介绍一下

思路：你好，我叫***，非常高兴能有机会来贵公司面试。大三时在软件公司实习了一年，期间参与开发了 3 个项目，包括管理系统项目和网站开发的项目。在技术方面我主要熟悉 JAVA、JSP、SSH，数据库主要熟悉 SQL 和 oracle，可以完成前台页面和后台业务逻辑功能，今天来贵公司面试，也希望能有一个好的结果。

2. 谈谈你做的项目（部分示范）

OA 项目

OA 项目主要是针企业的自动化办公,提高办公效率，减少人事理成本的项目类型。本项

目主要分为六大块：人事管理，日程管理，文档管理，消息传递，系统管理，考勤管理

人事管理：主要实现对机构部门员工增删改查，打印员工信息等操作.

日程管理：实现了公司部门及个人日程的管理.运用一个简单的日历显示日程，让人一目了然，有阳历转换阴历， 适合大众使用。

文档管理：实现了对公司文档增删改查，及上传下载文件等操作.

消息传递：实现了公司内部人员相互通信，及即使通报工作会议等操作.有即时消息通知。

系统管理：该功能主要用于管理者本身，可以对员工身份权限管理，日志等，进行更好的管理。

考勤管理：主要用于考察员工出勤情况，并统计员工出勤率。

酒店管理系统

武汉**商务酒店是一家集住宿、点餐、娱乐于一体的商务酒店，为更好的对酒店进行管理，同时能够大大节省酒店的人力资源，特委托本公司为其量身打造一款软件。项目主要包括：

一、房间预订：信息录入、资料调出、取消预订

二、宾客入住：入住登记、房态图

三、业务管理：加床退床、宾客转房、用户留言

四、客房收费：退房结算、住客赔偿

五、员工管理：员工信息、添加功能

六、统计报表：收银数据、客房数据、历史数据

七、系统管理：用户管理、更改密码、系统设置

电子商务网站项目

武汉市鲜花联盟电子商务有限公司以经营鲜花礼品电子商务网站为主,承诺本城 2 小时送货,全国 24 小时到货,为增加业务量,特委托我公司对其旗下鲜花礼品网重新开发。项目具体如下:

一．店面管理

1.前台管理：网站主页商品显示、用户注册/登陆/找回密码、查询/修改个人信息、商品搜索、订单管理、买家留言

2.商品浏览：对各种不同种类的商品信息的分类浏览

3.商品检索：对网店内所售商品的查询；可通过商品名、商品类型、花语等进行查询

4.购物车的实现：增加、修改、删除购物车中的数据以及提交购物车的数据生成订单号

二．后台管理

1.客户信息管理：通过管理员身份可以对系统的所有用户进行修改、删除、查看等操作，同时可以根据不同的信息查看用户的详细信息

2.鲜花信息管理：通过管理员的身份可以对网店的所有鲜花进行增删改查的操作

3.订单信息管理：通过管理员的身份可以对订单的信息进行处理，如修改状态等

3. 你有什么业余爱好?

思路：1、业余爱好能在一定程度上反映应聘者的的性格、观念、心态，这是招聘单位问该问题的主要原因。2、最好不要说自己没有业余爱好。3、不要说自己有那些庸俗的、令人感觉不好的业余爱好。4、最好不要说自己仅限于读书、听音乐、上网，否则可能令面试官怀疑应聘者性格孤僻。5、最好能有一些户外的业余爱好来点缀你的形象。

篮球、羽毛球、乒乓球、徒步登山、攀爬登山、游泳、钓鱼、摄影写生等

4. 你最崇拜谁?

思路：1、最崇拜的人能在一定程度上反映应聘者的性格、观念、心态，这是面试官问该问题的主要原因。2、不宜说自己谁都不崇拜。3、不宜说崇拜自己。4、不宜说崇拜一个虚幻的、或是不知名的人。5、不宜说崇拜一个明显具有负面形象的人。6、所崇拜的人最好与自己所应聘的工作能搭上关系。7、最好说出自己所崇拜的人的那些品质、哪些思想鼓舞着自己。

李彦宏、李想，我很崇拜他们的大胆创新精神，同时很佩服他们的博学和不断挑战自我的精神。

5. 你的座右铭是什么?

思路：1、座右铭能在一定程度上反映应聘者的性格、观念、心态，这是面试官问这个问题的主要原因。2、不宜说容易引起不好联想的座右铭。3、不宜说太抽象的座右铭。4、不宜说太长的座右铭。5、座右铭最好能反映出自己的某种优秀品质。

1、只要地球没有停止转动，我就不会停止奋斗。2、知识是掌控未来的权利。3、在哪摔了，就在那哭完了再爬起来。4、自己选择的路，跪着也要走完。

6. 谈谈你的缺点

思路：1、不亦说自己没有缺点。2、不宜把那些明显的优点说成缺点。3、不宜说出严重影响所应聘工作的缺点。4、不宜说出令人不放心、不舒服的缺点。5、说出一些对于所应聘工作无关紧要的缺点，甚至是一些表面上看是缺点，从工作的角度看却是优点的缺点.

1、我不太善于过多的交际，尤其是和陌生人交往有一定的难度。这虽然是缺点，但是说明你交友慎重； 2、我办事比较死板，有时容易和人较真。这虽然是缺点，但是说明你比较遵守单位既定的工作规范，有一定的原则性； 3、我什么知识或专业都想学，什么也没学精。这虽然是缺点，但是说明你比较爱学习，知识面比较广； 4、我对社会上新兴的生活方式或流行的东西接受比较慢。这虽然是缺点，但是说明你比较传统，不盲目跟随潮流； 5、我对我认为不对的人或事，容易提出不同意见，导致经常得罪人。这虽然是缺点，但是说明你比较有主见，有一定的原则性； 6、我办事比较急，准确性有时不够。这虽然是缺点，但是说明你完成工作速度较快； 7、对自己从事工作存在的困难，自己琢磨的多，向同事或领导请教的少。这虽然是缺点，但是说明你独立完成工作任务的能力较强

7. 谈一谈你的一次失败的经历

思路：1、不宜说自己没有失败的经历。2、不宜把那些明显的成功说成失败。3、不宜说出严重影响所应聘工作的失败经历。4、所谈经历的结果应该是失败的。5、宜说明失败之前自己曾信心百倍，尽心尽力。6、说明仅仅是由于外在原因导致失败。7、失败后自己很快振作起来，以更加饱满

的热情面对以后的工作。

人只要不放弃，有信心，就可能战胜挫折取得成功。面对失败，我们要更有毅力。例如：一次暑期工的时候，因为没有经验而失去了一次机会等（避重就轻的回答这个问题，先谈你面对失败的态度，再谈失败的例子）

8. 你最大的成就是什么？为什么？

思路：1、主考官问这样的问题是在考察应聘者的价值观，应聘者回答时要透露自己的判断标准和崇尚的观点。2、回答忌讳缺乏有价值的内容、毫无特别之处；回答不要太空泛，应找出自己经历中的亮点作为事例讲给主考官听。

1、我觉得学校里我最大的收获是结交了很多非常好的朋友，建立了非常好的人际关系。2、一次兼职获得的成就感。3、第一次做出的项目模块等

9. 你怎样影响其他人接受你的看法？

思路：1、判断一个人的人际沟通能力。

参考回对于好的想法，甚至是伟大的想法，人们有时并不接受。我现在认识到这样一个事实，那就是你表达想法的方式同想法本身一样重要。当我试图影响别人时，我一般会假设自己处在他们的位置上，让自己从他们的角度来看待问题。然后我就能够以一种更可能成功的方式向他们陈述我的想法。

10. 你的好友怎样评价你？

思路：1、通过过这个问题可以了解求职者的个性。2、有关的故事听起来也必须真实，如果不真实就不要使用因为这不会奏效。

参考回答 1：我的朋友对我很重要。在与朋友的交往中，最重要的是，彼此之间有相互依赖的感觉。我们都很忙，并不经常见面，但在我可以称之为亲密朋友的几个人中，我们都知道，大家可以相互依赖。

参考回答 2：我的朋友都说我是一个可以信赖的人，因为我一旦答应别人的事情，我一定会做到；如果我做不到，就不会轻易许诺。

11. 你为什么选择我们公司？

思路：1、面试官试图从中了解你求职的动机、愿望以及对此项工作的态度。2、建议从行业、企业和岗位三个角度来回答。

我十分看好贵公司所在的行业，我认为贵公司十分重视人才，而且这项工作很适合我，相信自己一定能做好。

12. 你在找工作时最看重的是什么？为什么？

思路： 通过这个开放性的问题，面试人可以了解你的关注重点，通过这个关注点又可以反映出你的理性思考能力。1、一定要表明自己对未来工作的看法，说明哪些方面能给自己带来最大程度的满足，这是回答这个问题的关键，当然回答问题的方法也同样重要。2、回答问题尽量简洁，但要实现三个目的：突出了求职者的技能；表明了求职者明白个人与企业的关系；同时也说明求职者理解变化与发展的重要性。

参考回我希望找到的工作能发挥我的长处，比如（说出具体技能）我认为还有一件事也很重要，那就是我在企业中的作用要与企业目标联系在一起。如果工作中偶尔有些挑战，让我超越自己目前的技能水平，那就再好不过了。

13. 在高薪、表彰和晋升之间，你认为哪种形式最有价值？

思路：1、这个问题是个陷阱。如果你顺着提问者的思路回答，那你就必败无疑。如果你选择金钱优于晋升，那就会被看成世俗和短见；相反，如果你认为金钱不重要，那你就会因为对金钱表现出的冷漠而被看做不真诚。2、回答要诚实，而且也不世俗。它反映了求职者的抱负，也反映了求职者对未来成功的合理推测。

参考回对我来说，这些东西都是紧密联系、不可分离的。尽管我对金钱并不着迷，但我认为，随着成功、晋升以及表彰的出现，它们一定也会给我带来更多的金钱回报。

14. 对这项工作，你有哪些可预见的困难？

思路：1、不宜直接说出具体的困难，否则可能令对方怀疑应聘者不行。2、可以尝试迂回战术，说出应聘者对困难所持有的态度，工作中出现一些困难是正常的，也是难免的，但是只要有坚韧不拔的毅力、良好的合作精神以及事前周密而充分的准备，任何困难都是可以克服的。

15. 如果我录用你，你将怎样开展工作？

思路：1、如果应聘者对于应聘的职位缺乏足够的了解，最好不要直接说出自己开展工作的具体办法。2、可以采用迂回战术来回答，如首先听取领导的指示和要求，然后就有关情况进行了解和熟悉，接下来制定一份近期的工作计划并报领导批准，最后根据计划开展工作。

16. 如果可以在企业内自主选择工作，你想选择什么样的工作?

思路：1、这个问题旨在了解求职者是否了解现代职场的复杂性。它的目的是考查求职者是否清楚，面试人要招聘的是能为企业做出一系列贡献的人。招聘者认为招聘到的人应该具有相应的技能，他（她）的责任应该非常清楚，他（她）的潜力也应该非常大。

参考回首先我希望找到的工作能够发挥我的特长和技能，具体地说（提到一些具体技能）我还希望自己的工作能够得到企业的认可，也就是说，这份工作对实现企业目标却是很重要。如果有一定的发展空间或者有多样化的可能，那份工作就更理想了。

17. 我们为什么要录用你?

思路：1、应聘者最好站在招聘单位的角度来回答。2、招聘单位一般会录用这样的应聘者：基本符合条件、对这份工作感兴趣、有足够的信心。3、如我符合贵公司的招聘条件，凭我目前掌握的技能、高度的责任感和良好的适应能力及学习能力，完全能胜任这份工作。我十分希望能为贵公司服务，如果贵公司给我这个机会，我一定能成为贵公司的栋梁！

18. 你认为自己在什么样的条件下工作最有效?

思路：1、此问题考察的是应聘者对工作条件的要求。2、主考官可以从中获取应聘者的工作方式、影响工作效率的因素等信息，还可以知道应聘者的工作方式、影响工作效率的因素等信息，还可以知道应聘者的不足在哪里。3、如果回不管在什么条件下，我都会努力把工作做到最好，这样并不是十分妥当，至少有喊口号的嫌疑，显得不够成熟。为了对自己负责，可以具体谈谈自己期望的工作条件。

19. 你能为我们做什么？你能给公司带来什么?

思路：1、基本原则上投其所好。2、回答这个问题前应聘者最好能先发制人，了解招聘单位期待这个职位所能发挥的作用。3、应聘者可以根据自己的了解，结合自己在专业领域的优势来回答这个问题。

参考回我可以做一个优秀的员工在组织中发挥能力，给组织带来高效率和更多的收益。

20. 你是应届毕业生，缺乏经验，如何能胜任这项工作?

思路：1、如果招聘单位对应聘者提出这个问题，说明招聘单位并不在乎经验，关键看应聘者怎样回答。2、这个问题真正的回答最好要体现出应聘者的诚恳、机智、果断及敬业。

我也发现，实际工作远比书本知识丰富、复杂。但我有较强的责任心、适应能力和学习能力，而且比较勤奋，所以在兼职中均能圆满完成各项工作，从中获取的经验也令我受益匪浅。请贵公司

放心，学校所学及兼职的工作经验也让我一定能胜任这个职位。

21. 与上级意见不一时，你将怎么办？

思路：1、这是一个陷阱问题；2 判断应聘者的沟通能力及执行力；

我绝不会和上级在公开场合发生激烈的争执。我和领导可能会存在意见不统一的时候，但是不会发生争执。如果我们的意见不一，我会和领导在一个私下的、只有 2 个人的场合进行诚挚的沟通，我会把问题事先想清楚，并表达出自己的看法，如果没有说服领导，我会坚持按照领导的指示执行，因为领导考虑问题的层次往往比下属高，所以意见不一时，很有可能是我错了，所以我要坚持执行。

22. 你希望与什么样的上级共事？

思路：1、通过应聘者对上级的希望可以判断出应聘者对自我要求的意识，这既是一个陷阱又是一次机会。2、最好回避对上级具体的希望，多谈对自己的要求。

参考回作为步入社会的新人，我应该多要求自己尽快熟悉环境、适应环境，而不应该对环境提出什么要求，只能发挥我的专长就可以了。

23. 上下级之间应该怎样交往？

思路：1、通过这个问题可以了解求职者在企业等级结构中的沟通方式。2、通过这一问题的回答，求职者可以展示自己在复杂领域工作的技能水平，求职者理解人际关系的复杂性以及多样性，求职者明确地表达了高效沟通技能的重要性，同时也显示了自己在这方面的自信。

我认为，能在企业各个层面上清楚地进行交流，这对企业的生存至关重要。我认为自己已经在这个方面培养了很强的能力。从上下级关系来说，我认为最重要的是应该意识到每个人以及每种关系都是不同的。对我来说最好的方式就是始终不带任何成见地来对待这种关系的发展。

24. 你和同事们怎么相处？

思路：1、通过这个问题以及前面上下级关系和朋友关系的问题，面试人可以对求职者的有效沟通技能得出一个总体印象。2、与同事的有效沟通技能将减少面试人的担心，避免他（她）认为你是又一个不合适的人选。3、回答需表明求职者的心理是稳定的，且具有很高的人际协调能力。4、通过将问题和人格区分开来，求职者表明过去在人际关系方面也很成功。

参考回我一般都能与同事相处的很好。当然有时候也可能会和某人发生冲突。这时，我一般会注意寻找冲突的根源，而不是转移到与对方的攻击上。我发现这种方法非常有效，它可以使我同任何人都维持一种相互尊重的关系。另外，通过过这样做，我往往都能解决问题，甚至会促进与同事

的关系。

25. 你在前一家公司离职的原因是什么?

思路：1、最重要的是：应聘者要使招聘单位相信，应聘者在过往单位的离职原因在此家招聘单位不存在。2、避免把离职原因说的太详细、太具体。3、不能掺杂主观的负面感受。4、但也不能躲闪、回避，如个人原因等。5、不能涉及自己负面的人格特征，如不诚实、缺乏责任感等。6、尽量使解释的理由为应聘者个人形象添彩。7、如我离职时因为这家公司倒闭。我在公司工作了三年多，有较深的感情。从去年开始，由于市场形式突变，公司的局面急转直下。到眼下这一步我觉得很遗憾，但还要面对现实，重新寻找能发挥我能力的舞台。同一个面试问题并非只有一个答案，而同一个答案并不是在任何面试场合都有效，关键在于应聘者掌握了规律后，对面试的具体情况进行把握，有意识地揣摩面试官提出问题的心理背景，然后投其所好。

26. 为了实现自己的目标你会怎样努力工作?

思路：1、面试人希望通过这个问题来确认求职者是否是一个努力工作的人。2、回答这个问题的关键在于，你一定要显示出自己履行责任的意愿和能力。它表明求职者有无限的能量，而且对工作也非常投入。它还表明，求职者解决问题是为了能更好地利用他（她）的资源这才是这个问题饿实质所在。

参考回对我来说，如何努力工作，不是问题。我的做事原则是，如果我制订了一个目标或者被分配了一项重要任务，我就会尽我所能地努力工作，实现预期目标。所以对我来说，重要的是怎样出色地完成工作，也就是说，怎样工作才能尽可能简单和顺利地完成任务，这样我就可以把精力转移到其他事情上。

27. 就你申请的这个职位，你认为你还欠缺什么?

思路：1、他们想问求职者的弱点，但精明求职者一般不直接回答。

回答建议：继续重复自己的优势，然后说：对于这个职位和我的能力来说，我相信自己是可以胜任的，只是缺乏经验，这个问题我想我可以进入公司以后以最短的时间来解决，我的学习能力很强，我相信可以很快融入公司的企业文化，进入工作状态。

28. 你期望的工资是多少?

思路：1、一般这样的企业工资水平是很灵活的，一般中小企业有很多仍以个人能力，面试评价作为议薪的标准。但这个问题却不能正面回答。2、依公司规定的回答是不被建议的，这样不但表示出自己对于工作的自信程度不高，在薪资无法符合个人要求时更会造成许多困扰。

参考回以我的能力和我的优势，我完全可以胜任这个职位，我相信我可以做的很好但是贵公司对这个职位的描述不是很具体，我想还可以延后再讨论。

29. 出于工作晋升的考虑你打算继续深造吗？

思路：1、用来衡量你的雄心，也可以判断企业对你的重视程度是否会影响你对自己未来的重视程度。

参考回作为一名刚毕业的学生，我学到了很多知识。如果有合适的机会，我当然会考虑继续深造。但是，我会认真考虑这件事情，我觉得很多人回学校学习是很盲目的。如果我发现自己所做的工作确实有价值，而且也需要获得更多的教育才能在这一领域做的出色，我当然会毫不犹豫地去学习。

30. 你是否认为大学的学习成绩能决定你在企业的成功程度？

思路：1、如果求职者在学校成绩很好，面试人希望通过这个问题让求职者知道，工作上的成功与学习上的成功并不一样。2、如果求职者在学校成绩不佳，面试人希望通过这个问题了解到，求职者是否认为自己解决问题的能力有所欠缺。

参考回（对那些成绩不佳的人）我认为有能力取得好成绩是很重要的。如果一个人在每个科目上成绩都不佳的话，那就会让人非常担心。然而，并非所有人都能在每一个科目上取得优异成绩。对我来说，重要的是在个人学习成绩中要有一些突出的地方，因为这些地方代表着一个人的潜力。

参考回（对于那些成绩较好的人）虽然规划学习生涯不会像管理高难度工作那么复杂，但是我认为两者之间存在着联系。我认为，取得优异的学习成绩的最大意义是它可以反映一个人追求卓越的决心。3、回答都以第三人称的形式进行了有效表达，从而避免使自己看起来过于谦卑或者过于傲慢。

31. 经过这次面试，我们认为你不合适，决定不录用你，你自己认为有哪些原因？

思路：1、该题的出题意图在于考察应试者的应变能力，并非真的对你不满，要沉着应对；不要中了圈套，暴露自己的弱点，回答时可以虚一点；重点要放在弥补弱点上，这可以看出你积极进取的品质；要诚恳地向考官讨教，可以博取他们的好感。

参考回

第一、我认为面试向来是 5 分靠实力，5 分靠运气的。我们不能指望一次面试就能对一个人的才能、品格有充分的了解。毕竟二十分钟时间太短，而要打分的项目很多；

第二、当然最大的原因可能是我的临场发挥不能得到各位的认可，比如我的心里的确感到紧张。吃一堑长一智，所以无论面试的结果如何，我的这段经历将为我提供一个自我审查的机会，发现自

己的不足，其中既有临场经验的不足，希望以后能有机会向各位考官讨教；

第三、我会好好总结经验教训，加强学习，弥补不足。另外，希望考官能对我进行全面、客观的考察，我一定会努力，使自己尽量适应岗位的要求。

32. 你认为你所受的教育对你生活的最大意义是什么？

思路：该题的出题意图在于考察应试者真实的教育程度及生活态度。

有意义，这样可以让人懂得一些做人的道理。如何去跟人相处，如何和人沟通。让人懂得很多道理。

33. 过去三年里，你为自我发展订立了什么样的目标？为什么要订立那样的目标？

思路：该题的出题意图在于考察应试者的职业规划。任何工作优秀的应试者都是那些不断更新自己知识和技能的人。自我发展是每个人自己的事，而不是老板要求去做的事。那些主动自我学习的人，是那些想不断提高自己的人。

1、可以谈一下工作、技术、生活等目标，切勿好高骛远，制定的目标要切实可行。

34. 现在软件规模越来越庞大，开发人数越来越多，形成以团队开发，提出团队精神，请说出你认为团队精神的本质和意义。

思路：该题的出题意图在于考察应试者的团队精神及团队配合能力。

凝聚力、编码的规范化、一个人的力量是有限的等。

35. 介绍一下你的课外活动。你为什么愿意从事那些课外活动？通过那些课外活动，你都学了些什么？

思路：该题的出题意图在于变向的考察应试者在学习中，是否有团队精神及很好的组织能力。

野炊、爬山、趣味运动会等。

36. 你为什么选择软件行业，而不选择其他行业呢？

思路：该题的出题意图在于考察应试者对于 IT 行业的关注度及了解度。

我个人对计算机比较感兴趣，但选择这个行业，我身边的家人朋友对我影响也很大。我的两个表哥都是从事这个行业的，做开发岗位三年多了。IT 行业是薪水高的行业，掌握了高科技技术，对于以后的人生道路也是一种保障。个人的兴趣爱好加上家人朋友帮我对就业前景的客观分析，我选

择了 IT 行业，并希望通过自己的努力在这个行业越做越好。

5.常规技术面试题（数据库）

1. 触发器的作用?

触发器是一个特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以联级运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

2. 什么是存储过程？用什么来调用?

存储过程是一个预编译的 SQL 语句，优点是允许模块化的设计，就是说只需创建一次，以后在程序中就可以调用多次。如果某次操作需要执行多次 SQL，使用存储过程比单纯 SQL 语句执行要快。可以用一个“execute 存储过程名 参数”命令来调用存储过程。

3. 索引的作用？和它的优点缺点是什么？。

索引就一种特殊的查询表，数据库的搜索引擎可以利用它加速对数据的检索。它很类似与现实生活中书的目录，不需要查询整本书内容就可以找到想要的数据。索引可以是唯一的，创建索引允许指定单个列或者是多个列。缺点是它减慢了数据录入的速度，同时也增加了数据库的尺寸大小。

4. 主键和索引的区别?

1.主键时为了标识数据库记录唯一性,不允许记录重复,且键值不能为空,主键也是一个特殊索引;

2.数据表中只允许有一个主键,但是可以有多个索引;

3.使用主键数据库会自动创建主索引,也可以在非主键上创建索引,方便查询效率;

4.索引可以提高查询速度,它就相当于字典的目录,可以通过它很快查询到想要的结果,而不需要进行全表扫描;

5.唯一索引则标识该索引值唯一,可以由一个或几个字段组成,一个表可以有多个唯一索引。

5. 什么是内存泄漏?

一般我们所说的内存泄漏指的是堆内存的泄漏。当应用程序用关键字 new 等创建对象时，就从堆中为它分配一块内存，使用完后由于某种原因程序未释放或无法释放，造成系统内存的浪费。导致程序运行速度减慢甚至系统崩溃等严重后果。

6. 维护数据库的完整性和一致性，你喜欢用触发器还是自写业务逻辑？为什么？

我是这样做的，尽可能使用约束，如 check,主键，外键，非空字段等来约束，这样做效率最高，也最方便。其次是使用触发器，这种方法可以保证，无论什么业务系统访问数据库都可以保证数据的完整新和一致性。最后考虑的是自写业务逻辑，但这样做麻烦，编程复杂，效率低下。

7. 什么是事务？

事务就是被绑定在一起作为一个逻辑工作单元的 SQL 语句组，如果任何一个语句操作失败那么整个操作就被失败，进而回滚到操作前状态，或者是上个节点。为了确保要么执行，要么不执行，就可以使用事务。要将一组语句作为事务考虑，就需要通过 ACID 测试，即原子性，一致性，隔离性和持久性。

8. 什么是锁？

在所有的 DBMS 中，锁是实现事务的关键，锁可以保证事务的完整性和并发性。与现实生活中锁一样，它可以使某些数据的拥有者，在某段时间内不能使用某些数据或数据结构。当然锁还分级别的。

9. 什么叫视图？

视图是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，视图通常是由一个表或者多个表的行或列的子集。它使得我们获取数据更容易，相比多表查询。

10. 视图创建和使用语法？

1.首先判断是否存在

```
if exists (select * from sysobjects where name = '视图名')
drop view View_EdsProd
Go
```

2. 创建视图

```
create view View_EdsProd as select * from Tab_EdsProd where Mid>1（条件） go
```

3. 使用视图

```
select *from View_EdsProd
```

11. 游标是什么？

游标是 SQL 的一种数据访问机制。可以将游标简单的看成是查询的结果集的一个指针，可以根据需要在结果集上面来回滚动，浏览需要的数据。

12. 你能向我简要叙述一下 SQL Server 中使用的一些数据库对象吗?

表、索引、视图、存储过程、触发器、用户定义函数、数据库关系图、全文索引。

13. NULL 是什么意思?

NULL(空)这个值表示 UNKNOWN(未知):它不表示“”(空字符串)。不能把任何值与一个 UNKNOWN 值进行比较，都会生产一个 NULL 值。您必须使用 IS NULL 操作符。

14. 什么是索引，有哪些索引，具体怎么用?

索引是与表或视图关联的磁盘上结构，可以加快从表或视图中检索行的速度。索引包含由表或视图中的一列或多列生成的键。这些键存储在一个结构（B 树）中，使 SQL Server 可以快速有效地查找与键值关联的行。

索引分为聚集索引和非聚集索引。

在数据库系统中建立索引主要有以下作用：

（1）快速取数据；

（2）保证数据记录的唯一性；

（3）实现表与表之间的参照完整性；

（4）在使用 ORDER by、group by 子句进行数据检索时，利用索引可以减少排序和分组的时间。

15. SQL Server 里有什么类型的索引?

在 SQL Server 里，它们有两种形式:聚集索引和非聚集索引。聚集索引在索引的叶级保存数据。每个表格只会有一个聚集索引。非聚集索引在索引的叶级有一个行标识符。每个表格有多个非聚集索引。

16. 什么是主键?

主键是表格里的(一个或多个)字段，只用来定义表格里的行;主键里的值总是唯一的。

17. 什么是外键?

外键是一个用来建立两个表格之间关系的约束。这种关系一般都涉及一个表格里的主键字段与另外一个表格(可能是同一个表格)里的一系列相连的字段。那么这些相连的字段就是外键。

18. 什么是触发器?

触发器是一种专用类型的存储过程，它被捆绑到 SQL Server 的表格或者视图上。

19. SQL Server 有什么不同类型的触发器?

有 INSTEAD-OF 和 AFTER 两种触发器。例如，如果有一个用于 TableA 的 INSTEAD-OF-UPDATE 触发器，同时对这个表格执行更新语句，那么 INSTEAD-OF-UPDATE 触发器里的代码会执行，而不是执行更新语句则不会执行操作。AFTER 触发器要在 DML 语句在数据库里使用之后才执行。这些类型的触发器对于监视发生在数据库表格里的数据变化十分好用。

20. 您如何确保一个带有名为 Fld1 字段的 TableB 表格里只具有 Fld1 字段里的那些值，而这些值同时在名为 TableA 的表格的 Fld1 字段里?

第一个答案是使用外键限制。外键限制用来维护引用的完整性。它被用来确保表格里的字段只允许有已经在另一表格里的某个字段里定义了的值。通常是另外一个表格的主键。

另外一种答案是触发器。触发器可以被用来保证以另外一种方式实现与限制相同的作用，但是它非常难设置与维护，而且性能一般都很糟糕。

21. 对一个投入使用的在线事务处理表格有过多索引需要有什么样的性能考虑?

对一个表格的索引越多，数据库引擎用来更新、插入或者删除数据所需要的时间就越多，因为在数据操控发生的时候索引也必须要维护。

22. 你可以用什么来确保表格里的字段只接受特定范围里的值?

可以使用 Check 约束，它在数据库表格里定义，用来限制输入该列的值。

触发器也可以被用来限制数据库表格里的字段能够接受的值，但是这种办法要求触发器在表格里被定义，可能会在某些情况下影响到性能。

23. 概述存储过程及其优缺点。

存储过程是一个预编译的 sql 语句 ，编译后可多次使用

优势：响应时间上来说有优势，可以给我们带来运行效率提高的好处，且使用存储过程的系统更加稳定

缺点：维护性较差，相对于简单 sql，存储过程并没有什么优势，并且在进行调试时比较困难

24. 什么是相关子查询?如何使用这些查询?

相关子查询是一种包含子查询的特殊类型的查询。查询里包含的子查询会请求外部查询的值，从而形成一个类似于循环的状况。

25. 什么是 SQL 注入式攻击?

所谓 SQL 注入式攻击，就是攻击者把 SQL 命令插入到 Web 表单的输入域或页面请求的查询字

符串，欺骗服务器执行恶意的 SQL 命令。比如：攻击者在用户名字和密码输入框中输入"'或'1'='1"之类的内容。最后得到的 SQL 命令可能变成：SELECT * from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'。这时，已经不能真正验证用户身份，所以系统会错误地授权给攻击者。

26. 如何防范 SQL 注入式攻击?

只要在利用表单输入的内容构造 SQL 命令之前，把所有输入内容过滤一番就可以了。过滤输入内容可以按多种方式进行。

⑴ 对于动态构造 SQL 查询的场合，可以使用下面的技术：

第一：替换单引号，即把所有单独出现的单引号改成两个单引号，防止攻击者修改 SQL 命令的含义。

第二：删除用户输入内容中的所有连字符。

第三：对于用来执行查询的数据库帐户，限制其权限。

⑵ 用存储过程来执行所有的查询。

⑶ 限制表单或查询字符串输入的长度。

⑷ 检查用户输入的合法性，确信输入的内容只包含合法的数据。

⑸ 将用户登录名称、密码等数据加密保存。

⑹ 检查提取数据的查询所返回的记录数量。

27. 默认的系统数据库有哪些?

1）master 数据库（主）；2）tempdb 数据库（临时）；3）model 数据库（模板）；4）msdb 数据库（计划任务）；

28. 默认创建一个数据库，会生成哪些文件?

1）主文件（.mdf），2）日志文件（.ldf），无次要文件（.ndf）。

29. 创建数据库时，能不能把数据文件和日志文件分开?

可以分开，起到优化作用。把数据文件放到高速读写区，把日志文件放到低速读写区。

30. 什么是索引覆盖(Index Covering)查询?

索引覆盖(Index Covering)查询是指数据可以只通过索引获取，而不用接触表。

31. 存储过程和触发器的区别?

触发器与存储过程的主要区别在于触发器的运行方式。存储过程必须有用户、应用程序或者触发器来显示的调用并执行，而触发器是当特定时间出现的时候，自动执行或者激活的，与连接用数据库中的用户、或者应用程序无关

32. 存储过程和函数的区别?

存储过程是用户定义的一系列 SQL 语句的集合,,而函数通常是数据库已定义的方法,具体区别如下:

1.对于存储过程来说可以返回参数,而函数只能返回值或者表对象.

2.函数必须有返回值,存储过程可有可无

3.存储过程一般是作为一个独立的部分来执行,而函数可以作为查询语句的一部分来调用.

33. 聚集索引和非聚集索引区别

聚集索引，数据按索引顺序存储，中子结点存储真实的物理数据

非聚集索引，存储指向真正数据行的指针

34. 索引的优缺点，什么时候使用索引，什么时候不能使用索引?

索引最大的好处是提高查询速度，

缺点是更新数据时效率低，因为要同时更新索引

对数据进行频繁查询进建立索引，如果要频繁更改数据不建议使用索引。

35. 数据库的优化

1.创建适当的索引

2.对 sql 语句优化

使用 exists 或 not exists 代替 in 或 not in

使用存储过程

用 union 替换 or（适用于索引列）

where 代替 having,having 检索完所有记录，才进行过滤

使用 select top 或 set rowcount 来限制操作的行

避免嵌套查询

对多个字段进行等值查询时，联合索引

36. 数据库的主从复制

默认异步复制，容易造成主库数据和从库不一致

一个数据库为 Master,一个数据库为 slave,通过 Binlog 日志来实现

slave 两个线程，一个线程去读 master binlog 日志，写到自己的中继日志

一个线程解析日志，执行 sql

master 启动一个线程，给 slave 传递 binlog 日志

半同步复制

只有把 master 发送的 binlog 日志写到 slave 的中继日志，这时主库才返回操作完成的反馈，性能有一定降低

并行复制

slave 多个线程去请求 binlog 日志

37. long_query 怎么解决

设置参数，开启慢日志功能，得到耗时超过一定时间的 sql

(1)slow_query_log 这句是开启记录慢查询功能，slow_query_log=0 关闭；slow_query_log=1 开启（这个 1 可以不写）

(2)long_query_time = 1 这句是记录超过 1 秒的 SQL 执行语句

38. varchar 和 char 的使用场景

用来存储字符

varchar 适用字符长度经常变的

char 适用字符长度固定的

39. 数据库连接池的作用

维护一定数量的连接，减少创建连接的时间

更快的响应时间

统一的管理

40. 分库分表，主从复制，读写分离

读写分离，读从库，写主库

spring 配置两个数据库，通过 AOP（面向切面编程），在写或读方法前面进行判断得到动态切换数据源。

41. 数据库三范式

1NF 属性不可分

2NF 非主键属性，完全依赖于主键属性

3NF 非主键属性无传递依赖

42. 数据库中 join 的 inner join, outer join, cross join

以 A，B 两张表为例 A left join B

选出 A 的所有记录，B 表中没有的以 null 代替

right join 同理

inner join

A,B 的所有记录都选出，没有的记录以 null 代替

cross join (笛卡尔积)

A 中的每一条记录和 B 中的每一条记录生成一条记录

例如 A 中有 4 条，B 中有 4 条，cross join 就有 16 条记录

43. 有哪些锁,select 时怎么加排它锁

乐观锁，悲观锁，排它锁，共享锁，更新锁，表锁，行级锁。

乐观锁：乐观锁不会锁住任何东西，也就是说，它不依赖数据库的事务机制，乐观锁完全是应用系统层面的东西。

悲观锁：悲观锁是指假设并发更新冲突会发生，所以不管冲突是否真的发生，都会使用锁机制

排它锁: 可以防止并发事务对资源进行访问。

共享锁: 允许并发事务在封闭式并发控制下读取资源。

更新锁:是共享锁和排他锁的结合。

行级锁：单独的一行记录加锁

表锁：锁住整个表，可以同时读，写不行

在 Select 语句中加 for update 是给相应的行增加排他锁。Select 出来的数据别的事务不能读取，不能修改、不能删除。

44. 死锁怎么解决

找到进程号，kill 进程

产生死锁的原因：

一是系统提供的资源数量有限，不能满足每个进程的使用；二是多道程序运行时，进程推进顺序不合理。

产生死锁的必要条件是：

1、互斥条件；

2、不可剥夺条件（不可抢占）；

3、部分分配；

4、循环等待。

根据产生死锁的四个必要条件，只要使其中之一不能成立，死锁就不会出现。为此，可以采取下列三种预防措施：

1、采用资源静态分配策略，破坏"部分分配"条件；

2、允许进程剥夺使用其他进程占有的资源，从而破坏"不可剥夺"条件；

3、采用资源有序分配法，破坏"环路"条件。

解除死锁常常采用下面两种方法：1、资源剥夺法；2、撤消进程法

45. 最左匹配原则

最左匹配原则是针对索引的

举例来说：两个字段（name,age）建立联合索引，如果 where age=12 这样的话，是没有利用到索引的，这里我们可以简单的理解为先是对 name 字段的值排序，然后对 age 的数据排序，如果直接查 age 的话，这时就没有利用到索引了，查询条件 where name=’xxx’ and age=xx 这时的话，就利用到索引了。因为创建复合索引的规则是首先会对复合索引的最左边的，也就是第一个 name 字段的数据进行排序，在第一个字段的排序基础上，然后再对后面第二个的 age 字段进行排序。其实就相当于实现了类似 order by name age 这样一种排序规则。所以：第一个 name 字段是绝对有序的，而第二字段就是无序的了。所以通常情况下，直接使用第二个 age 字段进行条件判断是用不到索引的，当然，可能会出现上面的使用 index 类型的索引。这就是所谓的为什么要强调最左匹配原则的原因。

46. SqlServer 是一种大型数据库，他的存储容量只受存储介质的限制，请问它是通过什么方式实现这种无限容量机制的。

它的所有数据都存储在数据文件中(*.dbf),所以只要文件够大,SQLServer 的存储容量是可以扩大的.

SQL Server 数据库有三种类型的文件：

主要数据文件

主要数据文件是数据库的起点，指向数据库中文件的其它部分。每个数据库都有一个主要数据文件。主要数据文件的推荐文件扩展名是 .mdf。

次要数据文件

次要数据文件包含除主要数据文件外的所有数据文件。有些数据库可能没有次要数据文件，而有些数据库则有多个次要数据文件。次要数据文件的推荐文件扩展名是 .ndf。

日志文件

日志文件包含恢复数据库所需的所有日志信息。每个数据库必须至少有一个日志文件，但可以不止一个。日志文件的推荐文件扩展名是 .ldf。

47. sqlserver 数据库中常用的聚合函数有哪些?

Max(),Avg(),Count(),Min(),Sum()。

中文：最大值，平均值，数据条数，最小值，总和。

48. 数据库主键、外键、约束、索引的作用是什么？有几种连表查询方式？

主键、外键及约束的作用：保证数据的完整性

索引的作用：索引是一个数据结构，用来快速访问数据库表格或者视图里的数据，加快数据库的搜索引擎对数据的检索效率

方式：左连接、右连接、内连接、自连接

49. 除了 sqlserver 存储过程实现分页，还有什么实现方法？

利用 select top 和 select not in 进行分页

利用 select top 和 select max(列)

利用 Row_number()给数据行加上索引

利用临时表及 Row_number

6. 常规技术面试题（.NET）

6.5　集群与分布式

1. 什么是 Zookeeper

ZooKeeper 是一个分布式的，开放源码的分布式应用程序协调服务，是 Google 的 Chubby 一个开源的实现，是 Hadoop 和 Hbase 的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

ZooKeeper 的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

ZooKeeper 包含一个简单的原语集，[1]　提供 Java 和 C 的接口。

2. 写出你对 zookeeper 的理解

随着大数据的快速发展，多机器的协调工作，避免主要机器单点故障的问题，于是就引入管理机器的一个软件，他就是 zookeeper 来协助机器正常的运行。

Zookeeper 有两个角色分别是 leader 与 follower ，其中 leader 是主节点，其他的是副节点，在安装配置上一定要注意配置奇数个的机器上，便于 zookeeper 快速切换选举其他的机器。

在其他的软件执行任务时在 zookeeper 注册时会在 zookeeper 下生成相对应的目录，以便 zookeeper 去管理机器。

3. zookeeper 的搭建过程

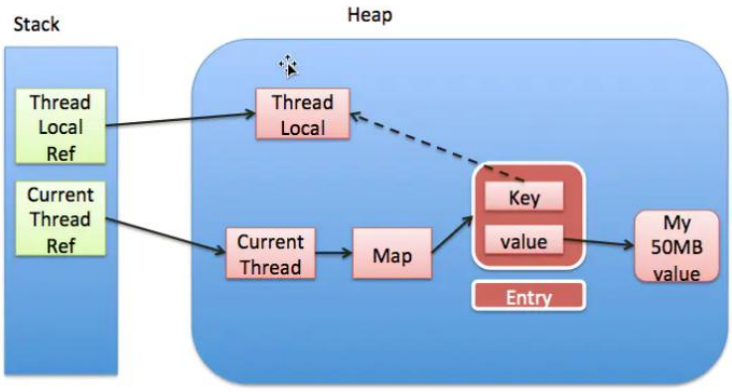
主要是配置文件 zoo.cfg 配置 dataDir 的路径一句 dataLogDir 的路径以及 myid 的配置以及 server 的配置，心跳端口与选举端口。

4. ThreadLocal 内存泄漏问题,如何防止

ThreadLocal 的作用是提供线程内的局部变量，这种变量在线程的生命周期内起作用，减少同一个线程内多个函数或者组件之间一些公共变量的传递的复杂度。但是如果滥用 ThreadLocal，就可能会导致内存泄漏。

ThreadLocal 的实现是这样的：每个 Thread 维护一个 ThreadLocalMap 映射表，这个映射表的 key 是 ThreadLocal 实例本身，value 是真正需要存储的 Object。

ThreadLocal 实现原理



ThreadLocal 内存泄漏的根源是：由于 ThreadLocalMap 的生命周期跟 Thread 一样长，如果没有手动删除对应 key 就会导致内存泄漏。综合上面的分析，我们可以理解 ThreadLocal 内存泄漏的前因后果，那么怎么避免内存泄漏呢？

防止方案：

每次使用完 ThreadLocal，都调用它的 remove()方法，清除数据。

在使用线程池的情况下，没有及时清理 ThreadLocal，不仅是内存泄漏的问题，更严重的是可能导致业务逻辑出现问题。所以，使用 ThreadLocal 就跟加锁完要解锁一样，用完就清理。

5. 简述高可用.

高可用（High Availability），是当一台服务器停止服务后，对于业务及用户毫无影响。 停止服务的原因可能由于网卡、路由器、机房、CPU 负载过高、内存溢出、自然灾害等不可预期的原因导致，在很多时候也称单点问题。

6.6 其他部分

1. 堆和栈的区别

数据结构的堆和栈 ：

1. 栈是一种可以实现“先进后出”（或者称为“后进先出”）的存储结构。

2. 堆则是一种经过排序的树形数据结构，常用来实现优先队列等

内存分配中的堆和栈

1. 栈空间的内存是由系统自动分配，一般存放局部变量，比如对象的地址等值，不需要程序员对这块内存进行管理

2. 堆空间的内存是动态分配的，一般存放对象，并且需要手动释放内存。

2. 请谈谈对正则表达式的看法?

正则表达式是一种字符规则。它是用来匹配字符串的。

打个比方，某警官学院要招生，要求：身高 170 以上，体重：70 以上，性别：男性。。。。

这些条件就是一种规则，通过这个规则，该警官学院就可以招到符合要求的一批学生。

正则表达式与上面这种规则一样，只不过它不是用来找人的，而是在程序里面用来找字符串的。

在编程的时候，我们要从一个字符串里找出特定的部分，就可以用这种规则来匹配。

3. UDP 连接和 TCP 连接的异同。

都可以实现远程通信，主要区别在于 TCP 需要保持连接而 UDP 不需要，因此 UDP 具有更高的效率和更少的资源占用，而 TCP 传输数据更加可靠.

4. 什么叫做 SQL 注入，如何防止？请举例说明。

利用现有应用程序，将(恶意)的 SQL 命令注入到后台数据库引擎执行的能力，这是 SQL 注入的标准释义。就是攻击者把 SQL 命令插入到 Web 表单的输入域或页面请求的查询字符串，欺骗服务器执行恶意的 SQL 命令。在某些表单中，用户输入的内容直接用来构造(或者影响)动态 SQL 命令，或作为存储过程的输入参数，这类表单特别容易受到 SQL 注入式攻击。

防止 SQL 注入：

1、对输入内容进行过滤，去掉有可能的威胁

2、对于用来执行查询的数据库帐户，限制其权限。

3、在查询语句中使用参数。

4、用存储过程来执行所有的查询。

5、将用户登录名称、密码等数据加密保存。

5. 请简述一下用 Socket 进行同步通讯编程的详细步骤

1.创建 socket

2.确定本地计算机端点（ip 和端口号）

3.socket 绑定端点

4.socket.receive(); 接收数据

5.socket.send();发送数据

6.socket.close();关闭 socket

6. float f=-123.567F;int i=(int)f;i 的值现在是?

-123。

7. 产生一个 int 数组，长度为 100，并向其中随机插入 1-100，并且不能重复。

```
int[] intArr=new int[100];
ArrayList myList=new ArrayList();
Random rnd=new Random();
while(myList.Count<100)
{
   int num=rnd.Next(1,101);
   if(!myList.Contains(num))
   myList.Add(num);
}
for(int i=0;i<100;i++)
intArr[i]=(int)myList[i];
```

8. 编程实现冒泡排序。

```
 int [] array = new int [*] ;
 int temp = 0 ;
 for (int i = 0 ; i < array.Length - 1 ; i++)
 {
```

```
for (int j =0 ; j < array.Length-i-1 ; j++)
{
 if (array[j] > array[j+1])
 {
  temp = array[j] ;
  array[j] = array[j+1] ;
  array[j+1] = temp ;
 }
}
```

9. 分层式结构究竟其优势何在?

