软件测试面试宝典

目录

1 测试理论

1.1 测试基础

1.1.1 什么是软件测试？

为了发现程序中的错误而执行程序的过程

1.1.2 软件测试的目的？

首先，测试并不仅仅是为了要找出错误。通过分析错误产生的原因和错误的分布特征，可以帮助项目管理者发现当前所采用的软件过程的缺陷，以便改进。同时，这种分析也能帮助我们设计出有针对性地检测方法，改善测试的有效性。

其次，没有发现错误的测试也是有价值的，完整的测试是评定测试质量的一种方法。详细而严谨的可靠性增长模型可以证明这一点。

测试的目的是按照用户所需软件的质量，检查开发软件过程出现的 bug，使得开发人员及时修改，可以避免在开发结束的时候发现软件存在质量问题，避免公司不必要的损失。赢得用户对公司产品的认可。

测试的目的是以最少人力、物力和时间找出软件中潜在各种错误和缺陷，通过修正种错误和缺陷提高软件质量，回避软件发布后由于潜在的软件缺陷和错误造成的隐患带来的商业风险。

测试的附带收获是，它能够证明软件的功能和性能与需求说明相符合。

实施测试收集到的测试结果数据为可靠性分析提供了依据。

测试不能表明软件中不存在错误，它只能说明软件中存在错误。

1.1.3 软件测试的目标

发现尽可能多的错误

测试是一个为了寻找错误而运行程序的过程。

一个好的测试案例是指很可能找到迄今为止尚未发现的错误的用例。

一个成功的测试是指揭示了迄今为止尚未发现的错误的测试。

1.1.4 软件测试的原则

1) 应当把“尽早地和不断地进行软件测试”作为软件开发者的座右铭。

2) 测试用例应由测试输入数据和对应的预期输出结果这两部分组成。

3) 程序员应避免检查自己的程序。

4) 在设计测试用例时，应包括合理的输入条件和不合理的输入条件。

5) 软件测试的原则

6) 充分注意测试中的群集现象。

经验表明，测试后程序中残存的错误数目与该程序中已发现的错误数目成正比。

7) 严格执行测试计划，排除测试的随意性。

8) 应当对每一个测试结果做全面检查。

9) 妥善保存测试计划，测试用例，出错统计和最终分析报告，为维护提供方便。

1.1.5 测试的工作流程

测试的功能点都是来自于需求文档从产品的需求文档中提炼出来的，等产品完成需求文档并完成需

求文档的评审会就开始测试用例的编写工作，一般项目半个月迭代一次的话设计测试的时间一般是 3 天就要完成，我们设计测试用例的时间还是比较充足案例设计一般都会和产品的开发并行。在案例完成编写之后大家会开会一起来评审你的案例。在评审的过程中大家会提出一些问题，会后要把这些遗漏的测试点补充上，但是这时并不是大功告成哦。痛苦的案例执行才刚刚开始，哈哈哈，在这个里我为什么用痛苦来形容呢，大家也知道开发一般只是把功能开发好自己可能都没有自测过就发给测试，这时候测试发现和自己想象中的 APP 差距太大，有的时候会发现一眼都能看到的问题为什么还要等着测试来发现，小编遇到这样的问题也表示无奈。但是只能硬着头皮测下去。在测试的过程中每天在下班的时候还需要发测试日报告诉项目中的成员现在案例执行的情况，当然了测试完成之后发测试报告也是必须的了，算是对这次项目跌代测试完成的一个交代。

1.1.6 测试工程师的职责

测试经理：

1、制定测试计划。

2、确保测试过程正常进行。

测试工程师

1、编写测试用例

2、搭建测试环境

3、执行测试

1.1.7 软件都有多少种分类？

根据功能的不同，电脑软件可以粗略地分成四个层次：

最贴近电脑硬件的是一些小巧的软件。它们实现一些最基本的功能，通常“固化”在只读存储器芯片中，因此称为固件。

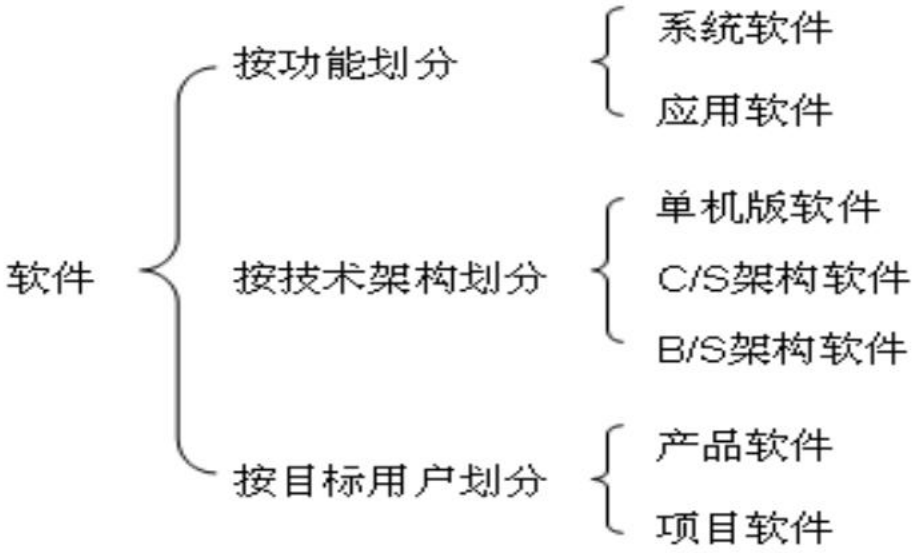
系统软件包括操作系统和编译器软件等。系统软件和硬件一起提供一个“平台”。它们管理和优化电脑硬件资源的使用。

支持软件。包括图形用户界面、软件开发工具、软件评测工具、数据库管理系统、中间件等。

应用软件种类最多，包括办公软件、电子商务软件、通信软件、行业软件，游戏软件等

等。

1.1.8 软件的分类



1.1.9 测试的主要方面

A、功能测试：a、链接测试 b、表单测试 c、Cookies 测试 d、设计语言测试 e、数据库测试

B、性能测试：a、连接速度测试 b、负载测试 c、压力测试

C、接口测试：a、服务器接口 b、外部接口 c、错误处理

D、可用性测试: a、导航测试 b、图形测试 c、内容测试 d、整体界面测试

E、兼容性测试：a、平台测试 b、浏览器测试 c、视频测试 d、Modem/连接速率测试 f、打印机测试 g、组合测试

F、安全测试：a、目录设置 b、登录 c、Session d、日志文件 e、加密 f、安全漏洞

G、代码合法性测试：a、程序代码合法性检查 b、显示代码合法性检查

H、文档测试：

1.1.10软件测试的对象

软件测试并不等于程序测试。软件测试应贯穿于软件定义与开发的整个期间。需求分析、概要设计、详细设计以及程序编码等各阶段所得到的文档，包括需求规格说明、概要设计规格说明、详细设计规格说明以及源程序，都应成为软件测试的对象

1.1.11什么是“测试案例”？

测试案例是一份文档，它描述了一个输入、反应、或者是与其相应的预期的响应，以便来判断应用软件的工作是否正常。测试案例应当包括测试标识、测试案例的名称、目标、测试条件/设置、输入数据要求、步骤、以及预期的结果。

注：开发一个应用软件的测试案例的过程，需要全面、深入地考虑该软件的操作，所以有助于发现在其需求或设计里面的问题。因此，如果有可能，在开发周期中应当尽早准备测试案例。

1.1.12怎么编写案例？

案例的编写与测试阶段的定义有很大的关系。系统测试和 unit 测试的案例可能不同。总体而言测试案例根据系统的需求而定。

1.1.13软件测试的两种方法

黑盒测试和白盒测试

黑盒：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。黑盒测试又叫做功能测试或数据驱动测试。

白盒：此方法把测试对象看做一个透明的盒子，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的状态，确定实际的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。

1.1.14测试结束的标准是什么？

1.用例全部执行。2.覆盖率达到标准。3.缺陷率达到标准。4.其他指标达到质量标准

1.1.15软件的生命周期

软件生命周期是指一个计算机软件从功能确定、设计，到开发成功投入使用，并在使用中不断地修改、增补和完善，直到停止该软件的使用的全过程（从酝酿到废弃的过程）

1.1.16什么是软件的生命周期？

生命周期从收到应用软件开始算起，到该软件不再使用为止。它有如下各方面的内容：初始构思、需求分析、功能设计、内部设计、文档计划、测试计划、文档准备、集成、测试、维护、升级、再测试、逐步淘汰 (phase-out)、等等。