1、开发人员可以只关注整个结构中的其中某一层；

2、可以很容易的用新的实现来替换原有层次的实现；

3、可以降低层与层之间的依赖；

4、有利于标准化；

5、利于各层逻辑的复用。

概括来说，分层式设计可以达至如下目的：分散关注、松散耦合、逻辑复用、标准定义。

10. 分层式结构也不可避免具有一些缺陷

1、降低了系统的性能。这是不言而喻的。如果不采用分层式结构，很多业务可以直接造访数据库，以此获取相应的数据，如今却必须通过中间层来完成。

2、有时会导致级联的修改。这种修改尤其体现在自上而下的方向。如果在表示层中需要增加一个功能，为保证其设计符合分层式结构，可能需要在相应的业务逻辑层和数据访问层中都增加相应的代码。

11. 请解释一下.NET 多层应用程序中层与层之间以哪几种方式进行数据传递。并解释你自己的项目中采用哪种方式进行。

三层架构一般指的是界面层，业务层，数据层。

界面层就是系统的操作界面，和用户直接交互的地方。

业务层又称为逻辑层，英文名称是 Business Logic Layer，简称 BLL，是执行业务逻辑的地方数据层也称为数据访问层，英文名称是 Database Access Layer，简称 DAL，这里是直接和数据库进行交互的地方，也是整个系统里唯一允许访问数据库的地方。

除此之外，还有一个业务实体层，这个层比较特殊，就是定义实体类的地方，有些人把这个层归属业务层，有些人把这个独立出来，变成一个公共层。

各个层次之间的访问关系：

界面层只能单向访问业务逻辑层，业务逻辑层只能单向访问数据层，这三层都可以访问公共模块（公共层）。

12. 对三层架构的理解。

答: 三层架构(3-tier application) 通常意义上的三层架构就是将整个业务应用划分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。区分层次的目的即为了”高内聚，低耦合”的思想。

1、表现层（UI）：通俗讲就是展现给用户的界面，即用户在使用一个系统的时候他的所见所得。

2、业务逻辑层（BLL）：针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。

3、数据访问层（DAL）：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等。

13. MVC 与三层架构比最主要的优势是什么?

三层是: UI 界面层

BLL 业务逻辑层

DAL 数据访问层

（特殊的 Model 实体层）

MVC 是：

M Model(模型层) 业务逻辑以及数据库的交互

V View(视图层) 显示数据和提交数据

C Controller(控制器) 负责从视图读取数据，控制用户输入，并向模型发送数据。

两者的区别

三层的 UI = MVC 的 View

三层的 Model+ BLL+DAL = MVC 的 Model

MVC 优点

易于进行单元测试

易于进行测试驱动开发

低耦合性、高重用性、可适用性

较低的生命周期成本

快速的部署

可维护性

有利于软件工程化管理

MVC 缺点

内部结构复杂

14. MVC 的生命周期是怎样的?

1) Request 请求到来

2) IIS 根据请求特征将处理权移交给 ASP.NET

3) UrlRoutingModule 将当前请求在 Route Table 中进行匹配

4) UrlRoutingModule 在 RouteCollection 中查找 Request 匹配的 RouteHandler,默认是 MvcRouteHandler MvcRouteHandler 创建 MvcHandler 实例.

5) MvcHandler 执行 ProcessRequest.

6) MvcHandler 使用 IControllerFactory 获得实现了 IController 接口的实例,找到对应的 HomeController

7) 根据 Request 触发 HomeController 的 Index 方法

8) Index 将执行结果存放在 ViewData

9) HomeController 的 Index 方法返回 ActionResult

10) Views/Home/Index.aspx 将 ViewData 呈现在页面上

11) Index.aspx 执行 ProcessRequest 方法

12) Index.aspx 执行 Render 方法 输出到客户端

15. 启用一个线程是用 run()还是 start()?

start()

7.常规技术面试题（Java）

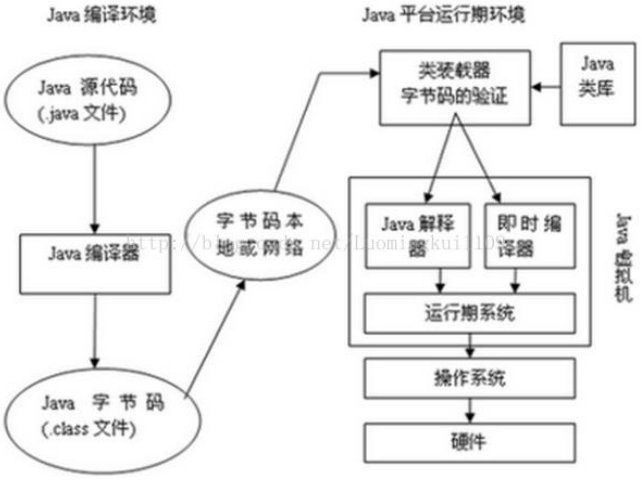
7.1 Java 基础部分

1. Java 的 “一次编写,处处运行”如何实现?

JAVA 之所以能实现 一次编译，到处运行，是因为 JAVA 在每个系统平台上都有 JAVA 虚拟机（JVM），JAVA 编译的中间文件 class 是由 JAVA 虚拟机在运行时动态转换为对应平台的机器代码.

2. 描述 JVM 运行原理。

Java 平台由 Java 虚拟机和 Java 应用程序接口搭建，Java 语言则是进入这个平台的通道，用 Java 语言编写并编译的程序可以运行在这个平台上。这个平台的结构如下图所示： 运行期环境代表着 Java 平台，开发人员编写 Java 代码(.java 文件)，然后将之编译成字节码(.class 文件)，再然后字节码被装入内存，一旦字节码进入虚拟机，它就会被解释器解释执行，或者是被即时代码发生器有选择的转换成机器码执行。



3. 为什么 Java 没有全局变量?

Global variables(全局变量) 是指可以全局访问的变量， Java 不支持全局变量，原因如下：

1. 全局变量破坏了引用的透明性。

2. 全局变量制造了命名空间冲突。

可以使用 properties 类将想要全局有效的变量值写在 properties 文件中,那么在何处用时都从此 properties 文件中读取这个变量的值就可以了,此值在任何时候都可以修改的

说明：我们平时在类中声明的只是相对类而言是全局变量，不是真正意义的全局变量

4. 说明一下 public static void main(String args[])这段声明里每个关键字的作用。

Public 是一个访问权限（访问修饰符）公共

static 修饰的成员称为类成员或者静态成员

void 当方法定义时用 void 修饰时，表示没有返回值

String 类来创建和操作字符串

5. Java 是否存在内存泄漏?

java 中内存泄露的发生场景，通俗地说，就是程序员可能创建了一个对象，以后一直不再使用这个对象，这个对象却一直被引用，即这个对象无用但是却无法被垃圾回收器回收的，这就是 java 中的内存泄露，一定要让程序将各种分支情况都完整执行到程序结束，然后看某个对象是否被使用过，如果没有，则才能判定这个对象属于内存泄露。

6. ==与 equals 的区别。

"= ="除了比较基本数据之外都是比较的内存地址

"equals"除了没有没有重写 equals 方法的类之外都是比较的内容

7. Java 中有几种类型的流?

Java 中的流分为两种，一种是字节流，另一种是字符流，分别由四个抽象类来表示（每种流包括输入和输出两种所以一共四个）:InputStream，OutputStream，Reader，Writer。

8. 如何用 Java 代码列出一个目录下所有的文件。

```
File file=new File("H:\\");
for(File temp:file.listFiles()){
    if(temp.isFile()){
        System.out.println(temp.toString());
    }
}
```

9. &和&&的区别。

1、&与&&都可以用作逻辑与的运算符，当两边的结果都为 true 时，运算结果才为 true，否则只要有一方运算结果是 false，运算结果就为 false。

2、&&在开发过程中用到的概率比&大，因为&&具有短路的功能，只要第一个表达式是 false，

就不会再进行判断。例如：if(str!=null && !str.equals("")){}

当 str==null 时，后面的表达式就不会继续执行，但是

if(str!=null & !str.equals("")){}当 str==null 时，程序会报空指针异常，因为&不具有短路的功能。

3、&同时也是位运算符，就是我们通常所说的按位与运算，当&的左右两边都是 Boolean 型表达式时或 Boolean 值时，就执行按位与运算。

10. 构造器（constructor）是否可被重写（override），其规范是什么?

构造器(构造方法)Constructor 不能被继承，因此不能重写 Override，但可以被重载 Overload（不同参数即可）。

每一个类必须有自己的构造函数，在创建对象时自动调用，如果添加有参构造函数后，默认无参构造函数则被覆盖。子类不会覆盖父类的构造函数，但是在创建子类对象的时候，会自动调用父类构造函数。

11. JAVA 的反射机制的原理。

JAVA 反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法；这种动态获取的信息以及动态调用对象的方法的功能称为 java 语言的反射机制。

Java 反射机制主要提供了以下功能： 在运行时判断任意一个对象所属的类；在运行时构造任意一个类的对象；在运行时判断任意一个类所具有的成员变量和方法；在运行时调用任意一个对象的方法；生成动态代理。

12. 静态嵌套类(Static Nested Class)和内部类(Inner Class)的不同?

内部类是类中类（内部类不为同一包的其他类可见，具有很好的封装性），分为静态内部类，成员内部类，局部内部类以及匿名内部类；局部内部类写在方法里面；用到最多的就是匿名内部类。

1. 静态的内部类称为嵌套类，嵌套类不能直接引用外部类的 non-static 属性和方法，创建嵌套类对象时不依赖外部类对象；

2. 静态内部类没有了指向外部的引用，和 C++的嵌套类很相像了，Java 内部类和 C++嵌套类最大的不同在于是否具有指向外部引用这点；

3. 在任何非静态内部类中，都不能有静态数据、静态方法或者又一个静态内部类（也就是不止一层），然后静态内部类可以拥有这一切。

13. 如何将 String 类型转化成 Number 类型。

Integer 类的 valueOf 方法可以将 String 转成 Number

14. 什么是值传递和引用传递?

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。

15. Java 的访问修饰符是什么?

在 Java 编程语言中有四种权限访问控制符，这四种访问权限的控制符能够控制类中成员的可见性。

public 是公共的，被 public 所修饰的成员可以在任何类中都能被访问到。

protected 是受保护的，受到该类所在的包所保护。

friendly 是友好的，即在成员的前面不写任何的访问修饰符的时候，默认就是友好的。所谓友好的，是对同一 package 的类友好。

private 是私有的，即只能在当前类中被访问到，它的作用域最小。

16. Java 基础数据类型有哪些?

byte(字节)　short(短整型)　int(整型)　long(长整型)　float(浮点型)　double(双精度)　char(字符型)　boolean(布尔型)

17. hashCode()和 equals()方法的重要性体现在什么地方?

Java 中的 HashMap 使用 hashCode()和 equals()方法来确定键值对的索引，当根据键获取值的时候也会用到这两个方法。如果没有正确的实现这两个方法，两个不同的键可能会有相同的 hash 值，因此，可能会被集合认为是相等的。而且，这两个方法也用来发现重复元素。所以这两个方法的实现对 HashMap 的精确性和正确性是至关重要的。

18. switch 中用于判断的表达式，可以用哪些数据类型?

1. int

2.char

3.byte

4.short

5.枚举

6. String: PS:对 JDK 版本有要求，必须为 1.7 及以上版本

19. char 型变量中能不能存贮一个中文汉字?为什么?

Java 里采用了 Unicode 编码格式，Unicode 编码中一个 char 型占用 2 个字节，而一个汉字也是占用 2 个字节，所以可以存储中文汉字。

备注：C 语言里，char 占用 1 个字节，不用存汉字。

20. 静态变量和实例变量的区别?

在语法定义上的区别：静态变量前要加 static 关键字，而实例变量前则不加。

在程序运行时的区别：实例变量属于某个对象的属性，必须创建了实例对象，其中的实例变量才会被分配空间，才能使用这个实例变量。静态变量不属于某个实例对象，而是属于类，所以也称为类变量，不用创建任何实例对象，静态变量就会被分配空间，静态变量就可以被使用了。总之，实例变量必须创建对象后才可以通过这个对象来使用，静态变量则可以直接使用类名来引用。

21. 是否可以从一个 static 方法内部发出对非 static 方法的调用?

不可以。因为非 static 方法是要与对象关联在一起的，必须创建一个对象后，才可以在该对象上进行方法调用，而 static 方法调用时不需要创建对象，可以直接调用。也就是说，当一个 static 方法被调用时，可能还没有创建任何实例对象，如果从一个 static 方法中发出对非 static 方法的调用，那个非 static 方法是关联到哪个对象上的呢？这个逻辑无法成立，所以，一个 static 方法内部发出对非 static 方法的调用。

22. Integer 与 int 的区别?

1、int 是基本数据类型，Integer 是包装类；

2、int 的默认值是 0，Interger 的默认值是 null；

23. Overload 和 Override 的区别。参数列表相同，返回值不同的方法，是否是重载方法?

Overload 是重载的意思，Override 是覆盖的意思，也就是重写。

重载是指在一个类里，方法名相同，参数不同；

重写是指子类继承父类，子类里重新编写了父类中的同名（同参数）方法，也就是覆盖了父类的方法；

不是！因为重载必须要求参数列表不同！

24. 接口是否可继承接口？抽象类是否可实现(implements)接口？抽象类是否可继承具体类(concrete class)?

接口可以继承多个接口。抽象类可以实现(implements)接口，抽象类是可继承具体类。

备注：只要明白了接口和抽象类的本质和作用，这些问题都很好回答。

只有记住抽象类与普通类的唯一区别就是不能创建实例对象和允许有 abstract 方法。

25. 面向对象有哪三大特性?

1、面向对象有三大特性，分别是：封装、继承和多态。

2、封装：面向对象的封装就是把描述一个对象的属性和行为的代码封装在一个类中，有些属性是不希望公开的，或者说被其他对象访问的，所以我们使用 private 修饰该属性，使其隐藏起来；类中提供了方法（用 public 修饰），常用的是 get、set 方法，可以操作这些被隐藏的属性，其他类可以通过调用这些方法，改变隐藏属性的值！

封装是保证软件部件具有优良的模块性的基础，封装的目标就是要实现软件部件的“高内聚、低耦合”，防止程序相互依赖性而带来的变动影响。在面向对象的编程语言中，对象是封装的最基本单位，面向对象的封装比传统语言的封装更为清晰、更为有力。

3、继承：在定义和实现一个类的时候，可以在一个已经存在的类的基础之上来进行，使用 extends 关键字实现继承；子类中可以加入若干新的内容，或修改原来的方法使之更适合特殊的需要，这就是继承。继承是子类自动共享父类数据和方法的机制，这是类之间的一种关系，提高了软件的可重用性和可扩展性。

4、多态：多态就是在声明时使用父类，在实现或调用时使用具体的子类；即不修改程序代码就可以改变程序运行时所绑定的具体代码，让程序可以选择多个运行状态，这就是多态性，多态增强了软件的灵活性和扩展性。这里可以举个例子，比如声明时使用的是动物类，调用时传递的是一个猫类（动物类的子类）的对象，具体执行父类里动物——吃的方法时，实际执行的是猫——吃的方法。

26. abstract class 和 interface 有什么区别?

1、抽象类里面可以用普通方法，而接口中的方法全部都是抽象的；

2、在应用范围上来说，接口往往在程序设计的时候，用来定义程序模块的功能，方便各模块协同工作；抽象类是对相似类进行抽象，形成一个抽象的父类可供重用！

27. 如何理解 Java 中的 Serialization 和 Deserialization。

串行化(serialization)是指将一个对象的当前状态转换成字节流(a stream of bytes)的过程，

而反串行化(deserialization)则指串行化过程的逆过程，将字节流转换成一个对象，打回原形。

28. String 是最基本的数据类型吗?

1、String 是个类，不是基本数据类型；

2、基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。

29. 如何实现字符串的反转及替换。

1.通过 jdk 自带 reverse 的方法

```
public class InvertString {
    public static void main(String[] args) {
        String a="abcde";
        StringBuilder b = new StringBuilder(a);
        System.out.print(b.reverse().toString());
    }
}
```

注：String 类本身没有反转类，需要包装成 Stringbuiler 或者是 StringBuffer 类。

2.通过自己写循环。

利用 String.toCharArray()方法，将 String 转成一个 char 型数组，然后用数组遍历的方式从后向前遍历。

```
public String reverse(String s){
char ch[] = s.toCharArray();
int start = 0, end = ch.length-1;
char temp;
while(start<end){
temp = ch[start];
ch[start] = ch[end];
ch[end] = temp;
start++;
end--;
}
String s1 = String.copyValueOf(ch);
return s1;
}
、
```

或者是这样

```
 public static String reverse2(String s) {
   int length = s.length();
   String reverse = "";
```

```
    for (int i = 0; i < length; i++)
      reverse = s.charAt(i) + reverse;
    return reverse;
  }
```

30. String s = new String("xyz");创建了几个字符串对象。

两个对象，要理解这个，就要知道 string 类的工作原理。

```
public class StringTest {
    public static void main(String[] args){
        String s1="Hello";

        String s2="Hello";

        String s3=new String("Hello");

        System.out.println("s1 和 s2 引用地址是否相同："+(s1 == s2));
        System.out.println("s1 和 s2 值是否相同："+s1.equals(s2));

        System.out.println("s1 和 s3 引用地址是否相同："+(s1 == s3));
        System.out.println("s1 和 s3 值是否相同："+s1.equals(s3));
    }
}
```

输出结果：

s1 和 s2 引用地址是否相同：true

s1 和 s2 值是否相同：true

s1 和 s3 引用地址是否相同：false

s1 和 s3 值是否相同：true

31. String 和 StringBuffer 的区别。

1、String 是个不可变长度的字符串，而 StringBuffer 是个可变长度的字符串；

2、在对 String 类进行操作的时候（例如增加字符），实际上是在内存中产生了一个新的 String 对象； 而 StringBuffer 是给原对象增加字符，不是新创建一个对象；

32. 数组有没有 length()这个方法? String 有没有 length()这个方法?

数组没有 length()这个方法，有 length 的属性。String 有 length()这个方法。

33. 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。JVM 不保证此方法总被调用

34. Errror , Exception,RuntimeException 区别

Error（错误）表示系统级的错误和程序不必处理的异常，是 java 运行环境中的内部错误或者硬件问题。比如：内存资源不足等。对于这种错误，程序基本无能为力，除了退出运行外别无选择，它是由 Java 虚拟机抛出的。

Exception（违例）表示需要捕捉或者需要程序进行处理的异常，它处理的是因为程序设计的瑕疵而引起的问题或者在外的输入等引起的一般性问题，是程序必须处理的。

Exception 又分为运行时异常，受检查异常。

运行时异常，表示无法让程序恢复的异常，导致的原因通常是因为执行了错误的操作，建议终止程序，因此，编译器不检查这些异常。

受检查异常，是表示程序可以处理的异常，也即表示程序可以修复（由程序自己接受异常并且做出处理）， 所以称之为受检查异常。

Exception 体系包括 RuntimeException 体系和其他非 RuntimeException 的体系：

① RuntimeException：RuntimeException 体系包括错误的类型转换、数组越界访问和试图访问空指针等等。处理 RuntimeException 的原则是：如果出现 RuntimeException，那么一定是程序员的错误。例如，可以通过检查数组下标和数组边界来避免数组越界访问异常。

②其他非 RuntimeException（IOException 等等）：这类异常一般是外部错误，例如试图从文件尾后读取数据等，这并不是程序本身的错误，而是在应用环境中出现的外部错误。

35. Java 语言如何进行异常处理，关键字：throws、throw、try、catch、finally 分别如何使用。

throws 是获取异常

throw 是抛出异常

try 是将会发生异常的语句括起来，从而进行异常的处理，

catch 是如果有异常就会执行他里面的语句，

而 finally 不论是否有异常都会进行执行的语句。

36. throw 和 throws 有什么区别?

throws 是用来声明一个方法可能抛出的所有异常信息，而 throw 则是指抛出的一个具体的异常类型。

37. 请说一下你常用的几种异常?

空指针异常； NullPointerException

数组下标越界； ArrayIndexOutOfBoundsException

类型转换异常； ClassCastException

算数异常，例如除数为零；ArithMeticException

IO 异常，比如说找不到文件； FileNotFoundException

找不到类异常； ClassNotFoundException

sql 异常，例如 sql 语句不能正常运行；

…

38. 线程的基本概念？线程的基本状态以及状态之间的关系

1、是程序执行流的最小单元。在单个程序中同时运行多个线程完成不同的工作，称为多线程。

2、开始时：就绪状态，等待 cpu 调用后进入运行状态，运行过程中遇到阻塞事件，进入阻塞状态，等待阻塞事件结束后，重新进入就绪状态；如果没有阻塞事件，运行结束后，则进入结束状态。

39. sleep() 和 wait() 有什么区别?

sleep 就是暂停当前线程一段时间，把 cpu 让给其他线程使用，到时后会自动恢复。调用 sleep 不会释放对象锁。 wait 方法导致本线程放弃对象锁，进入等待，只有等到本对象的 notify 方法（或 notifyAll）后本线程才进入就绪状态，等待执行。

40. 多线程有几种实现方法?

多线程有两种实现方法，分别是继承 Thread 类与实现 Runnable 接口

41. 启动一个线程是用 run()还是 start()? .

启动一个线程是调用 start()方法，使线程就绪状态，以后可以被调度为运行状态，一个线程必须关联一些具体的执行代码，run()方法是该线程所关联的执行代码。

42. 线程的基本状态以及状态之间的关系。

1、新建状态（New）：新创建了一个线程对象。

2、就绪状态（Runnable）：也叫可运行状态。线程对象创建后，其他线程调用了该对象的 start()方法。该状态的线程位于可运行线程池中，变得可运行，等待获取 CPU 的使用权。

3、运行状态（Running）：就绪状态的线程获取了 CPU，执行程序代码。

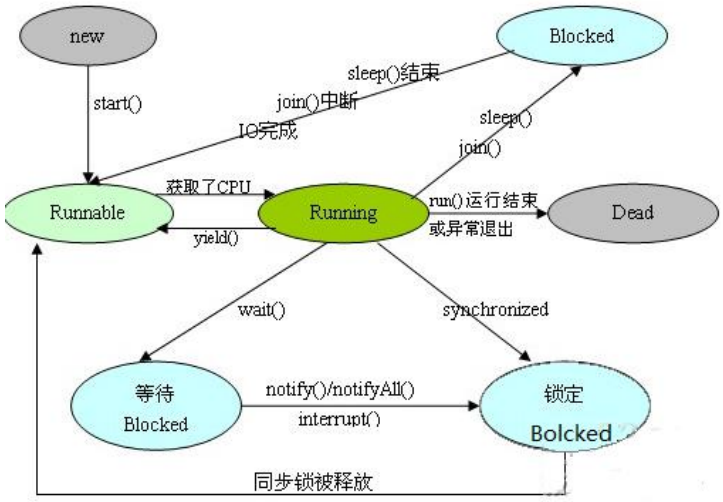
4、阻塞状态（Blocked）：阻塞状态是线程因为某种原因放弃 CPU 使用权，暂时停止运行。直到线程进入就绪状态，才有机会转到运行状态。阻塞的情况分三种：

①等待阻塞：运行的线程执行 wait()方法，JVM 会把该线程放入等待池中。

②同步阻塞：运行的线程在获取对象的同步锁时，若该同步锁被别的线程占用，则 JVM 会把该线程放入锁池中。

③其他阻塞：运行的线程执行 sleep()或 join()方法，或者发出了 I/O 请求时，JVM 会把该线程置为阻塞状态。当 sleep()状态超时、join()等待线程终止或者超时、或者 I/O 处理完毕时，线程重新转入就绪状态。

5、死亡状态（Dead）：线程执行完了或者因异常退出了 run()方法，该线程结束生命周期。



43. Set 和 List 的区别，List 和 Map 的区别?

1、Set 是无序的，元素不可重复；List 是有序的，元素可以重复；

2、List 存储的是单个对象的集合（有序的），Map 存储的是键值对为对象的集合（无序的）；

44. 同步方法、同步代码块区别?

1.同步方法

即有 synchronized 关键字修饰的方法。

由于 java 的每个对象都有一个内置锁，当用此关键字修饰方法时，

内置锁会保护整个方法。在调用该方法前，需要获得内置锁，否则就处于阻塞状态。

2.同步代码块

即有 synchronized 关键字修饰的语句块。

被该关键字修饰的语句块会自动被加上内置锁，从而实现同步

45. 描述 Java 锁机制。

java 中所说的锁就是指的内置锁，每个 java 对象都可以作为一个实现同步的锁，虽然说在 java 中一切皆对象， 但是锁必须是引用类型的，基本数据类型则不可以 。每一个引用类型的对象都可以隐式的扮演一个用于同步的锁的角色，执行线程进入 synchronized 块之前会自动获得锁，无论是通过正常语句退出还是执行过程中抛出了异常，线程都会在放弃对 synchronized 块的控制时自动释放锁。 获得锁的唯一途径就是进入这个内部锁保护的同步块或方法 。

46. Comparable 和 Comparator 接口是干什么的？列出它们的区别

它们都是用于对类的对象进行比较和排序使用的接口。

Comparable 是排序接口，位于 java.lang 包下，若一个类实现了 Comparable 接口，且重写了 compareTo 方法，就意味着该类支持排序，常结合 Collections.sort 或 Arrays.sort 对集合或数组内的元素进行排序。

Comparator 是比较接口，位于 java.util 包下，我们如果需要控制某个类对象的次序，而该类本身不支持排序(即没有实现 Comparable 接口)，那么我们就可以建立一个“该类的比较器”来进行排序，这个“比较器”只需要实现 Comparator 接口即可。

Comparable 相当于“内部比较器”，而 Comparator 相当于“外部比较器”。

47. Java 集合类框架的最佳实践有哪些?

首先 Java 中的集合框架体系非常强大和完善，主要用于程序中的数据存储，从最顶层主要分为了 Collection 和 Map 接口，我们平时使用的集合类都是从这两个类别中扩展开来，正确选择要使用的集合的类型对性能非常重要。

比如：

1）元素的大小是固定的，而且能事先知道，我们就应该用 Array 而不是 ArrayList。

2）如果我们大概知道存储的数量，可以在使用集合时先给予一个初始容量大小，从而有效避免集合自动增长的算法而造成的空间浪费，如 new ArrayList(30)。

3）为了类型安全，提高存取效率和可读性，我们优先使用泛型，并且还能有效避免 ClassCastException 类型转换异常。

4）有时为了提高数据的快速定位查找，可优先使用 Map 键值对集合，因为 Map 集合在数据的查找上效率非常高，但是如果要保证数据的顺序，最好使用 List

5）使用 JDK 提供的不变类作为 Map 的键可以避免为我们自己的类实现 hashCode()和 equals()方法。

6）底层的集合实际上是空的情况下，返回长度是 0 的集合或者是数组，不要返回 null。

48. HashMap 和 Hashtable 的区别。

1、HashMap 和 HashTable 都是键值对数据结构，且都实现了 Map 接口，存储的元素无序；

2、HashMap 非线程安全的，而 HashTable 是线程安全的（HashTable 里面的方法使用 Synchronize 关键字修饰），所以 HashMap 的效率高于 HashTable。

3、HashMap 允许空键空值，HashTable 则不允许

49. HashSet 和 TreeSet 有什么区别?

相同点：1、单列存储　2、元素不可重复

不同点：1、底层数据结构不同（HashSet===哈希表结构　TreeSet===二叉树结构）

2、数据唯一性依据不同（HashSet 通过重写 hashcode 和 equals　TreeSet 通过 compareable 接口）

3、有序性不同，HashSet 无序，TreeSet 有序

50. 说出 ArrayList,Vector, LinkedList 的存储性能和特性。

1、ArrayList 和 LinkedList、Vector 都实现了 List 接口；

2、ArrayList 和 Vector 底层是用数组实现的，而 LinkedList 使用双向链表实现的，在集合插入、删除元素时，ArrayList 需要移动数组元素性能较差；但是在查询时，因为是连续的数组，所以查询速度快；LinkedList 正好相反。

3、在容量增长上，ArrayList 增长原来 50%，Vector 集合增加容量原来的一倍。

4、安全性方面 Vector 能够保证线程安全，但是效率比 ArrayList 要低。

51. 在 Java 语言,怎么理解 goto。

goto 这个词是 C 语言中的，goto 语句通常与条件语句配合使用，可用来实现条件转移，构成循环，跳出循环体等功能。而在结构化程序语言中一般不主张使用 goto 语句， 以免造成程序流程的混乱，使理解和调试程序都产生困难。但是在 java 语言中，goto 这个词只是作为了保留字，不推荐使用，因为 java 语言讲究简单，方便。

52. 请描述一下 Java 5 有哪些新特性?

1、泛型

2、For-Each 循环

3、自动装包/拆包

4、枚举

5、静态导入

6、Annotation(框架配置,代码生成 的里程

7、可变参数

8、字符串格式化器(java.util.Formatter)

9、新的线程模型和并发库 Thread Frameword

53. Java 6 新特性有哪些。

1、引入了一个支持脚本引擎的新框架

2、UI 的增强

3、对 WebService 支持的增强（JAX-WS2.0 和 JAXB2.0）

4、一系列新的安全相关的增强

5、JDBC4.0

6、Compiler API

7、通用的 Annotations 支持

54. Java 7 新特性有哪些。

1.switch 中可以使用字串了

2.运用 List<String> tempList = new ArrayList<>(); 即泛型实例化类型自动推断

3.语法上支持集合，而不一定是数组

4.新增一些读取环境信息的工具方法

5.Boolean 类型反转，空指针安全,参与位运算

6.两个 char 间的 equals

7.安全的加减乘除

8.map 集合支持并发请求，且可以写成 Map map = {name:"xxx",age:18};

55. Java 8 新特性有哪些。

1.Lambda 表达式 − Lambda 允许把函数作为一个方法的参数，用更简洁的语法实现以前复杂的功能代码

2.方法引用提供了非常有用的语法，可以直接引用已有 Java 类或对象（实例）的方法或构造器。与 lambda 联合使用，方法引用可以使语言的构造更紧凑简洁，减少冗余代码。

3.接口中可以使用 default 关键字声明一个默认实现方法。

4.新的编译工具，如：Nashorn 引擎 jjs、 类依赖分析器 jdeps。

5.新添加的 Stream API（java.util.stream） 把真正的函数式编程风格引入到 Java 中。

6.加强对日期与时间的处理。

7.Optional 类用来解决空指针异常。

8.Java 8 提供了一个新的 Nashorn javascript 引擎，它允许我们在 JVM 上运行特定的 javascript 应用。

56. 描述 Java 动态代理。

代理模式是常用的 java 设计模式之一，java 中的代理实现主要有基于接口的 jdk 动态代理和基于类的 cglib 动态代理，代理的特点就是会为目标对象产生代理对象，从而在调用实际目标对象方法时实现无侵入式的代码扩展，比如一些框架中的拦截器机制、springAOP 面向切面机制都是基于动态代理模式实现的，其次也可以更安全的保护目标对象。动态代理的目的主要就是为了解决一些安全性问题（不让客户直接调用目标对象的功能，而是相当于通过中介），其次就是可以在调用目标功能方法之前、之后额外添加一些功能，比如日志、事务等，并且还能阻止调用功目标功能，从而实现权限控制等。

57. 为什么要使用单例模式?

1、避免在开发程序的时候，创建出一个类的多个实例（占用空间，性能问题），所以使用单例模式，保证该类只创建一个对象；

2、一般单例模式通常有两种形式：它的构造函数为 private 的，必须有一个静态方法，静态方法返回自己的实例；实行形式有两种，懒汉式和饿汉式；所谓的饿汉式就是在声明实例的时候直接初始化对象，而懒汉式是先声明一个空对象，在静态方法中实例化该对象并返回。

```
//构造方法私有化，让外界无法创建对象
private Test() {
    }
    //在当前类中声明静态对象，并提供公共静态方法让外界调用获取对象
    private static Test t;

    public static Test getInstance(){
        if(t==null){
            t=new Test();
```

```
        }
        return t;
    }
```

58. Java 中有哪些常用排序方式，使用 Java 实现冒泡排序。

排序主要用于将一组无需数据进行升序/降序排列，Java 中有很多种排序方法，如：冒泡排序，选择排序，插入排序，快速排序等，其中在大量数据无需数据中效率最高的属于快速排序，比如实际工作中如果需要对数据排序，我们可以借助 JDK 中自带的 Arrays.sort 方法，它内部采用了快速排序，效率非常高，当然也可以自己实现。

冒泡排序代码如下：

```
import java.util.Arrays;
public class BubbleSort {
    public static void BubbleSort(int[] arr) {
        int temp;//定义一个临时变量
        for(int i=0;i<arr.length-1;i++){          //循环的轮数
            for(int j=0;j<arr.length-i-1;j++){  //从前往后循环比较，从第 1 个数往后依次比较
                if(arr[j+1]<arr[j]){            //如果后面的数字小于前面的则交换
                    temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
    }
    public static void main(String[] args) {
        int arr[] = new int[]{1,6,2,2,5};
        BubbleSort.BubbleSort(arr);
        System.out.println(Arrays.toString(arr));
    }
}
```

59. Java 中垃圾回收有什么目的？什么时候进行垃圾回收?

Java 语言中一个显著的特点就是引入了垃圾回收机制（简称 GC），使 c 语言程序员最头疼的内存管理的问题迎刃而解，它使得 Java 程序员在编写程序的时候不再需要考虑内存管理。由于有个垃圾回收机制，Java 中的对象不再有“作用域”的概念，只有对象的引用才有“作用域”。垃圾回收可以有效的防止内存泄露，有效的使用空闲的内存。简单的理解，就是当对象失去引用后，系统会在调

度的时间对它所占用的内存进行回收。

60. 如何实现对象克隆。

实现对象克隆有两种方式：

1). 实现 Cloneable 接口并重写 Object 类中的 clone()方法；

2). 实现 Serializable 接口，通过对象的序列化和反序列化实现克隆，可以实现真正的深度克隆。

61. Java 设计模式有哪些?

Java 中有 23 种设计模式，我觉得理解设计模式对我们程序中的类和类的设计、依赖关系，扩展性，灵活性起着非常重要的作用，比如 spring 框架中都大量使用了设计模式，我比较熟悉的设计模式有：

1.单例设计模式

2.工厂设计模式（简单工厂、抽象工厂、工厂方法）

3.代理设计模式

4.观察者设计模式

5.适配器模式

6.原型模式

62. GC 是什么？为什么要有 GC?

GC 是垃圾回收的意思（gabage collection），内存处理器是编程人员容易出现问题的地方，忘记或者错误的内存回收导致程序或者系统的不稳定甚至崩溃，java 的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，java 语言没有提供释放已分配内存的显式操作方法。

63. Java 中是如何支持正则表达式。

Java 中的 String 类提供了支持正则表达式操作的方法，包括：matches()、replaceAll()、replaceFirst()、split()。此外，Java 中可以用 Pattern 类表示正则表达式对象，它提供了丰富的 API 进行各种正则表达式操作。

64. 比较一下 Java 和 JavaSciprt。

JavaScript 与 Java 是两个公司开发的不同的两个产品。Java 是 SUN 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发；而 JavaScript 是 Netscape 公司的产品，其目的是为了扩展 Netscape Navigator 功能,而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言,它的前身是 Live Script；而 Java 的前身是 Oak 语言。

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

11.5+0.5 后是 12 再向下取整是 12;-11.5+0.5 后是-11 再向下取整-11

66. JDBC 连接数据库的步骤？

JDBC 是 java 连接及操作数据库的一种技术，使用步骤如下：

1、加载 JDBC 驱动程序； Class.forName("com.mysql.jdbc.Driver")

2、创建数据库的连接对象；

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ 数据库名","root","123456");

3、创建一个执行 sql 命令的 Statement 或 PreparedStatement 或者 CallableStatement 对象

4、执行 SQL 语句； 增删改操作 executeUpdate()→返回 int 查询操作 executeQuery()→返回结果集 ResultSet

5、返回并处理结果； 如果是查询需要对结果集逐行处理：while(rst.next())

6、关闭连接； conn.close()

67. Class.forName()方法有什么作用？

通过一个字符串加载一个类到 java 虚拟机中，返回与给定的字符串名称相关联类或接口的 Class 对象，该方法使用时会抛出 ClassNotFoundException，即类无法找到异常。

68. JDBC 中如何进行事务处理

事务是为了保证一个业务下的多条更新语句处于同一个不可分割的单元，要么都成功执行要么都取消回滚，目的是保证数据的准确合理性。

JDBC 同样对事务进行了很好的支持，因为 JDBC 默认是开启事务的，所以需要通过数据库连接对象的 setAutoCommit(false)来禁止自动提交，然后在执行完相关操作之后，调用连接对象的 commit()方法提交事务，如果出现异常则调用 rollback()方法进行回滚。

一般在实际应用中，建议使用存储过程来进行事务的控制，因为存储过程更安全，高效。

69. JDBC 能否处理 Blob 和 Clob

Blob 是指二进制大对象(Binary Large Object)，而 Clob 是指大字符对象(Character Large Objec)，因此其中 Blob 是为存储大的二进制数据而设计的，而 Clob 是为存储大的文本数据而设计的。JDBC 的 PreparedStatement 和 ResultSet 都提供了相应的方法来支持 Blob 和 Clob 操作。

70. JDBC 中的 PreparedStatement 相比 Statement 的好处

1、PreparedStatement 是 Statement 的子接口；

2、PreparedStatement 支持 SQL 语句中使用占位符，能够避免 SQL 注入，安全性更好；

3、PreparedStatement 对 SQL 语句有预编译的功能，所以性能更好；

71. 解释内存中的栈(stack)、堆(heap)和静态区(static area)的用法。

通常我们定义一个基本数据类型的变量，还有就是函数调用的现场保存都使用内存中的栈空间；而通过 new 关键字和构造器创建的对象放在堆空间；程序中的字面量（literal）如直接书写的 100、”hello”和常量都是放在静态区中。

栈空间操作起来最快但是栈很小，通常大量的对象都是放在堆空间，理论上整个内存没有被其他进程使用的空间甚至硬盘上的虚拟内存都可以被当成堆空间来使用。

String str = new String("hello");

上面的语句中变量 str 放在栈上，用 new 创建出来的字符串对象放在堆上，而”hello”这个字面量放在静态区。

72. 怎样将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串?

代码如下所示：

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

73. 运行时异常与受检异常有何异同?

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，只要程序设计得没有问题通常就不会发生。受检异常跟程序运行的上下文环境有关，即使程序设计无误，仍然可能因使用的问题而引发。Java 编译器要求方法必须声明抛出可能发生的受检异常，但是并不要求必须声明抛出未被捕获的运行时异常。异常和继承一样，是面向对象程序设计中经常被滥用的东西，在 Effective Java 中对异常的使用给出了以下指导原则：

- 不要将异常处理用于正常的控制流（设计良好的 API 不应该强迫它的调用者为了正常的控制流而使用异常）
- 对可以恢复的情况使用受检异常，对编程错误使用运行时异常
- 避免不必要的使用受检异常（可以通过一些状态检测手段来避免异常的发生）
- 优先使用标准的异常
- 每个方法抛出的异常都要有文档
- 保持异常的原子性
- 不要在 catch 中忽略掉捕获到的异常

74. 列出一些你常见的运行时异常?

ArithmeticException（算术异常） 5/0

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）

NumberFormatException （数字格式异常 Integer.parseInt("a1b2")）

FileNotFoundException （文件找不到异常）

ClassNotFoundException （类找不到异常）

75. List、Set、Map 是否继承自 Collection 接口?

List、Set 是，Map 不是。Map 是键值对映射容器，与 List 和 Set 有明显的区别，而 Set 存储的零散的元素且不允许有重复元素（数学中的集合也是如此），List 是线性结构的容器，适用于按数值索引访问元素的情形。

76. Thread 类的 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?

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。

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。

77. 线程的 sleep()方法和 yield()方法有什么区别?

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。

78. 请说出与线程同步以及线程调度相关的方法。

- 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；

79. 举例说明同步和异步。

同步：指发送一个请求,需要等待返回,然后才能够发送下一个请求，有个等待过程，如果某个操作非常耗时，则会使后续的功能处于等待状态，产生假死/阻塞效果。

异步：指发送一个请求,不需要等待返回,随时可以再发送下一个请求，即不需要等待。

区别：一个需要等待，一个不需要等待，在部分情况下，我们的项目开发中都会优先选择不需要等待的异步交互方式。银行的转账系统，对数据库的保存操作等等，都会使用同步交互操作，其余情况都优先使用异步交互。

Java 中的某个方法可以通过 Synchronized 关键字使其变为同步，从而解决线程中的异步资源安全问题。

80. Java 中如何实现序列化，有什么意义？

Java 中将一个类实现 Serializable 接口（实际是空接口，起标识作用），则该类的对象就可以被序列化。

序列化就是将类的对象进行流化，被流化后的对象可以在网络中传输或者以文件的形式进行保存，然后在需要的时候可以进行反序列化，将流化的对象还原为原始对象，并且数据都保持原来的状态。具体操作是使用 writeObject()方法进行写，然后使用时再通过 readObject()方法进行对象读取还原。

81. 获得一个类的类对象有哪些方式？

- 方法 1：类型.class，例如：String.class
- 方法 2：对象.getClass()，例如：”hello”.getClass()
- 方法 3：Class.forName()，例如：Class.forName(“java.lang.String”)

82. 如何通过反射创建对象?

- 方法 1：通过类对象调用 newInstance()方法，例如：String.class.newInstance()

- 方法 2：通过类对象的 getConstructor()或 getDeclaredConstructor()方法获得构造器（Constructor）对象并调用其 newInstance()方法创建对象，例如：

String.class.getConstructor(String.class).newInstance(“Hello”);

83. Enumeration 接口和 Iterator 接口的区别有哪些?

Enumeration 速度是 Iterator 的 2 倍，同时占用更少的内存。但是，Iterator 远远比 Enumeration 安全，因为其他线程不能够修改正在被 iterator 遍历的集合里面的对象。同时，Iterator 允许调用者删除底层集合里面的元素，这对 Enumeration 来说是不可能的。

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

吞吐量收集器使用并行版本的新生代垃圾收集器，它用于中等规模和大规模数据的应用程序。

而串行收集器对大多数的小应用(在现代处理器上需要大概 100M 左右的内存)就足够了。

7.2 Java web 部分

1. 什么是 B/S 和 C/S?

B/S 是 Brower/Server(浏览器/服务器模式)的缩写，客户机上只要安装一个浏览器,实现与服务器交互；

C/S 是 Client/Server(客户端/服务器模式)的缩写,客户机需安装专用客户端,实现与服务器交互,如数据库产品；

B/S 代表,淘宝网、京东网站。

C/S 代表,腾讯 QQ

.主要区别如下：

1、 B/S 使用浏览器访问，安装维护的成本代价很小。

2、 C/S 需要安装客户端软件，安装维护成本较大。

2. 如何设置 servlet 初始化参数? 如何获取 servlet 初始化的参数?

1、在 web.xml 里为每一个 Servlet 配置初始化参数；

2、通过 ServletConfig 对象实现对 Servlet 初始化对象的获取；

3. Ajax 是什么?

1、Ajax 是采用了异步请求的方式，解决了页面无刷新式提交的问题，改善了页面的用户体验效果；常用自动完成提示，注册时用户名重复性校验。

2、常用的 Ajax 框架： Jquery 中的 ajax

答题技巧:解释概念,说明用途,举例

4. HTTP 请求的 GET 与 POST 方式的区别?

Get 和 Post 都是提交表单的方式之一；

1、[安全性]get 方式提交后，数据会在地址栏中显示出来，而 post 方式不会，所以 post 更安全；

2、[数据长度]get 方式在提交数据的时候，数据长度是有限制的；而 post 方式在理论上对提交数据的大小是无限制的；

3、[效率]get 方式效率更高

5. 说一说 Servlet 的生命周期? Servlet 里常用的方法有哪些?

1、Servlet 生命周期包括加载和实例化、初始化、处理请求以及销毁。

2、Servlet 被服务器调用构造方法实例化，第一次访问 Servlet 时，容器运行其 init 方法进行初始化，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。

6. Servlet API 中 forward() 与 redirect()的区别?

1、forward 是请求转发，而 redirect 是重定向；

2、请求转发是服务器将客户端的请求转发到另一个地址去处理，然后将响应返回给客户端；实际上是 1 次请求，1 次响应，对客户端而言是透明的；而重定向是服务器根据客户端发来的请求，返回给客户端一个新的地址，客户端根据这个返回的地址再发送请求，得到响应；实际上是 2 次请求，2 次响应，而且客户端的地址是第二次访问的地址；

3、转发只能转发到服务器自己的资源，重定向无限制。

4、转发的效率高于重定向

5、转发地址栏不会改变，重定向地址栏会变成第二次访问的地址。

转发核心代码：request.getRequestDispatcher(“资源地址”).forward(request,response);

重定向核心代码：response.sendRedirect(“资源地址”);

7. 什么是 JSP

JSP 是 Java Server Page 的简称，是由 Sun 公司倡导简历的一种动态网页技术标准，**用于开发动态网页。JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成**，服务器在页面被客户端请求后执行

Java 代码，将生成的 HTML 页面返回给客户端。

8. Jsp 优缺点。

1、JSP 技术的优点

（1）一次编写，到处运行。

（2）跨平台。

（3）强大的可伸缩性。

2、JSP 技术的弱势

（1）正是由于为了跨平台的功能，所以极大的增加了产品的复杂性。

（2）占用内存较大。

（3）代码调试不便。

9. jsp 有哪些内置对象?

JSP 共有以下 9 个内置的对象：

page、config、application、request、response、session、out、exception、pageContext

request：封装客户端的请求，其中包含来自 GET 或 POST 请求的参数；

response：封装服务器对客户端的响应；

session：封装用户会话的对象；

application：封装服务器运行环境的对象；

page：JSP 页面本身（相当于 Java 程序中的 this）；

pageContext：通过该对象可以获取其他对象；

out：输出服务器响应的输出流对象；

config：Web 应用的配置对象；

exception：封装页面抛出异常的对象。

10. 讲解 JSP 中的四种作用域。

JSP 的四大作用域：page、request、session、application

page：变量只能在**当前页面**上生效

request：变量能在**一次请求**中生效，一次请求可能包含一个页面，也可能包含多个页面，比如页面 A 请求转发到页面 B

session：代表变量能在**一次会话**中生效，基本上就是能在 web 项目下都有效。一般来说，只要浏览器不关闭，session 的使用就不会受到影响。

application：代表**整个应用程序**范围。

存入数据：作用域对象.setAttribute(“名字”,数据);

取出数据：作用域对象.getAttribute(“名字”);

11. Session 和 Cookie 的区别?

1、Session 和 Cookie 都是会话跟踪技术；

2、Session 是**保存在服务器端**的技术（保持用户登录状态的检查常用 Session），而 Cookie 是**保存在客户端**的技术；

3、Cookie **只能存字符串**，Session **可以存对象**；

12. 常用的 Web 服务器有哪些

1.Tomcat

目前非常流行的 Tomcat 服务器是 Apache-Jarkarta 开源项目中的一个子项目，是一个小型、轻量级的支持 JSP 和 Servlet 技术的 Web 服务器，也是初学者学习开发 JSP 应用的首选。

2.JBoss

JBoss 是一个种遵从 JavaEE 规范的、开放源代码的、纯 Java 的 EJB 服务器，对于 J2EE 有很好的支持。JBoss 采用 JML API 实现软件模块的集成与管理，其核心服务又是提供 EJB 服务器，不包含 Servlet 和 JSP 的 Web 容器，不过它可以和 Tomcat 完美结合

3.WebLogic

WebLogic 是 BEA 公司的产品(现在已经被 Oracle 收购)。WebLogic 支持企业级的、多层次的和完全分布式的 Web 应用，并且服务器的配置简单、界面友好。

13. 什么是 Servlet。

在 Java Web 程序中，**Servlet 主要负责接收用户请求 HttpServletRequest,在 doGet(),doPost()中做相应的处理，并将回应 HttpServletResponse 反馈给用户**。Servlet 可以设置初始化参数，供 Servlet 内部使用。一个 Servlet 类只会有一个实例，在它初始化时调用 init()方法，销毁时调用 destroy()方法。Servlet 需要在 web.xml 中配置，**一个 Servlet 可以设置多个 URL 访问**。

14. JSP 和 Servlet 是什么区别和联系。

1.jsp 经编译后就变成了 Servlet.

(**JSP 的本质就是 Servlet**，Web 容器将 JSP 的代码编译成 java 类)

2.jsp 更**擅长表现于页面显示**，Servlet 更**擅长于逻辑控制**.

3.Servlet 中没有内置对象，Jsp 中的内置对象都是必须通过 HttpServletRequest 对象，HttpServletResponse 对象以及 HttpServlet 对象得到。

15. web.xml 文件中可以配置哪些内容。

web.xml 文件是**用来配置：欢迎页、servlet、filter、listener、初始化信息、错误页面等的**。当你的 web 工程没用到这些时，你可以不用 web.xml 文件来配置你的 web 工程。

答题技巧：可根据个人实际情况，增加各类框架在 web.xml 内的配置代码,如 struts2 过滤器配置,spring 监听器配置,springmvc Servlet 配置等.

16. 说一下表达式语言（EL）的隐式对象及其作用。

EL 的隐式对象包括：

pageContext、

initParam（访问上下文参数）、

param（访问请求参数）、

paramValues、

header（访问请求头）、

headerValues、

cookie（访问 cookie）、

applicationScope（访问 application 作用域）、

sessionScope（访问 session 作用域）、

requestScope（访问 request 作用域）、

pageScope（访问 page 作用域）。

17. 使用标签库有什么好处

使用标签库的好处包括以下几个方面：

- **分离 JSP 页面的内容和逻辑**，简化了 Web 开发；
- 开发者可以创建自定义标签来封装业务逻辑和显示逻辑；
- 标签具有很好的可移植性、可维护性和可重用性；
- **避免了对 Scriptlet（小脚本）的使用**（很多公司的项目开发都不允许在 JSP 中书写小脚本）

```
<%
    Java 代码
%>
```

18. 表达式语言（EL）支持哪些运算符。

EL 表达式的操作符主要有: 算术运算符、关系运算符、逻辑运算符、验证运算符 empty、条件运算符

算术运算符主要有平时常用的“+”、“-”、“*”、“/”、“%”。

关系运算符主要有“==”、“!=”、“<”、“>”、“<=”、“>=”

逻辑运算符主要有“&&”、“||”、“!”。

验证运算符“empty”

条件运算符“?:”

答题技巧: 先说明支持分类,然后对每一类运行符说明具体运算符,

19. 你的项目中使用过哪些 JSTL 标签。

表达式控制标签：out、set、remove、catch

流程控制标签：if、choose、when、otherwise

循环标签：forEach、forTokens

URL 操作标签：import、url、redirect

答题技巧:不要求写出全部标签,写自己真实用过,理解最深的标签,适当附上示例代码

20. 静态 include 和动态 include 的区别?

1、静态包含属于编译期包含（包含页面和被包含的页面在编译期**形成一个 jsp 文件**），动态包含属于运行期包含（包含页面和被包含的页面**分别编译成两个文件**，然后运行时把两个文件组装起来）；

2、**动态包含可以带参数**；

21. 如何设置请求的编码以及响应内容的类型。

通过请求对象（ServletRequest）的 setCharacterEncoding(String)方法可以设置请求的编码，其实要彻底解决乱码问题就应该让页面、服务器、请求和响应、Java 程序都使用统一的编码，最好的选择当然是 UTF-8；**通过响应对象（ServletResponse）的 setContentType(“text/html;charset=utf-8”)方法可以设置响应内容的类型**，当然也可以通过 HttpServletResponse 对象的 setHeader(String, String)方法来设置。

7.3 数据持久化部分

1. ORM 框架的原理是什么?

ORM 是对象-关系映射（Object-Relational Mapping，简称 ORM）

它是 java 中持久层框架技术的一种实现思想，基于这种思想常见的框架有 MyBatis、Hibernate 等。其原理是建立 java 中类（实体类）和数据表之间的映射关系，然后通过反射的思想，动态获取类中的属性，此时属性对应了表中的列，所以能够动态产生 JDBC 代码从而达到操作数据库的目的。

2. 什么是 DAO 模式?

dao 全称是 data access object，数据库访问对象，主要的功能就是用于进行数据操作的，在程序的标准开发架构中属于数据层的操作

DAO 组成

在整个 DAO 中实际上都是以接口为操作标准的，即：客户端依靠 DAO 实现的接口进行操作，而服务端要将接口进行具体的实现。DAO 由以下几个部分组成。

1.DatabaseConnection：专门负责数据库的打开与关闭操作的类

2.VO：主要由属性、setter、getter 方法组成，VO 类中的属性与表中的字段相对应，每一个 VO 类的对象都表示表中的每一条记录；

3.DAO：主要定义操作的接口，定义一系列数据库的原子性操作，例如：增加、修改、删除、按 ID 查询等；

4.Impl ： DAO 接口的真实实现类，完成具体的数据库操作，但是不负责数据库的打开和关闭；

5 Proxy ：代理实现类，主要完成数据库的打开和关闭，并且调用真实实现类对象的操作

6 Factory ：工厂类，通过工厂类取得一个 DAO 的实例化对象

3. 举一个多对多关联的例子，并说明如何实现多对多关联映射。

例如：商品和订单、学生和课程都是典型的多对多关系。首先在数据库表中需要通过第三张关系表来进行数据的维护，该关系表中主要包含了两个外键。具体代码中实现根据框架的不同也有不同的实现方式，比如 JPA 中可以在实体类上通过@ManyToMany 注解配置多对多关联或者通过映射文件中的和标签配置多对多关联，但是实际项目开发中，很多时候都是将多对多关联映射转换成两个多对一关联映射来实现的。

4. 谈一下你对继承映射的理解。

继承关系的映射策略有三种：

① 每个继承结构一张表（table per class hierarchy），不管多少个子类都用一张表。

② 每个子类一张表（table per subclass），公共信息放一张表，特有信息放单独的表。

③ 每个具体类一张表（table per concrete class），有多少个子类就有多少张表。

第一种方式属于单表策略，其优点在于查询子类对象的时候无需表连接，查询速度快，适合多态查询；缺点是可能导致表很大。后两种方式属于多表策略，其优点在于数据存储紧凑，其缺点是需要进行连接查询，不适合多态查询。

5. 什么是 Hibernate，好处是什么?

1、Hibernate 是一个基于 ORM 思想的操作数据库的框架技术，实现了对 JDBC 的封装；

2、Hibernate 用的是面向对象的方法，但是在关系型数据库里，存的是一条条的记录；为了用纯面向对象的思想解决问题，所以需要将程序中的对象和数据库中的记录建立起映射关系，ORM 就是将对象和数据库中的记录建立起映射的技术；

3、Hibernate 通过方言来自动切换不同的数据库，无需在代码中编写；

4、使用 Hibernate 的基本流程是：配置 Configuration 对象、产生 SessionFactory、创建 session 对象，启动事务，完成 CRUD 操作，提交事务，关闭 session；

5、使用 Hibernate 时，先要配置 hibernate.cfg.xml 文件，其中配置数据库连接信息和方言等，还要为每个实体配置相应的 hbm.xml 文件，hibernate.cfg.xml 文件中需要登记每个 hbm.xml 文件。

6. Hibernate 的核心思想?

Hibernate 是一个采用 ORM（Object/Relation Mapping 对象关系映射）机制持久层的开源框架

其主要核心思想是面向对象，而非面向过程，而这个面向对象则主要通过 ORM 实现。

7. 什么是 Mybatis?

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java 对象)映射成数据库中的记录。

8. MyBatis 中使用#和$书写占位符有什么区别?

1.#将传入的数据都当成一个字符串，会对自动传入的数据加一个双引号。

2.$将传入的数据直接显示生成在 sql 中。

3.#方式能够很大程度防止 sql 注入。

4.$方式无法防止 Sql 注入。

5.$方式一般用于传入数据库对象

6.一般能用#的就别用$

注意: 一般如果将表名、列名作为参数的动态功能操作时，必须使用$，而不能使用#。如 MyBatis 排序时使用 order by 动态参数时需要使用$。

9. Hibernate 与 Mybatis 区别(MyBatis 与 Hibernate 有什么不同)。

相同点：都是基于 ORM 思想的 java 持久层框架技术，都封装了 JDBC 实现数据库操作，提高了开发数据库编码的效率。

不同点：

1） Hibernate 是一个全自动的 orm 映射工具，它可以自动生成 sql 语句；MyBatis 需要我们自己在 xml 配置文件中写 sql 语句。因为 hibernate 自动生成 sql 语句，我们无法控制该语句，我们就无法去写特定的高效率的 sql。对于一些不太复杂的 sql 查询，hibernate 可以很好帮我们完成，但是，对于特别复杂的查询，hibernate 就很难适应了，这时候用 MyBatis 就是不错的选择，因为 MyBatis 还是由我们自己写 sql 语句；

2） Mybatis 比 hibernate 对存储过程的支持更好一些。

10. 持久层设计要考虑的问题有哪些?

所谓"持久"就是将数据保存到存储设备中以便今后使用，简单的说，就是将内存中的数据保存到关系型数据库、文件系统、消息队列等提供持久化支持的设备中。持久层就是系统中专注于实现数据持久化的相对独立的层面。

持久层设计的目标包括：

- 数据存储逻辑的分离，提供抽象化的数据访问接口。
- 数据访问底层实现的分离，可以在不修改代码的情况下切换底层实现。
- 资源管理和调度的分离，在数据访问层实现统一的资源调度（如缓存机制）。
- 数据抽象，提供更面向对象的数据操作。

11. 你用过的持久层框架有哪些

持久层主要是负责将数据进行持久化，早期的 JDBC 代码太麻烦繁琐，所以出现了一系列基于 ORM 思想的持久层框架，我主要使用的是 MyBatis 框架，其他的比如 Hibernate 以及 JPA 有一些了解。

技巧：选自己最熟的持久化框架，说明其关键特点

12. @OneToMany 注解的 mappedBy 属性有什么作用?

技巧：mappedBy 属性的主要作用是告诉 hibernate 关联关系是通过对方外键引用完成关联映射，这一点必须说清楚。

One 的一方会成为主控方，加入 mappedBy 以后，就不会生成这两张关联表的中间表了，而是在 Many 的一方的表中生成一列作为外键。

13. MyBatis 中的动态 SQL 是什么意思?

MyBatis 的动态 SQL 是基于 OGNL 表达式的，它可以帮助我们方便的在 SQL 语句中实现某些逻辑，一般在我们的 Mapper.xml 映射文件中使用，MyBatis 中用于实现动态 SQL 的标签元素主要有：

if

choose（when，otherwise）

trim

where

set

foreach

14. 如何解决 Mybatis “n+1”问题。

N+1 问题主要是数据库表中的级联问题，就是在查询的时候使用了 association 或 collection 标签进行关联数据时，如果使用了嵌套的 select 方式，则会造成 N+1 的问题，解决方案是：

第一种方法是使用一条多表关联的 SQL 语句，一次性查询出需要的数据。

第二种方法是使用 MyBatis 的延迟加载机制，在具体使用到的时候再去加载数据。

15. Mybatis 动态 SQL 相关标签作用及含义?

1.MyBatis 的动态 SQL 是基于 OGNL 表达式的，它可以帮助我们方便的在 SQL 语句中实现某些逻辑。

2.MyBatis 中用于实现动态 SQL 的元素主要有

if

choose（when，otherwise）

trim

where

set

foreach

可以看出 MyBatis 的动态 SQL 的标签元素和接近 JSP 中的 JSTL 语法，下面我就分别详细的介绍一下

3.动态 SQL 中 if 的用法

4.动态 SQL 中 choose 用法

5.动态 SQL 中 where 语句的作用主要是简化 SQL 语句中 where 中的条件判断的

注意：where 元素的作用是给 SQL 语句添加一个条件判断. 如果输出后是 and 开头的，MyBatis

会把第一个 and 忽略，当然如果是 or 开头的，MyBatis 也会把它忽略；此外，在 where 元素中你不需要考虑空格的问题，MyBatis 会智能的帮你加上

6.动态 SQL 中 set 元素主要是用在更新操作的时候，它的主要功能和 where 元素其实是差不多的

7.动态 SQL 中 foreach

主要用在构建 in 条件中，它可以在 SQL 语句中进行迭代一个集合。

foreach 元素的属性主要有 item，index，collection，open，separator，close

8.动态 SQL 中 trim 语句

16. 什么是 JDBC,简述实现方法?

JDBC 是 java 进行数据库操作时的技术统称，提供了很多操作接口，但是编写的代码非常繁琐麻烦，创建一个以 JDBC 连接数据库的程序，包含 7 个步骤：

1、加载 JDBC 驱动程序： Class.forName("com.mysql.jdbc.Driver")

2、提供 JDBC 连接的 URL： "jdbc:mysql://localhost:3306/数据库名"

3、创建数据库的连接

Connection conn = DriverManager.getConnection("URL","root","123456")

4、创建一个执行 sql 命令的 Statement 对象

PreparedStatement pstmt = conn.prepareStatement(sql);

5、执行 SQL 语句

1）增删改 pstmt.executeUpdate() →返回 int 为影响的行数

2）查询 pstmt.executeQuery()→返回 ResultSet 为查询的结果集

6、循环读取结果： while(rst.next())

7、关闭 JDBC 相关资源对象

17. 请介绍一下数据库连接池技术?

1）数据库连接池技术主要是提高应用程序和数据库交互时的性能问题，因为连接数据库是非常耗费资源的，如果每个操作都去频繁连接和断开，则会造成巨大的开销，连接池技术即在系统初始化时，就创建一系列的连接对象放入集合中管理，需要连接数据库时从连接池中取出使用，使用完毕归还给连接池，这样的目的是提高同一个连接的可重用性。

2）java 中有很多连接池框架，比如 dbcp、c3p0，我个人比较喜欢使用阿里的 druid 连接池，因为性能稳定，功能强大。

注：一般项目使用时直接添加 maven 引用

18. MySQL、Oracle、SQL Server 各数据库服务的默认端口号?

1、MySQL：3306

2、Oracle：1521

3、SQL Server：1433

19. MyBatis 中如何实现分页?

分页的目的是减少一次性查询出数据的额外开销，提高应用的性能。

Mybatis 中可以使用如下几种方案：

1、 sql 语句分页，利用传递的页码和每页显示数，使用 limit 分页

2、 采用 mybatis 拦截器插件实现分页

20. MyBatis 中如何实现数据映射?

Mybatis 要求我们自己完成查询 sql 语句，然后对查询的类进行映射，默认情况下如果类和实体属性一致，则可以直接使用 reusltType 进行关联映射，如果不一致，则需要自定义 resultMap 进行映射说明。当然如果数据库列是下划线，而 java 中是驼峰命名，则可以在 mybatis 主配置中添加驼峰命名的全局 setting 配置即可。

21. MyBatis 中的缓存如何使用?

1） 缓存是为了提高程序的性能，把从数据库中查询出来和使用过的对象保存在内存中，当以后要使用某个对象时，先查询缓存中是否有这个对象，如果有则使用缓存中的对象，如果没有则去查询数据库，并将查询出来的对象保存在缓存中，以便下次使用。

2） Mybatis 中有一级缓存和二级缓存，其中一级缓存是 SqlSession 级别的缓存，在同一个 SqlSession 操作时都会默认使用，当会话关闭则会自动清空无法使用。二级缓存是 Mapper 级别的缓存，需要在主配置中声明开启，然后在 Mapper.xml 中使用<cache>标签声明后才能使用。

3） 缓存使用时需要注意，经常修改和准确度要求很高的数据不适合放入缓存中，因为很容易产生脏数据，从而造成程序中的数据不准确。

22. MyBatis 中的使用流程?

1） 首先项目中加入 mybatis 的依赖，如 maven 的 pom.xml 中添加

2） 创建并修改 mybatis-config.xml 核心配置文件（配置全局属性、别名、数据库连接、注册的 Mapepr 等）

3） 创建实体类和数据表映射

4） 创建功能操作接口，完成抽象方法

5） 针对抽象方法完成对应的 Mapper.xml 中的映射或者使用注解实现

6） 运行时加载配置，产生 SqlSessionFactory 对象，然后调用 openSession 方法获取 SqlSession，然后拿到 Mappper 接口代理对象，调用方法完成操作。

7） 关闭会话

23. MyBatis 中的关联如何实现?

1） 一般使用< association>标签进行一对一的对象关联映射

2） 使用< collection>标签进行一对多的集合关联映射

具体映射时候可以采用一次性查询出结果（推荐）然后映射，也可以采用嵌套的 select 查询方式。

24. MyBatis 中的插件如何实现?

插件也可以理解为拦截器，即在 mybatis 执行某个 sql 语句或者结果处理前后，均可以自己编写代码进行额外的操作，Mybatis 仅可以编写针对 ParameterHandler、ResultSetHandler、StatementHandler、Executor 这 4 种接口的插件。Mybatis 使用 JDK 的动态代理，为需要拦截的接口生成代理对象以实现接口方法拦截功能，每当执行这 4 种接口对象的方法时，就会进入拦截方法，具体就是 InvocationHandler 的 invoke()

方法，只会拦截那些你指定需要拦截的方法，具体使用时还需要在拦截器类上进行签名以及在主配置文件中声明才能使用。

25. 请描述 MyBatis 中的优缺点?

1） MyBatis 的优点：屏蔽 jdbc api 的底层访问细节；自己定制 sql 能自定义映射，和将 sql 语句与 java 代码进行分离;提供了将结果集自动封装称为实体对象和对象的集合的功能，queryForList 返回对象集合，用 queryForObject 返回单个对象；提供了自动将实体对象的属性传递给 sql 语句的参数。其次 mybatis 很好的跟 spring 集成。

2） Mybatis 的缺点：SQL 语句的编写工作量较大，尤其当字段多、关联表多时，对开发人员编写 SQL 语句的功底有一定要求，SQL 语句依赖于数据库，导致数据库移植性差，不能随意更换数据库。

26. 请描述 MyBatis 如何进行批量操作?

mybatis 中提供了<foreach>标签，可以对集合、数组进行循环，从而实现动态的批量操作。

27. 请描述 MyBatis 如何获取自增的主键值?

Mybatis 中可以在<insert>标签中添加 usegeneratedkeys=”true” keyproperty=”

id”属性获取自增后的主键值，比如增加操作时传入的实体对象没有 id，调用带有该属性的增加操作后，则会将自增主键放入实体对象中。

也可以在<insert>标签中使用<selectKey>标签查询获取自增主键列数据。

28. 请描述 MyBatis 的 Mapper 功能方法如何传递多个参数?

Mybatis 如果参数是一个，或者实体对象，则直接可以使用#{名称或属性}取出对应的数据，那么如果不是实体对象，但是有多个参数，实现如下：

1） 使用 Map 集合封装传递的多个参数，因为 map 有键作为名称。

2） 使用@Param 注解实现参数名的标注。

29. Hibernate 如何实现分页查询?

技巧:hibernate 实现分页的关键是通过方言的方式隔离数据差异,在分页处理上,由于各种数据库处理差异很大,未在 hql 中实现相应关键字,是利用查询接口两个关键方法完成(setFirstResult,setMaxResult),说明这两个方法是关键。说明之后，可根据场景使用代码示例，进一步可阐述分页实现原理。

Hibernate 实现分页查询的关键 是两个方法 setFirstResult: 设置第一数据开始位置

setMaxResult :设置查询总记录数

30. @OneToMany 注解的 mappedBy 属性有什么作用?

技巧：mappedBy 属性的主要作用是告诉 hibernate 关联关系是通过对方外键引用完成关联映射,这一点必须说清楚。

One 的一方会成为主控方，加入 mappedBy 以后，就不会生成这两张关联表的中间表了，而是在 Many 的一方的表中生成一列作为外键。

31. MyBatis 中的动态 SQL 是什么意思?

MyBatis 的动态 SQL 是基于 OGNL 表达式的，它可以帮助我们方便的在 SQL 语句中实现某些逻辑。

MyBatis 中用于实现动态 SQL 的元素主要有：

if

choose（when，otherwise）

trim

where

set

foreach

32. hibernate 中的 update()和 saveOrUpdate()的区别

1、update 针对的是已存在的实体对象，数据库中如果没有实体所对应的数据记录，将会抛出异常；

2、saveOrUpdate()对象存在与否都不会有任何影响，有则更新，没则插入。

33. hibernate 中对象的三种状态?

1、瞬时态：用 new 创建的对象，它没有持久化，没有处于 Session 中，处于此状态的对象叫临时对象；

2、持久态：已经持久化，加入到了 Session 缓存中。如通过 hibernate 语句保存的对象。处于此状态的对象叫持久对象；

3、游离态（托管态）：持久化对象脱离了 Session 的对象。如 Session 缓存被清空的对象。特点：已经持久化，但不在 Session 缓存中。处于此状态的对象叫游离对象；

34. 如何理解 Hibernate 的延迟加载机制？在实际应用中，延迟加载与 Session 关闭的矛盾是如何处理。

延迟加载就是并不是在读取的时候就把数据加载进来，而是等到使用时再加载。Hibernate 使用了虚拟代理机制实现延迟加载，我们使用 Session 的 load()方法加载数据或者一对多关联映射在使用延迟加载的情况下从一的一方加载多的一方，得到的都是虚拟代理，简单的说返回给用户的并不是实体本身，而是实体对象的代理。代理对象在用户调用 getter 方法时才会去数据库加载数据。但加载数据就需要数据库连接。而当我们把会话关闭时，数据库连接就同时关闭了。

延迟加载与 session 关闭的矛盾一般可以这样处理：

① 关闭延迟加载特性。这种方式操作起来比较简单，因为 Hibernate 的延迟加载特性是可以通过映射文件或者注解进行配置的，但这种解决方案存在明显的缺陷。首先，出现"no session or session was closed"通常说明系统中已经存在主外键关联，如果去掉延迟加载的话，每次查询的开销都会变得很大。

② 在 session 关闭之前先获取需要查询的数据，可以使用工具方法 Hibernate.isInitialized()判断对象是否被加载，如果没有被加载则可以使用 Hibernate.initialize()方法加载对象。

③ 使用拦截器或过滤器延长 Session 的生命周期直到视图获得数据。Spring 整合 Hibernate 提供的 OpenSessionInViewFilter 和 OpenSessionInViewInterceptor 就是这种做法。

35. session 的 load()和 get()的区别?

1、get()采用立即加载方式；而 load()采用延迟加载；

2、get()方法执行的时候，会立即向数据库发出查询语句，而 load()方法返回的是一个代理(此代理中只有一个 id 属性)，只有等真正使用该对象属性的时候,才会发出 sql 语句；

3、如果数据库中没有对应的记录，get()方法返回的是 null。而 load()方法出现异常 ObjectNotFoundException 。

36. 介绍一下 Hibernate 的二级缓存。

1、按照以下思路来回答：（1）首先说清楚什么是缓存，（2）再说有了 hibernate 的 Session 就是一级缓存，即有了一级缓存，为什么还要有二级缓存，（3）最后再说如何配置 Hibernate 的二级缓存。

2.1、缓存就是把以前从数据库中查询出来和使用过的对象保存在内存中，当以后要使用某个对象时，先查询缓存中是否有这个对象，如果有则使用缓存中的对象，如果没有则去查询数据库，并将查询出来的对象保存在缓存中，以便下次使用。

2.2、Hibernate 中 Session 内置了一种缓存，我们通常将之称为 Hibernate 的一级缓存（Session 级别的），当想使用 session 从数据库中查询出一个对象时，Session 也是先从自己内部查看是否存在这个对象，存在则直接返回，不存在才去访问数据库，并将查询的结果保存在自己内部。由于 Session 代表一次会话过程，一个 Session 与一个数据库连接相关连，所以 Session 最好不要长时间保持打开，通常仅用于一个事务当中，在事务结束时就应关闭。

2.3、二级缓存是 SessionFactory 级别的缓存，能被此 SessionFactory 里的所有 Session 共享。多个厂商和组织都提供有缓存产品，例如，EHCache 和 OSCache 等等。在 Hibernate 中使用二级缓存，首先就要在 hibernate.cfg.xml 配置文件中配置使用哪个厂家的缓存产品，接着需要配置该缓存产品自己的配置文件，最后要配置 Hibernate 中的哪些实体对象要纳入到二级缓存的管理中。

37. 谈一谈 Hibernate 的一级缓存、二级缓存和查询缓存?

Hibernate 的 Session 提供了一级缓存的功能，默认总是有效的，当应用程序保存持久化实体、修改持久化实体时，Session 并不会立即把这种改变提交到数据库，而是缓存在当前的 Session 中，除非显示调用了 Session 的 flush()方法或通过 close()方法关闭 Session。通过一级缓存，可以减少程序与数据库的交互，从而提高数据库访问性能。

SessionFactory 级别的二级缓存是全局性的，所有的 Session 可以共享这个二级缓存。不过二级缓存默认是关闭的，需要显示开启并指定需要使用哪种二级缓存实现类（可以使用第三方提供的实现）。一旦开启了二级缓存并设置了需要使用二级缓存的实体类，SessionFactory 就会缓存访问过的该实体类的每个对象，除非缓存的数据超出了指定的缓存空间。

一级缓存和二级缓存都是对整个实体进行缓存，不会缓存普通属性，如果希望对普通属性进行缓存，可以使用查询缓存。查询缓存是将 HQL 或 SQL 语句以及它们的查询结果作为键值对进行缓存，对于同样的查询可以直接从缓存中获取数据。查询缓存默认也是关闭的，需要显示开启。

38. 简述 Hibernate 常见优化策略?

① 制定合理的缓存策略（二级缓存、查询缓存）。

② 采用合理的 Session 管理机制。

③ 尽量使用延迟加载特性。

④ 设定合理的批处理参数。

⑤ 如果可以，选用 UUID 作为主键生成器。

⑥ 如果可以，选用基于版本号的乐观锁替代悲观锁。

⑦ 在开发过程中，开启 hibernate.show_sql 选项查看生成的 SQL，从而了解底层的状况；开发完成后关闭此选项。

⑧ 考虑数据库本身的优化，合理的索引、恰当的数据分区策略等都会对持久层的性能带来可观的提升，但这些需要专业的 DBA（数据库管理员）提供支持。

39. 锁机制有什么用？简述 Hibernate 的悲观锁和乐观锁机制。

有些业务逻辑在执行过程中要求对数据进行排他性的访问，于是需要通过一些机制保证在此过程中数据被锁住不会被外界修改，这就是所谓的锁机制。

Hibernate 支持悲观锁和乐观锁两种锁机制。悲观锁，顾名思义悲观的认为在数据处理过程中极有可能存在修改数据的并发事务（包括本系统的其他事务或来自外部系统的事务），于是将处理的数据设置为锁定状态。悲观锁必须依赖数据库本身的锁机制才能真正保证数据访问的排他性。乐观锁，顾名思义，对并发事务持乐观态度（认为对数据的并发操作不会经常性的发生），通过更加宽松的锁机制来解决由于悲观锁排他性的数据访问对系统性能造成的严重影响。最常见的乐观锁是通过数据版本标识来实现的，读取数据时获得数据的版本号，更新数据时将此版本号加 1，然后和数据库表对应记录的当前版本号进行比较，如果提交的数据版本号大于数据库中此记录的当前版本号则更新数据，否则认为是过期数据无法更新。Hibernate 中通过 Session 的 get()和 load()方法从数据库中加载对象时可以通过参数指定使用悲观锁；而乐观锁可以通过给实体类加整型的版本字段再通过 XML 或@Version 注解进行配置。

40. 阐述 Session 加载实体对象的过程。

Session 加载实体对象的步骤是：

① Session 在调用数据库查询功能之前，首先会在一级缓存中通过实体类型和主键进行查找，如果一级缓存查找命中且数据状态合法，则直接返回；

② 如果一级缓存没有命中，接下来 Session 会在当前 NonExists 记录（相当于一个查询黑名单，如果出现重复的无效查询可以迅速做出判断，从而提升性能）中进行查找，如果 NonExists 中存在同样的查询条件，则返回 null；

③ 如果一级缓存查询失败则查询二级缓存，如果二级缓存命中则直接返回；

④ 如果之前的查询都未命中，则发出 SQL 语句，如果查询未发现对应记录则将此次查询添加到 Session 的 NonExists 中加以记录，并返回 null；

⑤ 根据映射配置和 SQL 语句得到 ResultSet，并创建对应的实体对象；

⑥ 将对象纳入 Session（一级缓存）的管理；

⑦ 如果有对应的拦截器，则执行拦截器的 onLoad 方法；

⑧ 如果开启并设置了要使用二级缓存，则将数据对象纳入二级缓存；

⑨ 返回数据对象。

41. 描述 Hibernate 对象状态转换过程。

瞬时状态 (Transient)

持久状态 (Persistent)

脱管状态 (Detached)

Save() 方法将瞬时对象保存到数据库，对象的临时状态将变为持久化状态。当对象在持久化状态时，它一直位于 Session 的缓存中，对它的任何操作在事务提交时都将同步到数据库，因此，对一个已经持久的对象调用 save()或 update() 方法是没有意义的。

update() 方法两种用途重新关联脱管对象为持久化状态对象，显示调用 update() 以更新对象。调用 update() 只为了关联一个脱管对象到持久状态，当对象已经是持久状态时，调用 update() 就没有多大意义了。

saveOrUpdate() 方法兼具 save() 和 update() 方法的功能，对于传入的对象， saveOrUpdate() 首先判断其是脱管对象还是临时对象，然后调用合适的方法。

42. 如何解决 hibernate “N+1”问题。

1.延迟加载，当需要的时候才查询，不需要就不查询，但是感觉这种方式治标不治本，尤其是在那种报表统计查询的时候更为明显。

2.fetch="join"，默认是 fetch="select"，这个其实说白了就是一个做外连接，允许外键为空的情况之下。

3.二级缓存，第一次查询之后存在内存中，后面的相同查询就快了。但是有 2 个缺点：a. 二级缓存首先是有点浪费内存空间，如果多了的话浪费还比较严重，这是一个不好的方面，当然这不是主要的，主要的问题在于，二级缓存的特性决定的，那就是很少的增删改才做二级缓存，而对

于普通的 CRUD 系统，其实不太适合。所以感觉也不是首选。

43. 如何解决 Mybatis “n+1”问题。

第一种方法是使用一条 SQL 语句。

第二种方法是使用 MyBatis 的延迟加载机制。

44. Mybatis 动态 SQL 相关标签作用及含义?

1.MyBatis 的动态 SQL 是基于 OGNL 表达式的，它可以帮助我们方便的在 SQL 语句中实现某些逻辑。

2.MyBatis 中用于实现动态 SQL 的元素主要有

if

choose（when，otherwise）

trim

where

set

foreach

可以看出 MyBatis 的动态 SQL 的标签元素和接近 JSP 中的 JSTL 语法，下面我就分别详细的介绍一下

3.动态 SQL 中 if 的用法

4.动态 SQL 中 choose 用法

5.动态 SQL 中 where 语句的作用主要是简化 SQL 语句中 where 中的条件判断的

注意：where 元素的作用是给 SQL 语句添加一个条件判断. 如果输出后是 and 开头的，MyBatis 会把第一个 and 忽略，当然如果是 or 开头的，MyBatis 也会把它忽略；此外，在 where 元素中你不需要考虑空格的问题，MyBatis 会智能的帮你加上

6.动态 SQL 中 set 元素主要是用在更新操作的时候，它的主要功能和 where 元素其实是差不多的

7.动态 SQL 中 foreach

主要用在构建 in 条件中，它可以在 SQL 语句中进行迭代一个集合。

foreach 元素的属性主要有 item，index，collection，open，separator，close

8.动态 SQL 中 trim 语句

45. 什么是 JDBC,简述实现方法?

创建一个以 JDBC 连接数据库的程序，包含 7 个步骤：

1、加载 JDBC 驱动程序： Class.forName(“com.mysql.jdbc.Driver”)

2、提供 JDBC 连接的 URL “jdbc:mysql://localhost:3306/数据库名”

3、创建数据库的连接 Connection conn = DriverManager.getConnection(“URL”,”root”,”123456”)

4、创建一个 Statement PreparedStatement pstmt = conn.prepareStatement(sql);

5、执行 SQL 语句 增删改 pstmt.executeUpdate() → int 查询 pstmt.executeQuery()→ResultSet

6、处理结果 while(rst.next())

7、关闭 JDBC 对象

46. 什么是事务?

1、事务是一系列的数据库操作，是数据库应用的基本逻辑单位，事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。

2、事务具有如下特性：（其中原子性最为重要，必须掌握）

原子性（atomicity） 一个事务是一个不可分割的工作单位，事务中包括的诸操作要么都做，要么都不做。

一致性（consistency） 事务必须是使数据库从一个一致性状态变到另一个一致性状态。

隔离性（isolation） 一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

持久性（durability） 持续性也称永久性（permanence），指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。

47. 请介绍一下数据库连接池技术?

1、数据库连接池技术，就是数据库启动时会建立一定数量的数据库连接（也称为池连接），并一直维持不少于此数目的池连接。

2、客户端程序需要连接数据库时，数据库连接池会返回一个未使用的池连接给数据库使用。如果当前没有空闲连接，数据库连接池就新建一定数量的连接。当使用的池连接调用完成后，连接池将此连接表记为空闲，其他调用就可以使用这个连接。这样做的目的是提高了应用程序访问数据库的性能。

48. MySQL、Oracle、SQL Server 各数据库服务的默认端口号?

1、MySQL：3306

2、Oracle：1521

3、SQL Server：1433

7.4 流行框架与技术

1. 什么是 Maven。

Maven 是一个项目管理工具，它包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的逻辑。当你使用 Maven 的时候，你用一个明确定义的项目对象模型来描述你的项目，然后 Maven 可以应用横切的逻辑，这些逻辑来自一组共享的（或者自定义的）插件。

Maven 有一个生命周期，当你运行 mvn install 的时候被调用。这条命令告诉 Maven 执行一系列的有序的步骤，直到到达你指定的生命周期。遍历生命周期旅途中的一个影响就是，Maven 运行了许多默认的插件目标，这些目标完成了像编译和创建一个 JAR 文件这样的工作。

此外，Maven 能够很方便的帮你管理项目报告，生成站点，管理 JAR 文件，等等。

2. 什么是 SVN。

SVN 是 Subversion 的简称，是一个开放源代码的版本控制系统，相较于 RCS、CVS，它采用了分支管理系统，它的设计目标就是取代 CVS。互联网上很多版本控制服务已从 CVS 迁移到 Subversion。说得简单一点 SVN 就是用于多个人共同开发同一个项目，共用资源的目的。

3. 什么是 GIT。

Git(读音为/gɪt/。)是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。[1] Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

4. Java 常用构建工具区别(ant,maven,gradle 区别)。

Java 世界中主要有三大构建工具：Ant、Maven 和 Gradle

目前：Ant 已经销声匿迹、Maven 也没落了，而 Gradle 的发展则如日中天。

Maven 的主要功能主要有，分别是依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制。

第一 Maven 为 Java 世界引入了一个新的依赖管理系统。在 Java 世界中，可以用 groupId、artifactId、version 组成的 Coordination（坐标）唯一标识一个依赖。任何基于 Maven 构建的项目自身也必须定义这三项属性，生成的包可以是 Jar 包，也可以是 war 包或者 ear 包。一个典型的依赖引用如下所示：

Gradle 在设计的时候基本沿用了 Maven 的这套依赖管理体系。不过它在引用依赖时还是进行了一些改进。首先引用依赖方面变得非常简洁。

第二，Maven 和 Gradle 对依赖项的 scope 有所不同。在 Maven 世界中，一个依赖项有 6 种 scope，

分别是 complie(默认)、provided、runtime、test、system、import。而 grade 将其简化为了 4 种，compile、runtime、testCompile、testRuntime。那么如果想在 gradle 使用类似于 provided 的 scope 怎么办？别着急，由于 gradle 语言的强大表现力，我们可以轻松编写代码来实现类似于 provided scope 的概念（例如 How to use provided scope for jar file in Gradle build?）。

第三点是 Gradle 支持动态的版本依赖。在版本号后面使用+号的方式可以实现动态的版本管理。

第四点是在解决依赖冲突方面 Gradle 的实现机制更加明确。

5. 常用源码版本管理工具有几种,并说明其优缺点(svn,git)。

其实 Git 和 SVN 还是挺像的，都有提交，合并等操作，看来这是源码管理工具的基本操作。

1. Git 是分布式的，SVN 是集中式的，好处是跟其他同事不会有太多的冲突，自己写的代码放在自己电脑上，一段时间后再提交、合并，也可以不用联网在本地提交；

2. Git 下载下来后，在本地不必联网就可以看到所有的 log，很方便学习，SVN 却需要联网；

3. Git 鼓励分 Branch，而 SVN，说实话，我用 Branch 的次数还挺少的，SVN 自带的 Branch merge 我还真没用过，有 merge 时用的是 Beyond Compare 工具合并后再 Commit 的；

4. Tortoise 也有出 Git 版本，真是好东西；

5. SVN 在 Commit 前，我们都建议是先 Update 一下，跟本地的代码编译没问题，并确保开发的功能正常后再提交，这样其实挺麻烦的，有好几次同事没有先 Updata，就

Commit 了，发生了一些错误，耽误了大家时间，Git 可能这种情况会少些。

6. 什么是 MVC?常用的 MVC 框架有哪些?

1、M:Model 模型层 主要用来处理业务逻辑，承载数据；

2、V:View 视图层 主要用来做页面显示的

3、C:Control 控制层 主要用来进行业务流程控制；

4、常用的 MVC 框架包括：Struts、Struts2、SpringMVC；谈谈 Struts2 的工作流程（或运行原理）。

1、客户端发送一个指向 Servlet 容器（例如 Tomcat）的请求

2、这个请求经过一系列的过滤器（Filter）

3、接着核心控制器 FilterDispatcher 被调用（2.3 以后是 StrutsPrepareAndExecuteFilter 过滤器），将 request 中所携带的数据放入值栈（ValueStack）；

4、 核心控制器询问 ActionMapper 来决定这个请求是否需要调用某个 Action，把请求的处理交给 ActionProxy；

5、 ActionProxy 通过 Configuration Manager 询问框架的配置文件(struts.xml)，找到调用的 Action

类；

6、 ActionProxy 创建一个 ActionInvocation 的实例；

7、 ActionInvocation 在调用 Action 的过程前后，涉及到相关拦截器（Intercepter）的调用；

8、 一旦 Action 执行完毕，ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果（JSP 显示结果）。

7. Struts 框架的优缺点。

优点：

1、实现 MVC 模式，结构清晰，使开发者只关注业务逻辑的实现；

2、有丰富的 tag 可以用 ,Struts 的标记库(Taglib)，如能灵活动用，则能大大提高开发效率；

3、通过配置文件，使系统的脉络更加清晰。一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着很大的好处。

4、Struts 提供了两种异常处理方式，声明式异常处理和编程式异常处理

5、对国际化进行支持，支持 I18N；

缺点：

1、实现 MVC 模式，代码复杂度提升；

2、需要维护配置文件，比较麻烦；

8. 说说 struts2 和 springmvc 的区别?

1. 拦截机制的不同

Struts2 是类级别的拦截，每次请求就会创建一个 Action，和 Spring 整合时 Struts2 的 ActionBean 注入作用域是原型模式 prototype，然后通过 setter，getter 吧 request 数据注入到属性。Struts2 中，一个 Action 对应一个 request，response 上下文，在接收参数时，可以通过属性接收，这说明属性参数是让多个方法共享的。Struts2 中 Action 的一个方法可以对应一个 url，而其类属性却被所有方法共享，这也就无法用注解或其他方式标识其所属方法了，只能设计为多例。

SpringMVC 是方法级别的拦截，一个方法对应一个 Request 上下文，所以方法直接基本上是独立的，独享 request，response 数据。而每个方法同时又何一个 url 对应，参数的传递是直接注入到方法中的，是方法所独有的。处理结果通过 ModeMap 返回给框架。在 Spring 整合时，SpringMVC 的 Controller Bean 默认单例模式 Singleton，所以默认对所有的请求，只会创建一个 Controller，有应为没有共享的属性，所以是线程安全的，如果要改变默认的作用域，需要添加@Scope 注解修改。

Struts2 有自己的拦截 Interceptor 机制，SpringMVC 这是用的是独立的 Aop 方式，这样导致 Struts2 的配置文件量还是比 SpringMVC 大。

2、底层框架的不同

Struts2 采用 Filter（StrutsPrepareAndExecuteFilter）实现，SpringMVC（DispatcherServlet）则采用 Servlet 实现。Filter 在容器启动之后即初始化；服务停止以后坠毁，晚于 Servlet。Servlet 在是在调用时初始化，先于 Filter 调用，服务停止后销毁。

3、性能方面

Struts2 是类级别的拦截，每次请求对应实例一个新的 Action，需要加载所有的属性值注入，SpringMVC 实现了零配置，由于 SpringMVC 基于方法的拦截，有加载一次单例模式 bean 注入。所以，SpringMVC 开发效率和性能高于 Struts2。

4、配置方面

spring MVC 和 Spring 是无缝的。从这个项目的管理和安全上也比 Struts2 高。

9. 什么是 AOP？谈谈你对他的理解。

1.AOP 为 Aspect Oriented Programming 的缩写，意为：面向切面编程（也叫面向方面），可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种思想。

2.利用 AOP 可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。 当项目中需要编写大量相同功能重复代码时，可以考虑使用 AOP 的形式来实现。如声明式事务管理、权限校验等。

备注：如果面试官问你是否用过 AOP，一定要说用过（Spring 的事务处理机制就是 AOP 的思想来实现的）——非常重要！！！

10. 什么是 IOC 和 DI?

1.IOC 是控制反转，DI 是依赖注入，其基本含义差不多。

2.IOC：原来创建对象的控制权是由我们自己来创建的，现在不再是由程序员创建了，由 IOC 容器为我们提供，这样带来的好处是，降低代码的耦合度，更加符合开闭原则；

备注：学生需要知道怎么在配置文件里配置相关信息，面试官可能会问。

11. 依赖注入的两种方式?

1.Set 注入

2.构造器注入

12. Web 应用开发中,如何启用 Spring 框架支持,写出关键配置。

在 WebApp 中获得 XMLWebApplicationContext 的步骤

1.在 Web.xml 中配置上下文载入器.

2.指定上下文载入器的配置文件.

3.获得应用上下文.