1.1.17软件测试按过程分为三个步骤

单元测试：单元测试又称模块测试，是针对软件设计的最小单位 – 程序模块，进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

集成测试：在运行（可能是不完整）的应用中保证软件单元被结合后能正常操作的测试执行的阶段

系统测试：当应用作为整体运行时的测试执行阶段

1.1.18面向对象的设计如何影响测试？

好的面向对象的工程设计使得从代码追溯内部设计、再到功能测试，最后追溯到需求，成为一件容易的事。因为它对黑盒测试的影响很少 (不需要了解应用软件的内部设计) ，而白盒测试只需针对该应用软件的对象。如果该应用软件设计得好，就可简化测试设计。

1.1.19软件带来错误的原因很多。主要的原因有哪些？

1) 交流不够、交流上有误解或者根本不进行交流
2) 软件复杂性
3) 程序设计错误
4) 需求变化
5) 时间压力
6) 代码文档贫乏
7) 软件开发工具

1.1.20做好软件测试的一些关键点

1.测试人员必须经过测试基础知识和理论的相关培训。

2.测试人员必须熟悉系统功能和业务。

3.测试必须事先要有计划，而且测试方案要和整个项目计划协调好

4.必须事先编写测试用例，测试执行阶段必须根据测试用例进行

5.易用性，功能，分支，边界，性能等功能性和非功能性需要都要进行测试

6.对于复杂的流程一定要进行流程分支，组合条件分析，再进行等价类划分准备相关测试数据

7.测试设计的一个重要内容是要准备好具体的测试数据，清楚这个测试数据是测哪个场景或分支的

8.个人任务平均每三个测试用例至少应该发现一个 BUG，否则只能说明测试用例质量不好

9.除了每日构建的冒烟测试可以考虑测试自动化外，其它暂时都不要考虑去自动化。

1.1.21软件测试的步骤是什么？

1) 测试过程按 4 个步骤进行，即单元测试（Unit Testing）、集成测试（Integrated Testing）、确认测试（Validation Testing）和系统测试（System Testing）及发版测试。

2) 开始是单元测试，集中对用源代码实现的每一个程序单元进行测试，检查各个程序模块是否正确地实现了规定的功能。

3) 集成测试把已测试过的模块组装起来，主要对与设计相关的软件体系结构的构造进行测试。

4) 确认测试则是要检查已实现的软件是否满足了需求规格说明中确定了的各种需求，以及软件配置是否完全、正确。

1.1.22如何录制测试脚本？

1) 新建一个脚本（Web/HTML 协议）

2) 点击录制按钮，在弹出的对话框的 URL 中输入”about:blank”。

3) 在打开的浏览器中进行正常操作流程后，结束录制。

4) 调试脚本并保存。可能要注意到字符集的关联。

5) 设置测试场景

6) 针对性能设置测试场景，主要判断在正常情况下，系统的平均事务响应时间是否达标

7) 针对压力负载设置测试场景，主要判断在长时间处于满负荷或者超出系统承载能力的条件下，系统是否会崩溃

1.1.23应该考虑进行如何测试的测试方法

黑盒测试 (Black box testing) — 不考虑内部设计和代码，根据需求和功能进行测试。

白盒测试 (White box testing) — 根据应用软件的代码的内部逻辑，按照代码的语句、分支、路径和条件进行测试。

功能测试（functional testing）——对一个应用软件的功能模块进行黑盒测试。这种测试应当由测试人员进行。但这并不意味着程序员在推出软件之前不进行代码检查。（这一原则适用于所有的测试阶段。）

系统测试 — 针对全部需求说明进行黑盒测试，包括系统中所有的部件。

回归测试 (regression testing) — 每当软件经过了整理、修改、或者其环境发生变化，都重复进行测试。很难说需要进行多少次回归测试，特别是是到了开发周期的最后阶段。进行此种测试，特别适于使用自动测试工具。

负荷试验 (load testing) — 在大负荷条件下对应用软件进行测试。例如测试一个网站在不同负荷情况下的状况，以确定在什么情况下系统响应速度下降或是出现故障。

压力测试 (stress testing) — 经常可以与“负荷测试”或“性能测试”相互代替。这种测试是用来检查系统在下列条件下的情况：在非正常的巨大负荷下、某些动作和输入大量重复、输入大数、对数据库进行非常复杂的查询，等等。

性能测试 (performance testing) — 经常可以与“压力测试”或“负荷测试”相互代替。理想的“性能测试”(也包括其他任何类型的测试) 都应在质量保障和测试计划的文档终予以规定。

可用性测试 (usability testing) — 是专为“对用户友好”的特性进行测试。这是一种主观的感觉，取决于最终用户或顾客。可以进行用户会见、检查、对用户会议录像、或者使用其他技术。程序员和测试人员通常不参加可用性测试。

安装/卸载测试 (install/uninstall testing) — 对安装/卸载进行测试 (包括全部、部分、升级操作)。

安全测试 (security testing) — 测试系统在应付非授权的内部/外部访问、故意的损坏时的防护情况。这需要精密复杂的测试技术。

兼容性测试 (compatability testing) — 测试在特殊的硬件/软件/操作系统/网络环境下的软件表现。

α 测试 (alpha testing) — 在开发一个应用软件即将完成时所进行的测试。此时还允许有较小的设计修

改。通常由最终用户或其他人进行这种测试，而不是由程序员和测试人员来进行。

β 测试 (beta testing) — 当开发和测试已基本完成，需要在正式发行之前最后寻找毛病而进行的测试。通常由最终用户或其他人进行这种测试，而不是由程序员和测试人员来进行。

1.1.24怎样估计测试工作量？

效率假设：即测试队伍的工作效率。对于功能测试，这主要依赖于应用的复杂度，窗口的个数，每个窗口中的动作数目。对容量测试，主要依赖于建立测试所需数据的工作量大小。

测试假设：为了验证一个测试需求所需测试动作数目。应用的维数：应用的复杂度指标。例如要加入一个记录，测试需求的维数就是这个记录中域的数目。

所处测试周期的阶段：有些阶段主要工作都在设计，有些阶段主要是测试执行。

1.1.25测试设计的问题

1) 不做测试设计，测试过程也是胡乱建立的。

2) 测试设计不详细，不是基于可量度的测试策略，例如测试计划覆盖一个集合或者测试需求的一个子集。

3) 测试过程没有采用最好的技术来检验 Windows C/S 结构的测试需求

4) 测试用例的选择规则

5) 选择与测试需求的实质部分最相关的测试用例。

6) 选择的测试用例应该不容易应用程序的改变的影响。

1.1.26当测试过程发生错误时，有哪几种解决办法？

1) 跳转到别的测试过程

2) 调用一个能够清除错误的过程

3) 退出过程，启动另一个

4) 退出过程和应用程序，重新启动启动 Windows，在失败的地方重新开始测试

1.1.27测试执行的问题

测试执行的问题

1) 自动化测试没有有效的利用，使得手工测试太多。

2) 测试结果的捕获没有系统性，而且没有查看或调查

3) 缺陷报告必须用手工加入缺陷跟踪系统

错误分类

1、 测试用例失败

正常错误

2、 脚本命令失败

当测试过程不能不能执行录制过程中的某个功能时，回产生这种错误，如鼠标单击按钮或选择菜单项等。它也能指示是缺陷还是测试过程的设计问题。

3、 致命错误

导致测试停止，这种情况最好重起 Windows。

具体步骤：

1) 建立测试系统

2) 准备测试过程

3) 运行初始化过程

4) 执行测试

5) 从终止的测试恢复

6) 验证预期结果

7) 调查突发结果

8) 记录缺陷日记

1.1.28测试评估的目标

1) 量化测试进程

2) 生成缺陷和测试覆盖率的总结报告

1. 测试评估的问题

3) 没有把测试覆盖率作为报告测试进程的根据，使得不知测试是否结束；

4) 没有做缺陷评估，缺陷评估是量度软件可行性的重要指标；

5) 不使用专门的软件工具进行数据输入任务和相应的评估活动，使得这些任务变得繁重累人。

1.1.29如何提高测试？

提高测试需要从几个方面着手，其实只是自己的一些感觉，不一

定就需要按部就班，需要找自己适合的点。

制定完备的测试计划

清楚的认识测试计划，测试计划是一个文档，能够保证整个研

发过程中顺利执行的一个指导性文档，它描述了几个方面的问题。

1）描述了项目的的

2）描述了项目的开发周期

3）描述了在测试中遇到的技术

4）描述了测试案例的设计周期

5）描述测试案例的执行周期

6）描述了测试过程中用到的工具或者技术

7）描述了测试过程中用到的资源情况

8）描述了测试过程中可能遇到的风险以及规避方法

9） 提高案例设计水平

1.1.30C/S 模式的优点和缺点

C/S 模式的优点

1) 由于客户端实现与服务器的直接相连，没有中间环节，因此响应速度快。

2) 操作界面漂亮、形式多样，可以充分满足客户自身的个性化要求。

3) C/S 结构的管理信息系统具有较强的事务处理能力，能实现复杂的业务流程。

C/S 模式的缺点

1) 需要专门的客户端安装程序，分布功能弱，针对点多面广且不具备网络条件的用户群体，不能够实

2) 现快速部署安装和配置。

3) 兼容性差，对于不同的开发工具，具有较大的局限性。若采用不同工具，需要重新改写程序。

4) 开发成本较高，需要具有一定专业水准的技术人员才能完成。

1.1.31B/S 模式的优点和缺点

B/S 模式的优点

1) 具有分布性特点，可以随时随地进行查询、浏览等业务处理。

2) 业务扩展简单方便，通过增加网页即可增加服务器功能。

3) 维护简单方便，只需要改变网页，即可实现所有用户的同步更新。

4) 开发简单，共享性强。

B/S 模式的缺点

1) 个性化特点明显降低，无法实现具有个性化的功能要求。

2) 操作是以鼠标为最基本的操作方式，无法满足快速操作的要求。

3) 页面动态刷新，响应速度明显降低。

4) 无法实现分页显示，给数据库访问造成较大的压力。

5) 功能弱化，难以实现传统模式下的特殊功能要求。

1.1.32测试结束的标准是什么?

1) 第一类标准：测试超过了预定时间，则停止测试。

2) 第二类标准：执行了所有的测试用例，但并没有发现故障，则停止测试。

3) 第三类标准：使用特定的测试用例设计方案作为判断测试停止的基础。

4) 第四类标准：正面指出停止测试的具体要求，即停止测试的标准可定义为查出某一预订数目的故障。

5) 第五类标准：根据单位时间内查出故障的数量决定是否停止测试。

1.1.33怎么才能够全面的测试到每一个点?

测试的全面性主要需要在设计测试计划的时候考虑，从测试策略，产品需求等等多个角度考虑从而定义全部的测试点

1.1.34开发与测试的关系

开发和测试是一个有机的整体！在产品的发布之前，开发和测试是循环进行的， 测出的缺陷要经开发人员修改后继续测试。在开发的同时测试经理开始编写测试用例，测试文档要参考开发文档，所以开发和测试是不可分割的，少了任何一个都不能开发出产品。

从角色方面看，像理论和实验的关系，开发人员通过自己的想象创造出一套思想，之后测试人员再对它进行检验、证伪，开发人员再修改的过程从而不断丰富产品。从方法方面看，是演绎和归纳的关系，一个要掌握大量的技术，一个要不断的从实例中学习。因这 两方面的不同，所以开发和测试看上去做的工作很不一样。

开发与测试是相辅相承、密不可分的，开发人员开发出新的产品后要通过测试判断产 品是否完全满足用户的需求。如果发现缺陷，提交给开发人员进行修复，然后再转交测试人员进行回归测试，直到产品符合需求规格说明。一个符合用户需求的产品是开发和测试共同努力的成果。

1.1.35怎么和开发沟通

测试和开发沟通大部分都在讨论 bug，测试说是 bug 但是开发认为这个不是 bug，对于测试来说就很头痛了明明是问题但是为什么开发不主动修改呢？这时候测试应该去需求文档中找出有关这个功能的描述或者去询问产品经理，总之不要正面冲突，要拿出证据来说服开发。

1.1.36测试过程

1) 制定系统测试计划
2) 编写系统测试用例
3) 执行系统测试用例
4) 跟踪管理缺陷
5) 总结测试

1.1.37测试出口准则

1) 所有的缺陷已经解决
2) 项目规定测试阶段时间结束
3) 执行完成测试计划中的系统测试内容，修正了所发现的错误，未修正的错误被项

目经理允许留到下一版本

4) 高级经理和项目经理均同意结束测试

5) 测试结果经过了专门的评审

1.1.38测试完成标准

1) 系统功能与用户需求说明书一致

2) 功能性测试用例通过率达到 100%

3) 非功能性测试用例通过率达到 95%

4) 一、二级错误修复率应达到 100%。

5) 三、四级错误修复率应达到 80%以上。

6) 五级错误修复率应达到 60%以上。

1.1.39测试活动中统计了哪些数据？

工作量 bug 数量

1.1.40如何选择用户测试的工作产品？

在用户有需求得到签字确认之后，我们选择用户测试的工作产品。我们几乎所有的项目都进行了测试，我们是在项目立项公告中得知需要对工作产品进行测试。

1.1.41测试环境描述在哪儿？

测试环境在测试计划里面进行描述，测试计划是由测试经理编写，我们在测试计划中了解到自己是此次项目组的测试工程师。

1.1.42进行测试时产生了哪些文档或记录？

测试的整个过程有系统测试计划、系统测试用例、系统测试报告、缺陷报告、产品发布说明

在执行测试的过程中 只有缺陷报告，这个还是用在缺陷管理工具中进行的，最后在工具中导出缺陷报告

1.1.43测试人员需要何时参加需求分析？

如果条件循序 原则上来说 是越早介入需求分析越好 因为测试人员对需求理解越深刻 对测试工作的开展越有利 可以尽早的确定测试思路 减少与开发人员的交互 减少对需求理解上的偏差

1.1.44产品测试完以后由谁来发布？

这个不定 开发发布 还是技术支持发布都有可能

1.1.45软件测试与调试的关系

1) 测试条件已知，规程可定义，结果可预知
2) 测试可以计划，过程可控
3) 测试是检验，调试是推理过程
4) 测试表明程序失败，调试表明正确
5) 测试可不了解设计细节
6) 测试由非设计人员完成
7) 测试有理论依据
8) 测试可自动化

1.1.46质量的八大特性是什么？各种特性的定义？

1）功能性：软件所实现的功能达到它的设计规范和满足用户需求的程度

2）性能：在规定条件下,实现软件功能所需的响应时间和计算机资源(CPU、内存、磁盘空间和数据吞吐量)的使用程度

3）可靠性：在满足一定条件的应用环境中，软件能够正常维持其工作的能力，在出现一些错误操作时，软件可以具有容错性，如果软件意外退出，重新启动后可以恢复最近的软件数据

4）安全性：为了防止意外或人为的破坏，软件应具备的自身保护能力

5）使用性：用户在理解、学习和操作软件的过程中的付出的努力的难易程度

6）维护性：软件在运行维护过程中，如果出现了运行故障或者扩展新功能和性能，软件系统是否具有可分析性和良好的扩展性，重新设计后的软件的稳定性和可测试性

7）移植性：软件从现有运行平台向另一个运行平台过度的适应程度和平台可替换性

8）重用性：整个软件或其中一部分能作为软件包而被再利用的程度

1.1.47什么是软件的“质量”？

高质量的软件是适当的、无错误的，能在预算内按时交货，满足需求/或期望，并且是可维护的。所以，质量是一个主观的术语。它取决于谁是客户以及客户对项目计划的影响。对一个软件开发项目来说，“客户”的范围很广，包括最终用户、客户所接受的测试者、与客户合同有关的官员、客户管理、开发机构的管理者/会计/测试人员/销售人员、未来的软件维护工程师、股票持有者、杂志专栏记者，等等。每一类客户对“质量”都有自己的倾向性 – 会计部门判断质量会从其收益来考虑，而最终用户则重视友好的用户界面和没有错误。