13. 简述 Spring Aop 实现方法(cglib,java proxy 差别)。

springAOP 主要通过 Java 动态代理的方式进行实现。

GCLIB 代理

cglib（Code Generation Library）是一个强大的,高性能,高质量的 Code 生成类库。它可以在运行期扩展 Java 类与实现 Java 接口。

cglib 封装了 asm，可以在运行期动态生成新的 class。

cglib 用于 AOP，jdk 中的 proxy 必须基于接口，cglib 却没有这个限制。

原理区别：

java 动态代理是利用反射机制生成一个实现代理接口的匿名类，在调用具体方法前调用 InvokeHandler 来处理。而 cglib 动态代理是利用 asm 开源包，对代理对象类的 class 文件加载进来，通过修改其字节码生成子类来处理。

1.如果目标对象实现了接口，默认情况下会采用 JDK 的动态代理实现 AOP

2.如果目标对象实现了接口，可以强制使用 CGLIB 实现 AOP

3.如果目标对象没有实现了接口，必须采用 CGLIB 库，spring 会自动在 JDK 动态代理和 CGLIB 之间转换

14. Spring 框架为企业级开发带来的好处有哪些?

1. IoC 容器：IoC 容器帮助应用程序管理对象以及对象之间的依赖关系，对象之间的依赖关系如果发生了改变只需要修改配置文件而不是修改代码，因为代码的修改可能意味着项目的重新构建和完整的回归测试。有了 IoC 容器，程序员再也不需要自己编写工厂、单例，这一点特别符合 Spring 的精神“不要重复的发明轮子”。

2. AOP：面向切面编程，将所有的横切关注功能封装到切面（aspect）中，通过配置的方式将横切关注功能动态添加到目标代码上，进一步实现了业务逻辑和系统服务之间的分离。另一方面，有了 AOP 程序员可以省去很多自己写代理类的工作。

3. MVC：Spring 的 MVC 框架是非常优秀的，从各个方面都可以甩 Struts 2 几条街，为 Web 表示层提供了更好的解决方案。

4. 事务管理：Spring 以宽广的胸怀接纳多种持久层技术，并且为其提供了声明式的事务管理，在不需要任何一行代码的情况下就能够完成事务管理。

15. SpringMVC 执行流程。

1.用户请求首先发送到前端控制器 DispatcherServlet，DispatcherServlet 根据请求的信息来决定使用哪个页面控制器 Controller（也就是我们通常编写的 Controller）来处理该请求。找到控制器之后，DispatcherServlet 将请求委托给控制器去处理。

2.接下来页面控制器开始处理用户请求，页面控制器会根据请求信息进行处理，调用业务层等等，处理完成之后，会把结果封装成一个 ModelAndView 返回给 DispatcherServlet。

3.前端控制器 DispatcherServlet 接到页面控制器的返回结果后，根据返回的视图名选择相应的试图模板，并根据返回的数据进行渲染。

4.最后前端控制器 DispatcherServlet 将结果（如 jsp）返回给用户。

16. 描述 Spring 事务的概念。

1. 事务简介：

事务管理是企业级应用程序开发中必不可少的技术，用来确保数据的完整性和一致性

事务就是一系列的动作，它们被当作一个单独的工作单元。这些动作要么全部完成，要么全部不起作用

2. 事务的四个关键属性(ACID)

① 原子性(atomicity):事务是一个原子操作，有一系列动作组成。事务的原子性确保动作要么全部完成，要么完全不起作用

② 一致性(consistency):一旦所有事务动作完成，事务就被提交。数据和资源就处于一种满足业务规则的一致性状态中

③ 隔离性(isolation):可能有许多事务会同时处理相同的数据，因此每个事物都应该与其他事务隔离开来，防止数据损坏

④ 持久性(durability):一旦事务完成，无论发生什么系统错误，它的结果都不应该受到影响。通常情况下，事务的结果被写到持久化存储器中

17. 描述 Spring 事务管理机制。

作为企业级应用程序框架，Spring 在不同的事务管理 API 之上定义了一个抽象层。而应用程序开发人员不必了解底层的事务管理 API，就可以使用 Spring 的事务管理机制。

Spring 既支持编程式事务管理(也称编码式事务)，也支持声明式的事务管理

编程式事务管理：将事务管理代码嵌入到业务方法中来控制事务的提交和回滚，在编程式事务中，必须在每个业务操作中包含额外的事务管理代码

声明式事务管理：大多数情况下比编程式事务管理更好用。它将事务管理代码从业务方法中分离出来，以声明的方式来实现事务管理。事务管理作为一种横切关注点，可以通过 AOP 方法模块化。Spring 通过 Spring AOP 框架支持声明式事务管理。

18. 描述 Spring 事务的方式。

如何在 Spring 配置文件中定义事务管理器：

声明对本地事务的支持：

a)JDBC 及 iBATIS、MyBatis 框架事务管理器

```
<bean id="txManager" class="org.springframework.jdbc.datasource.
        DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>
<tx:advice id=”txAdvice” transaction-manager=”txManager”>
    <tx:attributes>
        <tx:method name=”delete*” propagation=”REQUIRED” />
    </tx:attributes>
</tx:advice>

<aop:config>
    <aop:cutpoint expression=”execution(* com.muke.mytest.service.impl*.*(..))” id=”pointcut1”>
        <aop:advisor advice-ref=”txAdvice” pointcut-ref=”pointcut1” />
    </aop:cutpoint>
</aop:config>
```

19. 描述 Spring 事务的属性。

在 Spring 中，声明式事务是通过事务属性来定义的，事务属性描述了事务策略如何应用到方法上。尽管 Spring 提供了多种声明式事务的机制，但是所有的方式都依赖这五个参数来控制如何管理事务策略。声明式事务通过传播行为，隔离级别，只读提示，事务超时及回滚规则来进行定义。

Spring 事务的传播行为：

当事务方法被另一个事务方法调用时，必须指定事务应该如何传播。例如：方法可能继续在现有事务中运行，也可能开启一个新事务，并在自己的事务中运行。

事务的传播行为可以由传播属性指定。Spring 定义了 7 种传播行为：

1.Propagation.REQUIRED

方法被调用时自动开启事务，在事务范围内使用则使用同一个事务，否则开启新事务。

2.Propagation.REQUIRES_NEW

无论何时自身都会开启事务

3.Propagation.SUPPORTS

自身不会开启事务，在事务范围内则使用相同事务，否则不使用事务

4.Propagation.NOT_SUPPORTED

自身不会开启事务，在事务范围内使用挂起事务，运行完毕恢复事务

5.Propagation.MANDATORY

自身不开启事务，必须在事务环境使用否则报错

6.Propagation.NEVER

自身不会开启事务，在事务范围使用抛出异常

7.Propagation.NESTED

如果一个活动的事务存在，则运行在一个嵌套的事务中. 如果没有活动事务，则按 TransactionDefinition.PROPAGATION_REQUIRED 属性执行。需要 JDBC3.0 以上支持。

20. 描述 Spring 事务的隔离级别。

1. ISOLATION_DEFAULT： 这是一个 PlatfromTransactionManager 默认的隔离级别，使用数据库默认的事务隔离级别。另外四个与 JDBC 的隔离级别相对应

2. ISOLATION_READ_UNCOMMITTED： 这是事务最低的隔离级别，它充许令外一个事务可以看到这个事务未提交的数据。这种隔离级别会产生脏读，不可重复读和幻像读。

3. ISOLATION_READ_COMMITTED： 保证一个事务修改的数据提交后才能被另外一个事务读取。另外一个事务不能读取该事务未提交的数据

4. ISOLATION_REPEATABLE_READ： 这种事务隔离级别可以防止脏读，不可重复读。但是可能出现幻像读。

它除了保证一个事务不能读取另一个事务未提交的数据外，还保证了避免下面的情况产生(不可重复读)。

5. ISOLATION_SERIALIZABLE 这是花费最高代价但是最可靠的事务隔离级别。事务被处理为顺序执行。

除了防止脏读，不可重复读外，还避免了幻像读。

21. 描述 Spring 事务的只读、超时、回滚的原则。

1. Spring 事务的只读

“只读事务”并不是一个强制选项，它只是一个“暗示”，提示数据库驱动程序和数据库系统，这个事务并不包含更改数据的操作，那么 JDBC 驱动程序和数据库就有可能根据这种情况对该事务进行一些特定的优化，比方说不安排相应的数据库锁，以减轻事务对数据库的压力，毕竟事务也是要消耗数据库的资源的。“只读事务”仅仅是一个性能优化的推荐配置而已，并非强制你要这样做不可。

2. Spring 事务的事务超时

为了使应用程序更好的运行，事务不能运行太长的时间。因此，声明式事务的第四个特性就是超时。

3. Spring 事务的回滚规则

默认情况下，事务只有在遇到运行期异常时才会回滚，而在遇到检查型异常时不会回滚，但是

也可以声明事务在遇到特定的检查型异常时像遇到运行期异常那样回滚。同样，你还可以声明事务遇到特定的异常不回滚，即使这些异常是运行期异常。

22. Spring Boot 的优点。

1.创建独立的 Spring 应用程序

2.嵌入式的 Tomcat，不需要部署 war 包

3.简化 Maven 配置

4.自动配置 Spring

5.提供生产就绪型功能，如指标，健康检查，和外部配置

6.开箱即用，没有代码生成，也无需 XML 配置

23. Spring 中自动装配的方式有哪些?

- no：不进行自动装配，手动设置 Bean 的依赖关系。

- byName：根据 Bean 的名字进行自动装配。

- byType：根据 Bean 的类型进行自动装配。

- constructor：类似于 byType，不过是应用于构造器的参数，如果正好有一个 Bean 与构造器的参数类型相同则可以自动装配，否则会导致错误。

- autodetect：如果有默认的构造器，则通过 constructor 的方式进行自动装配，否则使用 byType 的方式进行自动装配。

24. Spring 中的自动装配有哪些限制?

- 如果使用了构造器注入或者 setter 注入，那么将覆盖自动装配的依赖关系。
- 基本数据类型的值、字符串字面量、类字面量无法使用自动装配来注入。
- 优先考虑使用显式的装配来进行更精确的依赖注入而不是使用自动装配。

25. Spring JavaConfig 与 Xml 配置优缺点。

xml 配置

优势:集中配置,代码配置分离更加方便管理

劣势:繁杂,编译期不容易发现错误

javaConfig 配置

优势:代码简洁,

劣势:国内 xml 配置方式比较多,不容易被人接受

26. Spring IOC 容器中 Bean 范围（scope）有几个?

singleton

prototype 多例即原型

Request 和 http 请求关联

Session 会话

application 应用程序

27. 阐述 Spring 框架中 Bean 的生命周期?

① Spring IoC 容器找到关于 Bean 的定义并实例化该 Bean。

② Spring IoC 容器对 Bean 进行依赖注入。

方法被调用。

⑧ 当销毁 Bean 实例时，如果 Bean 实现了 DisposableBean 接口，则调用其 destroy 方法。

8.常规技术面试题（Web 前端）

8.1 HTML、CSS 基础

1. 简述 html、http、www、URL、web、W3C 的意思

Html:超文本标记语言

http:超文本传输协议

www: 是环球信息网（World Wide Web）的缩写，也可以简称为 Web，中文名字为“万维网”。

URL:统一资源定位器

web: 万维网

W3C：万维网联盟（World Wide Web Consortium，简称 W3C）

2. 介绍一下 HTML 的基本结构

```
<html>
    <head>
        <title>标题</title>
    </head>
    <body>
      ......
    </body>
</html>
```

3. 谈谈你了解的图片格式

主要有：jpg、gif、bmp、png 等等，其中 jpg 的颜色最丰富，gif 支持动态图片，png 支持透明

4. HTML 有哪些基本标签?

格式标签：段落<p>、居中 、编号列表<ol>、项目列表<ul>、自定义列表<dl>、预定义格式<pre>、环绕标签<marquee>、文本：h1-h6、font 等等、图像：<img/>、超链接 :<a>

5. marquee 标签的作用？如何使用?

marquee 标签的主要作用是做滚动字幕,格式如下：

<marquee direction=”滚动方向” scrolldelay=”滚动间隔时间”>……</marquee>

但是现在网页中很少使用 marquee 标签了，都会是使用 js 脚本或是前端的框架实现，效果会比 marquee 好的多。

6. HTML 表格的基本结构与常用属性有哪些?

布局基本结构如下：

```
<table>
  <caption>标题</caption>
  <tr>
    <th>列标题</th>
    ......
  </tr>
  <tr>
    <td>单元格内容</td>
    .....
  </tr>
  .....
</table>
```

常用属性如下：

1）border:边框

2）bordercolor:边框的颜色

3）width:宽度

4）height:高度

5）background:背景图片

6）bgcolor:背景颜色

7）align:水平对齐：可以取 left|center|right

8）valign:垂直对齐：可以取 top|middle|bottom

9）colspan:跨列合并

10）rowspan:跨行合并

11）cellpadding:内容与边框之间的距离，边距

12）cellspacing:边框与边框之间的距离，间距

7. CSS 是什么？使用 CSS 有哪些优势?

CSS 是 Cascading Style Sheets 的英文缩写,即层叠样式表

CSS 的优势：

1．实现了内容与表现的分离

2．实现了许多网页一起更新

3．减少网页的代码量,增加网页的浏览速度

8. CSS 样式规则有哪几个部分组成?

由选择器、属性名、属性值组成

eg:

```
p    /*  选择器  */
{
    color:red;  /*  一条样式规则  */
    font-size:20px;/*  一条样式规则  */
}
```

注意：样式规则之间用;隔开，属性名和属性值用:隔开

9. 写 CSS 有哪几个位置?

写 CSS 有 3 个位置，

（1）行内样式：<标签名 style=”属性 1:值 1; 属性 2:值 2;…”>

（2）内部样式：在 head 标签里写

<style type=”text/css”> CSS 的语法 </style>

（3）外部样式：<link href=”css 文件路径” rel=”stylesheet” />

10. CSS 样式的优先级

1.按 css 样式的位置分：行内>内部>外部

2.按选择器分：id 选择器样式>class 类选择器样式>标签选择器样式

11. CSS 样式文件引入的方式有哪些? link 和@import 的区别是?

link 连接方式和 import 导入方式，两者区别如下：

1）link 链接时，不用写<style></style>

2）import 导入时，一定要写<style></style>

3）一般而言，链接的效率要高于导入（因为导入样式表相当于是将所有的样式规则原样复制到当前源文件中，增加了代码量，从而影响了加载的速度）

12. 边框样式常用属性有哪些?

border-width:边框宽度

border-style:边框样式(虚线、实线、双线等)

border-color:边框颜色

13. display 属性有什么作用？它有哪些值?

display:none　　隐藏标签元素

display:inline　在同一行显示

display:block　　以块状方式独占一行显示

14. 常见的页面布局方式有哪些?

表格布局、框架布局、DIV+CSS 布局

15. 什么是框架布局？如何进行框架布局?

框架布局是采用 frameset 将浏览器窗口分成几个独立的窗口，分别显示不同的页面的布局方式。格式如下（假定做一个上下框架，并将下框架拆分为左右框架）：

```
<frameset rows="93,*" frameborder="0">
    <frame    src="top.html" />
    <frameset    cols="180,*">
        <frame src="left.html" name="left"/>
        <frame src="sys.html" name="right"/>
    </frameset>
</frameset>
```

16. 什么是 DIV 布局？DIV+CSS 布局有哪些优点?

DIV 布局用 DIV 盒模型结构给各部分内容划分到不同的区块，然后用 css 来定义盒模型的位置、大小、边框、内外边距、排列方式等

优点如下：

表现和内容相分离

提高搜索引擎对网页的索引效率

代码简洁，提高页面浏览速度

易于维护和改版

17. 分别介绍一下页面元素定位中的绝对定位、相对定位、流动定位、浮动定位，并阐述他们的区别

1）流动定位：

保持在标准文档流中原有的位置，元素是按从左往右，从上往下的顺序摆放， left,top 等属性无效

2）相对定位：

相对定位是指相对于盒子元素原有的位置进行偏移，不会脱离标准文档流，也不对其他元素产

生任何影响。需设定垂直方向和水平向的偏移量，分别是 left、top、right、bottom。

3）绝对定位：

绝对定位是以某一个点为基准进行偏移。绝对定位是以最近的一个已定位的父级元素为基准，如果父级元素没有定位或没有父级元素，则以浏览器窗口为基准；会脱离标准文档流，不影响同一级的盒子元素位置。

4）浮动定位：只能设置水平方向上的定位，指内容停靠在容器的左边还是右边。垂直方向上不能设浮动定位。

18. 如何使用 Dreamweaver 建立本地站点和 WEB 站点

存放在本地磁盘上的网站称为本地站点，存放在 Web 服务器中的网站称为远程站点。

使用 Dreamweaver 在“站点”菜单中选中相应的菜单项根据提示选择路径确定即可

19. 如何创建模板页?

创建步骤如下：

1.使用面板工具创建模板

2.将网页另存为模板

3.将需要改变的位置设置为可编辑区域

20. 盒子模型由哪几个部分组成?

盒子模型包含：margin(外补丁)、border(边框)、padding(内补丁)、content(内容)

21. 谈一下 DIV+CSS 常见的背景特效?

鼠标悬停时使用 background 属性动态改变背景。滑动门特效 ，CSS Sprites 等等，除了常见的特效外，H5 和各种前端框架中也包含大量的背景特效并且实现也比较简单。

22. GBK,gb2312,UTF-8 的区别

utf-8 是基于 unicode 的国际化的场合适合使用

gb2312 和 gb2312 都是国标码，出现的较早。主要用于编解码常用汉字。gb2312 和 gbk 的一个区别是 :gb2312 出来得比较早，所以有一些汉字和繁体好象不支持。gbk 是一个改进版。所以能用 gbk 的时候一般不用 gb2312

23. 谈谈你见过的浏览器不兼容问题？如何使用 IETest?

1）以火狐代表的浏览器不支持如下用法：

1.document.all

2.unload 事件

3.innerText 属性

2）很多版本的浏览器不支持 clear:both 来清除浮动

3）各浏览器默认的 margin 和 padding 不同。解决方法： *{margin:0;padding:0;}

4）对盒子的解析标准不同

IETest 安装之后，即可以使用，可用来检查当前代码在不同浏览器版本中的兼容性

24. 一个 200*200 的 div 在不同分辨率屏幕中上下左右居中，用 css 实现

```
body{margin: 0;}
#div1{ width:200px;height: 200px;position: absolute;background-color: aqua;
left: 50%;top:50%;margin-left: -100px;margin-top: -100px;}
```

25. display:none 和 visibility:hidden 的区别是

1.display:none 是彻底消失，不在文档流中占位，浏览器也不会解析该元素；visibility:hidden 是视觉上消失了，可以理解为透明度为 0 的效果，在文档流中占位，浏览器会解析该元素；

2.使用 visibility:hidden 比 display:none 性能上要好，display:none 切换显示时 visibility，页面产生回流（当页面中的一部分元素需要改变规模尺寸、布局、显示隐藏等，页面重新构建，此时就是回流。所有页面第一次加载时需要产生一次回流），而 visibility 切换是否显示时则不会引起回流。

26. 在 HTML 中，table 标签的 cellpadding 属性和 cellspacing 属性的区别?

cellpadding:表示单元格边框与单元格内容的距离

cellspacing:表示单元格边框与另一个单元格边框的距离

27. html5 有哪些新特性，如何处理 html5 对低版本的兼容问题

新特性：

1. 拖拽释放(Drag and drop) API
2. 语义化更好的内容标签（header,nav,footer,aside,article,section）
3. 音频、视频 API(audio,video)
4. 画布(Canvas) API
5. 地理(Geolocation) API
6. 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
7. sessionStorage 的数据在浏览器关闭后自动删除
8. 表单控件，calendar、date、time、email、url、search
9. 新的技术 webworker, websocket, Geolocation

HTML5 标签的兼容性解决方式：

1. 使用 document.createElement(tag)来解决
2. 使用 html5shiv 库来兼容

28. 对 JSON 的理解

JSON：JavaScript Object Notation(JavaScript 对象表示法)

JSON 是存储和交换文本信息的语法,类似 XML。

JSON 比 XML 更小、更快、更容易解析。

29. px 和 em 和 rem 的区别

px 特点

1. IE 无法调整那些使用 px 作为单位的字体大小；

2. 国外的大部分网站能够调整的原因在于其使用了 em 或 rem 作为字体单位；

3. Firefox 能够调整 px 和 em，rem，但是 96%以上的中国网民使用 IE 浏览器(或内核)。

px 像素（Pixel）。相对长度单位。像素 px 是相对于显示器屏幕分辨率而言的。

em 特点

1. em 的值并不是固定的；

2. em 会继承父级元素的字体大小。

3. em 是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。

（任意浏览器的默认字体高都是 16px。所有未经调整的浏览器都符合: 1em=16px。那么 12px=0.75em,10px=0.625em。为了简化 font-size 的换算，需要在 css 中的 body 选择器中声明 Font-size=62.5%，这就使 em 值变为 16px*62.5%=10px, 这样 12px=1.2em, 10px=1em, 也就是说只需要将你的原来的 px 数值除以 10，然后换上 em 作为单位就行了。）

rem 特点

rem 是 CSS3 新增的一个相对单位（root em，根 em）。与 em 的区别在于使用 rem 为元素设定字体大小时，仍然是相对大小，但相对的只是 HTML 根元素。rem 集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了 IE8 及更早版本外，所有浏览器均已支持 rem。对于不支持它的浏览器，多写一个绝对单位的声明。这些浏览器会忽略用 rem 设定的字体大小。下面就是一个例子：p {font-size:14px; font-size:.875rem;}

选择使用什么字体单位主要由你的项目来决定，如果你的用户群都使用最新版的浏览器，那推荐使用 rem，如果要考虑兼容性，那就使用 px,或者两者同时使用。

30. 页面有大量图片怎么优化

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为 css 图片，可以使用 CSSsprite，SVGsprite，Iconfont、Base64 等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

31. CSS 选择器有哪几种并作出说明，以及优先级是怎么样的

CSS 选择器：

1.标签名选择器 div { color：Red;} /即页面中的各个标签名的 css 样式

2.类选择器 .divClass {color:Red;} /即定义的每个标签的 class 中的 css 样式

3.ID 选择器 #myDiv {color:Red;} /即页面中的标签的 id

4.后代选择器（类选择器的后代选择器） .divClass span { color:Red;} /即多个选择器以逗号的格式分隔 命名找到准确的标签

5.群组选择器 div,span,img {color:Red} /即具有相同样式的标签分组显示

选择器的优先级:

行类样式>ID 选择器>类选择器>标签选择器

32. CSS 有主要有哪些优点(列举不少于 2 条)?

1.减少代码量，提高网页的浏览速度

2.内容与表现分离，便于更新

3.有利于搜索引擎的搜索

33. 表单的 method 方法有哪 2 种，解释他们的区别?

post:可以发送大量数据，安全。

get :只能发送少量数据，不够安全，但是速度快。

34. 列举常用的字体样式(不少于 4 个)?

font-size:大小

color:颜色

font-family:字体

font-weight:粗细

font-style:样式

35. 行级标签和块状标签的区别?

行级标签：不会独占一行；大小根据内容自适应，设置 width 属性和 height 属性无效

块状标签：独占一行；可以设置 width 属性和 height 属性

36. HTML 及 CSS 注释怎么表示?

html 注释：<!-- 注释内容 -->

css 注释：/* 注释内容 */

37. css 中内补丁和外补丁是什么意思，有什么作用

内补丁指：padding，对应内边距

外补丁指：margin，对应外边距

38. 在 HTML 中，创建提交按钮、重置按钮和普通按钮的语句分别是?

提交按钮：<input type="submit"/>

重置按钮：<input type="reset" />

普通按钮：<input type="button"/>

39. 在 HTML 中，样式规则：margin:1px 2px;表示什么意思?

margin 表示修饰元素的外边距

上下外边距为 1px，左右外边距为 2px

40. 在 HTML 中，样式规则：padding:1px 2px 3px 4px;表示什么意思?

padding:表示修饰元素的内间距

上内间距为 1px，右内间距为 2px，下内间距为 3px，左内间距为 4px

41. 在 HTML 中，样式规则：margin:1px 2px 3px;表示什么意思?

margin 表示修饰元素的外边距

上外边距为 1px，左右外边距为 2px，下外边距为 3px

42. 行级标签和块状标签的区别?

1）行级标签：不会独占一行；大小根据内容自适应，设置 width 属性和 height 属性无效

2）块状标签：独占一行；可以设置 width 属性和 height 属性

43. display 的三个常见属性值 none,inline,block 的意思分别是?

none:隐藏标签元素

inline:在同一行显示

block:以块状方式独占一行显示

44. 常见的页面布局主要有哪三种?

1）表格布局

2）框架布局

3）div+css 布局

8.2 JavaScript 基础

1. JS 有哪些数据类型

Number

Boolean

Null

Undefined

2. 谈一下你对 JS 的理解

特点：

1）寄生虫语言，必须嵌套在 HTML 中，以 HTML 为宿主

2）弱类型语言

3）基于对象（不是完全的 OOP 语言）

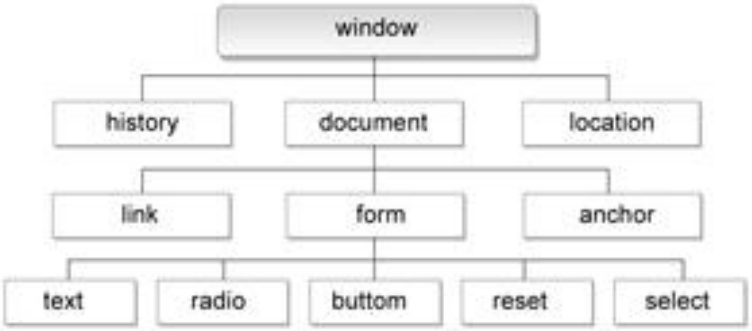
作用：

1）将以前必须在服务器端做的验证处理放在客户端运行，从而减轻了服务器的负担

2）增强了页面的交互性

3. 说一下你对浏览器模型的理解

浏览器对象模型是用于描述对象与对象之间层次关系的模型，该对象模型提供了独立于内容的、可以与浏览器窗口进行互动的对象结构层次如下：



4. JavaScript 获取页面元素有哪四种常见的访问形式?

方法一：document.all.页面元素名称

方法二：document.表单名.元素名称

方法三：document.getElementBId("ID 名称")

方法四：document.getElementsByName("元素名称")

5. alert 和 confim 有什么区别?

答：Alert 是弹出提示消息框；Confirm 是弹出确认框，根据用户的选择将返回 ture 或 false

6. 什么是 JS 内存泄露？如何处理?

答：内存泄露是指一块被分配的内存既不能使用，又不能回收，直到浏览器进程结束，由于浏览器垃圾回收方法有 bug，会产生内存泄露。

常见的处理方法如下：

对浏览器对优化

对大规模的循环代码进行优化

尽量避免过多的引用层级和不必要的多次方法调用

尽量使用语言本身的构造和内建函数

尽量减少不必要的对象创建

7. 谈一下你知道的 JS 事件，做项目时一般你在什么时候使用这些事件?

1.onload 和 onunload 事件 ，页面加载和页面卸载时使用

2.onfocus、onblur、onsubmit 和 onchange 事件,得到焦点，失去焦点，提交，内容改变时使用

3.onmouseover 和 onmouseout 事件，鼠标进入和移出时使用

4.onClick 事件，单击时使用

8.3 Ajaxl\jquery 基础

1. 什么是 ajax，为什么要使用 Ajax（请谈一下你对 Ajax 的认识）

什么是 ajax：AJAX 直译为异步的 JavaScript 和 XML，是一种异步提交、局部刷新的网页编程技术。

Ajax 应用程序的优势在于：

1. 通过异步模式，提升了用户体验。

2. 优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了带宽占用。

3. Ajax 引擎在客户端运行，承担了一部分本来由服务器承担的工作，从而减少了大用户量下的服务器负载。

2. Ajax 的最大的特点是什么。

Ajax 可以实现异步提交、局部刷新，这使得 Web 应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

3. HTTP 状态

100-199 用于指定客户端相应的某些动作。

200-299 用于表示请求成功

300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息

400-499 用于支出客户端的错误

500-599 用于支持服务错误

200 服务器成功返回页面

404 请求的网页不存在

503 服务不可用

4. 请介绍一下 XmlHttpRequest 对象?

Ajax 的核心是 JavaScript 对象 XmlHttpRequest，它是一种支持异步请求的技术。XmlHttpRequest 可以使得程序员使用 JavaScript 向服务器进行请求并处理响应，而不阻塞用户。通过 XMLHttpRequest 对象，Web 开发人员可以在页面加载以后进行页面的局部更新。

5. Ajax 技术体系的组成部分有哪些?

HTML，css，dom，xml，xmlHttpRequest，javascript

6. AJAX 应用和传统 Web 应用有什么不同?

在传统的 Web 应用的请求/响应为同步模式。即当服务器端在处理客户端请求时，客户端需要等待，直到服务器端响应返回后，客户端才能继续执行。

而 AJAX 应用的请求/响应为异步模式，即当服务器端在处理客户端请求时，客户端无须等待可以继续执行；当服务器端响应返回后，客户端进行局部刷新。

7. AJAX 请求总共有多少种 CALLBACK

Ajax 请求总共有八种 Callback

onSuccess

onFailure

onUninitialized

onLoading

onLoaded

onInteractive

onComplete

onException

8. Ajax 和 javascript 的区别?

javascript 是一种在浏览器端执行的脚本语言，Ajax 是一种创建交互式网页应用的开发技术 ，它是利用了一系列相关的技术其中就包括 javascript。

9. 在浏览器端如何得到服务器端响应的 XML 数据

XMLHttpRequest 对象的 responseXMl 属性

10. XMLHttpRequest 对象在 IE 和 Firefox 中创建方式有没有不同?

IE 中通过 new ActiveXObject()创建，Firefox 中通过 new XMLHttpRequest()创建

11. 介绍一下 XMLHttpRequest 对象的常用方法和属性（回答的越多越好）

open()方法：建立对服务器的调用。

send()方法：发送具体请求

abort()方法：停止当前请求

readyState 属性：返回请求的状态

responseText 属性：服务器端响应的文本

reponseXML 属性：服务器端响应的 XML

Status:服务器的 HTTP 状态码

12. Ajax 的优点和缺点

优点：

1、改善用户体验

2、减少带宽占用，及降低服务器端处理开销

缺点：

1、破坏了浏览器的前进、后退功能

2、对搜索引擎不友好

3、存在浏览器兼容性问题

13. 什么是 XML

XML 是可扩展标记语言，用于跨平台的数据存储和交互。

XML 文件格式是纯文本格式，在许多方面类似于 HTML，例如 XML 由 XML 元素组成，每个 XML 元素包括开始标记和结束标记以及两个标记之间的内容。

XML 与 HTML 的区别：

(1)可扩展性方面：HTML 不允许用户自行定义他们自己的标识或属性，而在 XML 中，用户能够根据需要自行定义新的标识及属性名，以便更好地从语义上修饰数据。

(2)结构性方面：HTML 不支持深层的结构描述，XML 的文件结构嵌套可以复杂到任意程度，能表示面向对象的等级层次。

(3)可校验性方面：HTML 没有提供规范文件以支持应用软件对 HTML 文件进行结构校验，而 XML 文件可以包括一个语法描述，使应用程序可以对此文件进行结构校验。

14. xml 的解析方式

常用的 dom 解析和 sax 解析。dom 解析是一次性读取 xml 文件并将其构造为 DOM 对象供程序使用，优点是操作方便，但是比较耗内存。Sax 是按事件驱动的方式解析的，占用内存少，但是编程复杂。

15. 你采用的是什么框架?

目前流行的 JS 框架：Node.Js、angular.Js、JQuery、Vue.js、React.js 等。

16. 如果熟悉某种 ajax 框架，他可能会问到怎样在程序中使用这种框架

ASP.NETAjax 集成了一套客户端脚本库使得与功能丰富的、基于服务器开发平台的 ASP.NET 结合在一起。其服务器端编程模型相对于客户端编程模型较为简单，而且容易与现有的 ASP.NET 程序相结合，通常实现复杂的功能只需要在页面中拖几个控件，而不必了解深层次的工作原理，除此之外服务器端编程的 ASP.NETAJAXControlToolkit 含有大量的独立 AJAX 控件和对 ASP.NET 原有服务

器控件的 AJAX 功能扩展。

16，介绍一下 Prototype 的$()函数，$F()函数，$A()函数都是什么作用？

$() 方法是在 DOM 中使用过于频繁的 document.getElementById() 方法的一个便利的简写，就像这个 DOM 方法一样，这个方法返回参数传入的 id 的那个元素。

$F()函数是另一个大收欢迎的“快捷键”，它能用于返回任何表单输入控件的值，比如 text box,drop-down list。这个方法也能用元素 id 或元素本身做为参数。

$A()函数能把它接收到的单个的参数转换成一个 Array 对象。

17. 介绍一下 XMLHttpRequest 对象

通过 XMLHttpRequest 对象，Web 开发人员可以在页面加载以后进行页面的局部更新。

AJAX 开始流行始于 Google 在 2005 年使用的”Google Suggest”。

“Google Suggest”就是使用 XMLHttpRequest 对象来创建动态的 Web 接口：

当用户开始输入 google 的搜索框，Javascript 发送用户输入的字符到服务器，然后服务器返回一个建议列表。

XMLHttpRequest 对象在 IE5.0+, Safari 1.2, Mozilla 1.0/Firefox, Opera 8+ 和 NetScapt7 开始被支持。

18. AJAX 应用和传统 Web 应用有什么不同?

在传统的 Javascript 编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。用户需要点击”Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。

因为服务器每次都会返回一个新的页面， 所以传统的 web 应用有可能很慢而且用户交互不友好。

使用 AJAX 技术， 就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。

通过 HTTP Request， 一个 web 页面可以发送一个请求到 web 服务器并且接受 web 服务器返回的信息(不用重新加载页面)，展示给用户的还是通一个页面，用户感觉页面刷新，也看不到到 Javascript 后台进行的发送请求和接受响应。

8.4 前端开发晋级

1. 你能描述一下渐进增强和优雅降级之间的不同吗?

- 定义:

- 优雅降级（graceful degradation): 一开始就构建站点的完整功能，然后针对浏览器测试和修复
- 渐进增强（progressive enhancement): 一开始只构建站点的最少特性，然后不断针对各浏览器追加功能。
- 都关注于同一网站在不同设备里不同浏览器下的表现程度

➢ 区别:

- “优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站. 而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。
- “渐进增强”观点则认为应关注于内容本身。

➢ 理解:

- "优雅降级"就是首先完整地实现整个网站，包括其中的功能和效果。然后再为那些无法支持所有功能的浏览器增加候选方案，使之在旧式浏览器上以某种形式降级体验却不至于完全失效。
- "渐进增强"则是从浏览器支持的基本功能开始, 首先为所有设备准备好清晰且语义化的 html 及完整内容, 然后再以无侵入的方法向页面增加无害于基础浏览器的额外样式和功能。当浏览器升级时, 它们会自动呈现并发挥作用。

2. 图片默认有间距

➢ 几个 img 标签放在一起的时候，有些浏览器会有默认的间距。

➢ 解决: 使用 float 属性为 img 布局。

3. 标签最低高度设置 min-height 不兼容

➢ 因为 min-height 本身就是一个不兼容的 CSS 属性，所以设置 min-height 时不能很好的被各个浏览器兼容

➢ 如果我们要设置一个标签的最小高度 200px，需要进行的设置为：{min-height:200px; height:auto !important; height:200px; overflow:visible;}

4. 透明度的兼容 CSS 设置

➢ IE：filter:alpha(opacity=50)

➢ 非 IE：opacity:0.5

5. IE ol 的序号全为 1, 不递增

➢ 解决: li 设置样式{display: list-item}

6. 如何对网站的文件和资源进行优化?

1. 文件合并
2. 文件最小化/文件压缩
3. 使用 CDN 托管
4. 使用缓存
5. css 文件放置在 head，js 放置在文档尾
6. css sprites
7. 图片延迟加载
8. 将资源放在不同的域名下
9. 书写代码的时候注意优化

7. 关于浏览器内核(渲染引擎)

- 现在的主要浏览器：IE、Firefox、Safari、Chrome、Opera。它们的浏览器内核（渲染引擎）：
  - IE--Trident、
  - FF(Mozilla)--Gecko、
  - Safari--Webkit、
  - Chrome--Blink（WebKit 的分支）、
  - Opera--原为 Presto，现为 Blink。

8. 浏览器加载文件(repaint/reflow)

- 文件加载顺序
  - 浏览器加载页面上引用的 CSS、JS 文件、图片时，是按顺序从上到下加载的，每个浏览器可以同时下载文件的个数不同，因此经常有网站将静态文件放在不同的域名下，这样可以加快网站打开的速度。
- reflow
  - 在加载 JS 文件时，浏览器会阻止页面的渲染，因此将 js 放在页面底部比较好。因为如果 JS 文件中有修改 DOM 的地方，浏览器会倒回去把已经渲染过的元素重新渲染一遍，这个回退的过程叫 reflow。
  - CSS 文件虽然不影响 js 文件的加载，但是却会造成 js 执行的阻塞。因为 js 代码执行前，浏览器必须保证 css 文件已下载和解析完成。
  - 办法：当 js 文件不需要依赖 css 文件时，可以将 js 文件放在头部 css 的前面。
- repaint
  - repaint（重绘）和 reflow 相似，但是在元素改变样式的时候触发，这个比 reflow 造成的影响要小，所以能触发 repaint 解决的时候就不要触发 reflow。

9. 为什么利用多个域名来请求网络资源会更有效?

- 动静分离的需求。使用不同的服务器处理请求，处理动态内容的只处理动态内容，提高效率。
- 突破浏览器并发限制：浏览器同一时间针对同一域名下的请求有一定数量限制，超过限制数目的请求会被阻止。
- Cookieless 节省带宽。因为客户端的每次请求，都会带上自己的 cookie 。
- 节约主域名的连接数，从而提高客户端网络带宽的利用率，优化页面响应。
- 避免不必要的安全问题(上传 js 窃取主站 cookie 之类的)

10. 进程和线程的区别

最根本区别：进程是操作系统资源分配的基本单位，而线程是 CPU 调度和执行的基本单位

1）在开销方面：每个进程都有独立的代码和数据空间（程序上下文），程序之间的切换会有较大的开销；线程可以看做轻量级的进程，同一类线程共享代码和数据空间，线程之间切换的开销小。

2）所处环境：在操作系统中能同时运行多个进程（程序）；而在同一个进程（程序）中有多个线程同时执行（通过 CPU 调度，在每个时间片中只有一个线程执行）

3）内存分配方面：系统在运行的时候会为每个进程分配不同的内存空间；而对线程而言，除了 CPU 外，系统不会为线程分配内存（线程所使用的资源来自其所属进程的资源），线程组之间只能共享资源。

4）包含关系：没有线程的进程可以看做是单线程的，如果一个进程内有多个线程，则执行过程不是一条线的，而是多条线（线程）共同完成的；线程是进程的一部分，所以线程也被称为轻权进程或者轻量级进程。

11. 前端开发的优化问题

- 前端开发的优化问题：

(1)减少 http 请求次数：css spirit,data uri

(2)JS，CSS 源码压缩

(3)前端模板 JS+数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数

(4)用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能

(5)用 setTimeout 来避免页面失去响应

(6)用 hash-table 来优化查找

(7)当需要设置的样式很多时设置 className 而不是直接操作 style

(8)少用全局变量

(9 缓存 DOM 节点查找的结果

(10)避免使用 CSS Expression

(11)图片预载

(12)避免在页面的主体布局中使用 table，table 要等其中的内容完全下载之后才会显示出来，显示比 div+css 布局慢

- 如何控制网页在网络传输过程中的数据量

(1)启用 GZIP 压缩

(2)保持良好的编程习惯，避免重复的 CSS、JavaScript 代码、多余的 HTML 标签和属性

12. Flash、Ajax 各自的优缺点，在使用中如何取舍?

- Ajax 的优势

(1) 可搜索型

(2) 开放性

(3) 易用性

(4) 易于开发

- Flash 的优势

(1)多媒体处理能力强

(2) 兼容性好

(3) 矢量图形比 SVG，Canvas 优势大很多

(4) 客户端资源调度，比如麦克风，摄像头

13. 什么是跨域?

由于浏览器同源策略，凡是发送请求 url 的协议、域名、端口三者之间任意一与当前页面地址不同即为跨域。如 http 协议访问 https 协议、80 端口访问 8080 端口等等。

14. 不同浏览器的标签默认的外补丁和内补丁不同.

- 即随便写几个标签, 在不加样式控制的情况下, 各自的 margin 和 padding 差异较大.
- 解决方法: CCS 里: *{margin:0; padding:0}

15. 行内属性标签，设置 display:block 后采用 float 布局，又有横行的 margin 的情况，IE6 间距 bug

- 在 display:block;后面加入 display:inline;display:table;

16. 图片默认有间距

- 几个 img 标签放在一起的时候，有些浏览器会有默认的间距，加了问题一中提到的通配符也不起作用。
- 解决: 使用 float 属性为 img 布局

17. 标签最低高度设置 min-height 不兼容

- 因为 min-height 本身就是一个不兼容的 CSS 属性，所以设置 min-height 时不能很好的被各个浏览器兼容
- 如果我们要设置一个标签的最小高度 200px，需要进行的设置为：{min-height:200px; height:auto !important; height:200px; overflow:visible;}

18. 透明度的兼容 CSS 设置

- IE `filter:alpha(opacity=0-100)`
- 非 IE `opacity:0-1`

19. ie6,7 不支持 display:inline-block

- 解决方法: 设置 inline 并触法 haslayout
- display:inline-block; *display:inline; *zoom:1

20. 如何对网站的文件和资源进行优化?

文件合并（同上题“假若你有 5 个不同的 CSS 文件，加载进页面的最好方式是？”）

减少调用其他页面、文件的数量。一般我们为了让页面生动活泼会大量使用 background 来加载背景图，而每个 background 的图像都会产生 1 次 HTTP 请求，要改善这个状况，可以采用 css 的 1 个有用的 background-position 属 性来加载背景图，我们将需要频繁加载的多个图片合成为 1 个单独的图片，需要加载时可以采用：background:url(....) no-repeat x-offset y-offset;的形式加载即可将这部分图片加载的 HTTP 请求缩减为 1 个。

每个 http 请求都会产生一次从你的浏览器到服务器端网络往返过程，并且导致推迟到达服务器端和返回浏览器端的时间，我们称之为延迟。

文件最小化/文件压缩

即将需要传输的内容压缩后传输到客户端再解压，这样在网络上传输的 数据量就会大幅减小。通常在服务器上的 Apache、Nginx 可以直接开启这个设置，也可以从代码角度直接设置传输文件头，增加 gzip 的设置，也可以 从 负载均衡设备直接设置。不过需要留意的是，这个设置会略微增加服务器的负担。建议服务器性能不是很好的网站，要慎重考虑。

js 和 css 文件在百度上搜一个压缩网站就能压缩，但是在实际开发的项目中，使用 gulp、webpack 等工具可以打包出合并压缩后的文件，小图片可以在打包时转换成 base64 方式引入，大图片可以被压缩，html 文件也是可以被压缩的

使用 CDN 托管

CDN 的全称是 Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。其目的是使用户可就近取得所需内容，解决 Internet 网络拥挤的状况，提高用户访问网站的响应速度。

缓存的使用

Ajax 调用都采用缓存调用方式，一般采用**附加特征参数方**式实现，注意其中的<script src=”xxx.js?{VERHASH}”，{VERHASH}就是特征参数，**这个参数不变化就使用缓存文件**，如果发生变化则重新下载新文件或更新信息。

css 文件放置在 head，js 放置在文档尾

在服务器端配置 control-cache last-modify-date

在服务器配置 Entity-Tag if-none-match

用更少的时间下载更多的文件，提高网站加载速度，提高用户体验，可以使用以下方法：

1.**css sprites**----将小图片合并为一张大图片，使用 background-position 等 css 属性取得图片位置

2.**将资源放在多个域名下**-----打开控制台，可以看到很多网站都是这么做的~

3.**图片延迟加载**-----很多电商网站、新闻网站，尤其是用到瀑布流展示图片的时候，很多都这么做了，这个技术已经很普遍~

书写代码的时候要注意优化：

1.css

将可以合并的样式合并起来，比如 margin-top、margin-bottom 等。

给 img 图片设置宽高，因为浏览器渲染元素的时候没有找到这两个参数，需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了。

2.js

少改变 DOM 元素，少触发 reflow，可以复用的代码提出来写成公共的等等……

3.img

优化图片，不失真的情况下尽量减少图片大小，使用 iconfont 等

21. 关于前后端分离:

- 前端：负责 View 和 Controller 层。

后端：只负责 Model 层，业务处理/数据等。

22. 跨域请求资源的方法是什么?

(1) porxy 代理

(2)CORS 【Cross-Origin Resource Sharing】

(3) json

8.5 CSS 扩展

1. CSS3 新特性有哪些。

- 新增各种 CSS 选择器
- 圆角属性：border-radiuis
- 多列布局。
- 阴影和反射。
- 文字特效。
- 线性渐变。
- 旋转、缩放、定位、倾斜、动画、多背景。

2. 什么是无样式内容闪烁?如何避免?

- 如果使用 import 方法对 CSS 进行导入,会导致某些页面在 Internet Explorer 中以无样式显示页面内容的瞬间闪烁，这种现象称之为文档样式短暂失效，简称为 FOUC。
- 原因在于：当样式表晚于结构性 html 加载，当加载到此样式表时，页面将停止之前的渲染。此样式表被下载和解析后，将重新渲染页面，也就出现了短暂的花屏现象。
- 解决办法：使用 LINK 标签将样式表放在文档 HEAD 中。

3. display:none 和 visibility: hidden 的区别

- display:none 隐藏对应的元素，在文档布局中不再给它分配空间，它各边的元素会合拢，就当他从来不存在。
- visibility:hidden 隐藏对应的元素，但是在文档布局中仍保留原来的空间。

4. 解释浮动和工作原理

- 浮动可以理解为让某个 div 元素脱离标准流，漂浮在标准流之上，和标准流不是一个层次。
- 假如某个 div 元素 A 是浮动的，如果 A 元素上一个元素也是浮动的，那么 A 元素会跟

随在上一个元素的后边(如果一行放不下这两个元素，那么 A 元素会被挤到下一行)；如果 A 元素上一个元素是标准流中的元素，那么 A 的相对垂直位置不会改变，也就是说 A 的顶部总是和上一个元素的底部对齐。

- 清除浮动是为了清除使用浮动元素产生的影响。浮动的元素，高度会塌陷，而高度的塌陷使页面后面的布局不能正常显示。清除浮动所使用的样式为： clear: none | left | right | both。这个样式只能影响使用清除的元素本身，不能影响其他元素。

5. 清除浮动

- 在浮动元素后面添加空标签 clear:both
- 给父标签使用 overflow: hidden/auto;zoom:1
- 父级 div 定义, 使用伪类:after 和 zoom

zoom 是 IE 浏览器设置或检索对象的缩放比例。当设置了 zoom 的值之后，所设置的元素就会扩大或缩小，高度宽度就会重新计算了，也就解决了 IE 下子元素浮动时候父元素不随着自动扩大的问题。

6. 解释 CSS Sprites, 以及你要如何使用?

CSS Sprites 其实就是把网页中一些背景图片整合到一张图片文件中，再利用 CSS 的“background-image”，“background- repeat”，“background-position”的组合进行背景定位，background-position 可以用数字能精确的定位出背景图片的位置。

7. 如何为有功能限制的浏览器提供网页?

功能限制的浏览器，比如低版本 IE，手机浏览器，等会在很多功能上不符合 Web 标准，而应对方式主要有:

- 只提供符合 Web 标准的页面
- 提供另一个符合那些浏览器标准的页面
- 兼容: 两种思路:
  - 渐进增强: 提供一个可用的原型，后来再为高级浏览器提供优化
  - 优雅降级: 据高级浏览器提供一个版本，然后有功能限制的浏览器只需要一个刚好能用的版本

8. 如何优化网页的打印样式?

针对打印机的样式: @media print{...}

9. 描述一下你曾经使用过的 CSS 预处理的优缺点

优点:

- 结构清晰, 便于扩展
- 可以方便屏幕浏览器私有语法差异
- 可以轻松实现多重继承
- 完全兼容 css 代码

缺点：降低了对最终代码的控制力。提高了维护、以及团队整体水平和规范的门槛。

10. 如果设计中使用了非标准的字体，你该如何去实现?

- 图片替代
- web : fonts 在线字库
- @font-face

11. 解释下浏览器是如何判断元素是否匹配某个 CSS 选择器?