1.1.48软件质量应该从哪些方面来评价？

可靠性、安全性、性能、易用性、外观、稳定性

1.1.49什么是“软件质量保障”？

软件质量保障涉及到整个软件开发过程，包括监视和改善过程、确保任何经过认可的标准和步骤都被遵循、并且保证问题被发现和被处理。从本质上说，软件质量保障是“预防”。

1.1.50为什么软件会有毛病？

1. 交流错误或者没有进行交流，

2. 软件的复杂性 编程错误

3. 需求变更 客户恐怕不明白改变需求的影响，也许是知道但依然需要变更 —会导致重新设计、重订工程进度表、对其他项目的影响、已完成的工作需要重做或者放弃、对硬件需求的影响等等。如果在项目中出现许多小的改变或一个大的改变，在项目各部分中出现已知或未知的相关的问题，可能会相互影响并导致出现问题。而且，不断地变更也会增加软件的复杂性，可能会导致错误的出现。这样就会影响技术人员的积极性。在一些快速变化的商业环境里，持续变更需求的影响是致命的。在这种情况下，管理者必须知道它的危险性。质量保障和测试工程师必须与此相适应，并安排持续的广泛的测试，以克服不可避免产生的问题。

4. 时间压力

因为有许多猜测成分，软件开发项目的进度很难安排得理想。当最后期限快到的时候，压力逐渐增大，错误随之产生

5. 自负心理、 代码文档质量差、 软件开发工具

1.1.51什么是 UML？

Unified Modeling Language 它是一种用于描述，构造软件系统以及商业建模的语言。简单的理解就是它可以以一种直观的方式表示出一个系统的各项内容

1.1.52什么是 CMM？

CMM：Capability Maturity Model，即“能力成熟度模型”。它是一个分 5 级的、可以描述结构完善程度的模型，用它来说明所交付的软件的效能。

1.1.5355. 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

黑盒测试：把测试对象当成一个黑盒子，测试人员完全不考虑逻辑结构和内部特性，只依据程式的需求说明书来检查程式的功能是否满足它的功能说明。

白盒测试：把测试对象当成一个透明的盒子，允许测试人员利用程序内部逻辑结构及相关信息，设计或选择测试用例，对程式所有逻辑路径进行测试。

单元测试：白盒测试的一种，对软件设计中的单元模块进行测试。

集成测试：在单元测试的基础上，对单元模块之间的连接和组装进行测试。

系统测试：在所有都考虑的情况下，对系统进行测试。

验收测试：第三方进行的确认软件满足需求的测试。

1.1.54比较负载测试、压力测试，容量测试和强度测试区别

负载测试：在一定的工作负荷下，系统的负荷及响应时间。通过逐步增加系统负载，最终确定在满足性能指标的情况下，系统能承受的最大负载量的测试。

强度测试：又称疲劳强度测试，在系统稳定运行的情况下能够支持的最大并发用户数，持续执行一段时间业务，通过综合分析，确定系统处理最大工作量强度性能的过程。一定负荷条件下，在较长时间

跨度内的系统连续运行给系统性能所造成的影响。

容量测试：容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且目的是显示系统可以处理目标内确定的数据容量。

压力测试：通过逐步增加系统负载，最终确定在什么负载条件下系统性能将处于崩溃状态，以此获得系统能提供的最大服务级别的测试。

1.1.55测试执行过程的三个阶段

（1）初测期

测试主要功能和关键的执行路径，排除主要障碍。

（2）细测期

依据测试计划和测试大纲、测试用例，逐一测试大大小小的功能、方方面面的特性、性能、用户界面、兼容性、可用性等等；预期可发现大量不同性质、不同严重程度的错误和问题。

（3）回归测试期

系统已达到稳定，在一轮测试中发现的错误已十分有限；复查已知错误的纠正情况，确认未引发任何新的错误时，终结回归测试。

1.1.56什么是验证、评价、预排、检查？

验证 (verification) 涉及了回顾和会议，以评估文档、计划、代码、需求和说明书。可以通过检查表、调查表、排练、和检查会来进行。

评价 (validation) 则指在检察完成之后的实际测试。术语“IV”和“V”分别代表验证和评价。

预排是一个非正式的会议，用来进行评估和信息交流。通常不需要或者只需很少一点准备。

检查比预排更正式一点，通常有 3-8 个人参加会议，包括一个仲裁者 (moderator)、读者 (可以是作者或者任何评论者)、一个记录员作记录。典型的检查对象是一个文件，例如需求说明或者测试计划，

目的在于发现问题和查找遗漏，而不是去对任何东西进行实际的修改。会议的参加者应当有准备，应当通读文件，大多数的问题会在准备的过程中被发现。检查会的结果应写成书面报告。对检查会进行全面准备是困难而艰苦的工作，但它是保证质量最有用的方法。在检查过程中，最有经验的雇员的作用就向‘大哥哥’一样，他们的技能也许不大显眼，但对任何软件开发机构是最重要的，这是因为预防错误要比发现错误在费用方面更加有效。

1.1.57什么是兼容性测试？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。

1.1.58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。

1.1.59测试的策略有哪些？

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）

1.1.60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。

1.1.61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。

1.1.62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。

1.1.63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；

1.1.64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

1.1.65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。

1.1.66QTP 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

1.1.67TestDirector 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。

1.1.68你所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）？

Compatibility Testing（兼容性测试），也称“Configuration testing（配置测试）”，测试软件是否和系统的其它与之交互的元素之间兼容，如：浏览器、操作系统、硬件等。验证测试对象在不同的软件和硬件配置中的运行情况。

Functional testing (功能测试)，也称为 behavioral testing（行为测试），根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。

Performance testing（性能测试），评价一个产品或组件与性能需求是否符合的测试。包括负载测试、强度测试、数据库容量测试、基准测试等类型。

1.1.69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？

参考答案：5C 标准

1.1.70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

1.1.71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。

1.1.72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

1.1.73阐述工作版本的定义？

参考答案：

构造号： BUILD

1.1.74什么是桩模块？什么是驱动模块？

参考答案：

桩模块：被测模块调用模块

驱动模块 调用被测模块

1.1.75什么是扇入？什么是扇出？

参考答案：

扇入：被调次数，扇出：调其它模块数目

1.1.76你认为做好测试计划工作的关键是什么？

参考答案：

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。

1.1.77你觉得对于测试有哪些基本素质要求

（1）细心，只有细心才能保证不遗漏测试点并及时发现问题。

（2）善于怀疑，在测试的过程中总会遇到开发会说这个功能肯定没有问题，这个时候就要小心开发给你埋下的坑了。

（3）要有追根究底的精神，我们有的时候发现一些不好复现的 bug，对于这样的 bug 一定要有不找出问题不罢休的精神。

（4）考虑问题要周到，需要测试结合需求业务流程，和不同手机的兼容等多个方面来考虑问题。

（5）要有良好的沟通能力，不要让开发说服你这个问题修补修改，如果你认为这个问题比较严重需要说服开发来修改他/她认为不用修改的问题。

1.1.78一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

1.1.79软件测试的流程是什么？

需求调查: 全面了解您的系统概况、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、时间和工作量估计及项目报价。

制定初步的项目计划: 在与您充分共同和协商的基础上制定我们的测试计划。

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及测试脚本的开发等。

测试实施: 按照测试计划进行实施测试。

测试评估: 根据测试的结果，出具测试评估报告。

1.1.80说说你对 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，必要是可以要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；

1.1.81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

1.1.82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。

1.1.83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。

1.1.84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

1.1.85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

1.1.86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。

1.1.87单元测试主要内容是什么？

参考答案：

单元测试大多数由开发人员来完成，测试人员技术背景较好或者开发系统软件时可能会安排测试人员进行单元测试，大多数进行的单元测试都是开发人员调试程序或者开发组系统联合调试的过程。讨论这个问题主要是扩充一下读者的视野。

单元测试一般包括五个方面的测试：

（1）模块接口测试：模块接口测试是单元测试的基础。只有在数据能正确流入、流出模块的前提下，其他测试才有意义。模块接口测试也是集成测试的重点，这里进行的测试主要是为后面打好基础。测试接口正确与否应该考虑下列因素：