从后往前判断：浏览器先产生一个元素集合，这个集合往往由最后一个部分的索引产生（如果没有索引就是所有元素的集合）。然后向上匹配，如果不符合上一个部分，就把元素从集合中删除，直到整个选择器都匹配完，还在集合中的元素就匹配这个选择器了。

12. 描述下 reset.css 文件的作用和使用它的好处

reset.css 能够重置浏览器的默认属性。不同的浏览器具有不同的样式，重置能够使其统一。比如说 ie 浏览器和 FF 浏览器下 button 显示不同，通过 reset 能够统一样式，显示相同的效果。

13. block, inline 和 inline-block 的区别

- block 元素会独占一行, 默认宽度为 100%，在标准文档流中垂直排列；inline 元素不会独占一行, 宽度随元素的内容而变化，在标准文档流中水平排列。
- block 元素可以设置 width, height 属性；inline 元素设置 width, height 无效。
- block 元素可以设置 margin 和 padding 属性；inline 元素的 margin 和 padding 属性在垂直方向无效。
- block 可以包含 inline 和 block 元素；而 inline 元只能包含 inline 元素
- display: inline-block, 则是将对象呈现为 inline 对象, 但是对象的内容作为 block 对象呈现。比如我们可以给一个 link(a 元素)inline-block 的属性, 使其既有 block 的高宽特性又有 inline 的同行特性。

14. css 动画和 js 动画的优缺点

CSS3 的动画

- 优点：

1.在性能上会稍微好一些，浏览器会对 CSS3 的动画做一些优化

2.代码相对简单

- 缺点：

1.在动画控制上不够灵活

2.兼容性不好

3.部分动画功能无法实现（如滚动动画，视差滚动等）

JavaScript 的动画

- 优点：

1.控制能力很强，可以单帧的控制、变换

2.兼容性好，且功能强大。

- 缺点：

1.计算没有 css 快，另外经常需要依赖其他的库。

15. 你用过媒体查询，或针对移动端的布局/CSS 吗?

通过媒体查询可以为不同大小和尺寸的媒体定义不同的 css，适合相应的设备显示；即响应式布局

- @media screen and (min-width: 400px) and (max-width: 700px) { … }
- @media handheld and (min-width: 20em), screen and (min-width: 20em) { … }

16. 有哪些隐藏内容的方法

display:none 或者 visibility:hidden，overflow:hidden。

17. img 设置属性 title 和 alt 的区别?

- alt 是 img 的特有属性，或与<input type="image">配合使用，规定图像的替代文本。如果无法显示图像，浏览器将显示替代文本。
- title 为元素提供附加的提示信息，用于鼠标滑到元素上的时候显示。其值可以比 alt 属性值设置的更长。

8.6 HTML 扩展

1. h5 的改进

- 新元素

- ■画布 canvas
- ■音频 audio
- ■视频 video
- ■语义性: article, nav , footer, section, aside, hgroup 等.
- ■时间 time

- 新属性
  - ■拖放: draggable
  - ■可编辑: contenteditable
- 新事件
  - ■拖放 ondrag ondrop
  - ■关闭页面 onunload
  - ■窗口大小改变 onresize
- 取消了一些元素: font center 等
- 新的 DOCTYPE 声明 <!DOCTYPE html>
- 完全支持 CSS3
- Video 和 Audio
- 2D/3D 制图
- 本地存储
- 本地 SQL 数据
- Web 应用

2. 从前端角度出发谈谈做好 seo 需要考虑什么?

- 语义化 html 标签
- 合理的 title, description, keywords;
- 重要的 html 代码放前面
- 少用 iframe, 搜索引擎不会抓取 iframe 中的内容
- 图片加上 alt

3. 文档类型(DOCTYPE)

- **作用:** doctype 声明位于文档中最前面的位置，处于标签之前，告知浏览器使用的是哪种规范。
- **类型:** 三种: Strict、Transitional 以及 Frameset
- **如果不声明:** 浏览器不引入 w3c 的标准，那么早期的浏览器会按照自己的解析方式渲

染页面。

4. 使用 XHTML 的局限有哪些?

- XHTML 较为严格，标签必须闭合，必须要 body，head 等
- 如果页面使用 'application/xhtml+xml' 一些老的浏览器并不兼容

5. 如果网页内容需要多语言,要怎么做?

采用统一编码 utf-8 模式

6. data-*属性的作用

data-*为前端开发者提供自定义的属性，这些属性集可以通过对象的 dataset 属性获取，不支持该属性的浏览器可以通过 getAttribute 方法获取。

7. 如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些?

1）Web Storage API

2）基于位置服务 LBS

3）无插件播放音频视频

4）调用相机和 GPU 图像处理单元等硬件设备

5）拖拽和 Form API

8. 请描述一下 cookies，sessionStorage 和 localStorage 的区别?

- cookie:

  ■cookie 是网站为了标示用户身份而储存在用户本地上的数据，cookie 数据始终在同源的 http 请求中携带（即使不需要），即会在浏览器和服务器间来回传递。

  ■sessionStorage 和 localStorage 不会自动把数据发给服务器，仅在本地保存。

- 存储大小：

  ■cookie 数据大小不能超过 4k。

  ■sessionStorage 和 localStorage 虽然也有存储大小的限制，但比 cookie 大得多，可以达到 5M 或更大。

- 有效时间：

  ■localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

  ■sessionStorage 数据在当前浏览器窗口关闭后自动删除。

  ■cookie 设置的 cookie 过期时间之前一直有效，即使窗口或浏览器关闭

- 作用域不同:

■sessionStorage 不在不同的浏览器窗口中共享，即使是同一个页面；

■localStorage 在所有同源窗口中都是共享的；cookie 也是在所有同源窗口中都是共享的。

- Web Storage 支持事件通知机制，可以将数据更新的通知发送给监听者。
- Web Storage 的 api 接口使用更方便。

9. 浏览器本地存储与服务器端存储之间的区别

- 服务器端保存数据占用服务器端的资源，而浏览器端保存则把不同用户需要的数据分布保存在用户各自的浏览器中。
- 浏览器端一般只用来存储小数据，而服务器则可以存储大数据。
- 服务器存储数据安全性高，浏览器只适合存储对安全不敏感的一般数据。

10. sessionStorage 和页面 js 数据对象的区别

- 页面中一般的 js 对象或数据的生存期是仅在当前页面有效，因此刷新页面或转到另一页面这样的重新加载页面的情况，数据就不存在了。
- 而 sessionStorage 只要同源的同窗口（或 tab）中，刷新页面或进入同源的不同页面，数据始终存在。

11. canvas 和 svg 的区别?

- SVG 是一种使用 XML 描述 2D 图形的语言
- Canvas 通过 js 来绘制 2D 图形
- **区别**

■Canvas 支持分辨率, SVG 不支持

■Canvas 不支持事件处理器, SVG 支持

■Canvas 是基于位图的图像,它不能够改变大小，只能缩放显示; SVG 是基于矢量的,所以它能够很好地处理图形大小的改变

■Canvas 适合像素处理，动态渲染和大数据量绘制; SVG 适合静态图片显示，高保真文档查看和打印的应用场景

■绘制 Canvas 对象后，不能使用脚本和 CSS 对它进行修改；而 SVG 对象是文档对象模型的一部分，所以可以随时使用脚本和 CSS 修改它们

12. src 和 href 的区别?

- src 指向外部资源的位置，用于替换当前元素，比如 js 脚本，图片等元素
- href 指向网络资源所在的位置，用于在当前文档和引用资源间确定联系，比如加载 css

8.7 JavaScript 扩展

1. apply 和 call 的用法和区别:

JavaScript 中的每一个 function 对象都会有 call 和 apply 方法

作用：改变函数体内部 this 指向

举例：

B.apply(A, arguments);即 A 对象指向 B 对象的方法

B.call(A, args1,args2);即 A 对象指向 B 对象的方法

区别

apply 最多只能有两个参数——新 this 对象和一个数组 argArray

call 则可以传递多个参数，第一个参数和 apply 一样，是用来替换的对象，后边是参数列表

2. 变量的作用域

在 js 中，变量的作用域并**不是以代码块作为作用域**的，而是**以函数作为作用域**。

全局作用域：定义在所有函数之外的变量或者在函数中没有用 var 关键字定义（不是重新赋值）的变量

变量局部作用域：在函数体内部通过 var 关键词申明

注意：

局部变量的优先级高于同名的全局变量。如果在函数内声明一个局部变量或者函数中带有的变量和全局变量重名，那么全局变量就被局部变量所遮盖

3. bind 函数的兼容性

用法:

bind()函数会创建一个新函数，为绑定函数。当调用这个绑定函数时,绑定函数会以创建它时传入 bind 方法的第一个参数作为 this,传入 bind 方法的第二个以及以后的参数加上绑定函数运行时本身的参数按照顺序作为原函数的参数来调用原函数.

一个绑定函数也能使用 new 操作符创建对象：这种行为就像把原函数当成构造器。提供的 this 值被忽略，同时调用时的参数被提供给模拟函数。

4. 解释下事件代理

事件代理又称之为事件委托。是 JavaScript 中绑定事件的常用技巧，“事件代理”即是把原本需要绑定在子元素的响应事件（click、keydown.....）委托给父元素，让父元素担当事件监听的职务。事件代理的原理是 DOM 元素的事件冒泡

5. 解释下 js 中 this 是怎么工作的?

this 在 JavaScript 中主要有以下五种使用场景。

- 作为函数调用，this 绑定全局对象，浏览器环境全局对象为 window 。
- 内部函数的 this 也绑定全局对象，应该绑定到其外层函数对应的对象上，这是 JavaScript 的缺陷，用 that 替换。
- 作为构造函数使用，this 绑定到新创建的对象。
- 作为对象方法使用，this 绑定到该对象。
- 使用 apply 或 call 调用 this 将会被显式设置为函数调用的第一个参数。

6. AMD vs. CommonJS?

1. commonjs 的目标是制定一个 js 模块化的标准，它的目标制定一个可以同时在客服端和服务端运行的模块

2. AMD 是为了弥补 commonjs 规范在浏览器中目前无法支持 ES6 的一种解决方案。异步模块定义规范（AMD）制定了定义模块的规则，这样模块和模块的依赖可以被异步加载

3. cmd 是 commonjs 另外的一种模块加载方案，这个规范本身偏向于 common

js 的规范它以一个文件就是一个模块和 ES6 中标准的 commonjs 规范类似

7. 什么是闭包？闭包有什么作用?

- 闭包是指有权访问另一个函数作用域中的变量的函数. 创建闭包常见方式,就是在一个函数内部创建另一个函数.
- 作用:
  - 匿名自执行函数 (function (){ ... })(); 创建了一个匿名的函数，并立即执行它，由于外部无法引用它内部的变量，因此在执行完后很快就会被释放，关键是这种机制不会污染全局对象。
  - 缓存, 可保留函数内部的值
  - 实现封装
  - 实现模板

8. 什么是伪数组?

- 伪数组
  - 具有 length 属性
  - 按索引方式存储数据
  - 不具有数组的方法

- 伪数组存在的意义，是可以让普通的对象也能正常使用数组的很多算法
- 我们可以通过 Array.prototype.slice.call(fakeArray)将伪数组转变为真正的 Array 对象: 返回新数组而不会修改原数组

9. undefined 和 null 的区别, 还有 undeclared:

- null 表示没有对象, 即此处不该有此值. 典型用法:
  - （1） 作为函数的参数，表示该函数的参数不是对象。
  - （2） 作为对象原型链的终点。
  - （3）null 可以作为空指针，只要意在保存对象的值还没真正保存对象,就应该明确地让该对象保存 null 值.
- undefined 表示缺少值, 即此处应该有值, 但还未定义.
  - （1）变量被声明了，但没有赋值时，就等于 undefined。
  - （2) 调用函数时，应该提供的参数没有提供，该参数等于 undefined。
  - （3）对象没有赋值的属性，该属性的值为 undefined。
  - （4）函数没有返回值时，默认返回 undefined。
- undeclared 即为被污染的命名, 访问没有被声明的变量, 则会抛出异常, 终止执行. 即 undeclared 是一种语法错误

10. 简述事件冒泡机制。

从目标元素开始，往顶层元素传播。途中如果有节点绑定了相应的事件处理函数，这些函数都会被一次触发。如果想阻止事件起泡，可以使用 e.stopPropagation()（Firefox）或者 e.cancelBubble=true（IE）来组织事件的冒泡传播。

11. 解释下为什么接下来这段代码不是 IIFE(立即调用的函数表达式)：function foo(){ }();?

以 **function** 关键字开头的语句会被解析为**函数声明**，而函数声明是不允许直接运行的。

只有当解析器把这句话解析为函数表达式，才能够直接运行

```
(function foo(){

    // code..

})()
```

12. "attribute" 和 "property" 的区别是什么?

Attribute 就是 dom 节点自带的属性，例如 html 中常用的 id、class、title、align 等，它的

值只能够是字符串

Property 是这个 DOM 元素作为对象，其附加的内容，例如 childNodes、firstChild 等，是 JavaScript 里的对象

13. 请指出 document load 和 document ready 两个事件的区别。

- 执行时间不同
  - ■ready，表示文档结构已经加载完成（不包含图片等非文字媒体文件）
  - ■onload，指示页面包含图片等文件在内的所有元素都加载完成
- 可以被执行的次数不同
  - ■ready()可以在 JavaScript 代码中出现多次
  - ■load()只能在 JavaScript 代码中出现一次
- 执行的效率不同
  - ■如要在 dom 的元素节点中添加 onclick 属性节点，这时用$(document).ready()就要比用$(window).load()的效率高

14. 什么是 use strict? 其好处坏处分别是什么?

在所有的函数（或者所有最外层函数）的开始处加入 "use strict"; 指令启动严格模式。"严格模式"有两种调用方法

1）将"use strict"放在脚本文件的第一行，则整个脚本都将以"严格模式"运行。如果这行语句不在第一行，则无效，整个脚本以"正常模式"运行。如果不同模式的代码文件合并成一个文件，这一点需要特别注意。

2）将整个脚本文件放在一个立即执行的匿名函数之中。

好处

- 消除 Javascript 语法的一些不合理、不严谨之处，减少一些怪异行为;
- 消除代码运行的一些不安全之处，保证代码运行的安全；
- 提高编译器效率，增加运行速度；
- 为未来新版本的 Javascript 做好铺垫。

坏处

同样的代码，在"严格模式"中，可能会有不一样的运行结果；一些在"正常模式"下可以运行的语句，在"严格模式"下将不能运行

15. 浏览器端的 js 包括哪几个部分?

核心(ECMAScript) , 文档对象模型(DOM), 浏览器对象模型(BOM)

16. DOM 包括哪些对象?

DOM 是针对 HTML 和 XML 文档的一个 API(应用程序编程接口). DOM 描绘了一个层次化的节点树，允许开发人员添加，移除和修改页面的某一部分.

Document 对象

Node 对象

NodeList 对象

Element 对象

attribute 对象

- 常用的 DOM 方法:
  - getElementById(id)
  - getElementsByTagName()
  - appendChild(node)
  - removeChild(node)
  - replaceChild()
  - insertChild()
  - createElement()
  - createTextNode()
  - getAttribute()
  - setAttribute()
- 常用的 DOM 属性
  - innerHTML　节点(元素)的文本值
  - parentNode　节点(元素)的父节点
  - childNodes
  - attributes　节点(元素)的属性节点

17. js 有哪些基本类型?

Undefined, Null, Boolean, Number, String

18. 基本类型与引用类型有什么区别?

- 基本类型如上题所示. 引用类型则有: Object, Array, Date, RegExp, Function
- 存储
  - 基本类型值在内存中占据固定大小的空间,因此被保存在栈内存中
  - 引用类型的值是对象，保存在堆内存中. 包含引用类型的变量实际上包含的并不是对象本身，而是一个指向改对象的指针
- 复制
  - 从一个变量向另一个变量复制基本类型的值，会创建这个值的一个副本

■从一个变量向另一个变量复制引用类型的值，复制的其实是指针， 因此两个变量最终都指向同一个对象

- 检测类型

■确定一个值是哪种基本类型可以用 typeof 操作符，

■而确定一个值是哪种引用类型可以使用 instanceof 操作符

19. 关于 js 的垃圾收集例程

js 是一门具有自动垃圾回收机制的编程语言,开发人员不必关心内存分配和回收问题

- 离开作用域的值将被自动标记为可以回收，因此将在垃圾收集期间被删除
- "标记清除"是目前主流的垃圾收集算法，这种算法的思路是给当前不使用的值加上标记，然后再回收其内存
- 另一种垃圾收集算法是"引用计数"，这种算法的思想是跟踪记录所有值被引用的次数. js 引擎目前都不再使用这种算法，但在 IE 中访问非原生 JS 对象(如 DOM 元素)时，这种算法仍然可能会导致问题
- 当代码中存在循环引用现象时，"引用计数" 算法就会导致问题
- 解除变量的引用不仅有助于消除循环引用现象，而且对垃圾收集也有好处. 为了确保有效地回收内存，应该及时解除不再使用的全局对象，全局对象属性以及循环引用变量的引用

20. js 有几种函数调用方式?

- 方法调用模型 var obj = { func : function(){};} obj.func()
- 函数调用模式 var func = function(){} func();
- 构造器调用模式
- apply/ call 调用模式

21. 描述事件模型?

三种事件模型：

1.原始事件模型（DOM0）

2.DOM2 事件模型

3.IE 事件模型

区别：

1.原始事件模型（没有兼容性问题）

2.DOM2 级（IE8 以下不支持），主要特点：有一个事件的传播过程：捕获->处于目标->冒泡

3. IE 事件模型：Event 对象不是事件处理程序的函数参数，而是 window 的全局变量

22. js 动画有哪些实现方法?

三种分别为：setTimeout、setInterval、requestAnimationFrame

requestAnimationFrame 方法是 h5 提供的一种专门解决动画更新的 API，浏览器会自动以最合适的频率去刷新动画

23. 还有什么实现动画的方法?。

CSS3: animation

HTML5:

canvas 元素结合 JS

CSS3 结合 Jquery 实现

24. 如何判断属性来自自身对象还是原型链?

hasOwnPrototype

25. ES6 新特性

1) **箭头操作符** inputs=>outputs: 操作符左边是输入的参数，而右边则是进行的操作以及返回的值

2) **支持类**, 引入了 **class** 关键字. ES6 提供的类实际上就是 JS 原型模式的包装

3) **增强的对象字面量**.

1. 可以在对象字面量中**定义原型 __proto__**: xxx //设置其原型为 xxx,相当于继承 xxx
2. 定义方法可以**不用 function** 关键字
3. **直接调用父类方法**

4) **字符串模板**: ES6 中允许使用反引号 ` 来创建字符串，此种方法创建的字符串里面可以包含由美元符号加花括号包裹的变量${vraible}。

5) **自动解析数组或对象中的值**。比如若一个函数要返回多个值，常规的做法是返回一个对象，将每个值做为这个对象的属性返回。但在 ES6 中，利用解构这一特性，可以直接返回一个数组，然后数组中的值会自动被解析到对应接收该值的变量中。

6) **默认参数值:** 现在可以在定义函数的时候指定参数的默认值了，而不用像以前那样通过逻辑或操作符来达到目的了。

7) **不定参数**是在函数中使用命名参数同时接收不定数量的未命名参数。在以前的 JavaScript 代码中我们可以通过 arguments 变量来达到这一目的。不定参数的格式是三个句点后跟代表所有不定参数的变量名。比如下面这个例子中，...x 代表了所有传入 add 函数的参数。

8) **拓展参数**则是另一种形式的语法糖，它允许传递数组或者类数组直接做为函数的参数而不用通过 apply。

9) **let 和 const 关键字:** 可以把 let 看成 var，只是它定义的变量被限定在了特定范围内才能使用，而离开这个范围则无效。const 则很直观，用来定义常量，即无法被更改值的变量。

10) **for of 值遍历** 每次循环它提供的不是序号而是值。

11) **Map, Set, WeakMap, WeakSet**

12) **Proxy** 可以监听对象身上发生了什么事情，并在这些事情发生后执行一些相应的操作。一下子让我们对一个对象有了很强的追踪能力，同时在数据绑定方面也很有用处。

13) **Symbols** Symbol 通过调用 symbol 函数产生，它接收一个可选的名字参数，该函数返回的 symbol 是唯一的。之后就可以用这个返回值做为对象的键了。Symbol 还可以用来创建私有属性，外部无法直接访问由 symbol 做为键的属性值。

14) **Math, Number, String, Object 的新 API**

15) **Promises** 是处理异步操作的一种模式

26. 如何获取某个 DOM 节点，节点遍历方式

获取 DOM 节点

getElementsByClassName

getElementsByTagName

getElementById

节点遍历

1.用 firstChild，lastChild 进行元素遍历

2.使用 firstElementChild，nextElementSibling

27. 图片预加载的实现

1. 使用 jQuery 图片预加载插件 Lazy Load
   (1) 加载 jQuery，与 jquery.lazyload.js
   (2) 设置图片的占位符为 data-original，给图片一个特别的标签,比如 class=".lazy"
   (3) 然后延迟加载: $('img.lazy').lazyload();这个函数可以选择一些参数:

① 图片预先加载距离：threshold，通过设置这个值，在图片未出现在可视区域的顶部距离这个值时加载。

② 事件绑定加载的方式：event

③ 图片限定在某个容器内：container

2. 使用 js 实现图片加载: 就是 new 一个图片对象，绑定 onload 函数，赋值 url
3. 用 CSS 实现图片的预加载
   (1) 写一个 CSS 样式设置一批背景图片，然后将其隐藏

(2) 改进: 使用 js 来推迟预加载时间, 防止与页面其他内容一起加载

4. 用 Ajax 实现预加载

(1) 其实就是通过 ajax 请求请求图片地址. 还可以用这种方式加载 css,js 文件等

28. 构造函数里定义 function 和使用 prototype.func 的区别?

1. 直接调用 function，每一个类的实例都会拷贝这个函数，弊端就是**浪费内存**（如上）。prototype 方式定义的方式，函数不会拷贝到每一个实例中，所有的实例共享 prototype 中的定义，节省了内存。

2. 但是如果 prototype 的属性是对象的话，所有实例也会共享一个对象（这里问的是函数应该不会出现这个情况），如果其中一个实例改变了对象的值，则所有实例的值都会被改变。同理的话，如果使用 prototype 调用的函数，一旦改变，所有实例的方法都会改变。——不可以对实例使用 prototype 属性，只能对类和函数用。

29. js 实现对象的深克隆

因为 js 中数据类型分为基本数据类型(number, string, boolean, null, undefined)和引用类型值(对象, 数组, 函数). 这两类对象在复制克隆的时候是有很大区别的. 原始类型存储的是对象的实际数据, 而对象类型存储的是对象的引用地址(对象的实际内容单独存放, 为了减少数据开销通常放在内存中). 此外, 对象的原型也是引用对象, 它把原型的属性和方法放在内存中, 通过原型链的方式来指向这个内存地址.

于是克隆也会分为两类:

1.浅度克隆: 原始类型为值传递, 对象类型仍为引用传递

2.深度克隆: 所有元素或属性均完全复制, 与原对象完全脱离, 也就是说所有对于新对象的修改都不会反映到原对象中

30. js 中如何检测一个变量是一个 String 类型?

```
typeof(obj) === "string"

typeof obj === "string"

obj.constructor === String
```

31. 请用 js 去除字符串空格?

1. **方法一：使用 replace 正则匹配的方法**

去除所有空格: str = str.replace(/\s*/g,"");

去除两头空格: str = str.replace(/^\s*|\s*$/g,"");

去除左空格： str = str.replace(/^\s*/, “”);

去除右空格： str = str.replace(/(\s*$)/g, "");

str 为要去除空格的字符串，实例如下：

```
var str = " 23 23 ";var str2 = str.replace(/\s*/g,"");

console.log(str2); // 2323
```

2. **方法二：使用 str.trim()方法**

str.trim()局限性：无法去除中间的空格，实例如下：

```
var str = "   xiao  ming   ";var str2 = str.trim();

console.log(str2);   //xiao  ming
```

同理，str.trimLeft()，str.trimRight()分别用于去除字符串左右空格。

3. **方法三：使用 jquery,$.trim(str)方法**

$.trim(str)局限性：无法去除中间的空格，实例如下：

```
var str = "   xiao  ming   ";var str2 = $.trim(str)

console.log(str2);   //  xiao  ming
```

32. js 字符串操作函数

我这里只是列举了常用的字符串函数

concat() - 将两个或多个字符的文本组合起来，返回一个新的字符串。

indexOf() - 返回字符串中一个子串第一处出现的索引。如果没有匹配项，返回 -1 。

charAt() - 返回指定位置的字符。

lastIndexOf() - 返回字符串中一个子串最后一处出现的索引，如果没有匹配项，返回 -1 。

match() - 检查一个字符串是否匹配一个正则表达式。

substr() 函数 -- 返回从 string 的 startPos 位置，长度为 length 的字符串

substring() - 返回字符串的一个子串。传入参数是起始位置和结束位置。

slice() - 提取字符串的一部分，并返回一个新字符串。

replace() - 用来查找匹配一个正则表达式的字符串，然后使用新字符串代替匹配的字符串。

search() - 执行一个正则表达式匹配查找。如果查找成功，返回字符串中匹配的索引值。否则返回 -1 。

split() - 通过将字符串划分成子串，将一个字符串做成一个字符串数组。

length - 返回字符串的长度，所谓字符串的长度是指其包含的字符的个数。

toLowerCase() - 将整个字符串转成小写字母。

toUpperCase() - 将整个字符串转成大写字母。

33. 怎样添加、移除、移动、复制、创建和查找节点?

1)创建新节点

createDocumentFragment() //创建一个 DOM 片段

createElement() //创建一个具体的元素

createTextNode() //创建一个文本节点

2）添加、移除、替换、插入

appendChild() //添加

removeChild() //移除

replaceChild() //替换

insertBefore() //插入

3）查找

getElementsByTagName() //通过标签名称

getElementsByName() //通过元素的 Name 属性的值

getElementById() //通过元素 Id，唯一性

34. 比较 typeof 与 instanceof?

相同点：JavaScript 中 typeof 和 instanceof 常用来判断一个变量是否为空，或者是什么类型的。

区别：

1.typeof 判断所有变量的类型，返回值有 number，boolean，string，function，object，undefined。

2.typeof 对于丰富的对象实例，只能返回"Object"字符串。

3.instanceof 可以对不同的对象实例进行判断

35. Array 相关的属性和方法

1. Array 对象属性

constructor 返回对创建此对象的数组函数的引用。

length 设置或返回数组中元素的数目。

prototype 使您有能力向对象添加属性和方法。

2. Array 对象方法

concat() 连接两个或更多的数组，并返回结果。

join() 把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。

pop() 删除并返回数组的最后一个元素。

shift() 删除并返回数组的第一个元素

push() 向数组的末尾添加一个或更多元素，并返回新的长度。

unshift() 向数组的开头添加一个或更多元素，并返回新的长度。

reverse() 颠倒数组中元素的顺序。

slice() 从某个已有的数组返回选定的元素

sort() 对数组的元素进行排序

splice() 删除元素，并向数组添加新元素。

toSource() 返回该对象的源代码。

toString() 把数组转换为字符串，并返回结果。

toLocaleString() 把数组转换为本地数组，并返回结果。

valueOf() 返回数组对象的原始值

36. jQuery 库中的 $() 是什么?

$() 函数是 jQuery() 函数的别称。$() 函数用于将任何对象包裹成 jQuery 对象，接着你就被允许调用定义在 jQuery 对象上的多个不同方法。你可以将一个选择器字符串传入 $() 函数，它会返回一个包含所有匹配的 DOM 元素数组的 jQuery 对象。

37. $(this) 和 this 关键字在 jQuery 中有何不同?

$(this) 返回一个 jQuery 对象

this 代表上下文中的当前 DOM 元素

38. jquery 怎么移除标签 onclick 属性?

获得 a 标签的 onclick 属性: $("a").attr("onclick")

删除 onclick 属性：$("a").removeAttr("onclick");

设置 onclick 属性：$("a").attr("onclick","test();");

39. jquery 中 addClass,removeClass,toggleClass 的使用。

$(selector).addClass(class)：为每个匹配的元素添加指定的类名

$(selector).removeClass(class)：从所有匹配的元素中删除全部或者指定的类，删除 class 中某个值；

$(selector).toggleClass(class)：如果存在（不存在）就删除（添加）一个类

40. JQuery 有几种选择器?

(1)、基本选择器：#id，class,element,*;

(2)、层次选择器：parent > child，prev + next ，prev ~ siblings

(3)、基本过滤器选择器：:first，:last ，:not ，:even ，:odd ，:eq ，:gt ，:lt

(4)、内容过滤器选择器： :contains ， :empty ， :has ， :parent

(5)、可见性过滤器选择器：:hidden ， :visible

(6)、属性过滤器选择器：[attribute] ， [attribute=value] ， [attribute!=value] ， [attribute^=value] ， [attribute$=value] ， [attribute*=value]

(7)、子元素过滤器选择器：:nth-child ， :first-child ， :last-child ， :only-child

(8)、表单选择器： :input ， :text ， :password ， :radio ， :checkbox ， :submit 等；

(9)、表单过滤器选择器：:enabled ， :disabled ， :checked ， :selected

41. jQuery 中的 Delegate()函数有什么作用?

delegate()会在以下两个情况下使用到：

1、如果你有一个父元素，需要给其下的子元素添加事件，这时你可以使用 delegate()了，代码如下：

```
$("ul").delegate("li", "click", function(){ $(this).hide(); });
```

2、当元素在当前页面中不可用时，可以使用 delegate()

42. jquery 中$.get()提交和$.post()提交有区别吗?

相同点：都是异步请求的方式来获取服务端的数据；

异同点：

1、请求方式不同：$.get() 方法使用 GET 方法来进行异步请求的。$.post() 方法使用 POST 方法来进行异步请求的。

2、参数传递方式不同：get 请求会将参数跟在 URL 后进行传递，而 POST 请求则是作为 HTTP 消息的实体内容发送给 Web 服务器的，这种传递是对用户不可见的。

3、数据传输大小不同：get 方式传输的数据大小不能超过 2KB 而 POST 要大的多

4、安全问题： GET 方式请求的数据会被浏览器缓存起来，因此有安全问题。

43. Ajax 的优缺点及工作原理?

定义和用法:

AJAX = Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。Ajax 是一种用于创建快速动态网页的技术。Ajax 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。

传统的网页（不使用 Ajax）如果需要更新内容，必须重载整个网页页面。

优点：

1.减轻服务器的负担,按需取数据,最大程度的减少冗余请求

2.局部刷新页面,减少用户心理和实际的等待时间,带来更好的用户体验

3.基于 xml 标准化,并被广泛支持,不需安装插件等,进一步促进页面和数据的分离

缺点：

1.AJAX 大量的使用了 javascript 和 ajax 引擎,这些取决于浏览器的支持.在编写的时候考虑对浏览器的兼容性.

2.AJAX 只是局部刷新,所以页面的后退按钮是没有用的.

3.对流媒体还有移动设备的支持不是太好等

AJAX 的工作原理：

1.创建 ajax 对象（XMLHttpRequest/ActiveXObject(Microsoft.XMLHttp)）

2.判断数据传输方式(GET/POST)

3.打开链接 open()

4.发送 send()

5.当 ajax 对象完成第四步（onreadystatechange）数据接收完成，判断 http 响应状态（status）200-300 之间或者 304（缓存）执行回调函数

8.8 编程题

1. 题目 1

var obj = {a : 1}; (function (obj) { obj = {a : 2}; })(obj); //问 obj 怎么变?

答: 外部的 obj 不变. 因为匿名函数中 obj 传入参数等于是创建了一个局部变量 obj, 里面的 obj 指向了一个新的对象 . 如果改成(function () { obj = {a : 2}; })(obj); 就会变了

2. 题目 2

var obj = { a:1, func: function() { (function () { a=2; }(); }} ; obj.fun() //a 怎么变? 匿名函数里的 this 是什么?

答: obj 里的 a 不会变. 匿名函数里的 this 指向全局对象 window. 这等于是给 window 加了一个名为 a 的属性

要改变 obj 中 a 的值 , 应当:

(function() { this.a = 2}).call(this);

或者 obj 中定义 func : func: function() { var self = this; (function(){self.a=2;})();}

3. 题目 3

编写 JavaScript 代码，判断一个字符串中出现次数最多的字符，统计这个次数

```
var str = 'asdfssaaasasasasaa';var json = {};
for (var i = 0; i < str.length; i++) {
    if(!json[str.charAt(i)]){
       json[str.charAt(i)] = 1;
    }else{
       json[str.charAt(i)]++;
    }
};
var iMax = 0;var iIndex = '';
for(var i in json){
    if(json[i]>iMax){
        iMax = json[i];
        iIndex = i;
    }
}
console.log('出现次数最多的是:'+iIndex+'出现'+iMax+'次');
```

4. 题目 4

编写 JavaScript 代码，点击一个 ul 的五个 li 元素，分别弹出他们的序号，怎么做？

方法 1：

```
1   for(var i=0; i<oLis.length; i++){
2       oLis[i].onclick = (function(j){
3           return function(){
4               alert(j);
5           }
6       })(i);
7   }
```

这样的话，给每个 li 绑定 onclick 事件时，其实绑的是一个立即执行函数，这个立即执行函数的参数是 i，因为它是立即执行的，循环时已经把 i 的值赋给了 li 的 onclick 事件，所以在外部函数里的 i 改变后并不会影响 i 的值

```
    for(var i=0; i<oLi.length; i++){
        (function(j){
            oLi[j].onclick
= function(){
                alert(j);
            };
        })(i);
    }
```

另一种实现方法:(立即执行函数)

或者不用闭包

方法 2:

```
var oLi = document.getElementsByTagName('li');
    function func(obj,i){
       obj.onclick = function(){
          alert (i);
       }
    }
      for (var i = 0; i<oLi.length; i++){
          func(oLi[i], i);
      }
```

方法 3: 设置属性

```
    var oLi = document.getElementsByTagName('li');
    for(var i=0; i<oLi.length; i++){
        oLi[i].setAttribute("onclick", "alert("+i+");");
    }
```

方法 4: 设置 index 保存

```
    for(var i=0; i<oLi.length; i++){
        oLi[i].index = i;
        oLi[i].onclick = function(){
            alert(this.index);
        }
    }
```

5. 题目 5

js 实现数组去重怎么实现?

方法 1. 创建一个新的临时数组来保存数组中已有的元素

```
var a = new Array(1,2,2,2,2,5,3,2,9,5,6,3);
Array.prototype.unique1 = function(){
    var n = [];     //一个新的临时数组
    for(var i=0; i<this.length; i++){
        //如果把当前数组的第 i 已经保存进了临时数组，那么跳过
        if(n.indexOf(this[i]) == -1){
            n.push(this[i]);
        }
    }
    return n;
}
console.log(a.unique1());
```

方法 2. 使用哈希表存储已有的元素

```
Array.prototype.unique2 = function(){
    var hash = {};
    var  n = [];     //hash 作为哈希表，n 为临时数组
    for(var i=0; i<this.length; i++){
        if(!hash[this[i]]){          //如果 hash 表中没有当前项
            hash[this[i]] = true;    //存入 hash 表
            n.push(this[i]);         //当前元素 push 到临时数组中
        }
    }
    return n;
}
```

方法 3. 使用 indexOf 判断数组元素第一次出现的位置是否为当前位置

```
Array.prototype.unique3 = function(){
    var n = [this[0]];
    for(var i=1; i<this.length; i++)    //从第二项开始遍历
    {
        //如果当前数组元素在数组中出现的第一次的位置不是 i
        //说明是重复元素
        if(this.indexOf(this[i]) == i){
            n.push(this[i]);
        }
    }
    return n;
}
```

方法 4. 先排序再去重

```
Array.prototype.unique4 = function(){
    this.sort(function(a, b){ return a - b;});
    var n = [this[0]];
    for(var i=1; i<this.length; i++){
        if(this[i] != this[i-1]){
            n.push(this[i]);
        }
    }
    return n;
}
```

第一种方法和第三种方法都使用了 indexOf(), 这个函数的执行机制也会遍历数组

第二种方法使用了一个哈希表, 是最快的.

第三种方法也有一个排序的复杂度的计算.

然后做了个测试, 随机生成 100 万个 0-1000 的数组结果如下:

```
方法一执行时间:240
方法二执行时间:5
方法三执行时间:477
方法四执行时间:246
```

第三种方法总是第二种方法的将近两倍，而第四种方法与数组的范围有关，

如果是 0-100 的数组

```
方法一执行时间:40
方法二执行时间:4
方法三执行时间:64
方法四执行时间:176
```

而如果是 0-10000，方法四看着就效果还不错了

```
方法一执行时间:2199
方法二执行时间:6
方法三执行时间:4876
方法四执行时间:298
```

而第二种方法永远是最好的，但是是以空间换时间

全部代码如下：

```
var a = [];
for(var i=0; i<1000000; i++){
    a.push(Math.ceil(Math.random()*10000));
}

Array.prototype.unique1 = function(){
    var n = [];     //一个新的临时数组
    for(var i=0; i<this.length; i++){
        //如果把当前数组的第 i 已经保存进了临时数组，那么跳过
        if(n.indexOf(this[i]) == -1){
            n.push(this[i]);
        }
    }
    return n;
}
```

```
Array.prototype.unique2 = function(){
    var hash = {},
        n = [];     //hash 作为哈希表, n 为临时数组
    for(var i=0; i<this.length; i++){
        if(!hash[this[i]]){          //如果 hash 表中没有当前项
            hash[this[i]] = true;    //存入 hash 表
            n.push(this[i]);         //当前元素 push 到临时数组中
        }
    }
    return n;
}

Array.prototype.unique3 = function(){
    var n = [this[0]];
    for(var i=1; i<this.length; i++)    //从第二项开始遍历
    {
        //如果当前数组元素在数组中出现的第一次的位置不是 i
        //说明是重复元素
        if(this.indexOf(this[i]) == i){
            n.push(this[i]);
        }
    }
    return n;
}

Array.prototype.unique4 = function(){
    this.sort(function(a, b){ return a - b;});
    var n = [this[0]];
    for(var i=1; i<this.length; i++){
        if(this[i] != this[i-1]){
            n.push(this[i]);
        }
    }
    return n;
}
var begin1 = new Date();
```

```
    a.unique1();
    var end1 = new Date();

    var begin2 = new Date();
    a.unique2();
    var end2 = new Date();

    var begin3 = new Date();
    a.unique3();
    var end3 = new Date();

    var begin4 = new Date();
    a.unique4();
    var end4 = new Date();

    console.log("方法一执行时间:" + (end1 - begin1));
    console.log("方法二执行时间:" + (end2 - begin2));
    console.log("方法三执行时间:" + (end3 - begin3));
    console.log("方法四执行时间:" + (end4 - begin4));
```

```
7
6
8
6
9
7
0
7
1
```

6. 题目 6

写出一个简单的$.ajax()的请求方式？

```
$.ajax({
    url:'http://www.baidu.com',          --请求地址
    type:'POST',                         --Post 请求
    data:data,                           --请求的同时传递过去的数据
    cache:true,                          --是否高速缓存此页
    headers:{},
    dataType:'json',                     --返回数据类型：xml/json/html/script…
    beforeSend: function(){},            -在发送请求前执行的函数
    success:function(){},                --请求成功后的回调函数
    error:function(){},                  --请求失败时调用的函数
    complete:function(){}                --请求完成后回调函数
});
```

7. 题目 7

编写一个方法 去掉一个数组的重复元素

方法一:

```
var arr = [0,2,3,4,4,0,2];var obj = {};var tmp = [];for(var i = 0 ;i<
arr.length;i++){
```

```
    if( !obj[arr[i]] ){
        obj[arr[i]] = 1;
        tmp.push(arr[i]);
    }
  }
  console.log(tmp);
```

结果如下： [0, 2, 3, 4]

方法二：

```
  var arr = [2,3,4,4,5,2,3,6],
    arr2 = [];for(var i = 0;i< arr.length;i++){
     if(arr2.indexOf(arr[i]) < 0){
         arr2.push(arr[i]);
     }
  }
  console.log(arr2);
```

结果为：[2, 3, 4, 5, 6]

方法三：

```
  var arr = [2,3,4,4,5,2,3,6];
var arr2 = arr.filter(function(element,index,self){
return self.indexOf(element) === index;
});
console.log(arr2);
```

结果为：[2, 3, 4, 5, 6]

8. 题目 8

判断一个字符串中出现次数最多的字符，统计这个次数

```
  var str = 'asdfssaaasasasasaa';var json = {};for (var i = 0; i <
str.length; i++) {
```

```
    if(!json[str.charAt(i)]){
       json[str.charAt(i)] = 1;
    }else{
       json[str.charAt(i)]++;
    }
};var iMax = 0;var iIndex = '';for(var i in json){
    if(json[i]>iMax){
        iMax = json[i];
        iIndex = i;
    }
}       console.log('出现次数最多的是:'+iIndex+'出现'+iMax+'次');
```

9. 题目 9

点击一个 ul 的五个 li 元素，分别弹出他们的序号，怎么做？

方法 1：

```
for(var i=0; i<oLis.length; i++){
    oLis[i].onclick = (function(j){
        return function(){
            alert(j);
        }
    })(i);
}
```

这样的话，给每个 li 绑定 onclick 事件时，其实绑的是一个立即执行函数，这个立即执行函数的参数是 i，因为它是立即执行的，循环时已经把 i 的值赋给了 li 的 onclick 事件，所以在外部函数里的 i 改变后并不会影响 i 的值.

另一种实现方法:(立即执行函数)

```
for(var i=0; i<oLi.length; i++){
    (function(j){
        oLi[j].onclick = function(){
```

```
                alert(j);
            };
        })(i);
    }
```

或者不用闭包

方法 2:

```
    var oLi = document.getElementsByTagName('li');
    function func(obj,i){
        obj.onclick = function(){
            alert (i);
        }
    }
    for(var i = 0; i<oLi.length; i++){
        func(oLi[i], i);
    }
```

方法 3：设置属性:

```
    var oLi = document.getElementsByTagName('li');
    for(var i=0; i<oLi.length; i++){
        oLi[i].setAttribute("onclick", "alert("+i+");");
    }
```

方法 4：设置 index 保存

```
    for(var i=0; i<oLi.length; i++){
        oLi[i].index = i;
        oLi[i].onclick = function(){
            alert(this.index);
        }
```

```
    }
```

或者也可以用事件代理来做.

10. 题目 10

js 实现数组去重怎么实现?

方法 1. 创建一个新的临时数组来保存数组中已有的元素

```
var a = new Array(1,2,2,2,2,5,3,2,9,5,6,3);
Array.prototype.unique1 = function(){
    var n = [];     //一个新的临时数组
    for(var i=0; i<this.length; i++){
        //如果把当前数组的第 i 已经保存进了临时数组, 那么跳过
        if(n.indexOf(this[i]) == -1){
            n.push(this[i]);
        }
    }
    return n;
}
console.log(a.unique1());
```

方法 2. 使用哈希表存储已有的元素

```
Array.prototype.unique2 = function(){
    var hash = {},
        n = [];     //hash 作为哈希表, n 为临时数组
    for(var i=0; i<this.length; i++){
        if(!hash[this[i]]){         //如果 hash 表中没有当前项
            hash[this[i]] = true;   //存入 hash 表
            n.push(this[i]);        //当前元素 push 到临时数组中
        }
    }
    return n;
}
```

方法 3. 使用 indexOf 判断数组元素第一次出现的位置是否为当前位置

```
Array.prototype.unique3 = function(){
    var n = [this[0]];
    for(var i=1; i<this.length; i++)    //从第二项开始遍历
```

```
        {
            //如果当前数组元素在数组中出现的第一次的位置不是 i
            //说明是重复元素
            if(this.indexOf(this[i]) == i){
                n.push(this[i]);
            }
        }
        return n;
    }
```

方法 4. 先排序再去重

```
    Array.prototype.unique4 = function(){
        this.sort(function(a, b){ return a - b;});
        var n = [this[0]];
        for(var i=1; i<this.length; i++){
            if(this[i] != this[i-1]){
                n.push(this[i]);
            }
        }
        return n;
    }
```

第一种方法和第三种方法都使用了 indexOf(), 这个函数的执行机制也会遍历数组

第二种方法使用了一个哈希表, 是最快的.

第三种方法也有一个排序的复杂度的计算.

然后做了个测试, 随机生成 100 万个 0-1000 的数组结果如下:

```
方法一执行时间:240
方法二执行时间:5
方法三执行时间:477
方法四执行时间:246
```

第三种方法总是第二种方法的将近两倍, 而第四种方法与数组的范围有关,

如果是 0-100 的数组

```
方法一执行时间:40
方法二执行时间:4
方法三执行时间:64
方法四执行时间:176
```

而如果是 0-10000, 方法四看着就效果还不错了

```
方法一执行时间:2199
方法二执行时间:6
方法三执行时间:4876
方法四执行时间:298
```

而第二种方法永远是最好的，但是是以空间换时间

全部代码如下：

```
var a = [];
for(var i=0; i<1000000; i++){
    a.push(Math.ceil(Math.random()*10000));
}

Array.prototype.unique1 = function(){
    var n = [];     //一个新的临时数组
    for(var i=0; i<this.length; i++){
        //如果把当前数组的第 i 已经保存进了临时数组，那么跳过
        if(n.indexOf(this[i]) == -1){
            n.push(this[i]);
        }
    }
    return n;
}

Array.prototype.unique2 = function(){
    var hash = {},
        n = [];     //hash 作为哈希表，n 为临时数组
    for(var i=0; i<this.length; i++){
        if(!hash[this[i]]){         //如果 hash 表中没有当前项
            hash[this[i]] = true;   //存入 hash 表
            n.push(this[i]);        //当前元素 push 到临时数组中
        }
    }
    return n;
}

Array.prototype.unique3 = function(){
    var n = [this[0]];
```

```
        for(var i=1; i<this.length; i++)    //从第二项开始遍历
        {
            //如果当前数组元素在数组中出现的第一次的位置不是 i
            //说明是重复元素
            if(this.indexOf(this[i]) == i){
                n.push(this[i]);
            }
        }
        return n;
    }

    Array.prototype.unique4 = function(){
        this.sort(function(a, b){ return a - b;});
        var n = [this[0]];
        for(var i=1; i<this.length; i++){
            if(this[i] != this[i-1]){
                n.push(this[i]);
            }
        }
        return n;
    }
    var begin1 = new Date();
    a.unique1();
    var end1 = new Date();

    var begin2 = new Date();
    a.unique2();
    var end2 = new Date();

    var begin3 = new Date();
    a.unique3();
    var end3 = new Date();

    var begin4 = new Date();
    a.unique4();
    var end4 = new Date();

```

```
69      console.log("方法一执行时间:" + (end1 - begin1));
70      console.log("方法二执行时间:" + (end2 - begin2));
71      console.log("方法三执行时间:" + (end3 - begin3));
        console.log("方法四执行时间:" + (end4 - begin4));
```

11. 题目 11

写出一个简单的$.ajax()的请求方式?

```
$.ajax({
    url:'http://www.baidu.com',
    type:'POST',
    data:data,
    cache:true,
    headers:{},
    beforeSend: function(){},
    success:function(){},
    error:function(){},
    complete:function(){}
});
```

9.常用技术面试题（系统与软件实施）

9.1 计算机基础

1. 你熟悉的远程有哪些方法？各种方法应该怎么配置？

（1）最简单的 QQ 上有，打开对话框　上边有个 “应用”图标　点击“远程协助”。

（2）系统自带的远程桌面服务，右击我的电脑—属性，点远程，把两个够都打上去。

（3）远程协助软件，在要远程的主机安装代理程序后，即可使用远程服务。

（4）专业通信系统，即时通、 OA 之类的。

2. 在你进行实施的过程中，公司制作的一款软件系统缺少某一项功能，而且公司也明确表示不会再为系统做任何的修改或添加任何的功能，而客户也坚决要求需要这一项功能！对于实施人员来说，应该怎么去合理妥善处理这个问题？

先看客户要求合不合理，不合理就可以坚决退还需求。如果需求合理的话，可以考虑做以下选项：

（1）申请做二次开发，并且收取一定的费用,这个两边都要沟通好。

（2）第二种方法,使用第三方软件做补助。

3. 在项目实施过程中，使用者对产品提出了适合自己习惯的修改意见，但多个使用者相互矛盾，应该如何去处理？

对于客户提出的修改意见，我们实施人员应该有自己的方案。当使用者之间意见出现不一致时，我们应当引导他们内部之间的意见统一，和客户经过沟通或确认后，找到切实可行的方案，双方认可并达成共识。

4. 同一个网络环境中，A 电脑访问不到 B 电脑的共享文件。此现象可能是哪些方面所导致？怎样处理？

首先检查网络是否有问题，再确定是不是在一个工作组内，只有在一个工作组内才可以共享文件，然后看有没有被防火墙阻止，最后确定文件是不是已经被共享。

5. 什么是 DHCP？如何快速为多台（20 台）电脑安装操作系统？多台电脑如何组网？

（1）、DHCP：动态主机设置协议，是一个局域网的网络协议，使用 UDP 协议工作，主要有两个用途：给内部网络或网络服务供应商自动分配 IP 地址、给用户给内部网络管理员作为对所有计算机作中央管理的手段。

（2）、可以通过网络硬盘克隆，过程为：在装有软驱的工作站上，用一张引导盘来启动机器，连接到服务器，使用 Ghost 多播服务（Multicast Server）将硬盘或分区的映像克隆到工作站，这样就实现了不拆机、安全、快速的网络硬盘克隆。

（3）、多台电脑组网可以分为两个类型：

①、少于 250 台：可以采用用户接入层和核心接入层这二层网络结构，通过普通二层交换机与核心交换机的堆叠连接组成单位局域网，以满足单位各种上网访问需求。普通电脑通过双绞线连接到普通百兆二层交换机。

②、超过 250 台：我们就需要通过交换机的 VLAN 功能，将它们划分到不同的子网中。为了让两网段中的所有电脑都能实现共享上网目的，我们还需要在核心路由交换机或者双 WAN 端口路由器设备中对两个网关参数进行合适配置，确保各个子网中的电脑能通过局域网路由功能访问 Internet 网络。

6. 局域网内，一台机器不能上网，而其他机器可以。所有的机器都安装的 WinXP 系统，且该电脑可以访问局域网内电脑，试分析原因？

可能由如下原因导致：

（1）、检查有无 Microsoft 网络客户端、 Microsoft 网络的文件和打印机共享、 Internet 协议(TCP/IP) 。

（2）、检查 IP 地址、网关、DNS 、网络是否连上等。

（3）、查杀木马、病毒。

7. 如果有一个不太懂电脑的客户，你应该采取什么样的方法去教他用公司的软件产品？

（1）、如果软件产品比较难懂，你就可以先教一些简单的。再告诉他需要再了解哪些知识来掌握这个软件。

（2）、如果软件产品比较简单，就可以直接一步一步的教他怎么操作，一直操作熟练就行了。

8. 当你觉得工作的付出和你的收入不成正比的时候你会怎么想？

无论干什么工作，必须干一行爱一行，脚踏实地、用心去钻研， 只要真正有能力，只要有思想和技术，终会出头。砖石总会发光的。接受你不能接受的，改变你能改变的。会争取到更高的薪水

的。

如果当初进来的时候公司有晋升调薪的承诺，那就看你的表现是否达到了要求，可以主动和相关领导沟通。

9. 系统启动后，不能连接数据库，可能是哪些方面的原因?

（1）和数据库有关的服务没启动；

（2）防火墙可能阻挡了数据库的端口；

（3）如数据库可以启动，而登陆不了，可能是密码错误或连接参数配置错误；

（4）数据库文件已被破坏或不存在；

10. 你认为客户服务的重点是什么?

随着市场的竞争进一步加剧，服务已经成为企业核心竞争力的要素之一，服务的重点是沟通，沟通可以消除客户的误会和不满，沟通可以提高客户的感知度。因此，我认为我们客户服务管理工作就应该从做好沟通的管理开始。

自己一定要理解服务，理解服务能干什么，能做到什么，结合公司的业务能给客户提供什么服务。服务过程中是否能给客户提供优秀的服务，倾听客户的意见，持续改进服务方式。尽量在事件发生之前，避免或杜绝客户的投诉，投诉发生后，认真处理。

11. 说明静态路由和动态路由的区别?

静态路由：就是由管理员在路由器中手工设置的固定的路由信息，静态路由不能对网络的改变做出反映，一般用于规模不大、拓扑结构固定的网络中，其优点是设置简单、高效，在所有路由中，静态路由优先级最高，当动态路由与静态路由发生冲突时，以静态路由为准。

动态路由：就是由网络中的路由器之间互相通信，传递路由信息，利用收到的路由信息更新路由表的过程，它能实时地适应网络结构的变化。主要用于规模大、拓扑结构复杂的网络。

12. 简述有哪些常用的虚拟机软件?

Vmware、VirtualBox、KVM (Kernel-based Virtual Machine)

13. 简要说明 Active Directory 和 Active Directory Domain Services（ADDS）

Active Directory（活动目录）是面向 Windows Standard Server、Windows Enterprise Server 以及 Windows Datacenter Server 的目录服务。

Active Directory 域内的 directory database（目录数据库）被用来存储用户账户、计算机账户、打印机与共享文件夹等对象，而提供目录服务的组件就是 Active Directory 域服务（Active

Directory Domain Services，ADDS），它负责目录数据库的存储、添加、删除、修改与查询等操作

14. 简要说明域和站点的联系和区别

域是逻辑的分组，站点是物理的分组。在 Active Directory 内，每个站点可能包含多个域；而一个域内的计算机也可能分别属于不同的站点。

15. 简要说明 Windows Server 系统中用户有效权限具备哪几个特点

（1）权限可以被继承。例如，设置用户 A 对甲文件夹拥有读取的权限，则用户 A 对甲文件夹内的文件也会拥有读取的权限。

（2）权限可以累加。例如，若用户 A 同时属于业务部与经理组，用户 A 自身的权限为“写入”，业务部权限为“读取”，经理组权限为“读取、执行”，则用户 A 最后的有效权限为：写入+读取+执行

（3）“拒绝”权限的优先级比较高。虽然用户对某个文件的有效权限是其所有权限来源的总和，但是只要其中有一个权限来源被设置为“拒绝”，则用户将不会拥有此权限。例如，例如，若用户 A 同时属于业务部与经理组，并且其权限分别如下表所示，则用户 A 的读取权限会被拒绝，也就是无法读取此文件。

“拒绝”权限的优先级比较高示例表

用户或组	权限
用户 A	读取
业务部组	读取被拒绝
经理组	修改
用户 A 的读取权限为：拒绝	

16. 访问 Windows Server 脱机文件的好处是什么？如何设置网络计算机的脱机文件?

脱机文件让你的计算机在与公司网络未连接的情况下（脱机），仍然可以访问原本位于网络计算机内的文件。图中用户在脱机的情况下，其所访问的文件并不是真正位于网络计算机内的文件，而是存储在本地计算机硬盘内的缓存版本。由于访问的是本地计算机资源，因此访问速度比访问远程的网络计算机要快。

网络计算机的文件夹必须是共享文件夹才具备被脱机使用的功能。

在网络计算机端设置脱机文件的步骤是：

选中共享文件夹并单击鼠标右键→【属性】，单击【共享】标签下的【高级共享】按钮，单击【缓存】按钮。

17. Windows Server 内置的备份和还原工具是哪个？备份有哪几种形式

Windows Server 内置的备份和还原工具是 Backup。备份形式有：一次性备份、自动备份

18. 简要说明 Windows Server 组策略。组策略有哪两种形式?

Windows Server 组策略是 Microsoft Windows 系统管理员为计算机和用户定义的，用来控制应用程序、系统设置和管理模板的一种机制。通俗一点说，是介于控制面板和注册表之间的一种修改系统、设置程序的工具。

组策略有两种形式:

（1）**计算机配置**

当计算机开机时，系统会根据计算机配置的内容来设置计算机环境。包括桌面外观、安全设置、应用程序分配和计算机启动和关机脚本运行等

（2）**用户配置**

当用户登录时，系统会根据用户配置的内容来设置计算机环境。包括应用程序配置、桌面配置、应用程序分配和计算机启动和关机脚本运行等。

19. 简要说明 ftp、smtp、pop3、telnet

（1）ftp

ftp（File Transfer Protocol，文件传输协议）是 TCP/IP 协议组中的协议之一。默认情况下 FTP 协议使用 TCP 端口中的 20 和 21 这两个端口，其中 20 用于传输数据，21 用于传输控制信息。

（2）smtp

smtp 是一种提供可靠且有效的电子邮件传输的协议。SMTP 是建立在 FTP 文件传输服务上的一种邮件服务，主要用于系统之间的邮件信息传递，并提供有关来信的通知。smtp 默认端口 25。

（3）pop3

pop 协议支持“离线”邮件处理。其具体过程是：邮件发送到服务器上，电子邮件客户端调用邮件客户机程序以连接服务器，并下载所有未阅读的电子邮件。即 smtp 负责发送邮件，pop3 负责读取邮件。pop3 协议默认的端口号为 110。

（4）telnet

telnet 协议是 Internet 远程登录服务的标准协议和主要方式，它为用户提供了在本地计算机上完成远程主机工作的能力。telnet 默认端口为 23。

20. mysql 用户管理常用相关 ml

（1）创建用户 finley，该用户由本机连接数据库，登录密码为“123@#”

```
create user 'finley'@'localhost' identified by '123@#'
```

（2）创建用户 jack，该用户由 192.168.6.82 连接数据库，登录密码为“123@#”

```
create user 'jack'@'192.168.6.82' identified by '123@#'
```

（3）将用户 finley 变成管理员用户

```
grant all privileges on *.* to 'finley'@'localhost' with grant option
```

with grant option 选项表示该用户可以将自己拥有的权限授权给别人。

注意：经常有人在创建操作用户的时候不指定 with grant option，选致后来该用户不能使用 grant 命令创建用户或者给其它用户授权。

（4）创建用户 smith，该用户在所有的库和表上具有 reload 和 process 权限，登录密码为“123@#”

```
grant reload,process on *.* to 'smith'@'localhost' identified by '123@#'
```

（5）创建用户 keme，该用户在 test 库 temp 表上的 id 列具有 select 权限，登录密码为“123@#”

```
grant select(id) on test.temp to keme@'localhost' identified by '123@#'
```

（6）创建用户 sherry，该用户在 test 库 product 表具有 select、insert 权限，登录密码为“123@#”

```
grant select,insert on test.product to sherry@'localhost' identified by '123@#'
```

（7）查看用户 admin 的权限信息

```
show grants for admin@'localhost'
```

（8）通过 revoke 命令收回用户 admin 在所有库中所有表的 process 权限

```
revoke process on *.* from admin@'localhost'
```

21. 简要说明 mysql 主从复制的原理

MySQL 复制是指将主数据库的 DDL 和 DML 操作通过二进制日志传到从服务器上，然后在从服务器上将这些日志文件重新执行，从而使从服务器与主服务器的数据保持同步

9.2 Linux 操作系统

1. 命令汇总

uname -a # 查看内核/操作系统/CPU 信息

cat /proc/version # 查看操作系统版本

cat /proc/cpuinfo # 查看 CPU 信息

hostname # 查看计算机名

lspci -tv # 列出所有 PCI 设备

lsusb -tv # 列出所有 USB 设备

lsmod # 列出加载的内核模块

env # 查看环境变量资源

free -m # 查看内存使用量和交换区使用量

df -h # 查看各分区使用情况

du -sh <目录名> # 查看指定目录的大小

grep MemTotal /proc/meminfo # 查看内存总量

grep MemFree /proc/meminfo # 查看空闲内存量

uptime # 查看系统运行时间、用户数、负载

cat /proc/loadavg # 查看系统负载磁盘和分区

mount | column -t # 查看挂接的分区状态

fdisk -l # 查看所有分区

swapon -s # 查看所有交换分区

dmesg | grep IDE # 查看启动时 IDE 设备检测状况网络

ifconfig # 查看所有网络接口的属性

iptables -L # 查看防火墙设置(centos7 版本以下)

firewall-cmd --list-all # 查看防火墙规则（centos7 版本及以上）

route -n # 查看路由表

netstat -lntp # 查看所有监听端口

netstat -antp # 查看所有已经建立的连接

netstat -s # 查看网络统计信息进程

ps -ef # 查看所有进程

top # 实时显示进程状态用户（q 键退出当前状态）

w # 查看活动用户

id <用户名> # 查看指定用户信息

last # 查看用户登录日志

cut -d: -f1 /etc/passwd # 查看系统所有用户

cut -d: -f1 /etc/group # 查看系统所有组

crontab -l # 查看当前用户的计划任务服务

chkconfig –list # 列出所有系统服务（centos7 版本以下）

systemctl list-unit-files # 列出所有系统服务（centos7 版本及以上）

chkconfig –list | grep on # 列出所有启动的系统服务程序（centos7 版本以下）

systemctl list-units --type=service #列出所有启动的系统服务程序（centos7 版本及以上）

rpm -qa # 查看所有安装的软件包

rpm -qa | grep <软件包名> # 查看是否安装了某 rpm 包软件

Pwd # 该命令用于显示用户当前所在的工作目录位置

Cd # 该命令用于切换工作目录

Ls # 列表显示目录

-l # 以长格式显示文件或目录的详细信息

-a # 显示所有文件，包括隐藏文件

-d # 显示目录本身的属性而不是显示目录的内容

-h # 以 K,M,G 等单位显示文件大小提高可读性

Mkdir # 创建目录 -p 创建多个目录

Touch # 建立空文件或修改时间戳

rm # 删除文件或目录

-r # 删除目录时必须使用此选项表示递归删除整个目录

-f # 强制删除，不需要用户确认

cp # 复制文件或目录

-r # 复制目录必须使用此选项表示递归复制所有文件及子目录

mv # 移动文件或目录

cat # 显示文件内容

-n ：-n 或 --number # 由 1 开始对所有输出的行数编号

more/less # 分页显示文件内容

head/tail # 查看文件开头或末尾的部分内容

wc # 文件内容统计

-l # 统计行数

find #文件或目录查找

-name # 按名称查找

-type # 按类型查找

-size # 按大小查找

grep #文件内容查询命令

-v # 反转查找即输出与查找条件不相符的行

-I # 忽略大小写

which # 查找外部命令所对应的程序文件

ln # 为文件或目录建立链接

-s # 创建软链接

alias # 设置命令别名

history # 查看历史命令

man/help # 查看命令帮助手册

2. 在 Linux 中有一文件列表内容格式如下：

63 lrwxrwxrwx 1 hawkeye users 6 Jul 18 09:41 nurse2 -> nurse1

问题一：要完整显示如上文件列表信息，应该使用什么命令。请写出完整的命令行。

问题二：上述文件列表内容的第一列内容“63”是什么含义？

问题三：上述文件列表内容的第二列内容“lrwxrwxrwx”中的“l”是什么含义？对于其它类型的文件或目录等还可能会出现什么字符，它们分别表示什么含义？

问题四：上述文件列表内容的第二列内容“lrwxrwxrwx”中的第一、二、三个“rwx”分别代表什么含义？其中的“r”、“w”、“x”分别表示什么含义？

问题五：上述文件列表内容的第三列内容“1”是什么含义？

问题六：上述文件列表内容的第四列内容“hawkeye”是什么含义？

问题七：上述文件列表内容的第五列内容“users”是什么含义？

问题八：上述文件列表内容的第六列内容“6”是什么含义？

问题九：上述文件列表内容中的“Jul 18 09:41”是什么含义？

问题十：上述文件列表内容的最后一列内容“nurse2>nurse1”是什么含义？

答：

问题一	ls －il nurse2
问题二	为文件 nurse2的索引节点号
问题三	表示文件类型，该文件为符号链接文件 其他文件类型有：-普通文件，d 目录，b 特殊块文件，c 特殊字符文件
问题四	分别表示对文件 nurse2的所有者、同组成员、其他人员都具有读/写/执行权限 r/w/x 分别表示读/写/执行权限
问题五	表示文件 nurse2的链接数
问题六	表示文件 nurse2的所有者
问题七	表示文件 nurse2的属组
问题八	表示文件 nurse2的字节数
问题九	表示文件 nurse2被创建的日期和时间
问题十	表示 nurse2文件被符号链接到 nurse1文件

3. 简述 Linux 的主要特点?

答：a.LUNIX 是一个分时、多用户、多任务的系统；

b、内核和核外程序的有机结合；

c、良好的用户界面；

d、树形结构的文件系统；

e、文件和设备的统一处理；

f、丰富的应用程序；

g、系统用 C 语言写成，具有良好的可移植性。

4. Linux 有哪些启动方式?

答：可以从软盘或硬盘引导 Linux。用 GRUB 引导。

也可以回答为：命令行启动模式和图形界面启动模式。

5. vi 编辑器有哪些模式？不同模式如何切换?

答：命令模式，插入模式，末行模式。不论任何模式按下 ESC 即进入命令模式　在命令模式下

i,o,a 或 insert 健就可以切换到插入模式　在命令模式下按：即可以进入末行模式。

6. 安装 Linux 系统有哪些方式?

答：图形安装或文本终端安装。

7. 管道的作用

答：管道就是把一个程序（进程）的输出连接到另一个程序（进程）的输入。

8. 简述标准的 Linux 运行级?

答：linux 下有 7 个运行级（Linux7 版本以下）

0 - 停机 ，机器关闭。

1 - 单用户模式 。就像 Win9x 下的安全模式类似

2 - 多用户，但是没有 NFS　进入无网络服务的多用户模式

3 - 完全多用户模式 ，是标准的运行级。

4 - 没有用到 ，一般不用。

5 - X11　，进到 X Window 系统了。

6 - 重新启动 ，运行 init 6 机器就会重启

这些运行级别是通过 init 调用不同的脚本来切换的。

CentOS 7 运行级别的切换

由命令行级别切换到窗口级别的命令未变：init 5 或 startx

由窗口级别切换到命令行级别的命令未变：init 3

新版本的运行级别都定义在 /lib/systemd/system 下:

运行级别	systemctl target
0	poweroff.target
1	rescure.target
2	multi-user.target
3	multi-user.target
4	multi-user.target
5	graphical.target
6	reboot.target

9. GRUB 是什么，它有什么作用?

答：在 Linux 系统安装中，GRUB 已经在 MBR 中写入引导代码，使得 RHEL 获得系统的控制权

引导代码加载启动分区 boot 或在 / 分区中加载 boot 目录，读取第二阶段引导代码

第二阶段引导代码加载系统内核

内核在引导过程中加载系统所需的模块

内核最后执行/sbin/init 程序(系统根进程)

init 根据/etc/inittab 文件，运行系统进行初始化程序(/etc/rc.d/rc.sysinit)

rc.sysinit 根据/etc/inittab 决定操作系统运行的级别，并且根据运行级别启动相应的服务，然后根据/etc/fstab 加载系统分区

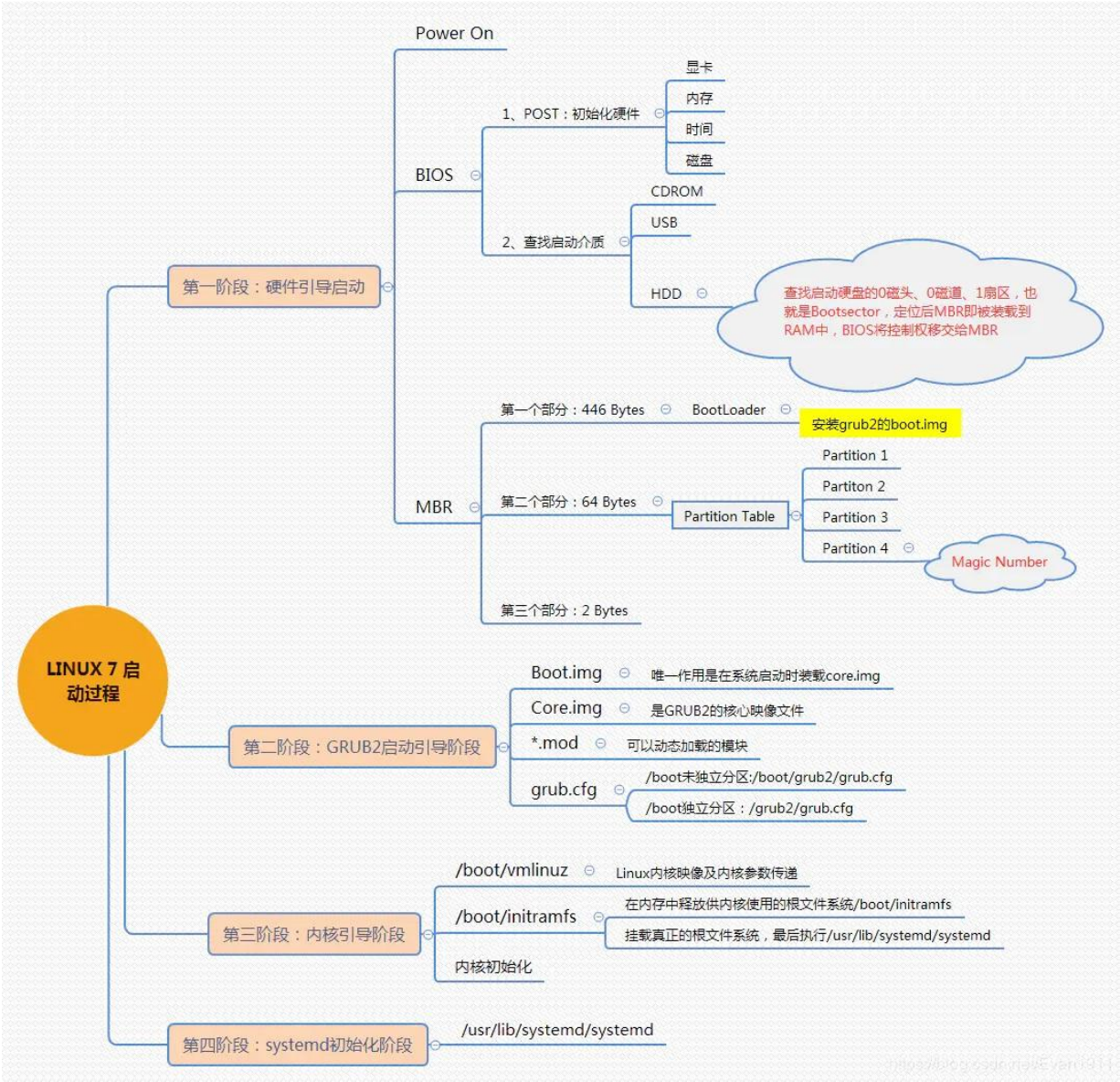
标准运行级别通长在最后运行本地初始化脚本 (/etc/rc.d/rc.local)

显示登录信息

10. 如何设置自动挂载文件系统?

答：利用 vi 编辑器修改/etc/fstab 文件 在最下面一行增加所挂载文件、所要挂载到的目录、文件类型，使用 mount -a 自动挂载

11. centos7 的启动顺序



12. 什么是 Shell，它的作用是什么?

答：Shell 是人机交互用的一个程序。其实就是字面意思--壳. 一个系统有内核(Kernel)自然就有外壳(Shell). 在 Linux 上广义的 Shell 就是与用户交互的界面, 图形化界面是 Shell, 命令行界面也是 Shell. 狭义上的 Shell 就是 Linux 上的命令行界面. 一般发行版上的命令行界面是 bash, 而 debian 的是 dash. 下面的讨论基于侠义的 Shell.

Shell 是用来执行计算机程序的. 例如执行 mkdir 就是创建一个文件夹, 执行 touch 就是创建一个文件. 执行 ps 就是查看进程信息. 其中 mkdir, touch, ps 都是用 C 语言编写编译后的程序

13. 简述 Swap 分区的作用?

答：当系统执行的任务临时要求大内存而得不到满足时，将部分不活跃的内存内容移到硬盘上

的一个缓存区域，这个专门预留的区域就是 swap 分区（相当于 Windows 的页面文件）。内存够用的情况下，如果不用睡眠功能，则 swap 分区可以小一点，甚至没有；如果要使用睡眠功能，则 swap 分区尽量不要比内存小。

14. 简述 Linux 文件系统通过 i 节点把文件的逻辑结构和物理结构转换的工作过程。

Linux 通过 i 节点表将文件的逻辑结构和物理结构进行转换。i 节点是一个 64 字节长的表，表中包含了文件的相关信息，其中有文件的大小、文件所有者、文件的存取许可方式以及文件的类型等重要信息，在 i 节点表中最主要的内容是磁盘地址表。在磁盘地址表中有 13 个块号，文件将以块号在磁盘地址表中出现的顺序依次读取相应的块。Linux 文件系统通过把 i 节点和文件名进行连接，当需要读取该文件时，文件系统在当前目录表中查询该文件名对应的项，由于此得到该文件相对应的 i 节点号，通过该 i 节点的磁盘地址表把分散存放的文件物理块连接成文件的逻辑结构。

15. 简述进程的启动、终止的方式以及如何进行进程的查看。

在 Linux 中启动一个进程有手工启动和调度启动两种方式：

（1）手工启动用户在输入端发出命令，直接启动一个进程的启动方式。可以分为： ①前台启动：直接在 SHELL 中输入命令进行启动。 ②后台启动：启动一个目前并不紧急的进程，如打印进程。

（2）调度启动系统管理员根据系统资源和进程占用资源的情况，事先进行调度安排，指定任务运行的时间和场合，到时候系统会自动完成该任务。

经常使用的进程调度命令为：at、batch、crontab。

16. Linux 系统中软件安装方法有哪些？简述优缺点。

有三种方法：源代码安装，rpm 包安装，yum 安装

源码安装：通过源码安装，用户可以获得最新的应用程序，可以定制灵活，丰富的功能，而且使软件可以跨越计算机平台，在所有版本的 Linux 系统中都能使用。但是这种安装方式过于复杂，耗时又长，对用户的软件开发能力要求也比较高。

rpm 软件包只能在 RPM 机制的 Linux 操作系统中使用，如 RHEL,Fdeora，Suse 等，RPM 安装包现在基本成为 Linux 系统中软件安装包事实上的标准但是 RPM 也有一个很大的缺点，即 RPM 软件包之间存在复杂的依赖关系。在多数情况下，一个软件都是由多个相互依赖的 RPM 软件包组成，也就是安装一个软件需要使用到许多软件包，而大部分的 RPM 又有相互之间的依赖关系。例如安装 A 软件包需要 B 软件包的支持，安装 B 需要 C 的支持

yum 是一个基于 RPM 却胜于 RPM 的软件管理工具，它的最大优点是可以自动解决 RPM 软件包的依赖性问题，从而可以更轻松的管理 Linux 系统中的软件。

17. 如何安装和卸载 rpm 格式的软件包?

利用 rpm 安装首先必须进入存放 rpm 软件包的目录.

rpm -ivh 软件包名(-i 安装软件包，-v 显示安装过程，-h 显示安装进度，rpm 每 2%就会显示一个#号)

rpm -e 软件名（删除一个已安装的软件，当删除成功时，没有任何提示；当再次删除时，会提示软件包没有安装）

18. 简述安装 RHEL6 的步骤。

准备 RHEL6 镜像从光盘引导在引导界面选择第一项“Install or upgrade an existing system”然后检查光盘兼容性这里选择 Skip 跳过安装语言选择中文简体键盘选择美国英语式存储设备选择基本存储设备弹出存储设备警告提示是否要将整个硬盘重新分区格式化这里选择是定义主机名和时间为用户设置密码将硬盘分区然后进行软件包的定制最后完成安装。

19. yum 源的配置方式有哪些？说明以下 yum 命令的作用。

如果是合法的 RHEL 用户会自动使用 RHN 作为默认 yum 源，也就是说安装的所有软件都可以直接来自红帽官网。但这要缴纳一定的费用，所以更多的情况下需要自己来配置 yum 源，由于在 RHEL6 的系统光盘中已经集成了绝大多数应用软件的 rpm 包因此一般可以指定系统光盘作为 yum 源，或者是指定网络中的某台 ftp 或 web 服务器作为 yum 源

yum list 列出软件清单

yum info 查看软件包的信息

yum install 安装软件

yum remove 卸载软件

yum clean all 清除 yum 缓存

20. 企业使用最多的 Linux 操作系统有哪些?

RHEL、Centos、ubuntu, Debian, suse、目前中国企业用的比较多的 linux 操作系统。

21. 简述 DNS 进行域名正向解析的过程。

首先，客户端发出 DNS 请求翻译 IP 地址或主机名。DNS 服务器在收到客户机的请求后：

（1）检查 DNS 服务器的缓存，若查到请求的地址或名字，即向客户机发出应答信息；

（2）若没有查到，则在数据库中查找，若查到请求的地址或名字，即向客户机发出应答信息；

（3）若没有查到，则将请求发给根域 DNS 服务器，并依序从根域查找顶级域，由顶级查找二级域，二级域查找三级，直至找到要解析的地址或名字，即向客户机所在网络的 DNS 服务器发出应答信息，DNS 服务器收到应答后现在缓存中存储，然后，将解析结果发给客户机。

（4）若没有找到，则返回错误信息。

22. 系统管理员的职责包括那些？管理的对象是什么？

系统管理员的职责是进行系统资源管理、设备管理、系统性能管理、安全管理和系统性能监测。管理的对象是服务器、用户、服务器的进程及系统的各种资源等。

23. 简述安装 Slackware Linux 系统的过程。

（1）对硬盘重新分区。

（2）启动 Linux 系统（用光盘、软盘等）。

（3）建立 Linux 主分区和交换分区。

（4）用 setup 命令安装 Linux 系统。

（5）格式化 Linux 主分区和交换分区

（6）安装 Linux 软件包

（7）安装完毕，建立从硬盘启动 Linux 系统的 LILO 启动程序，或者制作一张启动 Linux 系统的软盘。重新启动 Linux 系统。

24. 什么是静态路由，其特点是什么？什么是动态路由，其特点是什么？

静态路由是由系统管理员设计与构建的路由表规定的路由。适用于网关数量有限的场合，且网络拓朴结构不经常变化的网络。其缺点是不能动态地适用网络状况的变化，当网络状况变化后必须由网络管理员修改路由表。

动态路由是由路由选择协议而动态构建的，路由协议之间通过交换各自所拥有的路由信息实时更新路由表的内容。动态路由可以自动学习网络的拓朴结构，并更新路由表。其缺点是路由广播更新信息将占据大量的网络带宽。

25. 进程的查看和调度分别使用什么命令？

进程查看的命令是 ps 和 top。

进程调度的命令有 at，crontab，batch，kill。

26. Linux 系统中用户的分类？区别？

分类：超级用户、普通用户、虚拟用户

区别：超级用户：对本主机有至高无上的完全权限

普通用户：由 root 用户或其他管理员用户创建，拥有的权限受到一定限制，一般只有在用户自己的宿主目录中有完全权限

虚拟用户：不能登陆系统，主要是为了方便系统管理，用于维持系统或某个程序的正常运行，大多是在安装系统及部分应用程序时自动添加的。

27. 当文件系统受到破坏时，如何检查和修复系统?

成功修复文件系统的前提是要有两个以上的主文件系统，并保证在修复之前首先卸载将被修复的文件系统。

使用命令 fsck 对受到破坏的文件系统进行修复。fsck 检查文件系统分为 5 步，每一步检查系统不同部分的连接特性并对上一步进行验证和修改。在执行 fsck 命令时，检查首先从超级块开始，然后是分配的磁盘块、路径名、目录的连接性、链接数目以及空闲块链表、i-node。

28. 解释 i 节点在文件系统中的作用。

在 linux 文件系统中，是以块为单位存储信息的，为了找到某一个文件在存储空间中存放的位置，用 i 节点对一个文件进行索引。i 节点包含了描述一个文件所必须的全部信息。所以 i 节点是文件系统管理的一个数据结构。

29. 什么是符号链接，什么是硬链接？符号链接与硬链接的区别是什么?

链接分硬链接和符号链接。

符号链接可以建立对于文件和目录的链接。符号链接可以跨文件系统，即可以跨磁盘分区。符号链接的文件类型位是 l，链接文件具有新的 i 节点。

硬链接不可以跨文件系统。它只能建立对文件的链接，硬链接的文件类型位是一，且硬链接文件的 i 节点同被链接文件的 i 节点相同。

30. 在对 linux 系统分区进行格式化时需要对磁盘簇（或 i 节点密度）的大小进行选择，请说明选择的原则。

磁盘簇（或 i 节点密度）是文件系统调度文件的基本单元。磁盘簇的大小，直接影响系统调度磁盘空间效率。当磁盘分区较大时，磁盘簇也应选得大些；当分区较小时，磁盘簇应选得小些。通常使用经验值。

31. 简述网络文件系统 NFS，并说明其作用。

网络文件系统是应用层的一种应用服务，它主要应用于 Linux 和 Linux 系统、Linux 和 Unix 系统之间的文件或目录的共享。对于用户而言可以通过 NFS 方便的访问远地的文件系统，使之成为本地文件系统的一部分。采用 NFS 之后省去了登录的过程，方便了用户访问系统资源。

32. 某/etc/fstab 文件中的某行如下：

/dev/had5 /mnt/dosdata msdos defaults, usrquota 1 2

请解释其含义。

（1）第一列：将被加载的文件系统名；

（2）第二列：该文件系统的安装点；

（3）第三列：文件系统的类型；

（4）第四列：设置参数；

（5）第五列：供备份程序确定上次备份距现在的天数；

（6）第六列：在系统引导时检测文件系统的顺序。

33. Apache 服务器的配置文件 httpd.conf 中有很多内容，请解释如下配置项：

（1）MaxKeepAliveRequests 200

（2）UserDir public_html

（3）DefaultType text/plain

（4）AddLanguare en.en

（5）DocumentRoot“/usr/local/httpd/htdocs”

（6）AddType application/x-httpd-php.php.php.php4

答：

（1）允许每次连接的最大请求数目，此为 200；

（2）设定用户放置网页的目录；

（3）设置服务器对于不认识的文件类型的预设格式；

（4）设置可传送语言的文件给浏览器；

（5）该目录为 Apache 放置网页的地方；

（6）服务器选择使用 php4。

34. 某 Linux 主机的/etc/rc.d/rc.inet1 文件中有如下语句，请修正错误，并解释其内容。

/etc/rc.d/rc.inet1：

……

ROUTE add －net default gw 192.168.0.101 netmask 255.255.0.0 metric 1

ROUTE add －net 192.168.1.0 gw 192.168.0.250 netmask 255.255.0.0 metric 1

（1）ROUTE 应改为小写：route；

（2）netmask 255.255.0.0 应改为:netmask 255.255.255.0；

（3）缺省路由的子网掩码应改为:netmask 0.0.0.0；

（4）缺省路由必须在最后设定,否则其后的路由将无效。

解释内容:

（1）route：建立静态路由表的命令；

（2）add：增加一条新路由；

（3）-net 192.168.1.0：到达一个目标网络的网络地址；

（4）default：建立一条缺省路由；

（5）gw 192.168.0.101：网关地址；

（6）metric 1：到达目标网络经过的路由器数（跳数）。

35. 试解释 apache 服务器以下配置的含义：

（1）port 1080 （2）UserDir userdoc

（3）DocumentRoot “/home/htdocs”

（4） Options Indexes FollowSymLinks

AllowOverride None

Order deny,allow

deny from all

allow from 192.168.1.5

（5）Server Type Standlone

Apache 服务器配置行含义如下：

（1）将 apache 服务器的端口号设定为 1080；

（2）设定用户网页目录为 userdoc；

（3）设定 apache 服务器的网页根目录:/home/htdocs；

（4）在此 apache 服务器上设定一个目录/home/htdocs/inside，且此目录只允许 IP 地址为 192.168.1.5 的主机访问；

（5）定义 apache 服务器以独立进程的方式运行。

36. 简述使用 ftp 进行文件传输时的两种登录方式？它们的区别是什么？常用的 ftp 文件传输命令是什么？

（1）ftp 有两种登录方式：匿名登录和授权登录。使用匿名登录时，用户名为：anonymous，密码为：任何合法 email 地址；使用授权登录时，用户名为用户在远程系统中的用户帐号，密码为用户在远程系统中的用户密码。区别：使用匿名登录只能访问 ftp 目录下的资源，默认配置下只能下载；而授权登录访问的权限大于匿名登录，且上载、下载均可。

（2）常用的 ftp 文件传输命令为：put、get、bye 等

37. 简述 Linux 文件基本权限的表现形式，以及如何更改文件和文件夹的权限

（1）Linux 文件的基本权限分别是 owner/group/others 三种身份各有的 read/write/execute 权限的组合。如文件的权限字符为[-rwxrwxrwx]，这九个权限是每三个一组构成，可以使用数字来

代表各个权限，各权限对应的分数如下：

1）r:4

2）w:2

3）x:1

每种身份（owner/group/others）各自的三个权限（r/w/x）分数是需要累加的，例如，当权限为[-rwxrwx---]，分数如下：

1）owner = rwx = 4+2+1 = 7

2）group = rwx = 4+2+1 = 7

3）others= --- = 0+0+0 = 0

因此根据上述权限，设定权限的变更时，该文件的权限数字就是 770

（2）

1）采用数字法更改文件属性。

例如，执行 `chmod 765 .bashrc`，则文件.bashrc 的权限字符为[-rwxrw-r-x]。

2）采用符号法更改文件属性

a.u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体（group）者，o 表示其他以外的人，a 表示这三者皆是。

b.+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。

例如，`chmod ugo+rw .bashrc`，　`chmod a+rw .bashrc`，　`chmod a+r file1.txt`

38. 简述 RPM 的作用和使用方法

RPM 是 Red-Hat Package Manager（RPM 软件包管理器）的缩写。它作为一个软件包管理工具，它管理着系统已安装的所有 RPM 程序组件的资料。我们也可以使用 RPM 来卸载相关的应用程序。在 Linux 系统中使用 RPM 管理软件包安装最为简便。

rpm 常用选项的说明见下表。

rpm 常用选项的说明

选项	说明
a	查询所有的软件包
e	删除指定的软件包
h	软件包安装时列出标记
i	显示软件包的相关信息
l	显示软件包的文件列表
q	使用询问模式，当遇到任何问题时，rpm 指令会先询问用户
v	显示指令执行过程

使用 rpm 命令直接安装软件包 mypackage。命令如下：

```
# rpm -ivh mypackage
```

使用 rpm 命令强制安装软件包 mypackage（忽略报错）。命令如下：

```
# rpm --force -ivh mypackage
```

使用 rpm 命令查询是否安装 mypackage。命令如下：

```
# rpm -ql mypackage
```

使用 rpm 命令删除 mypackage（如果存在 mypackage）。命令如下：

```
# rpm -e mypackage
```

39. 简述 YUM 的作用和使用方法

yum（全称为 Yellow dog Updater, Modified）是一个在 Fedora 和 RedHat 以及 CentOS 中的 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

Yum 常用命令如下：

```
安装软件(以 foo-x.x.x.rpm 为例）：yum install foo-x.x.x.rpm
删除软件：yum remove foo-x.x.x.rpm 或者 yum erase foo-x.x.x.rpm
升级软件：yum upgrade foo 或者 yum update foo
查询信息：yum info foo
搜索软件（以包含 foo 字段为例）：yum search foo
显示软件包依赖关系：yum deplist foo
```

40. 简述在 Linux 上安装 MySQL 服务

（1）下载 MySQL 安装包文件。

（2）卸载 CentOS-7 中 MySQL 相关的依赖。

（3）分别安装 MySQL 服务端和客户端，命令如下：

```
# rpm -ivh MySQL-server-5.1.73-1.glibc23.x86_64.rpm
# rpm -ivh MySQL-client-5.1.73-1.glibc23.x86_64.rpm
```

（4）启动 MySQL 服务，命令如下：

```
# service mysql start
```

（5）登录 MySQL，命令如下（初次使用时 MySQL 是没有密码的）：

```
# mysql –u root
```

（6）修改 root 账号的密码为“123456”，命令如下：

```
set password for ‘root’@’localhost’ =password(‘123456’);
```

（7）支持 root 用户允许远程连接 MySQL 数据库，命令如下：

```
grant all privileges on *.* to 'root'@'%' identified by '123456' with grant option;
flush privileges;
```

41. 写出在 Linux 上启动、停止和重启 Tomcat 服务的命令

（1）启动 Tomcat：`service Tomcat start`

（2）停止 Tomcat：`service Tomcat stop`

（3）重启 Tomcat：`service Tomcat restart`

9.3 网络系统集成与信息安全

1. 简述下计算机网络的概念。

计算机网络就是利用通信线路和通信设备将分布在不同位置的、具有独立功能的计算机系统连接起来而形成的计算机集合，计算机之间可以借助于通信线路传递信息，共享软件、硬件和数据资源。

2. 计算机网络主要由哪些功能?

（1）资源共享。

（2）快速传输信息。

（3）提高系统可靠性。

（4）易于进行分布式处理。

（5）综合信息服务。

3. 计算机网络按网络覆盖的地理范围分为哪几类?

（1）局域网（Local Area Network，LAN）。

（2）城域网（Metropolitan Area Network，MAN）。

（3）广域网（Wide Area Network，WAN）。

（4）Internet。Internet 即国际互联网。

4. 网络层次结构有哪些特点?

（1）按照结构化设计方法，计算机网络将其功能划分为若干个层次，较高层次建立在较低层次的基础上，并为更高层次提供必要的服务功能。

（2）网络中的每一层都起到隔离作用，使得低层功能具体实现方法的变更不会影响到高层所执行的功能。

5. OSI 参考模型自下而上分为哪七个层次?

物理层、数据链路层、网络层、传输层、会话层、表示层和应用层

6. 简述下 OSI 各层的主要功能。

（1）物理层（Physical Layer），物理层（Physical Layer）处于 OSI 参考模型的最低层。物理层的主要功能是利用物理传输介质为数据链路层提供物理连接，以便透明地传送“比特”流。

（2）数据链路层（Data Link Layer），数据链路层是 OSI 模型中极其重要的一层，它把从物理层来的原

始数据打包成帧。

（3）网络层（Network Layer），网络层定义网络层实体通信的协议，它确定从源结点沿着网络

到

目的结点的路由选择，并处理相关的控制问题。

（4）传输层（Transport Layer），传输层的任务是向用户提供可靠的、透明的和端到端的数据传输，以及差错控制和流量控制机制。

（5）会话层（Session Layer），会话层的任务是建立、管理和终止应用程序进程之间的会话和数据交换。这种会话关系是由两个或多个表示层实体之间的对话构成的。

（6）表示层（Presentation Layer），表示层包含了处理网络应用程序数据格式的协议。

（7）应用层（Application Layer），应用层是最终用户应用程序访问网络服务的地方，是 OSI 参考模型的最高层，是 OSI 参考模型中最靠近用户的一层，它为用户的应用程序提供网络服务。

7. 介绍下星型拓扑结构的特点

（1）结点通过点-点通信线路与中心结点连接。

（2）中心结点控制全网的通信，任何两结点之间的通信都要通过中心结点。

（3）结构简单，易于实现，便于管理。

（4）网络的中心结点是全网可靠性的瓶颈，中心结点的故障可能造成全网瘫痪。

8. 简述下 DNS 的作用

通过 DNS 建立 IP 地址与域名之间的映射关系，将域名解析为 IP 地址或者将 IP 地址解析为域名。

9. 介绍下 Http 协议和 Https 协议的区别

Http 协议运行在 TCP 之上，明文传输，客户端与服务器端都无法验证对方的身份；Https 是身披 SSL(Secure Socket Layer)外壳的 Http，运行于 SSL 上，SSL 运行于 TCP 之上，是添加了加密和认证机制的 HTTP。

10. 介绍下 FTP 协议和 TFTP 协议的特点

（1）FTP 协议提供可靠的文件传输服务，具有认证和权限等功能。

（2）TFTP 协议提供不可靠的文件传输服务，常用于网络设备的配置文件和系统文件传输。

11. 介绍下 TCP 协议和 UDP 协议的特点

（1）TCP 协议是一种可靠的、面向连接的协议，传输效率低，类似打电话。

（2）UDP 协议是一种不可靠、面向无连接的协议，传输效率高，类似群聊。

12. 12、介绍下 MAC 地址的组成。

MAC 地址由两部分组成，分别是供应商代码和序列号。其中前 24 位代表该供应商代码，由 IEEE 管理和分配。剩下的 24 位序列号由厂商自己分配。

13. 简述下 ARP 协议的作用。。

ARP（地址解析协议）负责将相应的 IP 地址解析成 MAC 地址。网络层以上的协议用 IP 地址来标识网络接口，但以太数据帧传输时，以物理地址来标识网络接口。因此需要进行 IP 地址与物理地址之间的转化。ARP 协议提供了网络层地址（IP 地址）到物理地址（mac 地址）之间的动态映射。

14. 简述下 ARP 协议的作用。

ARP（地址解析协议）负责将相应的 IP 地址解析成 MAC 地址。网络层以上的协议用 IP 地址来标识网络接口，但以太数据帧传输时，以物理地址来标识网络接口。因此需要进行 IP 地址与物理地址之间的转化。ARP 协议提供了网络层地址（IP 地址）到物理地址（mac 地址）之间的动态映射。

15. 介绍下 IP 地址的组成及其分类。

IP 地址在形式上是一个 32 位的二进制无符号数，为了表示方便，国际通行一种“点分十进制表示法”。IP 地址用来标识网络中的设备，具有 IP 地址的设备可以在同一网段内或跨网段通信。IP 地址包括两部分，第一部分是网络号，表示 IP 地址所属的网段，第二部分是主机号，用来唯一标识本网段上的某台网络设备。

IP 地址分为 5 类。A、B、C 三类是常用地址。IP 地址的编码规定：全 0 表示本地网络，全 1 表示广播地址。

16. 简述下什么是网关。

网关是指接收并处理本地网段主机发送的报文并转发到目的网段的设备。位于不同网络间的主机要实现通讯，必须把数据包发送给网关。网关通常就是一台三层网络设备（路由器、三层交换机、防火墙或服务器），网关地址就是设备的接口地址。

17. 数据的转发过程包括那几步?