-输入的实际参数与形式参数的个数是否相同；

-输入的实际参数与形式参数的属性是否匹配；

-输入的实际参数与形式参数的量纲是否一致；

-调用其他模块时所给实际参数的个数是否与被调模块的形参个数相同；

-调用其他模块时所给实际参数的属性是否与被调模块的形参属性匹配；

-调用其他模块时所给实际参数的量纲是否与被调模块的形参量纲一致；

-调用预定义函数时所用参数的个数、属性和次序是否正确；

-是否存在与当前入口点无关的参数引用；

-是否修改了只读型参数；

-对全程变量的定义各模块是否一致；

-是否把某些约束作为参数传递。

如果模块功能包括外部输入输出，还应该考虑下列因素：

-文件属性是否正确；

-OPEN/CLOSE 语句是否正确；

-格式说明与输入输出语句是否匹配；

-缓冲区大小与记录长度是否匹配；

-文件使用前是否已经打开；

-是否处理了文件尾；

-是否处理了输入/输出错误；

-输出信息中是否有文字性错误。

-局部数据结构测试；

-边界条件测试；

-模块中所有独立执行通路测试；

（2）局部数据结构测试：检查局部数据结构是为了保证临时存储在模块内的数据在程序执行过程中完整、正确，局部功能是整个功能运行的基础。重点是一些函数是否正确执行，内部是否运行正确。局部数据结构往往是错误的根源，应仔细设计测试用例，力求发现下面几类错误：

-不合适或不相容的类型说明；

-变量无初值；

-变量初始化或省缺值有错；

-不正确的变量名（拼错或不正确地截断）；

-出现上溢、下溢和地址异常。

（3）边界条件测试：边界条件测试是单元测试中最重要的一项任务。众所周知，软件经常在边界上失效，采用边界值分析技术，针对边界值及其左、右设计测试用例，很有可能发现新的错误。边界条件测试是一项基础测试，也是后面系统测试中的功能测试的重点，边界测试执行的较好，可以大大提高程序健壮性。

（4）模块中所有独立路径测试：在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条语句至少执行一次。测试目的主要是为了发现因错误计算、不正确的比较和不适当的控制流造成的错误。具体做法就是程序员逐条调试语句。常见的错误包括：

-误解或用错了算符优先级；

-混合类型运算；

-变量初值错；

-精度不够；

-表达式符号错。

比较判断与控制流常常紧密相关，测试时注意下列错误：

-不同数据类型的对象之间进行比较；

-错误地使用逻辑运算符或优先级；

-因计算机表示的局限性，期望理论上相等而实际上不相等的两个量相等；

-比较运算或变量出错；

-循环终止条件或不可能出现；

-迭代发散时不能退出；

-错误地修改了循环变量。

模块的各条错误处理通路测试：程序在遇到异常情况时不应该退出，好的程序应能预见各种出错条件，并预设各种出错处理通路。如果用户不按照正常操作，程序就退出或者停止工作，实际上也是一种缺陷，因此单元测试要测试各种错误处理路径。一般这种测试着重检查下列问题：

-输出的出错信息难以理解；

-记录的错误与实际遇到的错误不相符；

-在程序自定义的出错处理段运行之前，系统已介入；

-异常处理不当；

-错误陈述中未能提供足够的定位出错信息。

1.1.88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

1.1.89如何理解压力、负载、性能测试测试？

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。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。

1.1.90什么是系统瓶颈？

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。

1.1.91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。

1.1.92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。

1.1.93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。

1.1.94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。

1.1.95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。

1.1.96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。

1.1.97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。

1.1.98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。

1.1.99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。

1.1.100 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。

1.1.101 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。

1.1.102 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。

1.1.103 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。

1.1.104 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。

1.1.105 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。

1.1.106 和用户共同测试（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。

1.1.107 如何编写提交给用户的测试报告？

参考答案：

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。

1.1.108 测试工具在测试工作中是什么地位？

参考答案：

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。

1.1.109 什么是软件测试，软件测试的目的？

参考答案：

1.1.110 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。

1.1.111 写出 bug 报告流转的步骤，每步的责任人及主要完成的工作。

参考答案：（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。

1.1.112 写出 bug 报告当中一些必备的内容。

参考答案：

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）。

1.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。

2.Bug 报告优先级

3.Bug 状态

4.Bug 的编号

5.发现人

6.提交人

7.指定处理人

8.概述

9.从属关系

10.详细描述

11.严重程度

12.所属模块

13.附件

14.提交日期

1.1.113 开发人员老是犯一些低级错误怎么解决？

参考答案：

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。

1.1.114 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。

1.1.115 测试在开发阶段的作用

1) 项目规划阶段：负责从单元测试到系统测试的整个测试阶段的监控。

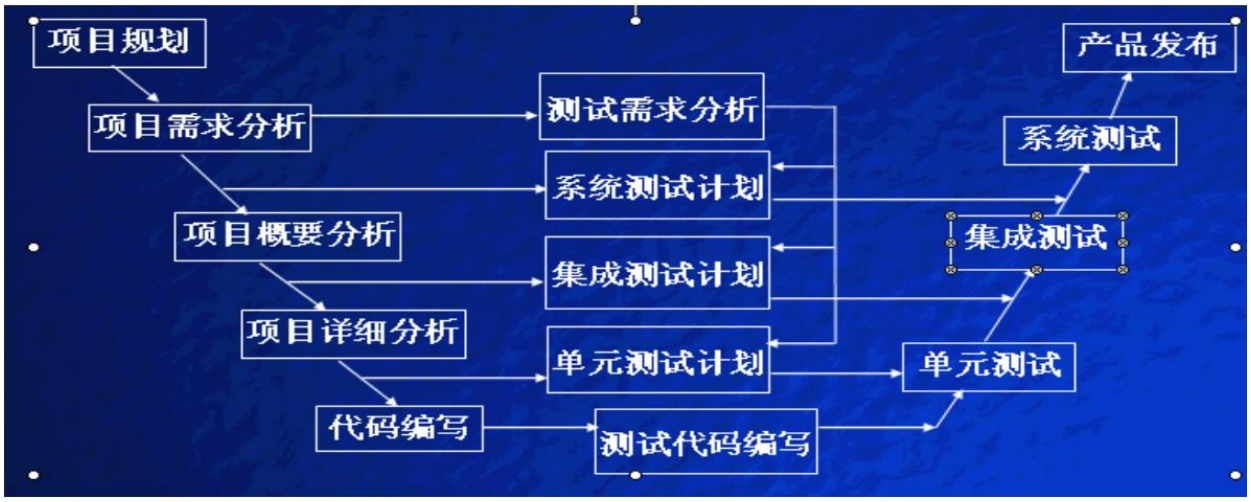
2) 需求分析阶段：确定测试需求分析、系统测试计划的制定，评审后成为管理项目。

3) 详细设计和概要设计阶段：确保集成测试计划和单元测试计划完成。

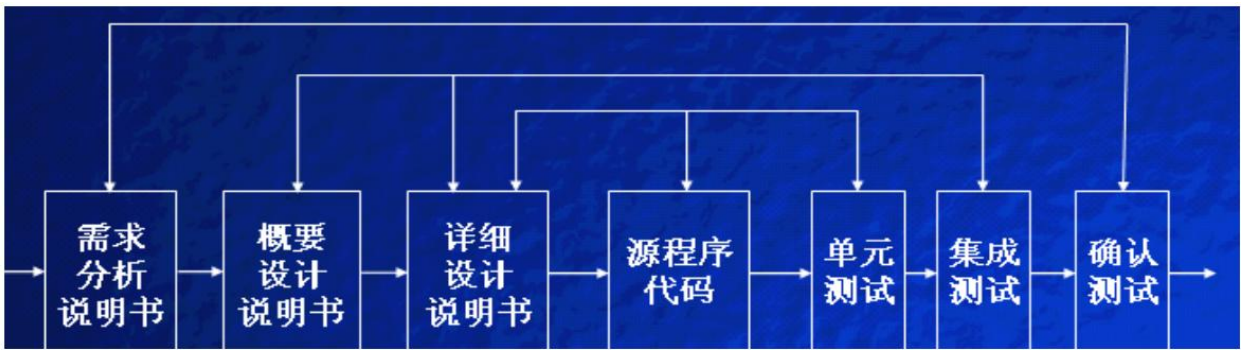
4) 编码阶段：由开发人员进行自己负责部分的测试代码。在项目较大时，由专人进行编码阶段的测试任务。

5)测试阶段（单元、集成、系统测试）：依据测试代码进行测试，并提交相应的测试状态报告和测试结束报告。

1.1.116 一个完整的开发流程是什么样的？



1.1.117 测试与开发各个阶段的关系



1.1.118 在软件开发过程中 5 个常见的问题是什么?