（1）TCP 封装
（2）IP 封装
（3）查找路由
（4）ARP
（5）以太网封装
（6）数据帧转发
（7）数据包转发

18. 简述下单播，广播和组播。

（1）单播是主机之间一对一的通讯模式，网络中的交换机和路由器对数据只进行转发不进行复制。

（2）广播主机之间一对所有的通讯模式，网络对其中每一台主机发出的信号都进行无条件复制并转发，所有主机都可以接收到所有信息。

（3）组播是主机之间一对一组的通讯模式，也就是加入了同一个组的主机可以接受到此组内的所有数据，网络中的交换机和路由器只向有需求者复制并转发其所需数据。

19. 简述下单播，广播和组播。

（1）单播是主机之间一对一的通讯模式，网络中的交换机和路由器对数据只进行转发不进行复制。

（2）广播主机之间一对所有的通讯模式，网络对其中每一台主机发出的信号都进行无条件复制并转发，所有主机都可以接收到所有信息。

（3）组播是主机之间一对一组的通讯模式，也就是加入了同一个组的主机可以接受到此组内的所有数据，网络中的交换机和路由器只向有需求者复制并转发其所需数据。

20. VRP 命令行包括哪几个视图。

（1）用户视图，查看运行状态或其他参数。

（2）系统视图，配置系统的参数。

（3）接口视图，配置接口参数。

（4）协议视图，配置路由协议。

21. 介绍下路由的分类。

直连路由、静态路由和动态路由

22. 介绍路由器的工作内容。

（1）收到数据包查看目标 IP 地址。

（2）在路由表中选择最佳路径。

（3）维护路由表。

23. 介绍路由的最长匹配原则。

路由器在转发数据时，需要选择路由表中的最优路由。当数据报文到达路由器时，路由器首先提取出报文的目的 IP 地址，然后查找路由表，将报文的目的 IP 地址与路由表中某表项的掩码字段做“与”操作，“与”操作后的结果跟路由表该表项的目的 IP 地址比较，相同则匹配上，否则就没有匹配上。当与所有的路由表项都进行匹配后，路由器会选择一个掩码最长的匹配项。

24. 介绍 RIP 路由的概念及其特点。

RIP（Routing Informatioin Protocol，路由信息协议）是一种内部网关协议，也是一种基于距离矢量算法的协议，使用跳数作为度量来衡量到达目的网络的距离。在带宽、配置和管理方面要求较低，主要应用于规模较小的网络中。

RIP 的特点介绍如下：

（1）距离矢量路由协议，属于 IGP 协议。

（2）适用于中小型网络，有 RIPv1 和 RIPv2 两个版本。

（3）基于 UDP，目标端口号为 520。

（4）周期性更新。

（5）支持水平分割、毒性逆转和触发更新等放环特性。

25. 介绍 RIPv1 和 RIPv2 的区别。

RIPv1	RIPv2
RIPv1 是有类别路由协议，不支持 VLSM 和 CIDR	RIPv2 是无类别路由协议，支持 VLSM、支持路由聚合和 CIDR
以广播的形式发送报文	支持以广播或者组播方式发送报文
不支持认证	支持明文认证和 MD5 密文认证

26. RIP 的防环机制一般包括几种机制?

（1）水平分割。

（2）毒性反转。

（3）触发更新。

27. 介绍下 OSPF 的工作流程。

（1）发现并建立邻居。

（2）传播 LSA（区别于距离矢量的路由表更新）

（3）将 LSA 泛洪到区域中的所有 OSPF 路由器，而不仅是直连的路由器。

（4）手机 LSA 创建 LSDB。

（5）使用 SPF 算法计算到每个目标网络的最短距离，并将其置于路由表中。

28. 在整个 OSPF 过程中会建立哪几张表?

邻居表（记录所有邻居关系）、链路状态数据库（记录所有链路状态信息）和路由表（记录最佳路由信息）。

29. 介绍下 OSPF 的 5 种报文类型。

（1）Hello 报文：最常用的一种报文，用于发现、维护邻居关系。并在广播和 NBMA（None-Broadcast Multi-Access）类型的网络中选举指定路由器 DR（Designated Router）和备份指定路由器 BDR（Backup Designated Router）。

（2）DD（Database Description）报文：两台路由器进行 LSDB 数据库同步时，用 DD 报文来描述自己的 LSDB。DD 报文的内容包括 LSDB 中每一条 LSA 的头部（LSA 的头部可以唯一标识一条

LSA）。LSA 头部只占一条 LSA 的整个数据量的一小部分，所以，这样就可以减少路由器之间的协议报文流量。

（3）LSR（Link State Request）报文：两台路由器互相交换过DD报文之后，知道对端的路由器有哪些LSA是本地LSDB所缺少的，这时需要发送LSR报文向对方请求缺少的LSA，LSR只包含了所需要的LSA的摘要信息。

（4）LSU（Link State Update）报文：用来向对端路由器发送所需要的 LSA。

（5）LSACK（Link State Acknowledgment）报文：用来对接收到的 LSU 报文进行确认。

30. 介绍下 OSPF 邻居建立必须满足的条件。

（1）RID 唯一。

（2）Hello/Dead 时间间隔一致。

（3）区域 ID 一致。

（4）认证一致。

（5）链路 MTU 大小一致（华为默认不开启检查，思科默认开启）。

（6）子网掩码一致。

（7）网络地址一致。

（8）末梢区域设置一致。

31. 介绍下工作中常用的 OSPF 配置命令。

命令	备注
ospf xx router-id xx.xx.xx.xx	开启 OSPF，进程号默认为 1，手动配置 Router ID
area XX	配置区域
network xx.xx.xx.xx xx.xx.xx.xx	宣告网络，即指定运行 OSPF 的接口；使用反掩码来匹配（255.255.255.255-掩码）
display ospf peer [brief]	显示 OSPF 邻居信息
ospf timer hello xx	修改 Hello 包发送间隔
ospf timer dead xx	修改 Hello 包超时时间
display ospf interface XXXX	显示 ospf 接口信息
ospf dr-priority XX	修改 OSPF 接口优先级
ospf cost 10	修改开销值，范围为 1~655335，缺省值为 1
bandwidth-reference XX	调整带宽参考值，默认为 100Mbps
reset ospf process	重启 OSPF 进程

32. 介绍下以太网接口的双工模式。

（1）半双工：通信双方都能发送和接收数据，但不能同时进行，并存在最大传输距离的限制。

（2）全双工：通信双方都能接收和发送数据，最大吞吐量可达到双倍速率，且消除了半双工的物理距离限制。

33. 介绍下 VLAN 的概念及其作用。

VLAN（虚拟局域网），是将一个物理的局域网在逻辑上划分多个广播域的技术。通过在交换机上配置 VLAN，可以实现在同一个 VLAN 内的用户可以进行二层访问，而不同 VLAN 间的用户被二层隔离。这样既能够隔离广播域，又能够提升网络的安全性。

34. 介绍下 VLAN 链路的两种类型。

VLAN 链路分为两种类型：Access 链路和 Trunk 链路。

（1）接入链路（Access Link）：连接用户主机和交换机的链路称为接入链路。接入链路上传输的帧都是无标记帧。

（2）干道链路（Trunk Link）：连接交换机和交换机或路由器的链路称为干道链路。干道链路上通过的帧几乎都是有标记帧用于两端识别。

35. 介绍下实现 VLAN 间路由的方式。

（1）解决 VLAN 间通信问题可以采用单臂路由的方法，即在交换机和路由器之间仅使用一条物理链路连接。在交换机上，把连接到路由器的端口配置成 Trunk 类型的端口，并允许相关 VLAN 的帧通过。在路由器上需要创建子接口，逻辑上把连接路由器的物理链路分成了多条。

（2）解决 VLAN 间通信问题可以在三层交换机上配置 VLANIF 接口来实现 VLAN 间路由。如果网络上有多个 VLAN，则需要给每个 VLAN 配置一个 VLANIF 接口，并给每个 VLANIF 接口配置一个 IP 地址。用户设置的缺省网关就是三层交换机中 VLANIF 接口的 IP 地址。

36. 介绍下 STP 的主要作用。

（1）消除环路：通过阻断冗余链路来消除网络中可能存在的环路。

（2）链路备份：当活动路径发生故障时，激活备份链路，及时恢复网络连通性。

37. 介绍下运行 STP 协议的设备上端口的 5 种状态.

（1）Forwarding：转发状态。端口既可转发用户流量也可转发 BPDU 报文，只有根端口或指定端口才能进入 Forwarding 状态。

（2）Learning：学习状态。端口可根据收到的用户流量构建 MAC 地址表，但不转发用户流量。增加 Learning 状态是为了防止临时环路。

（3）Listening：侦听状态。端口可以转发 BPDU 报文，但不能转发用户流量。

（4）Blocking：阻塞状态。端口仅仅能接收并处理 BPDU，不能转发 BPDU，也不能转发用户流量。此状态是预备端口的最终状态。

（5）Disabled：禁用状态。端口既不处理和转发 BPDU 报文，也不转发用户流量。

38. 介绍下 DHCP 各个报文类型的工作流程.

（1）DHCP 客户端初次接入网络时，会发送 DHCP 发现报文（DHCP Discover），用于查找和定位 DHCP 服务器。

（2）DHCP 服务器在收到 DHCP 发现报文后，发送 DHCP 提供报文（DHCP Offer），此报文中包含 IP 地址等配置信息。

（3）在 DHCP 客户端收到服务器发送的 DHCP 提供报文后，会发送 DHCP 请求报文（DHCP Request），另外在 DHCP 客户端获取 IP 地址并重启后，同样也会发送 DHCP 请求报文，用于确认分配的 IP 地址等配置信息。DHCP 客户端获取的 IP 地址租期快要到期时，也发送 DHCP 请求报文向服务器申请延长 IP 地址租期。

（4）收到 DHCP 客户端发送的 DHCP 请求报文后，DHCP 服务器会回复 DHCP 确认报文（DHCP ACK）。客户端收到 DHCP 确认报文后，会将获取的 IP 地址等信息进行配置和使用。

（5）如果 DHCP 服务器收到 DHCP REQUEST 报文后，没有找到相应的租约记录，则发送 DHCP NAK 报文作为应答，告知 DHCP 客户端无法分配合适 IP 地址。

（6）DHCP 客户端通过发送 DHCP 释放报文（DHCP Release）来释放 IP 地址。收到 DHCP 释放报文后，DHCP 服务器可以把该 IP 地址分配给其他 DHCP 客户端。

39. 简述下 ACL 的 3 中分类.

（1）基本 ACL 可以使用报文的源 IP 地址、分片标记和时间段信息来匹配报文，其编号取值范围是 2000-2999。

（2）高级 ACL 可以使用报文的源/目的 IP 地址、源/目的端口号以及协议类型等信息来匹配报文。高级 ACL 可以定义比基本 ACL 更准确、更丰富、更灵活的规则，其编号取值范围是 3000-3999。

（3）二层 ACL 可以使用源/目的 MAC 地址以及二层协议类型等二层信息来匹配报文，其编号取值范围是 4000-4999。

40. 简述下 NAT 的概念及其作用.

NAT（网络地址转换）主要用于实现位于内部网络的主机访问外部网络的功能。当局域网内的主机需要访问外部网络时，通过 NAT 技术可以将其私网地址转为公网地址，并且多个私网用户可以共用一个公网地址，这样既可以保证网络互通，又节省了公网地址。

41. 常用的传输介质有哪些.

（1）同轴电缆。

（2）双绞线。

（3）光纤。

42. 综合布线有哪些优点.

（1）结构清晰，便于管理维护。

（2）便于扩展，节约成本。

（3）灵活性强，适应各种需求。

43. 综合布线系统划分为哪几个子系统?

工作区子系统、水平子系统、垂直子系统、设备间子系统、管理子系统和建筑群子系统。

10.常用技术面试题（软件测试）

10.1 常规及职业选择问题

1. 你的测试职业发展是什么?

测试经验越多，测试能力越高。所以我的职业发展是需要时间积累的，一步步向着高级测试工程师奔去。而且我也有初步的职业规划，前 3 年积累测试经验，按如何做好测试工程师的要点去要求自己，不断更新自己改正自己，做好测试任务。

2. 你认为测试人员需要具备哪些素质

做测试应该要有一定的协调能力，因为测试人员经常要与开发接触处理一些问题，如果处理不好的话会引起 2、些冲突，这样的话工作上就会不好做。还有测试人员要有一定的耐心，有的时候做测试很枯燥乏味。除了耐心，测试人员不能放过每一个可能的错误。

3. 你为什么能够做测试这一行

虽然我的测试技术还不是很成熟，但是我觉得我还是可以胜任软件测试这个工作的，因为做软件测试不仅是要求技术好，还有有一定的沟通能力，耐心、细心等外在因素。综合起来看我认为我是胜任这个工作的。

4. 测试的目的是什么?

测试的目的是找出软件产品中的错误，是软件尽可能的符合用户的要求。当然软件测试是不可能找出全部错误的。

5. 测试分为哪几个阶段?

一般来说分为 5 个阶段：单元测试、集成测试、确认测试、系统测试、验收测试

6. 单元测试的测试对象、目的、测试依据、测试方法?

测试对象是模块内部的程序错误，目的是消除局部模块逻辑和功能上的错误和缺陷。测试依据是模块的详细设计，测试方法是采用白盒测试。

7. 怎样看待加班问题

加班的话我没有太多意见，但是我还是觉得如果能够合理安排时间的话，不会有太多时候加班的。

8. 结合你以前的学习和工作经验，你认为如何做好测试。

根据我以前的工作和学习经验，我认为做好工作首先要有一个良好的沟通，只有沟通无障碍了，才会有好的协作，才会有更好的效率，再一个就是技术一定要过关，做测试要有足够的耐心，和一个良好的工作习惯，不懂的就要问，实时与同事沟通这样的话才能做好测试工作。

9. 你为什么选择软件测试行业

因为之前了解软件测试这个行业，觉得他的发展前景很好。

10. 根据你以前的工作或学习经验描述一下软件开发、测试过程，由哪些角色负责，你做什么

要有架构师、开发经理、测试经理、程序员、测试员。我在里面主要是负责所分到的模块执行测试用例。

11. 根据你的经验说说你对软件测试/质量保证的理解

软件质量保证与测试是根据软件开发阶段的规格说明和程序的内部结构而精心设计的一批测试用例(即输入数据和预期的输出结果)，并根据这些测试用例去运行程序，以发现错误的过程。它是对应用程序的各个方面进行测试以检查其功能、语言有效性及其外观排布。

12. 软件测试的流程是什么?

需求调查：全面了解系统概况、应用领域、软件开发周期、软件开发环境、开发组织、时间安排、功能需求、性能需求、质量需求及测试要求等。根据系统概况进行项目所需的人员、时间和工作量估计以及项目报价。

制定初步的项目计划。

测试准备：组织测试团队、培训、建立测试和管理环境等。

测试设计：按照测试要求进行每个测试项的测试设计，包括测试用例的设计和测试脚本的开发等。

测试实施：按照测试计划实施测试。

测试评估：根据测试的结果，出具测试评估报告。

13. 你对 SQA 的职责和工作活动(如软件度量)的理解?

SQA 就是独立于软件开发的项目组，通过对软件开发过程的监控，来保证软件的开发流程按照指定的 CMM 规程(如果有相应的 CMM 规程),对于不符合项及时提出建议和改进方案，必要时可以向高层经理汇报以求问题的解决。通过这样的途径来预防缺陷的引入，从而减少后期软件的维护成本。SQA 主要的工作活动包括制定 SQA 工作计划，参与阶段产物的评审，进行过程质量、功能配置及物理配置的审计等;对项目开发过程中产生的数据进行度量等等。

14. 说说你对软件配置管理的理解

项目在开发过程中要用相应的配置管理工具对配置项(包括各个阶段的产物)进行变更控制，配置管理的使用取决于项目规模和复杂性及风险的水平。软件的规模越大，配置管理就越显得重要。还有在配置管理中，有一个很重要的概念，那就是基线，是在一定阶段各个配置项的组合，一个基线就提供了一个正式的标准，随后的工作便基于此标准，并只有经过授权后才能变更这个标准。配置管理工具主要有 CC，VSS,CVS,SVN 等，我只用过 SVN，对其他的工具不是很熟悉。

15. 怎样写测试计划和测试用例

简单点，测试计划里应有详细的测试策略和测试方法，合理详尽的资源安排等，至于测试用例，那是依赖于需求(包括功能与非功能需求)是否细化到功能点，是否可测试等。

16. 你在测试中发现了一个 bug ，但是开发经理认为这不是一个 bug ，你应该怎样解决。

首先，将问题提交到缺陷管理库里面进行备案。然后，要获取判断的依据和标准：根据需求说明书、产品说明、设计文档等，确认实际结果是否与计划有不一致的地方，提供缺陷是否确认的直接依据；如果没有文档依据，可以根据类似软件的一般特性来说明是否存在不一致的地方，来确认是否是缺陷；根据用户的一般使用习惯，来确认是否是缺陷；与设计人员、开发人员和客户代表等相关人员探讨，确认是否是缺陷；合理的论述，向测试经理说明自己的判断的理由，注意客观、严谨，不参杂个人情绪。等待测试经理做出最终决定，如果仍然存在争议，可以通过公司政策所提供的渠道，向上级反映，并有上级做出决定。

17. 给你一个网站，你如何测试?

首先，查找需求说明、网站设计等相关文档，分析测试需求。制定测试计划，确定测试范围和测试策略，一般包括以下几个部分：功能性测试；界面测试；性能测试；数据库测试；安

全性测试；兼容性测试

设计测试用例,链接测试、提交功能的测试和界面测试、性能测试。

数据库测试：要具体决定是否需要开展。数据库一般需要考虑连结性，对数据的存取操作，数据内容的验证等方面。

安全性测试：

1 基本的登录功能的检查

2 是否存在溢出错误，导致系统崩溃或者权限泄露

3 相关开发语言的常见安全性问题检查，例如 SQL 注入等。

4 如果需要高级的安全性测试，确定获得专业安全公司的帮助，外包测试，或者获取支持兼容性测试，根据需求说明的内容，确定支持的平台组合：浏览器的兼容性；操作系统的兼容性；软件平台的兼容性；数据库的兼容性开展测试，并记录缺陷。

18. 软件生存周期及其模型是什么?

软件生存周期是软件开发全部过程、活动和任务的结构框架，是从可行性研究到需求分析、软件设计、编码、测试、软件发布维护的过程。在经历需求、分析、设计、实现、部署后，软件将被使用并进入维护阶段，直到最后由于缺少维护费用而逐渐消亡。这样的一个过程，称为"生命周期模型"（Life Cycle Model）。

19. 什么是软件测试？软件测试的目的与原则

使用人工或自动手段，来运行或测试某个系统的过程。其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。

软件测试的目的：

测试是程序的执行过程，目的在于发现错误;

软件测试的原则：

1、软件测试应尽早执行，并贯穿于整个软件生命周期

2、软件测试应追溯需求

3、必须确定预期输出（或结果）

4、必须彻底检查每个测试结果

5、严格执行测试计划，排除测试的随意性

6、注意合法合理的输入，也要注意非法的非预期的输入

7、检查程序是否做了不该做的

8、测试应从“小规模”开始，逐步转向“大规模”

9、反复使用同样的测试会使软件具有抵抗力

10、关注缺陷的修复

20. 软件配置管理的作用？软件配置包括什么？

软件配置管理作为软件开发过程的必要环节和软件开发管理的基础，贯穿整个软件生命周期，同时对软件开发过程的宏观管理即项目管理也有重要的支持作用。一个软件开发组织真正有效的实施软件配置管理，将会使软件开发过程有更好的可预测性，使系统具有可重复性，大大提高软件组织的竞争力。

软件配置包括如下内容：配置项识别、工作空间管理、版本控制、变更控制、状态报告和配置审计

21. 什么是软件质量?

软件质量：软件产品的特性可以满足用户的功能、性能需求的能力。

22. 目前主要的测试用例设计方法是什么?

1、白盒测试：

逻辑覆盖、循环覆盖、基本路径覆盖

2、黑盒测试：

边界值分析法、等价类划分、错误猜测法、因果图法、状态图法、测试大纲法、随机测试和场景法

23. 软件的安全性应从哪几个方面 去测试?

软件安全性测试包括程序、数据库安全性测试。

根据系统安全指标不同测试策略也不同。用户认证安全的测试要考虑问题：

1、明确区分系统中不同用户权限，系统中会不会出现用户冲突，系统会不会因用户的权限的改变造成混乱；

2、用户登陆密码是否是可见、可复制，是否可以通过绝对途径登陆系统（拷贝用户登陆后的链接直接进入系统）用户退出系统后是否删除了所有鉴权标记，是否可以使用后退键而不通过输入口令进入系统

3、系统网络安全的测试要考虑问题，测试采取的防护措施是否正确装配好，有关系统

的补丁是否打上，模拟非授权攻击，看防护系统是否坚固

4、采用成熟的网络漏洞检查工具检查系统相关漏洞（即用最专业的黑客攻击工具攻击试一下，现在最常用的是 NBSI 系列和 IPhacker IP ）

5、采用各种木马检查工具检查系统木马情况，采用各种防外挂工具检查系统各组程序的外挂漏洞

6、数据库安全考虑问题：

系统数据是否机密（比如对银行系统，这一点就特别重要，一般的网站就没有太高要求）

系统数据的完整性

系统数据可管理性

系统数据的独立性

系统数据可备份和恢复能力（数据备份是否完整，可否恢复，恢复是否可以完整）

24. 什么是测试用例 什么是测试脚本 两者的关系是什么?

为实施测试而向被测试系统提供的输入数据、操作或各种环境设置以及期望结果的一个特定的集合。测试脚本是为了进行自动化测试而编写的脚本。测试脚本的编写必须对应相应的测试用例

25. 简述什么是静态测试、动态测试、黑盒测试、白盒测试、α测试 β测试

静态测试：不运行程序本身而寻找程序代码中可能存在的错误或评估程序代码的过程。

动态测试是实际运行被测程序，输入相应的测试实例，检查运行结果与预期结果的差异，判

定执行结果是否符合要求，从而检验程序的正确性、可靠性和有效性，并分析系统运行效率

和健壮性等性能。

黑盒测试：一般用来确认软件功能的正确性和可操作性,目的是检测软件的各个功能是否能得

以实现,把被测试的程序当作一个黑盒,不考虑其内部结构,在知道该程序的输入和输出之间的关系或程序功能的情况下,依靠软件规格说明书来确定测试用例和推断测试结果的正确性。

白盒测试：根据软件内部的逻辑结构分析来进行测试,是基于代码的测试，测试人员通过阅读

程序代码或者通过使用开发工具中的单步调试来判断软件的质量，一般黑盒测试由项目经理

在程序员开发中来实现。

α测试：是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环

境下进行的受控测试，Alpha 测试不能由程序员或测试员完成。

β测试是：软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在

测试现场，Beta 测试不能由程序员或测试员完成。

26. 软件产品质量特性是什么？？

功能性：适应性、准确性、互操作性、依从性、安全性。

可靠性：成熟性、容错性、以恢复性。

可使用性：易理解性、易学习性、易操作性。

效率：时间特性、资源特性。

可维护性：易分析性、易变更性、稳定性、易测试性。

可移植性： 适应性、易安装性、遵循性、易替换性。

27. 软件测试的策略是什么？测试的策略有哪些?

软件测试策略：在一定的软件测试标准、测试规范的指导下，依据测试项目的特定环境约束

而规定的软件测试的原则、方式、方法的集合。

测试策略有黑盒/白盒，静态/动态，手工/自动，冒烟测试，回归测试，公测（Beta 测试的策略）。

28. 软件测试分为几个阶段

软件测试按阶段划分可以分为单元测试、集成测试、系统测试和验收测试 4 个阶段

29. 测试人员在软件开发过程中的任务是什么?

1、寻找 Bug；

2、避免软件开发过程中的缺陷；

3、衡量软件的品质；

4、关注用户的需求。

总的目标是：确保软件的质量。

30. 黑盒测试和白盒测试是软件测试的两种基本方法，请分别说明各自的优点和缺点

黑盒测试的优点有：

1、比较简单，不需要了解程序内部的代码及实现；与软件的内部实现无关；

2、从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题；

3、基于软件开发文档，所以也能知道软件实现了文档中的哪些功能；

4、在做软件自动化测试时较为方便。

黑盒测试的缺点有：

1、不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的 30%；

2、自动化测试的复用性较低。

白盒测试的优点有：

1、帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐藏的问题

白盒测试的缺点有：

1、程序运行会有很多不同的路径，不可能测试所有的运行路径；

2、测试基于代码，只能测试开发人员做的对不对，而不能知道设计的正确与否，可能会漏掉一

些功能需求；

3、系统庞大时，测试开销会非常大。

31. 如何测试一个纸杯?

1、功能度：用水杯装水看漏不漏；水能不能被喝到；

2、安全性：杯子有没有毒或细菌；

3、可靠性：杯子从不同高度落下的损坏程度；

4、可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用；

5、兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等；

6、易用性：杯子是否烫手、是否有防滑措施、是否方便饮用；

7、用户文档：使用手册是否对杯子的用法、限制、使用条件等有详细描述；

8、疲劳测试：将杯子盛上水（案例一）放 24 小时检查泄漏时间和情况；盛上汽油（案例二）放 24 小时检查泄漏时间和情况等

9、压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

32. 测试计划工作的目的是什么？测试计划文档的内容应该包括什么？其中哪些是最重要的?

答案：软件测试计划是指导测试过程的纲领性文件。

包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试

交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，

可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对

测试过程中的各种变更。

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划

测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

所以其中最重要的是测试测试策略和测试方法（最好是能先评审）。

33. 黑盒测试的测试用例常见设计方法都有哪些?

1） 等价类划分

划分等价类: 等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的.并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况:有效等价类和无效等价类.

2） 边界值分析法

边界值分析方法是对等价类划分方法的补充。测试工作经验告诉我,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以查出更多的错误.使用边界值分析方法设计测试用例,首先应确定边界情况.通常输入和输出等价类的边界,就是应着重测试的边界情况.应当选取正好等于,刚刚大于或刚刚小于边界的值

作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据.

3）错误猜测法

基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法

4）因果图方法

前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系, 相互组合等. 考虑输入条件之间的相互组合,可能会产生一些新的情况. 但要检查输入条件的组合不是一件容易的事情, 即使把所有输入条件划分成等价类,他们之间的组合情况也相当多. 因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例. 这就需要利用因果图（逻辑模型）. 因果图方法最终生成的就是判定表. 它适合于检查程序输入条件的各种组合情况.

5）正交表分析法

有时候，可能因为大量的参数的组合而引起测试用例数量上的激增，同时，这些测试用例并

没有明显的优先级上的差距，而测试人员又无法完成这么多数量的测试，就可以通过正交表

来进行缩减一些用例，从而达到尽量少的用例覆盖尽量大的范围的可能性。

6）场景分析方法

指根据用户场景来模拟用户的操作步骤，这个比较类似因果图，但是可能执行的深度和可行

性更好。

7）状态图法

通过输入条件和系统需求说明得到被测系统的所有状态，通过输入条件和状态得出输出条件；通过输入条件、输出条件和状态得出被测系统的测试用例。

34. 详细的描述一个测试活动完整的过程。

1、 项目经理通过和客户的交流，完成需求文档

2、 开发人员和测试人员共同完成需求文档的评审，

3、 项目经理通过综合开发人员，测试人员以及客户的意见，完成项目计划。

4、 SQA 进入项目，开始进行统计和跟踪开发人员根据需求文档完成需求分析文档，测试人员进行评审，评审的主要内容包括是否有遗漏或者双方理解不同的地方。测试人员完成测试计划文档，测试计划包括的内容上面有描述。

5、 写测试用例，同时开发人员完成概要设计文档，详细设计文档。此两份文档成为测

试人员撰写测试用例的补充材料。

6、 测试用例完成后，测试和开发需要进行评审。

7、 测试人员搭建环境,开发人员提交第一个版本，可能存在未完成功能，需要说明。测试人员进行测试，发现 BUG 后提交

8、 开发提交第二个版本,测试人员进行测试。重复上面的工作，一般是 3-4 个版本后 BUG 数量减少，达到出货的要求。如果有客户反馈的问题，需要测试人员协助重现并重新测试。

9、

35. 软件验收测试包括 ___ 、 ___ 、 ____ 三种类型。

软件验收测试包括正式验收测试、alpha 测试、beta 测试三种测试。

36. 系统测试的策略有 _____等 15 种方法。 (该题 15 个空)

性能测试、负载测试、强度测试、易用性测试、安全测试、配置测试、安装测试、文档测试、故障恢复测试、用户界面测试、恢复测试、分布测试、可用性测试。

37. 设计系统测试计划需要参考的项目文档有 ___ 、 ___ 和 ____ 。

软件测试计划、软件需求工件、和迭代计划。

38. 软件测试项目从什么时候开始？为什么？

软件测试应该在需求分析阶段就介入,因为测试的对象不仅仅是程序编码,应该对软件开发过程中产生的所有产品都测试,并且软件缺陷存在放大趋势.缺陷发现的越晚,修复它所花费的成本就越大.

39. 什么是回归测试？？

回归测试: (regression testing): 回归测试有两类：用例回归和错误回归；

1、 用例回归是过一段时间以后再回头对以前使用过的用例在重新进行测试,看看会重新发现问题。

2、 错误回归，就是在新版本中，对以前版本中出现并修复的缺陷进行再次验证，并以缺陷为核心，对相关修改的部分进行测试的方法。

40. 单元测试、集成测试、系统测试的侧重点是什么？

1、单元测试：针对的是软件设计的最小单元--程序模块（面向过程中是函数、过程；面向

对象中是类）,进行正确性检验的测试工作,在于发现每个程序模块内部可能存在的差错.一般有两个步骤:人工静态检查\动态执行跟踪

2、集成测试：针对的是通过了单元测试的各个模块所集成起来的组件进行检验,其主要内容是各个单元模块之间的接口,以及各个模块集成后所实现的功能.

3、系统测试：针对的是集成好的软件系统，作为整个计算机系统的一个元素,与计算机硬件\外设\某些支持软件\数据和人员等其他系统元素结合在一起,要在实际的运行环境中,对计算机系统进行一系列的集成测试和确认测试.

41. 设计用例的方法有哪些?

1、 白盒测试用例设计有如下方法：逻辑覆盖、循环覆盖和基本路径覆盖。

2、 黑盒测试用例设计方法：等价类划分、边界值分析、错误猜测、因果图、状态图、测试大纲、场景法、正交策略表。

42. 一个测试工程师应具备那些素质?

1、责任心

2、沟通能力

3、团队合作精神

4、耐心、细心、信心

5、时时保持怀疑态度，并且有缺陷预防的意识

6、具备一定的编程经验

43. 集成测试通常都有那些策略?

基于分解的集成:大爆炸集成\自顶向下集成\自底向上集成\ 三明治集成\基于调用图的集成\基于路径的集成\分层集成\基于功能的集成\高频集成\基于进度的集成\基于风险集成\基于事件集成\基于使用的集成\C/S 集成

44. 你所了解的的软件测试类型都有哪些，简单介绍一下。

1、 按测试 策略分类：1、静态与动态测试 2、黑盒与白盒测试 3、手工和自动测试 4、冒烟测试 5、回归测试；

2、 按测试阶段分类：单元测试、集成测试、系统测试；

3、 其他常见测试方法：1、功能测试 2、性能测试 3、压力测试 4、负载测试 5、易用性测试 6、安装测试 7、界面测试 8、配置测试 9、文档测试 10、兼容性测试 11、安全性测试 12、恢复测试

45. 分别概述创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

46. 您认为做好测试用例设计工作的关键是什么?

白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

47. 性能测试工作的目的是什么？做好性能测试工作的关键是什么?

性能测试的目的主要是发现在并发多用户和大数据量操作时是否会出现与需求有差异的地方。性能测试工作的关键是做好系统分析和功能分析，确定系统瓶颈所在

48. 你的测试职业发展目标是什么?

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔去。而且我也有初步的职业规划，前 3 年累积测试经验，不断的更新自己改正自己，做好测试任务。

49. 测试结束的标准是什么?

从微观上来说，在测试计划中定义，比如系统在一定性能下平稳运行 72 小时，目前错误跟踪系统中，本版本中没有一般严重的 BUG，普通 BUG 的数量在 3 以下，BUG 修复率 90%以上等等参数，然后由开发经理，测试经理，项目经理共同签字认同版本 Release。如果说宏观的，则是当这个软件彻底的消失以后，测试就结束了。

50. 软件测试分为黑盒和白盒，分别适合什么情况?

1、 白盒测试又称为结构测试、逻辑驱动测试或基于程序本身的测试，它着重于程序的内部结构及算法，通常不关心功能与性能指标；

2、 黑盒测试又被称为功能测试、数据驱动测试或基于规格说明的测试，它实际上是站在最终用户的立场，检验输入输出信息及系统性能指标是否符合规格说明书中有关功能需求及性能需求的规定。

51. 一套完整的测试应该由哪些阶段组成?

可行性分析、需求分析、概要设计、详细设计、编码、单元测试、集成测试、系统测试、验收测试

52. 软件测试用例通常包括那些内容?

软件测试用例的基本要素包括测试用例编号、测试标题、重要级别、测试输入、操作步骤、预期结果。

53. 你的测试职业发展目标是什么?

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔去。而且我也有初步的职业规划，前 3 年累积测试经验，按如何做好测试工程师的要求自己，不断的更新自己改正自己，做好测试任务。

54. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

1、 黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

1、是否有不正确或遗漏的功能？

2、在接口上，输入是否能正确的接受？能否输出正确的结果？

3、是否有数据结构错误或外部信息（例如数据文件）访问错误？

4、性能上是否能够满足要求？

5、是否有初始化或终止性错误？

2、 白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查：

1、对程序模块的所有独立的执行路径至少测试一遍。

2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性，等等。

3、 单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。

4、 集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。

5、 系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。（常见的联调测试）系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

6、 验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

55. 当开发人员说不是 bug 时，你如何应付?

开发人员说不是 bug，有 2 种情况:

一是需求没有确定，所以我可以这么做，这个时候可以找来产品经理进行确认，需不需要改动，3 方商量确定好后再看要不要改。

二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是 BUG 的依据是什么,如果被用户发现或出了问题，会有什么不良结果,程序员可能会给你很多理由，你可以对他的解释进行反驳。

56. 为什么要在一个团队中开展软件测试工作?

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

57. 如果有机会转成开发人员，你会去做开发工作吗?

如果公司确实需要我可以从事开发，但我还是喜欢做测试，我认为我更适合做测试。

58. 软件测试分哪些阶段? 各阶段的含义?

分为单元测试、集成测试、确认测试、系统测试、验收测试。单元测试是最小单位的测试，测试独立模块；集成测试主要测试模块之间的接口是否正常，确认测试类似于冒烟测试通常在大规模系统测试之前验证版本主要功能是否实现，版本的稳定性是否可以进入系统测试，系统测试是全面测试验证系统是否满足用户需求包括功能、性能、兼容性等等。验收测试是用户参与的测试。

59. 一份测试计划应该包括哪些内容?

背景、项目简介、目的、测试范围、测试策略、人员分工、资源要求、进度计划、参考文档、常用术语、提交文档、风险分析。

60. 针对于软件的行业背景，你如何理解软件的业务?

阅读用户手册了解软件的功能和操作流程；看一些业务的专业书籍补充业务知识；如果有用户实际的数据，可以拿实际的数据进行参考；参考以前的用例和 BUG 报告；在使用软件的过程中多思考；多与产品经理交流。

61. 测试用例应包括哪些内容?

编号、模块名称、编写人、日期、操作说明、输入数据、预期结果等。

62. 什么是兼容性测试? 兼容性测试侧重哪些方面。

- 兼容测试主要是检查软件在不同的硬件平台、软件平台上是否可以正常的运行，即是通常说的软件的可移植性。
- 兼容的类型，如果细分的话，有平台的兼容，网络兼容，数据库兼容，以及数据格式的兼容。

- 兼容测试的重点是对兼容环境的分析。通常，是在运行软件的环境不是很确定的情况下，才需要做兼容。根据软件运行的需要，或者根据需求文档，一般都能够得出用户会在什么环境下使用该软件，把这些环境整理成表单，就得出做兼容测试的兼容环境了。
- 兼容和配置测试的区别在于，做配置测试通常不是 Clean OS 下做测试，而兼容测试多是在 Clean OS 的环境下做的。。

63. 对某软件进行测试，发现在 WIN98 上运行得很慢，怎么判别是该软件存在问题还是其软硬件运行环境存在问题?

看软件的运行环境要求。如果符合要求则是程序存在问题，若不符合要求则是硬件系统存在问题

64. 我现在有个程序，发现在 Windows 上运行得很慢，怎么判别是程序存在问题还是软硬件系统存在问题?

1、检查系统是否有中毒的特征；

2、检查软件/硬件的配置是否符合软件的推荐标准；

3、确认当前的系统是否是独立，即没有对外提供什么消耗 CPU 资源的服务；

4、如果是 C/S 或者 B/S 结构的软件，需要检查是不是因为与服务器的连接有问题，或者访问有问题造成的；

5、在系统没有任何负载的情况下，查看性能监视器，确认应用程序对 CPU/内存的访问情况。

65. 正交表测试用例设计方法的特点是什么?

用最少的实验覆盖最多的操作，测试用例设计很少，效率高，但是很复杂；

对于基本的验证功能，以及二次集成引起的缺陷，一般都能找出来；但是更深的缺陷，更复杂的缺陷，还是无能为力的；具体的环境下，正交表一般都很难做的。大多数，只在系统测试的时候使用此方法。

66. 描述使用 bugzilla 缺陷管理工具对软件缺陷（BUG）跟踪的管理的流程?

就是 Bugzilla 的状态转换图。

67. 你觉得 bugzilla 在使用的过程中，有什么问题?

界面不稳定；

根据需要配置它的不同的部分，过程很烦琐。

流程控制上，安全性不好界定，很容易对他人的 Bug 进行误操作；

没有综合的评分指标，不好确认修复的优先级别。

68. 描述测试用例设计的完整过程?

需求分析 + 需求变更的维护工作；

根据需求 得出测试需求；

设计测试方案，评审测试方案；

方案评审通过后，设计测试用例，再对测试用例进行评审；

69. 单元测试的策略有哪些?

逻辑覆盖、循环覆盖、同行评审、桌前检查、代码走查、代码评审、景泰数据流分析

70. LoadRunner 分哪三部分?

用户动作设计；场景设计；测试数据分析；

71. LoadRunner 进行测试的流程?

1. 测试测试
2. 创建虚拟用户脚本
3. 创建运行场景
4. 运行测试脚本
5. 监视场景
6. 分析测试的结果

以上，最好是结合一个案例，根据以上流程来介绍。

72. 什么是并发？在 LoadRunner 中，如何进行并发的测试？集合点失败了会怎么样?

在同一时间点，支持多个不同的操作。

LoadRunner 中提供 IP 伪装，集合点，配合虚拟用户的设计，以及在多台电脑上设置，可以比较好的模拟真实的并发。

集合点，即是多个用户在某个时刻，某个特定的环境下同时进行虚拟用户的操作的。集合点失败，则集合点的才操作就会取消，测试就不能进行。

73. 使用 QTP 做功能测试，录制脚本的时候，要验证多个用户的登录情况/查询情况，如何操作?

分析用户登录的基本情况，得出一组数据，通过性测试/失败性测试的都有（根据 TC 来设计这些数据），然后录制登录的脚本，将关键的数据参数化，修改脚本，对代码进行加强，调试脚本。

74. QTP 中的 Action 有什么作用？有几种?

Action 的作用

- 用 Action 可以对步骤集进行分组
- 步骤重组，然后被整体调用
- 拥有自己的 sheet
- 组合有相同需求的步骤，整体操作
- 具有独立的对象仓库

Action 的种类

- 可复用 Action
- 不可复用 Action
- 外部 Action

75. TestDirector 有些什么功能，如何对软件测试过程进行管理?

需求管理

- 定义测试范围
- 定义需求树
- 描述需求树的功能点

测试计划

- 定义测试目标和测试策略。
- 分解应用程序，建立测试计划树。
- 确定每个功能点的测试方法。
- 将每个功能点连接到需求上，使测试计划覆盖全部的测试需求。
- 描述手工测试的测试步骤
- 指明需要进行自动测试的功能点

测试执行

- 定义测试集合。
- 为每个测试人员制定测试任务和测试日程安排。
- 运行自动测试。

缺陷跟踪

- 记录缺陷
- 查看新增缺陷，并确定哪些是需要修正的
- 相关技术人员修改缺陷
- 回归测试
- 分析缺陷统计图表，分析应用程序的开发质量。

76. 软件缺陷（或者叫 Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

5C 标准

77. Beta 测试与 Alpha 测试有什么区别？

Beta testing(β测试),测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场

Alpha testing (α测试),是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试

78. 软件的评审一般由哪些人参加？其目的是什么？

在正式的会议上将软件项目的成果（包括各阶段的文档、产生的代码等）提交给用户、客户或有关部门人员对软件产品进行评审和批准。其目的是找出可能影响软件产品质量、开发过程、维护工作的适用性和环境方面的设计缺陷，并采取补救措施，以及找出在性能、安全性和经济方面的可能的改进。

人员：用户、客户或有关部门开发人员，测试人员，需求分析师都可以，就看处于评审那个阶段

79. 测试活动中，如果发现需求文档不完善或者不准确，怎么处理？

测试需求分析 发现需求文档不完善或者不准确，应该立即和相关人员进行协调交流。

80. 阶段评审与项目评审有什么区别？

阶段评审 对项目各阶段评审：对阶段成果和工作

项目评审 对项目总体评审：对工作和产品

81. 阐述工作版本的定义？

构造号： BUILD

82. 什么是桩模块？什么是驱动模块？

桩模块：被测模块调用模块

驱动模块 调用被测模块

83. 什么是扇入？什么是扇出？

扇入：被调次数，扇出：调其它模块数目

84. 你认为做好测试计划工作的关键是什么?

软件测试计划就是在软件测试工作正式实施之前明确测试的对象，并且通过对资源、时间、风险、测试范围和预算等方面的综合分析和规划，保证有效的实施软件测试；

做好测试计划工作的关键 ：目的，管理，规范

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2．坚持“5W”规则，明确内容与过程

“5W”规则指的是“What（做什么）”、“Why（为什么做）”、“When（何时做）”、“Where（在哪里）”、“How（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的（Why），明确测试的范围和内容（What），确定测试的开始和结束日期（When），指出测试的方法和工具（How），给出测试文档和软件的存放位置（Where）。

3．采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

4. 分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

85. 你认为做好测试用例工作的关键是什么?

需求和设计文档的理解程度，对系统的熟悉程度

86. 简述一下缺陷的生命周期?

提交->确认->分配->修复->验证->关闭

87. 软件的安全性应从哪几个方面去测试?

1. 用户认证机制：如数据证书、智能卡、双重认证、安全电子交易协议
2. 加密机制
3. 安全防护策略：如安全日志、入侵检测、隔离防护、漏洞扫描

4. 数据备份与恢复手段：存储设备、存储优化、存储保护、存储管理

5. 防病毒系统

88. 软件配置管理工作开展的情况和认识?

软件配置管理贯穿于软件开发、测试活动的始终，覆盖了开发、测试活动的各个环节，它的重要作用之一就是要全面的管理保存各个配置项，监控各配置项的状态，并向项目经理及相关的人员报告，从而实现对软件过程的控制。

软件测试配置管理包括 4 个最基本的活动：

配置项标识

配置项控制

配置项状态报告

配置审计

软件配置管理通常借助工具来辅助，主要有 MS SourceSafe、Rational ClearCase 等

89. 你觉得软件测试通过的标准应该是什么样的?

缺陷密度值达到客户的要求

90. 引入测试管理的含义?

风险分析，进度控制、角色分配、质量控制

91. 一套完整的测试应该由哪些阶段组成?

测试计划、测试设计与开发、测试实施、测试评审与测试结论

92. 单元测试的主要内容?

模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试

93. 集成测试也叫组装测试或者联合测试，请简述集成测试的主要内容?

（1）在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；

（2）一个模块的功能是否会对另一个模块的功能产生不利的影响；

（3）各个子功能组合起来，能否达到预期要求的父功能；

（4）全局数据结构是否有问题；

（5）单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

94. 简述集成测试与系统测试关系?