需求说明差 不切实际的时间表 测试不充分 不断增加功能 交流问题

（1）需求说明差 (poor requirements) — 需求不清楚、不完整、太概括、或者不可测试，都会造成问题。

（2）不切实际的时间表 (unrealistic schedule) — 如果在很短的时间里要求做许多事，出现错误是不可避免的。

（3）测试不充分 (inadequate testing)— 只能根据客户意见或者系统崩溃来判断系统质量的高低。

（4）不断增加功能 (featuritis) — 在开发正在进行过程中要求增加许多新的功能。这是常见的问题。

（5）交流问题 (miscommunication) — 如果开发人员对客户的要求不了解，或者客户由不恰当的期望，必然会导致错误。

1.1.119 针对软件开发过程中的问题，有哪些解决方法?

（1）可靠的需求 (solid requirements) —— 应当有一个经各方一致同意的、清楚的、完整的、详细的、整体的、可实现的、可测试的需求。为帮助确定需求，可使用模型(prototypes)。

（2）合理的时间表（realistic schedules）—— 为计划、设计、测试、改错、再测试、变更、以及编制文档留出足够的时间。不应使用突击的办法来完成项目。

（3）适当的测试 (adequate testing) —— 尽早开始测试；每次改错或变更之后，都应重新测试。项目计划中要为测试和改错留出足够的时间。

（4）尽可能坚持最初的需求 (stick to initial requirements as much as possible) —— 一旦开发工作开始，

要准备防止修改需求和新增功能。要说明这样作的后果。如果必须进行变更，必须在时间表上有相应的反映。如果可能，在设计阶段使用快速的模型，以便使客户了解将会得到的东西。这将会使他们对他们的需求有较高的信心，减少以后的变更。

（5）沟通 (communication) —— 在适当时机进行预排和检查；充分利用团组通信工具 —— 电子邮件、群件 (groupware)、网络故障跟踪工具、变更管理工具、以及因特网的功能。要确保文件是可用的和最新的。优选电子版文档，避免纸介质文档；进行远距离联合作业及协作；尽早使用模型，使得客户的预想是清楚的。

1.1.120 阐述软件生命周期都有哪些阶段？常见的软件生命周期模型有哪些?

软件生命周期是指一个计算机软件从功能确定、设计，到开发成功投入使用，并在使用中不断地修改、增补和完善，直到停止该软件的使用的全过程（从酝酿到废弃的过程）

生命周期从收到应用软件开始算起，到该软件不再使用为止。它有如下各方面的内容： 初始构思、需求分析、功能设计、内部设计、文档计划、测试计划、文档准备、集成、测 试、维护、升级、再测试、逐步淘汰 (phase-out)、等等

瀑布模型，迭代式模型，快速原型模型，螺旋模型

1.1.121 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

1.1.122 你认为做好测试用例工作的关键是什么？

参考答案：

需求和设计文档的理解程度，对系统的熟悉程度

1.1.123 简述一下缺陷的生命周期？

参考答案：提交->确认->分配->修复->验证->关闭

1.1.124 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

1.1.125 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：

（1）配置项标识

（2）配置项控制

（3）配置项状态报告

（4）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

1.1.126 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

1.1.127 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

1.1.128 什么是版本控制，常用的版本控制系统有哪些？

版本控制（Revision control）是一种软体工程技巧，籍以在开发的过程中，确保由不同人所编辑的同一档案都得到更新。

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。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。https://git-scm.com/doc

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。https://tortoisesvn.net/support.html

1.1.129 简述软件测试与软件开发之间的关系？

（1）项目规划阶段：负责从单元测试到系统测试的整个测试阶段的监控。

（2）需求分析阶段：确定测试需求分析、系统测试计划的制定，评审后成为管理项目。测试需求分析是对产品生命周期中测试所需求的资源、配置、每阶段评判通过的规约；系统测试计划则是依据软件的需求规格说明书，制定测试计划和设计相应的测试用例。

（3）详细设计和概要设计阶段：确保集成测试计划和单元测试计划完成。

（4）编码阶段：由开发人员进行自己负责部分的代码的测试。在项目较大时，由专人进行编码阶段的测试任务。

（5）测试阶段（单元、集成、系统测试）：依据测试代码进行测试，并提交相应的测试状态报告和测试结束报告。

开发和测试是一个有机的整体！在产品的发布之前，开发和测试是循环进行的， 测出的缺陷要经开发人员修改后继续测试。在开发的同时测试经理开始编写测试用例，测 试文档要参考开发文档，所以开发和测试是不可分割的，少了任何一个都不能开发出产品。

从角色方面看，像理论和实验的关系，开发人员通过自己的想象创造出一套思想，

之　后测试人员再对它进行检验、证伪，开发人员再修改的过程从而不断丰富产品。从方法方面看，是演绎和归纳的关系，一个要掌握大量的技术，一个要不断的从实例中学习。因这两方面的不同，所以开发和测试看上去做的工作很不一样。

开发与测试是相辅相承、密不可分的，开发人员开发出新的产品后要通过测试判断产品是否完全满足用户的需求。如果发现缺陷，提交给开发人员进行修复，然后再转交测试人员进行回归测试，直到产品符合需求规格说明。一个符合用户需求的产品是开发和测试共同努力的成果。

1.1.130　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。

1.1.131　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

1.1.132　您所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）

测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。是把测试对象看作一个黑盒子。利用黑盒测试法进行动态测试时，需要测试软件产品的功能，不需测试软件产品的内部结构和处理过程。采用黑盒技术设计测试用例的方法有：等价类划分、边界值分析、错误推测、因果图和综合策略。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种

工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范（快捷键之类的），是否美观（能否吸引用户的注意力），是否安全（尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告）？做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

1.1.133 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

1.1.134 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。

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。

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。

软件的黑盒测试意味着测试要在软件的接口处进行。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？

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。

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。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

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。

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。

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。

1.1.135 测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要的？

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）

1.1.136 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.

1.1.137 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。

1.1.138 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

1.1.139　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。

参考答案：（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；

1.1.140 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。

刚开始进入测试行业时，对测试的认识是从无忧测试网上了解到的一些资料，当时是冲着做测试需要很多技能才能做的好，虽然入门容易，但做好很难，比开发更难，虽然当时我很想做开发（学校专业课我基本上不缺席，因为我喜欢我的专业），但看到测试比开发更难更有挑战性，想做好测试的意志就更坚定了。

不到一年半的测试工作中，当时的感动和热情没有减退一点（即使环境问题以及自身经验，技术的不足，做测试的你一定也能理解）。

我觉得做测试整个过程中有 2 点让我觉得很有难度（对我来说，有难度的东西我就非常感兴趣），第一是测试用例的设计，因为测试的精华就在测试用例的设计上了，要在版本出来之前，把用例写好，用什么测试方法写？（也就是测试计划或测试策略），如果你刚测试一个新任务时，你得花一定的时间去消化业务需求和技术基础，业务需求很好理解（多和产品经理和开发人员沟通就能达到目的），而技术基础可就没那么简单了，这需要你自觉的学习能力，比如说网站吧，最基本的技术知识你要知道网站内部是怎么运作的的，后台是怎么响应用户请求的？测试环境如何搭建？这些都需要最早的学好。至少在开始测试之前能做好基本的准备，可能会遇到什么难题？需求细节是不是没有确定好？这

些问题都能在设计用例的时候发现。

第二是发现 BUG 的时候了，这应该是测试人员最基本的任务了，一般按测试用例开始测试就能发现大部分的 bug，还有一部分 bug 需要测试的过程中更了解所测版本的情况获得更多信息，补充测试用例，测试出 bug。还有如何发现 bug？这就需要在测试用例有效的情况下，通过细心和耐心去发现 bug 了，每个用例都有可能发现 bug，每个地方都有可能出错，所以测试过程中思维要清晰（测试过程数据流及结果都得看仔细了，bug 都在里面发现的）。如何描述 bug 也很有讲究，bug 在什么情况下会产生，如果条件变化一点点，就不会有这个 bug，以哪些最少的操作步骤就能重现这个 bug，这个 bug 产生的规律是什么？如果你够厉害的话，可以帮开发人员初步定位问题。

1.1.141 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）。

1.1.142 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。

1.1.143 测试总结报告包括那些项

测试用例的通过数，测试用例的未通过数，以及测试用例的通过率，未通过的功能都集中在哪几个功能模块 ，根据测试经验以及测试结果进行一个缺陷的分析和建议。

1.1.144 测试工作进行到一半是，发现时间不够，你如何处理

1.与客户沟通本次发布的版本什么是最重要的，什么是其次，我会安排一个优先级来对整体测 试功能进行一个筛选。

2.我会和测试组原体人员一起加班

1.1.145 开发与测试的关系

开发和测试是一个整体，也可以说测试驱动着开发，开发配合着测试，相辅相成的，在一个完整的项目组中缺一不可。

1.1.146 如果你是测试组长你如何对项目及组员进行管理

首先要从需求开始，充分了解被测系统的功能以及业务需求，并在遇到问题的时候及时有效的与开发人员以及其他项目相关人员进行沟通，做到最被测系统的十分熟悉。并了解整个测试组的成员他们的测试技能以及擅长的工作，做到测试任务的合理分配，得以让测试工作快速，稳定高效的进行！

1.1.147 缺陷报告严重级别的划分

严重级别的错误：影响系统整体基本流程运行的错误，由于某一操作造成系统死循环或服务器崩溃的错误

较严重：功能实现错误、内部计算错误、

一般：UI 错误，一些易用性的错误或建

1.1.148 开发人员修复缺陷后，如何保证不影响其他功能

Bug 的修复以及新功能的添加都有可能对版本造成一些影响，为了避免，在新版本发布以后，首先会对新版本做一个基础的流程测试也叫做冒烟测试，如果测试基本流程都顺利通过没有任何问题，那么测试人员可以继续进行详细的测试，否 则就将冒烟测试中出现 的问题以及问题有可能出现的原因反馈给开发人员，由开发人员修正后再次发版，进行测试。这是一个迭代的过程。

1.1.149 发现问题后你是如何判断其是否是 BUG，你是如何提交的、

测试用例是经过评审组严格的评审，完全按照客户的需求规格说明书作为最终依据来评审的，如果测试过程中，测试结果与实际结果不符就很可能是 Bug，如果一些比较明显的问题就直接录入缺陷管理系统，如果是一些边界问题不容易确定的，可以通过和开发人员甚至是设计人员等进行沟通最后得出一个结果究竟是否是 Bug，如果是 Bug 就录入，如果是一个需要增加的新功能等，可以录入缺陷管理系统，类型为新需求。

1.1.150 修复一个 BUG 而导致其他的 BUG 出现，该如何处理

帮助开发人员分析问题锁定原因然后进行新 Bug 的修正。

1.1.151 缺陷处理流程

1.讲缺陷的详细信息录入缺陷管理系统，并分配给对应的开发人员

2.如果遇到一些难以再现的缺陷，在开发人员修正过程中配合开发人员进行 Bug 的再现。

3.开发人员修正 Bug 后，会在缺陷管理系统中将修正后的 Bug 状态更改，通常为 Fixed 状态。

4.新版本发布后，测试人员会讲 bug 状态已经更改为 Fixed 的 Bug 进行回归测试。如果测试通过，则将该 Bug 关闭，如果仍未通过，则将该 Bug 从 Fixed 更改为 Reopen 状态，继续让开 发人员来修正。并等待下一个新版本发布后的二次回归测试。

1.1.152 缺陷报告包括那些项

编写人、被测系统的版本号、测试环境、预期结果、实际结果、对于实际结果如有必要附上截图、测试用例数、测试用例通过 数，测试用例的通过率、对缺陷的一个分析汇总。

1.1.153 介绍一下整体项目流程

1. 搭建缺陷管理的环境和测试环境以及配置管理的环境搭建；
2. 编写测试计划；
3. 设计测试用例；
4. 编写测试用例；
5. 测试用例的评审；
6. 执行测试；
7. 缺陷管理；
8. 测试报告的输出

1.1.154 在实际项目中你是如何做测试计划

1.对客户提供的或需求分析人员编写的用户需求文档或需求规格说明书进行分析，提炼出测试要点；

2.根据测试要点编写测试用例。

3.由评审组对测试用例进行评审--修改--再次评审--初步定稿

4.执行测试

4.1 按照测试用例对系统进行功能验证及客户的需求验证

4.2 将测试过程中产生的 Bug 录入缺陷管理系统

4.3 新版本发布后，对本次版本新增加的功能以及开发人员修正的 Bug 进行回归测试

4.4 根据项目需要提交测试报告。

1.1.155 你是如何制定测试过程中的时间进度表的

根据项目的需求、开发周期、开发人员的开发进度等时间安排来制定一个测试时间进度初 稿，并

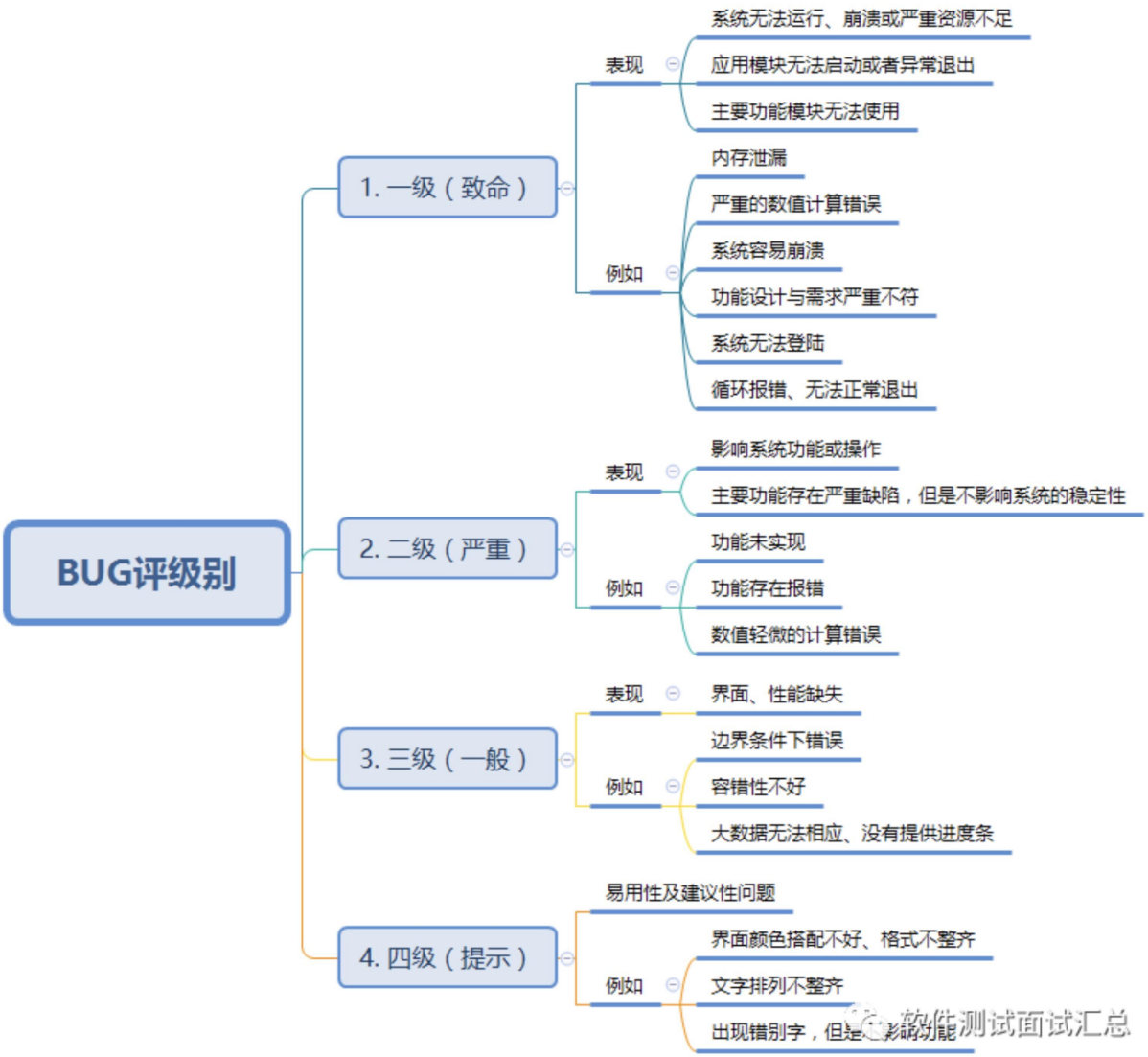
将测试时间进度表交与整个项目团队成员大家一起讨论和分析，最终和所有人达成共识制定出一个大家都可以执行的测试时间进度表。