（1）集成测试的主要依据概要设计说明书，系统测试的主要依据是需求设计说明书；

（2）集成测试是系统模块的测试，系统测试是对整个系统的测试，包括相关的软硬件平台、网络以及相关外设的测试。

95. 软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？

用户手册

安装和设置指导

联机帮助

指南、向导

样例、示例和模板

授权/注册登记表

最终用户许可协议

96. 软件系统中除用户文档之外，文档测试还应该关注哪些文档？

开发文档

软件需求说明书

数据库设计说明书

概要设计说明书

详细设计说明书

可行性研究报告

管理文档

项目开发计划

测试计划

测试报告

开发进度月报

开发总结报告

97. 简述软件系统中用户文档的测试要点？

（1）读者群。文档面向的读者定位要明确。对于初级用户、中级用户以及高级用户应该有不同的定位

（2）术语。文档中用到的术语要适用与定位的读者群，用法一致，标准定义与业界规范相吻合。

（3）正确性。测试中需检查所有信息是否真实正确，查找由于过期产品说明书和销售人员夸大事实而导致的错误。检查所有的目录、索引和章节引用是否已更新，尝试链接是否准确，产

品支持电话、地址和邮政编码是否正确。

（4）完整性。对照软件界面检查是否有重要的分支没有描述到，甚至是否有整个大模块没有描述到。

（5）一致性。按照文档描述的操作执行后，检查软件返回的结果是否与文档描述的相同。

（6）易用性。对关键步骤以粗体或背景色给用户以提示，合理的页面布局、适量的图表都可以给用户更高的易用性。需要注意的是文档要有助于用户排除错误。不但描述正确操作，也要描述错误处理办法。文档对于用户看到的错误信息应当有更详细的文档解释。

（7）图表与界面截图。检查所有图表与界面截图是否与发行版本相同。

（8）样例与示例。像用户一样载入和使用样例。如果是一段程序，就输入数据并执行它。以每一个模块制作文件，确认它们的正确性。

（9）语言。不出现错别字，不要出现有二义性的说法。特别要注意的是屏幕截图或绘制图形中的文字。

（10）印刷与包装。检查印刷质量；手册厚度与开本是否合适；包装盒的大小是否合适；有没有零碎易丢失的小部件等等。

98. 如何理解强度测试?

强度测试是为了确定系统在最差工作环境的工作能力,也可能是用于验证在标准工作压力下的各种资源的最下限指标。

它和压力测试的目标是不同的,压力测试是在标准工作环境下,不断增加系统负荷,最终测试出该系统能力达到的最大负荷(稳定和峰值),而强度测试则是在非标准工作环境下,甚至不断人为降低系统工作环境所需要的资源,如网络带宽,系统内存,数据锁等等,以测试系统在资源不足的情况下的工作状态,通过强度测试,可以确定本系统正常工作的最差环境.

强度测试和压力测试的测试指标相近,大多都是与时间相关的指标,如并发量(吞吐量),延迟(最大\最小\平均)以及顺序指标等

强度测试需要对系统的结构熟悉,针对系统的特征设计强度测试的方法

99. 如何理解压力、负载、性能测试测试?

1. 性能测试是一个较大的范围，实际上性能测试本身包含了性能、强度、压力、负载等多方面的测试内容。

压力测试是对服务器的稳定性以及负载能力等方面的测试，是一种很平常的测试。增大访问系统的用户数量、或者几个用户进行大数据量操作都是压力测试。

2. 而负载测试是压力相对较大的测试，主要是测试系统在一种或者集中极限条件下的相应能力，是性能测试的重要部分。100 个用户对系统进行连续半个小时的访问可以看作压力测试，那么连续访问 8 个小时就可以认为负载测试，1000 个用户连续访问系统 1 个

小时也可以看作是负载测试。

实际上压力测试和负载测试没有明显的区分。测试人员应该站在关注整体性能的高度上来对系统进行测试。

100. 什么是系统瓶颈?

瓶颈主要是指整个软硬件构成的软件系统某一方面或者几个方面能力不能满足用户的特定业务要求，“特定”是指瓶颈会在某些条件下会出现，因为毕竟大多数系统在投入前。

严格的从技术角度讲，所有的系统都会有瓶颈，因为大多数系统的资源配置不是协调的，例如 CPU 使用率刚好达到 100%时，内存也正好耗尽的系统不是很多见。因此我们讨论系统瓶颈要从应用的角度讨论：关键是看系统能否满足用户需求。在用户极限使用系统的情况下，系统的响应仍然正常，我们可以认为改系统没有瓶颈或者瓶颈不会影响用户工作。

因此我们测试系统瓶颈主要是实现下面两个目的：

-发现“表面”的瓶颈。主要是模拟用户的操作，找出用户极限使用系统时的瓶颈，然后解决瓶颈，这是性能测试的基本目标。

-发现潜在的瓶颈并解决，保证系统的长期稳定性。主要是考虑用户在将来扩展系统或者业务发生变化时，系统能够适应变化。满足用户目前需求的系统不是最好的，我们设计系统的目标是在保证系统整个软件生命周期能够不断适应用户的变化，或者通过简单扩展系统就可以适应新的变化。

101. 文档测试主要包含什么内容?

在国内软件开发管理中，文档管理几乎是最弱的一项，因而在测试工作中特别容易忽略文档测试也就不足为奇了。要想给用户提供完整的产品，文档测试是必不可少的。文档测试一般注重下面几个方面：

文档的完整性：主要是测试文档内容的全面性与完整性，从总体上把握文档的质量。例如用户手册应该包括软件的所有功能模块。

描述与软件实际情况的一致性：主要测试软件文档与软件实际的一致程度。例如用户手册基本完整后，我们还要注意用户手册与实际功能描述是否一致。因为文档往往跟不上软件版本的更新速度。

易理解性：主要是检查文档对关键、重要的操作有无图文说明，文字、图表是否易于理解。对于关键、重要的操作仅仅只有文字说明肯定是不够的，应该附有图表使说明更为直观和明了。

文档中提供操作的实例：这项检查内容主要针对用户手册。对主要功能和关键操作提供的应用实例是否丰富，提供的实例描述是否详细。只有简单的图文说明，而无实例的用户手册看起来就像是软件界面的简单拷贝，对于用户来说，实际上没有什么帮助。

印刷与包装质量：主要是检查软件文档的商品化程度。有些用户手册是简单打印、装订而

成，过于粗糙，不易于用户保存。优秀的文档例如用户手册和技术白皮书，应提供商品化包装，并且印刷精美。

102. 功能测试用例需要详细到什么程度才是合格的?

这个问题也是测试工程师经常问的问题。有人主张测试用例详细到每个步骤执行什么都要写出来，目的是即使一个不了解系统的新手都可以按照测试用例来执行工作。主张这类写法的人还可以举出例子：欧美、日本等软件外包文档都是这样做的。

另外一种观点就是主张写的粗些，类似于编写测试大纲。主张这种观点的人是因为软件开发需求管理不规范，变动十分频繁，因而不能按照欧美的高标准来编写测试用例。这样的测试用例容易维护，可以让测试执行人员有更大的发挥空间。

实际上，软件测试用例的详细程度首先要以覆盖到测试点为基本要求。举个例子：“用户登陆系统”的测试用例可以不写出具体的执行数据，但是至少要写出五种以上情况（），如果只用一句话覆盖了这个功能是不合格的测试用例。覆盖功能点不是指列出功能点，而是要写出功能点的各个方面（如果组合情况较多时可以采用等价划分）。

另一个影响测试用例的就是组织的开发能力和测试对象特点。如果开发力量比较落后，编写较详细的测试用例是不现实的，因为根本没有那么大的资源投入，当然这种情况很随着团队的发展而逐渐有所改善。测试对象特点重点是指测试对象在进度、成本等方面的要求，如果进度较紧张的情况下，是根本没有时间写出高质量的测试用例的，甚至有些时候测试工作只是一种辅助工作，因而不编写测试用例。

因此，测试用例的编写要根据测试对象特点、团队的执行能力等各个方面综合起来决定编写策略。最后要注意的是测试人员一定不能抱怨，力争在不断提高测试用例编写水平的同时，不断地提高自身能力。

103. 配置和兼容性测试的区别是什么?

配置测试的目的是保证软件在其相关的硬件上能够正常运行，而兼容性测试主要是测试软件能否与不同的软件正确协作。

配置测试的核心内容就是使用各种硬件来测试软件的运行情况，一般包括：

（1）软件在不同的主机上的运行情况，例如 Dell 和 Apple；

（2)软件在不同的组件上的运行情况,例如开发的拨号程序要测试在不同厂商生产的 Modem 上的运行情况；

（3）不同的外设；

（4）不同的接口；

（5）不同的可选项，例如不同的内存大小；

兼容性测试的核心内容：

（1）测试软件是否能在不同的操作系统平台上兼容；

（2）测试软件是否能在同一操作系统平台的不同版本上兼容；

（3）软件本身能否向前或者向后兼容；

（4）测试软件能否与其它相关的软件兼容；

（5）数据兼容性测试，主要是指数据能否共享；

配置和兼容性测试通称对开发系统类软件比较重要，例如驱动程序、操作系统、数据库管理系统等。具体进行时仍然按照测试用例来执行。

104. 软件文档测试主要包含什么？

随着软件文档系统日益庞大，文档测试已经成为软件测试的重要内容。文档测试对象主要如下：

-包装文字和图形；

-市场宣传材料、广告以及其它插页；

-授权、注册登记表；

-最终用户许可协议；

-安装和设置向导；

-用户手册；

-联机帮助；

-样例、示范例子和模板；

-……

文档测试的目的是提高易用性和可靠性，降低支持费用，因为用户通过文档就可以自己解决问题。因文档测试的检查内容主要如下：

-读者对象——主要是文档的内容是否能让该级别的读者理解；

-术语——主要是检查术语是否适合读者；

-内容和主题——检查主题是否合适、是否丢失、格式是否规范等；

-图标和屏幕抓图——检查图表的准确度和精确度；

-样例和示例——是否与软件功能一致；

-拼写和语法；

-文档的关联性——是否与其它相关文档的内容一致，例如与广告信息是否一致；

文档测试是相当重要的一项测试工作，不但要给予充分的重视，更要要认真的完成，象做功能测试一样来对待文档测试。

105. 没有产品说明书和需求文档地情况下能够进行黑盒测试吗？

这个问题是国内测试工程师经常遇到的问题，根源就是国内软件开发文档管理不规范，对

变更的管理方法就更不合理了。实际上没有任何文档的时候，测试人员是能够进行黑盒测试的，这种测试方式我们可以称之为探索测试，具体做法就是测试工程师根据自己的专业技能、领域知识等不断的深入了解测试对象、理解软件功能，进而发现缺陷。

在这种做法基本上把软件当成了产品说明书，测试过程中要和开发人员不断的进行交流。尤其在作项目的时候，进度压力比较大，可以作为加急测试方案。最大的风险是不知道有些特性是否被遗漏。

106. 测试中的“杀虫剂怪事”是指什么?

“杀虫剂怪事”一词由 BorisBeizer 在其编著的《软件测试技术》第二版中提出。用于描述测试人员对同一测试对象进行的测试次数越多，发现的缺陷就会越来越少的现象。就像老用一种农药，害虫就会有免疫力，农药发挥不了效力。这种现象的根本原因就是测试人员对测试软件过于熟悉，形成思维定势。

为了克服这种现象，测试人员需要不断编写新的测试程序或者测试用例，对程序的不同部分进行测试，以发现更多的缺陷。也可以引用新人来测试软件，刚刚进来的新手往往能发现一些意想不到的问题。

107. 在配置测试中，如何判断发现的缺陷是普通问题还是特定的配置问题?

在进行配置测试时，测试工程师仍然会发现一些普通的缺陷，也就是与配置环境无关的缺陷。因此判断新发现的问题，需要在不同的配置中重新执行发现软件缺陷的步骤，如果软件缺陷不出现了，就可能是配置缺陷；如果在所有的配置中都出现，就可能是普通缺陷。

需要注意的是,配置问题可以在一大类配置中出现。例如,拨号程序可能在所有的外置 Modem 中都存在问题，而内置的 Modem 不会有任何问题。

108. 为什么尽量不要让时间有富裕的员工去做一些测试?

表面上看这体现了管理的效率和灵活性，但实际上也体现了管理者对测试的轻视。测试和测试的人有很大关系。测试工作人员应该是勤奋并富有耐心，善于学习、思考和发现问题，细心有条理，总结问题，如果具备这样的优点，做其它工作同样也会很出色，因此这里还有一个要求，就是要喜欢测试这项工作。如果他是专职的，那么肯定更有经验和信心。国内的小伙子好象都喜欢做程序员，两者工作性质不同，待遇不同，地位不同，对自我实现的价值的认识也不同，这是行业的一个需要改善的问题。如果只是为了完成任务而完成任务，或者发现了几个问题就觉得满意了，这在任何其它工作中都是不行的。

109. 完全测试程序是可能的吗?

软件测试初学者可能认为拿到软件后需要进行完全测试，找到全部的软件缺陷，使软件“零

缺陷”发布。实际上完全测试是不可能的。主要有以下一个原因：

-完全测试比较耗时，时间上不允许；

-完全测试通常意味着较多资源投入，这在现实中往往是行不通的；

-输入量太大，不能一一进行测试；

-输出结果太多，只能分类进行验证；

-软件实现途径太多；

-软件产品说明书没有客观标准，从不同的角度看，软件缺陷的标准不同；

因此测试的程度要根据实际情况确定。

110.软件测试的风险主要体现在哪里?

我们没有对软件进行完全测试，实际就是选择了风险，因为缺陷极有可能存在没有进行测试的部分。举个例子，程序员为了方便，在调试程序时会弹出一些提示信息框，而这些提示只在某种条件下会弹出，碰巧程序发布前这些代码中的一些没有被注释掉。在测试时测试工程师又没有对其进行测试。如果客户碰到它，这将是代价昂贵的缺陷，因为交付后才被客户发现。

因此，我们要尽可能的选择最合适的测试量，把风险降低到最小。

111.发现的缺陷越多，说明软件缺陷越多吗?

这是一个比较常见的现象。测试工程师在没有找到缺陷前会绞尽脑汁的思考，但是找到一个后，会接二连三的发现很多缺陷，颇有个人成就感。其中的原因主要如下：

-代码复用、拷贝代码导致程序员容易犯相同的错误。类的继承导致所有的子类会包含基类的错误，反复拷贝同一代码意味可能也复制了缺陷。

-程序员比较劳累是可以导致某些连续编写的功能缺陷较多。程序员加班是一种司空见惯的现象，因此体力不只时容易编写一些缺陷较多的程序。而这些连续潜伏缺陷恰恰时测试工程师大显身手的地方。

“缺陷一个连着一个”不是一个客观规律，只是一个常见的现象。如果软件编写的比较好，这种现象就不常见了。测试人员只要严肃认真的测试程序就可以了。

112.所有的软件缺陷都能修复吗？所有的软件缺陷都要修复吗?

从技术上讲，所有的软件缺陷都是能够修复的，但是没有必要修复所有的软件缺陷。测试人员要做的是能够正确判断什么时候不能追求软件的完美。对于整个项目团队，要做的是对每一个软件缺陷进行取舍，根据风险决定那些缺陷要修复。发生这种现象的主要原因如下：

-没有足够的时间资源。在任何一个项目中，通常情况下开发人员和测试人员都是不够用的，而且在项目中没有预算足够的回归测试时间，再加上修改缺陷可能引入新的缺陷，因此在交付期限的强大压力下，必须放弃某些缺陷的修改。

-有些缺陷只是特殊情况下出现，这种缺陷处于商业利益考虑，可以在以后升级中进行修复。

-不是缺陷的缺陷。我们经常会碰到某些功能方面的问题被当成缺陷来处理，这类问题可以以后有时间时考虑再处理。

最后要说的是，缺陷是否修改要由软件测试人员、项目经理、程序员共同讨论来决定是否修复，不同角色的人员从不同的角度来思考，以做出正确的决定。

113.软件测试人员就是 QA 吗?

软件测试人员的职责是尽可能早的找出软件缺陷，确保得以修复。而质量保证人员（QA）主要职责是创建或者制定标准和方法，提高促进软件开发能力和减少软件缺陷。测试人员的主要工作是测试，质量保证人员日常工作重要内容是检查与评审，测试工作也是测试保证人员的工作对象。

软件测试和质量是相辅相成的关系，都是为了提高软件质量而工作。

114.如何减少测试人员跳槽带来的损失?

在 IT 行业里跳槽已经是一种司空见惯的现象，而且跳槽无论给公司还是给个人都会带来一定的损失。测试队伍也无疑会面临跳槽的威胁，作为测试经理管理者，只有从日常工作中开始做起，最能最大限度的减少损失。建议我们从以下两个方面做起：

-加强部门内员工之间的互相学习，互相学习是建立学习型组织的基本要求，是知识互相转移的过程。在此基础上，可以把个人拥有的技术以知识的形式沉积下来，也就完成了隐性知识到显性知识的转化。

-通常情况下，企业能为员工提供足够大的发展空间时，如果不是待遇特别低，员工都不会主动离开企业。因此我们要想留住员工，管理者就应该把员工的个人成长和企业的发展联系起来，为员工设定合理发展规划并付诸实现。不过这项要求做起来比较，要有比较好的企业文化为依托。

115.测试产品与测试项目的区别是什么?

习惯上把开发完成后进行商业化、几乎不进行代码修改就可以售给用户使用的软件成为软件产品，也就是可以买“卖拷贝”的软件，例如 Windows2000。而通常把针对一个或者几个特定的用户而开发的软件成为软件项目，软件项目是一种个性化的产品，可以是按照用户要求全部重新开发，也可以修改已有的软件产品来满足特定的用户需求。项目和产品的不同特点，决定我们测试产品和测试项目仍然会有很多不同的地方：

-质量要求不同。通常产品的质量要高一些，修复发布后产品的缺陷成本较高，甚至会带来很多负面的影响。而做项目通常面向某一用户，虽然质量越高越好，但是一般只要满足用户要求就可以了。

-测试资源投入多少不同。做软件产品通常是研发中心来开发，进度压力要小些。同时由于质量要求高，因此会投入较多的人力、物力资源。

-项目最后要和用户共同验收测试，这是产品测试不具有的特点。

此外，测试产品与测试项目在缺陷管理方面、测试策略制定都会有很大不同，测试管理者应该结合具体的环境，恰如其分的完成工作。

116.用户共同测试（UAT 测试）的注意点有哪些?

软件产品在投产前，通常都会进行用户验收测试。如果用户验收测试没有通过，直接结果就是那不到“Money”，间接影响是损害了公司的形象，而后者的影响往往更严重。根据作者的经验，用户验收测试一定要让用户满意。

实际上用户现场测试更趋于是一种演示。在不欺骗用户的前提下，我们向用户展示我们软件的优点，最后让“上帝”满意并欣然掏出“银子”才是我们的目标。因此用户测试要注意下面的事项：

（1）用户现场测试不可能测试全部功能，因此要测试核心功能。这需要提前做好准备，这些核心功能一定要预先经过测试，证明没有问题才可以和用户共同进行测试。测试核心模块的目的是建立用户对软件的信心。当然如果这些模块如果问题较多，不应该进行演示。

（2）如果某些模块确实有问题，我们可以演示其它重要的业务功能模块，必要时要向用户做成合理的解释。争得时间后，及时修改缺陷来弥补。

（3）永远不能欺骗用户，蒙混过关。道理很简单，因为软件是要给用户用的，问题早晚会暴露出来，除非你可以马上修改。

和用户进行测试还要注意各种交流技巧，争取不但短期利益得到了满足，还要为后面得合作打好基础。

117.如何编写提交给用户的测试报告?

随着测试工作越来越受重视，开发团队向客户提供测试文档是不可避免的事情。很多人会问：“我们可以把工作中的测试报告提供给客户吗？”答案是否定的。因为提供内部测试报告，可能会让客户失去信心，甚至否定项目。

测试报告一般分为内部测试报告和外部测试报告。内部报告是我们在测试工作中的项目文档，反映了测试工作的实施情况，这里不过多讨论，读者可以参考相关教材。这里主要讨论一下外部测试报告的写法，一般外部测试报告要满足下面几个要求：

-根据内部测试报告进行编写，一般可以摘录；

-不可以向客户报告严重缺陷，即使是已经修改的缺陷，开发中的缺陷也没有必要让客户知道；

-报告上可以列出一些缺陷，但必须是中级的缺陷，而且这些缺陷必须是修复的；

-报告上面的内容尽量要真实可靠；

-整个测试报告要仔细审阅，力争不给项目带来负面作用，尤其是性能测试报告。

总之，外部测试报告要小心谨慎的编写。

118.测试工具在测试工作中是什么地位?

国内的很多测试工程师对测试工具相当迷恋，尤其是一些新手，甚至期望测试工具可以取代手工测试。测试工具在测试工作中起的是辅助作用，一般用来提高测试效率。自动化测试弥补了手工测试的不足，减轻一定的工作量。实际上测试工具是无法替代大多数手工测试的，而一些诸如性能测试等自动化测试也是手工所不能完成的。

对于自动测试技术，应当依据软件的不同情况来分别对待，一般自动技术会应用在引起大量重复性工作的地方、系统的压力点、以及任何适合使用程序解决大批量输入数据的地方。然后再寻找合适的自动测试工具，或者自己开发测试程序。一定不要为了使用测试工具而使用。

119.简述负载测试与压力测试的区别。

压力测试（Stress Testing）

压力测试的主要任务就是获取系统正确运行的极限，检查系统在瞬间峰值负荷下正确执行的能力。例如，对服务器做压力测试时就可以增加并发操作的用户数量；或者不停地向服务器发送请求；或一次性向服务器发送特别大的数据等。看看服务器保持正常运行所能达到的最大状态。人们通常使用测试工具来完成压力测试，如模拟上万个用户从终端同时登录，这是压力测试中常常使用的方法。

负载测试（Volume Testing）

用于检查系统在使用大量数据的时候正确工作的能力，即检验系统的能力最高能达到什么程度。例如，对于信息检索系统，让它使用频率达到最大；对于多个终端的分时系统，让它所有的终端都开动。在使整个系统的全部资源达到“满负荷”的情形下，测试系统的承受能力。

120. 写出 bug 报告流转的步骤，每步的责任人及主要完成的工作。

（要结合自己实际的工作经验进行回答，不同公司略有区别）

测试人员提交新的 Bug 入库，错误状态为 New。

高级测试员/测试经理验证错误，如果确认是错误，分配给开发组。设置状态为 Open。如果不是错误，则拒绝，设置为 Declined 状态。

开发经理分配 bug 至对应的模块开发人员。

开发人员查询状态为 Open 的 Bug，如果不是错误，则置状态为 Declined；如果是 Bug 则修复并置状态为 Fixed。不能解决的 Bug，要留下文字说明及保持 Bug 为 Open 状态。

对于不能解决和延期解决的 Bug，不能由开发人员自己决定，一般要通过某种会议（评审会）

通过才能认可。

测试人员查询状态为 Fixed 的 Bug，然后验证 Bug 是否已解决，如解决，置 Bug 的状态为 Closed，如没有解决,置 bug 状态为 Reopen。

121. 写出 bug 报告当中一些必备的内容。

硬件平台和操作系统

测试应用的硬件平台（Platform），通常选择“PC”。

测试应用的操作系统平台（OS）。

a) 版本

提交缺陷报告时通过该字段标识此缺陷存在于被测试软件的哪个版本。

b) Bug 报告优先级

c) Bug 状态

d) Bug 的编号

e) 发现人

f) 提交人

g) 指定处理人

h) 概述

i) 从属关系

j) 详细描述

k) 严重程度

l) 所属模块

m) 附件

n) 提交日期

122. 开发人员老是犯一些低级错误怎么解决?

这种现象在开发流程不规范的团队里特别常见，尤其是一些“作坊式”的团队里。解决这种问题一般从两个方面入手：

一方面从开发管理入手，也就是从根源来解决问题。可以制定规范的开发流程，甚至可以制定惩罚制度，还有就是软件开发前做好规划设计。

另一方面就是加强测试，具体做法就是加强开发人员的自己测试，把这些问题“消灭”在开发阶段，这是比较好的做法，读者可以参考第 13 章试案例分析的“13.1.2 缺陷反复出现，谁的责任”小节，13.1.2 专门讨论了这类问题的方法。

此外，还可以通过规范的缺陷管理来对开发人员进行控制，比如测试部门整理出常见的缺陷，让开发人员自己对照进行检查，以减少这类低级错误的发生。

开发人员犯错误是正常的现象，作为测试人员一定不能抱怨，要认认真真的解决问题才是上策。

123. 画出软件测试的 V 模型图。

124. 为什么要在一个团队中开展软件测试工作?

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

125. 您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作?

（根据项目经验不同，灵活回答即可）

我曾经做过 web 测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

126. 您所熟悉的测试用例设计方法都有哪些？请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

1．等价类划分

划分等价类: 等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的.并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况:有效等价类和无效等价类.

2．边界值分析法

边界值分析方法是对等价类划分方法的补充。测试工作经验告诉我,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以查出更多的错误.

使用边界值分析方法设计测试用例,首先应确定边界情况.通常输入和输出等价类的边界,就是应着重测试的边界情况.应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据.

3．错误推测法

基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法.

错误推测方法的基本思想: 列举出程序中所有可能有的错误和容易发生错误的特殊情况,根据他们选择测试用例. 例如, 在单元测试时曾列出的许多在模块中常见的错误. 以前产品测试中曾经发现的错误等, 这些就是经验的总结. 还有, 输入数据和输出数据为 0 的情况. 输入表格为空格或输入表格只有一行. 这些都是容易发生错误的情况. 可选择这些情况下的例子作为测试用例.

4．因果图方法

前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系, 相互组合等. 考虑输入条件之间的相互组合,可能会产生一些新的情况. 但要检查输入条件的组合不是一件容易的事情, 即使把所有输入条件划分成等价类,他们之间的组合情况也相当多. 因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例. 这就需要利用因果图（逻辑模型）. 因果图方法最终生成的就是判定表. 它适合于检查程序输入条件的各种组合情况.

127. 请以您以往的实际工作为例，详细的描述一次测试用例设计的完整的过程。

就说最近的这次网站功能的测试吧

首先：得到相关文档（需求文档和设计文档），理解需求和设计设计思想后，想好测试策略（测试计划简单点就 OK 了），考虑到测试环境，测试用例，测试时间等问题。

第二步：设计测试用例，测试策略是：把网站部分的功能点测试完，然后在进行系统测试（另外个模块呢有另一个测试人员负责，可以进行联调测试），网站模块的测试基本是功能测试和界面测试（用户并发的可能性很小，所以不考虑）：这次的网站的输入数据呢是使用数据库中的某张表记录，如果表中某一数据记录中新加进来的（还没有被处理的，有个标志位），网站启动后会立刻去刷那张表，得到多条数据，然后在进行处理。处理过程中，会经历 3 个步骤，网站才算完成了它的任务。有 3 个步骤呢，就可以分别对 这 3 个步骤进行测试用例的设计,尽量覆盖到各种输入情况（包括数据库中的数据，用户的输入等），得出了差不多 50 个用例。界面测试，也就是用户看的到的地方，包括发送的邮件和用户填写资料的页面展示。

第三步：搭建测试环境（为什么这个时候考虑测试环境呢？因为我对网站环境已经很熟了，只有有机器能空于下来做该功能测试就可以做了），因为网站本身的环境搭建和其他的系统有点不同，它需要的测试环境比较麻烦，需要 web 服务器（Apache,tomcat），不过这次需求呢，网站部分只用到了 tomcat，所以只要有 tomcat 即可

第四步：执行测试

128. 您以往是否曾经从事过性能测试工作？如果有，请尽可能的详细描述您以往的性能测试工作的完整过程。

（以自己最熟悉的性能测试项目为例）

是的，曾经做过网站方面的性能测试，虽然做的时间并不久（2 个月吧），当时呢，是有位网站性能测试经验非常丰富的前辈带着我一起做。

性能测试类型包括负载测试，强度测试，容量测试等

负载测试：负载测试是一种性能测试指数据在超负荷环境中运行，程序是否能够承担。

强度测试： 强度测试是一种性能测试，他在系统资源特别低的情况下软件系统运行情况

容量测试：确定系统可处理同时在线的最大用户数

在网站流量逐渐加大的情况下，开始考虑做性能测试了，首先要写好性能测试计划，根据运营数据得出流量最大的页面（如果是第一次的话，一般是首页，下载页，个人帐户页流量最大，而且以某种百分比），

Web 服务器指标指标：

* Avg Rps: 平均每秒钟响应次数＝总请求时间 / 秒数；

* Successful Rounds：成功的请求；

* Failed Rounds ：失败的请求；

* Successful Hits ：成功的点击次数；

* Failed Hits ：失败的点击次数；

* Hits Per Second ：每秒点击次数；

* Successful Hits Per Second ：每秒成功的点击次数；

* Failed Hits Per Second ：每秒失败的点击次数；

* Attempted Connections ：尝试链接数；

129. 你对测试最大的兴趣在哪里？为什么？

最大的兴趣就是测试有难度，有挑战性！做测试越久越能感觉到做好测试有多难。曾经在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列了 11，12 点，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的 1，2 点我没有把握，其他点我都很有信心做好它。

130. 你以前工作时的测试流程是什么?

（灵活回答）

公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正（自己总结，吸取同行的方法）后的流程吧。需求评审（有开发人员，产品经理，测试人员，项目经理）—>需求确定(出一份确定的需求文档)—>开发设计文档（开发人员在开始写代码前就能输出设计文档）—>想好测试策略，写出测试用例—>发给开发人员和测试经理看看（非正式的评审用例）—>接到测试版本—>执行测试用例（中间可能会补充用例）—>提交 bug（有些 bug 需要开发人员的确定（严重级别的，或突然发现的在测试用例范围之外的，难以重现的），有些可以直接录制进 TD）—>开发人员修改（可以在测试过程中快速的修改）—>回归测试（可能又会发现新问题，再按流程开始跑）。

131. 当开发人员说不是 BUG 时，你如何应付?

开发人员说不是 bug，有 2 种情况，一是需求没有确定，所以我可以这么做，这个时候可以找来产品经理进行确认，需不需要改动，3 方商量确定好后再看要不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是 BUG 的依据是什么？如果被用户发现或出了问题，会有什么不良结果？程序员可能会给你很多理由，你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来,跟开发经理和测试经理进行确认,如果要修改就改,如果不要修改就不改。其实有些真的不是 bug，我也只是建议的方式写进 TD 中，如果开发人员不修改也没有大问题。如果确定是 bug 的话，一定要坚持自己的立场，让问题得到最后的确认。

132. 软件的构造号与版本号之间的区别？BVT(BuildVerificationTest)

版本控制命名格式: 主版本号.子版本号[.修正版本号[.编译版本号]]

Major.Minor [.Revision[.Build]]

应根据下面的约定使用这些部分：

Major ：具有相同名称但不同主版本号的程序集不可互换。例如，这适用于对产品的大量重写，这些重写使得无法实现向后兼容性。

Minor ：如果两个程序集的名称和主版本号相同，而次版本号不同，这指示显著增强，但照顾到了向后兼容性。例如，这适用于产品的修正版或完全向后兼容的新版本。

Build ：内部版本号的不同表示对相同源所作的重新编译。这适合于更改处理器、平台或编译器的情况。

Revision ：名称、主版本号和次版本号都相同但修订号不同的程序集应是完全可互换的。这适用于修复以前发布的程序集中的安全漏洞。

BVT(BuildVerificationTest)：

作为 Build 的一部分，主要是通过对基本功能、特别是关键功能的测试，保证新增代码没有导

致功能失效，保证版本的持续稳定。实现 BVT 方式是有以下几种：1、测试人员手工验证关键功能实现的正确性。特点：这是传统开发方法中，通常采用的方式。无需维护测试脚本的成本，在测试人力资源充足，测试人员熟悉业务、并对系统操作熟练情况下效率很高，比较灵活快速。缺点：人力成本较高；对测试人员能力有一定要求；测试人员面对重复的工作，容易产生疲倦懈怠，从而影响测试质量。2、借助基于 GUI 的自动化功能测试工具来完成，将各基本功能操作录制成测试脚本，每次回放测试脚本验证功能实现的正确性。特点：能够模拟用户操作完成自动的测试，从 UI 入口到业务实现，每一层的代码实现都经过验证；节约人力成本；降低测试人员重复劳动的工作量，机器不会疲倦；缺点：对于 UI 变动比较频繁的系统来说，这种方式的维护成本很高，实施起来非常困难。另外，在项目周期较短且后续无延续性或继承的情况下，也不推荐使用此方式。3、由开发人员通过自动化测试工具完成业务层的 BVT 测试。特点：通过对业务层关键功能的持续集成测试，保证系统功能的持续稳定。可以结合 DailyBuild，做为 Build 的一部分，自动实现并输入 BVT 报告。缺点：仅对业务规则实现的正确性进行了测试，对表现层无法测试到，对于诸如：前台页面控件各种事件响应、页面元素变化等方面的问题无法保证。

10.2 其他概括性问题

1. 对测试的理解

——考查点：基本的测试知识，对测试是否认可

2. 谈一谈过去自己的工作

——考查点：了解经历、提供进一步提问的素材，表达能力、测试技能

3. 测试设计的方法并举例说明

——考查点：测试技术的使用

4. 测试工具

——考查点：熟悉程度，能否与当前工作匹配？

5. 如何做计划？如何跟踪计划？

——考查点：日常工作能力

6. 如果开发人员提供的版本不满足测试的条件，如何做？

——考查点：与开发人员协作的能力

7. 熟悉 unix 系统、oracle 数据库吗?

——考查点：是否具备系统知识

8. 做过开发吗？写过哪些代码?

——考查点：开发技能

9. 阅读英语文章，给出理解说明?

——考查点：部分英语能力

10. 文档的意义

——考查点：是否善于思考？（最简单的概念，不同层次的理解）

11. 假如进入我们公司，对我们哪些方面会有帮助?

——考查点：讲讲自己的特长

12. 随便找一件物品，让其测试

——考查点：测试的实际操作能力

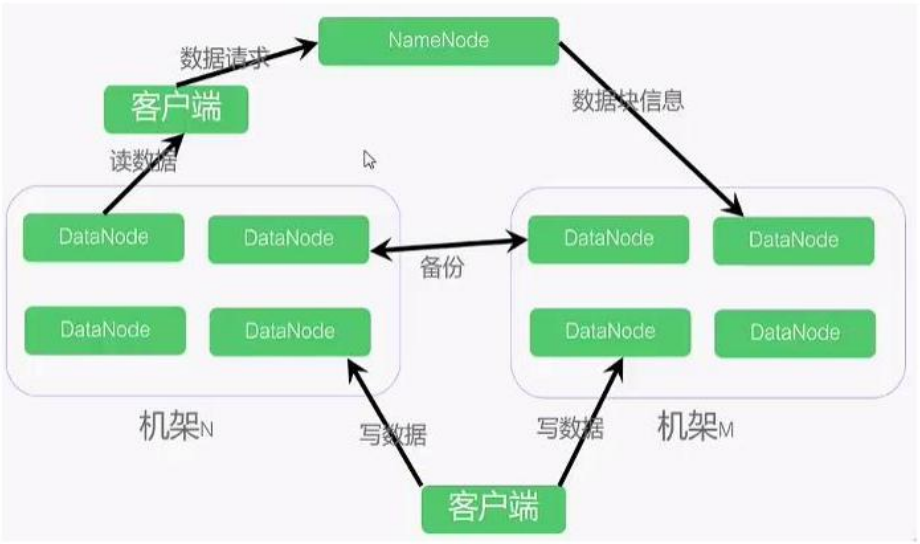
13. 有一个新的软件，假如你是测试工程师，该如何做

——考查点：实际项目经验、是否有带领测试团队的经验和潜力

11　常用技术面试题（云计算与大据数）

1. hdfs 的体系结构

HDFS体系结构



HDFS 有 namenode、secondraynamenode、datanode 组成。

namenode 负责管理 datanode 和记录元数据

secondraynamenode 负责合并日志

datanode 负责存储数据

2. hdfs 的存储机制

HDFS 主要是一个分布式的文件存储系统，由 namenode 来接收用户的操作请求，然后根据文件大小，以及定义的 block 块的大小，将大的文件切分成多个 block 块来进行保存。

在 HDFS 中，文件的读写过程就是 client 和 NameNode 以及 DataNode 一起交互的过程。我们已经知道 NameNode 管理着文件系统的元数据，DataNode 存储的是实际的数据，那么 client 就会联系 NameNode 以获取文件的元数据，而真正的文件读取操作是直接和 DataNode 进行交互的。

3. mapreduce 开发中使用过哪些算法

单词统计 pv

数据去重 uv

topn 最受欢迎的排序

4. 使用 mapreduce 处理数据,数据倾斜指什么?

map /reduce 程序执行时，reduce 节点大部分执行完毕，但是有一个或者几个 reduce 节点运行很慢，导致整个程序的处理时间很长，这是因为某一个 key 的条数比其他 key 多很多（有时是百倍或者千倍之多），这条 key 所在的 reduce 节点所处理的数据量比其他节点就大很多，从而导致某几个节点迟迟运行不完，此称之为数据倾斜。

5. mapreduce 的原理

"Map（映射）"和"Reduce（归约），以及中间的 shuffle 过程

6. spark 集群运算的模式

Spark 有很多种模式，最简单就是单机本地模式，还有单机伪分布式模式，复杂的则运行在集群中，目前能很好的运行在 Yarn 和 Mesos 中，当然 Spark 还有自带的 Standalone 模式，对于大多数情况 Standalone 模式就足够了，如果企业已经有 Yarn 或者 Mesos 环境，也是很方便部署的。

standalone(集群模式)：典型的 Mater/slave 模式，不过也能看出 Master 是有单点故障的；Spark 支持 ZooKeeper 来实现 HA

on yarn(集群模式)： 运行在 yarn 资源管理器框架之上，由 yarn 负责资源管理，Spark 负责任务调度和计算

on mesos(集群模式)： 运行在 mesos 资源管理器框架之上，由 mesos 负责资源管理，Spark 负责任务调度和计算

on cloud(集群模式):比如 AWS 的 EC2,使用这个模式能很方便的访问 Amazon 的 S3;Spark 支持多种分布式存储系统：HDFS 和 S3

7. HDFS 读写数据的过程

读：

1、跟 namenode 通信查询元数据，找到文件块所在的 datanode 服务器

2、挑选一台 datanode（就近原则，然后随机）服务器，请求建立 socket 流

3、datanode 开始发送数据（从磁盘里面读取数据放入流，以 packet 为单位来做校验）

4、客户端以 packet 为单位接收，现在本地缓存，然后写入目标文件

写：

1、根 namenode 通信请求上传文件，namenode 检查目标文件是否已存在，父目录是否存在

2、namenode 返回是否可以上传

3、client 请求第一个 block 该传输到哪些 datanode 服务器上

4、namenode 返回 3 个 datanode 服务器 ABC

5、client 请求 3 台 dn 中的一台 A 上传数据（本质上是一个 RPC 调用，建立 pipeline），A 收到请求会继续调用 B，然后 B 调用 C，将真个 pipeline 建立完成，逐级返回客户端

6、client 开始往 A 上传第一个 block（先从磁盘读取数据放到一个本地内存缓存），以 packet 为单位，A 收到一个 packet 就会传给 B，B 传给 C；A 每传一个 packet 会放入一个应答队列等待应答

7、当一个 block 传输完成之后，client 再次请求 namenode 上传第二个 block 的服务器。

8. RDD 中 reduceBykey 与 groupByKey 哪个性能好，为什么

reduceByKey：reduceByKey 会在结果发送至 reducer 之前会对每个 mapper 在本地进行 merge，有点类似于在 MapReduce 中的 combiner。这样做的好处在于，在 map 端进行一次 reduce 之后，数据量会大幅度减小，从而减小传输，保证 reduce 端能够更快的进行结果计算。

groupByKey：groupByKey 会对每一个 RDD 中的 value 值进行聚合形成一个序列(Iterator)，此操作发生在 reduce 端，所以势必会将所有的数据通过网络进行传输，造成不必要的浪费。同时如果数据量十分大，可能还会造成 OutOfMemoryError。

通过以上对比可以发现在进行大量数据的 reduce 操作时候建议使用 reduceByKey。不仅可以提高速度，还是可以防止使用 groupByKey 造成的内存溢出问题。

9. spark2.0 的了解

更简单：ANSI SQL 与更合理的 API

速度更快：用 Spark 作为编译器

更智能：Structured Streaming

10. rdd 怎么分区宽依赖和窄依赖

宽依赖：父 RDD 的分区被子 RDD 的多个分区使用　　例如 groupByKey、reduceByKey、sortByKey 等操作会产生宽依赖，会产生 shuffle

窄依赖：父 RDD 的每个分区都只被子 RDD 的一个分区使用　例如 map、filter、union 等操作会产生窄依赖

11. spark streaming 读取 kafka 数据的两种方式

这两种方式分别是：

Receiver-base

使用 Kafka 的高层次 Consumer API 来实现。receiver 从 Kafka 中获取的数据都存储在 Spark Executor 的内存中，然后 Spark Streaming 启动的 job 会去处理那些数据。然而，在默认的配置下，

这种方式可能会因为底层的失败而丢失数据。如果要启用高可靠机制，让数据零丢失，就必须启用 Spark Streaming 的预写日志机制（Write Ahead Log，WAL）。该机制会同步地将接收到的 Kafka 数据写入分布式文件系统（比如 HDFS）上的预写日志中。所以，即使底层节点出现了失败，也可以使用预写日志中的数据进行恢复。

Direct

Spark1.3 中引入 Direct 方式，用来替代掉使用 Receiver 接收数据，这种方式会周期性地查询 Kafka，获得每个 topic+partition 的最新的 offset，从而定义每个 batch 的 offset 的范围。当处理数据的 job 启动时，就会使用 Kafka 的简单 consumer api 来获取 Kafka 指定 offset 范围的数据。

12. kafka 的数据存在内存还是磁盘

Kafka 最核心的思想是使用磁盘，而不是使用内存，可能所有人都会认为，内存的速度一定比磁盘快，我也不例外。在看了 Kafka 的设计思想，查阅了相应资料再加上自己的测试后，发现磁盘的顺序读写速度和内存持平。

而且 Linux 对于磁盘的读写优化也比较多，包括 read-ahead 和 write-behind，磁盘缓存等。如果在内存做这些操作的时候，一个是 JAVA 对象的内存开销很大，另一个是随着堆内存数据的增多，JAVA 的 GC 时间会变得很长，使用磁盘操作有以下几个好处：

磁盘缓存由 Linux 系统维护，减少了程序员的不少工作。

磁盘顺序读写速度超过内存随机读写。

JVM 的 GC 效率低，内存占用大。使用磁盘可以避免这一问题。

系统冷启动后，磁盘缓存依然可用。

13. 采集数据为什么选择 kafka

采集层 主要可以使用 Flume, Kafka 两种技术。

Flume：Flume 是管道流方式，提供了很多的默认实现，让用户通过参数部署，及扩展 API.

Kafka：Kafka 是一个可持久化的分布式的消息队列。

Kafka 是一个非常通用的系统。你可以有许多生产者和很多的消费者共享多个主题 Topics。相比之下,Flume 是一个专用工具被设计为旨在往 HDFS,HBase 发送数据。它对 HDFS 有特殊的优化，并且集成了 Hadoop 的安全特性。所以，Cloudera 建议如果数据被多个系统消费的话，使用 kafka；如果数据被设计给 Hadoop 使用，使用 Flume。

14. 怎么解决 kafka 的数据丢失

producer 端：

宏观上看保证数据的可靠安全性，肯定是依据分区数做好数据备份，设立副本数。

broker 端：

topic 设置多分区，分区自适应所在机器，为了让各分区均匀分布在所在的 broker 中，分区数要大于 broker 数。

分区是 kafka 进行并行读写的单位，是提升 kafka 速度的关键。

Consumer 端

consumer 端丢失消息的情形比较简单：如果在消息处理完成前就提交了 offset，那么就有可能造成数据的丢失。由于 Kafka consumer 默认是自动提交位移的，所以在后台提交位移前一定要保证消息被正常处理了，因此不建议采用很重的处理逻辑，如果处理耗时很长，则建议把逻辑放到另一个线程中去做。为了避免数据丢失，现给出两点建议：

enable.auto.commit=false　关闭自动提交位移

在消息被完整处理之后再手动提交位移

15. Redis 性能优化，单机增加 CPU 核数是否会提高性能

1、根据业务需要选择合适的数据类型，并为不同的应用场景设置相应的紧凑存储参数。

2、当业务场景不需要数据持久化时，关闭所有的持久化方式可以获得最佳的性能以及最大的内存使用量。

3、如果需要使用持久化，根据是否可以容忍重启丢失部分数据在快照方式与语句追加方式之间选择其一，不要使用虚拟内存以及 diskstore 方式。

4、不要让你的 Redis 所在机器物理内存使用超过实际内存总量的 3/5。

我们知道 Redis 是用"单线程-多路复用 io 模型"来实现高性能的内存数据服务的，这种机制避免了使用锁，但是同时这种机制在进行 sunion 之类的比较耗时的命令时会使 redis 的并发下降。因为是单一线程，所以同一时刻只有一个操作在进行，所以，耗时的命令会导致并发的下降，不只是读并发，写并发也会下降。而单一线程也只能用到一个 cpu 核心，所以可以在同一个多核的服务器中，可以启动多个实例，组成 master-master 或者 master-slave 的形式，耗时的读命令可以完全在 slave 进行。

16. 怎样快速的杀死一个 job

1、执行 hadoop　job -list　拿到 job-id

2、Hadoop job kill hadoop-id

17. hdfs 的数据压缩算法

Hadoop 的压缩算法有很多，其中比较常用的就是 gzip 算法与 bzip2 算法，都可以可通过 CompressionCodec 来实现

18. flume 不采集 Nginx 日志,通过 log4j 采集日志,优缺点是什么?

在 nginx 采集日志时无法获取 session 的信息，然而 logger4j 则可以获取 session 的信息，logger4j 的方式比较稳定，不会宕机。缺点：不够灵活，logger4j 的方式和项目结合过滤紧密，二 flume 的方式就比较灵活，便于插拔式比较好，不会影响项目的性能。

19. flume 和 kafka 有集日志区别,采集日志时中间停了,怎么记录之前的日志

Flume 采集日志是通过流的方式直接将日志收集到存储层，而 kafka 试讲日志缓存在 kafka

集群，待后期可以采集到存储层。Flume 采集中间停了，可以采用文件的方式记录之前的日志，而 kafka 是采用 offset(偏移量) 的方式记录之前的日志。

20. kafka 的 message 包括哪些信息

一个 Kafka 的 Message 由一个固定长度的 header 和一个变长的消息体 body 组成

header 部分由一个字节的 magic(文件格式)和四个字节的 CRC32(用于判断 body 消息体是否正常)构成。当 magic 的值为 1 的时候，会在 magic 和 crc32 之间多一个字节的数据：attributes(保存一些相关属性，比如是否压缩、压缩格式等等)；如果 magic 的值为 0，那么不存在 attributes 属性

body 是由 N 个字节构成的一个消息体，包含了具体的 key/value 消息