时间表中包括了开发人员提交功能或功能模块的时间，以及为了更好的执行测试，配合测试人员进行功能培训的时间，以及测试执行时间等，都详细的写到 WBS 中，并按照这个时间进度表来执行项目的测试任务。

1.1.156 测试计划都包括那些项

1. 测试计划目标
2. 测试参考文档
3. 测试术语与定义
4. 测试内容
5. 测试人员的分工
6. 测试进度
7. 测试流程
8. 测试工具
9. 测试缺陷管理
10. 测试的风险分析

1.1.157 bug 有哪些等级？



1.1.158 说说你对软件配置管理的理解。

项目在开发的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能及风险的水平。软件的规模越大，配置管理就显得越重要。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，并且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 等，偶只用过 CVS，对其它的不熟悉

1.1.159 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），并利用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、语言有效性及外观排布.。

1.1.160 QA 和 QC 的区别是什么？

质量保证（QA）:是指确保产品符合预定质量要求而作出的所有有组织、有计划活动的总和。

质量控制（QC）：即实验室控制系统，它涉及取样、质量标准、检验、产品批准放行程序等方面内容。

一般是 QA 为领导地位。

QA：主要是事先的质量保证类活动，以预防为主，期望降低错误的发生几率。

QC：主要是事后的质量检验类活动为主，默认错误是允许的，期望发现并选出错误。

QA 是为满足顾客要求提供信任，即使顾客确信你提供的产品能满足他的要求，因此需从市场调查开始及以后的评审客户要求、产品开发、接单及物料采购、进料检验、生产过程控制及出货、售后服务等各阶段留下证据，证实工厂每一步活动都是按客户要求进行的。

QC 是为使产品满足质量要求所采取的作业技术和活动，它包括检验、纠正和反馈，比如 QC 进行检验发现不良品后将其剔除，然后将不良信息反馈给相关部门采取改善措施。

用通俗的话来说，QA 比作起草法律的法官，QC 比作警察。

QA 比作产品经理，QC 比作测试人员

1.1.161 软件测试的目的是什么？

软件测试的定义：为了发现程序中的错误而执行程序的过程

测试的目的：

1.发现程序员在开发中存在的代码以及逻辑错误

2.审核产品的完成是否符合用户需求

3.提高用户体验

4.交付更高质量的产品

1.1.162 如何定义所提交 bug 的严重等级和优先等级的？

Bug 有四种级别，分别为：致命的（Fatal），严重的（Critical），一般的（Major），微小的（Minor）。

A 类-致命的（Fatal）：

造成系统或应用程序崩溃、死机、系统挂起，或造成数据丢失，主要功能完全丧失，导致本模块以及相关模块异常等问题。如代码错误，死循环，数据库发生死锁、与数据库连接错误或数据通讯错误，未考虑异常操作，功能错误等

B 类-严重错误（critical）：系统的主要功能部分丧失、数据不能保存，系统的次要功能完全丧失。问题局限在本模块，导致模块功能失效或异常退出。如致命的错误声明，程序接口错误，数据库的表、业务规则、缺省值未加完整性等约束条件

C 类-一般错误（major）：次要功能没有完全实现但不影响使用。如提示信息不太准确，或用户界面差，操作时间长，模块功能部分失效等，

打印内容、格式错误，删除操作未给出提示，数据库表中有过多的空字段等

D 类-较小错误（Minor），使操作者不方便或遇到麻烦，但它不影响功能过的操作和执行，如错别字、界面不规范（字体大小不统一，文字排列不整齐，可输入区域和只读区域没有明显的区分标志），辅助说明描述不清楚。

常用的缺陷的优先级表示方法可分为：立即解决 P1、高优先级 P2、正常排队 P3、低优先级 P4.立即解决是指缺陷导致系统几乎不能使用或者测试不能继续，需立即修复；高优先级是指缺陷严重影响测试，需要优先考虑；正常排队是指缺陷需要正常排队等待修复；而低优先级是指缺陷可以在开发人员有时间的时候再被纠正。

1.1.163 Web 和 APP 测试的异同有哪些？

单纯从**功能测试的层面**上来讲的话，APP 测试、web 测试 在流程和功能测试上是没有区别的。

根据两者载体不一样，则区别如下：

系统结构方面

web 项目，b/s 架构，基于浏览器的；web 测试只要更新了服务器端，客户端就会同步会更新。

app 项，c/s 结构的，必须要有客户端；app 修改了服务端，则客户端用户所有核心版本都需要进行一遍。

性能方面

web 项目 需监测 响应时间、CPU、Memory

app 项目 除了监测 响应时间、CPU、Memory 外，还需监测 流量、电量等

兼容方面

web 项目：

1. 浏览器（火狐、谷歌、IE 等）

2. 操作系统（Windows7、Windows10、Linux 等）

app 项目：

1. 设备系统: iOS（ipad、iphone）、Android（三星、华为、联想等） 、Windows（Win7、Win8）、OSX （Mac）

2. 手机设备可根据 手机型号、分辨率不同

相对于 web 项目，APP 有专项测试

1. 干扰测试：中断，来电，短信，关机，重启等

2. 弱网络测试（模拟 2g、3g、4g，wifi 网络状态以及丢包情况）；网络切换测试（网络断开后重连、3g 切换到 4g/wifi 等）

3. 安装、更新、卸载

安装：需考虑安装时的中断、弱网、安装后删除安装文件等情况

1.1.164 怎么理解回归测试？是否思考过如何减少回归测试工作量？

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

回归测试作为软件生命周期的一个组成部分，在整个软件测试过程中占有很大的工作量比重,自动

回归测试将大幅降低系统测试、维护升级等阶段的成本。

1.1.165 一条软件缺陷（或 BUG）包括哪些内容？请完整列出

1.和 BUG 对应的软件版本

2.开发的借口人员，测试人员

3.BUG 的优先级

4.BUG 的严重程度

5.BUG 可能属于的模块

6.BUG 的标题

7.BUG 的描述

8.BUG 的截图

9.BUG 的状态

10.BUG 的错误类型（数据，界面。。。。）

1.1.166 软件测试方法有哪些分类？

软件测试方法分类：白盒、黑盒、灰盒；单元测试、集成测试、系统测试、验收测试、回归测试、Alpha 测试、Beta 测试；静态测试和动态测试。

1.1.167 设计测试用例的主要方法有哪些？

设计测试用例的主要方法有：等价类划分；边界值分析法；因果图法；场景法。

1.1.168 单元测试、集成测试、系统测试的侧重点是什么？

单元测试的重点是系统的模块，包括子程序的正确性验证等。

集成测试的重点是模块间的衔接以及参数的传递等。

系统测试的重点是整个系统的运行以及与其他软件的兼容性。

1.1.169 怎样才能成为一个优秀的测试工程师

软件测试工程师需要具有一些逆向思维的能力，想别人所不想，测别人所不测，这样才可以找到更多的软件中的错误。这是作为一名优秀的软件测试工程师最基本的素质。善于同软件开发人员沟通沟 通是当今软件项目中需要掌握的最关键技术之一。软件测试人员要善于同软件开发人员沟通，软件测试人员与开发人员搞好关系，使测试人员不成为开发人员的眼中钉，这对于提高整个软件项目质量是十分重要的。沟通主要包括：讨论软件的需求，设计：通过这样的沟通，你可以更好的了解所测试的软件系统，以至于尽可能少的测试出软件中不是错误的“错误”，从而降低给软件开发人员带来的压力。报告好的测试结果：作为一个测试人员，发现错误往往是测试人员最愿意而且引以自豪的结果，但是一味地给开发人员报告软件错误，会给他们造成厌恶感，降低整个软件的质量和开发进度。所以作为一名软件测试工程师，当你测试的模块没有严重的错误或者错误很少的时候，你不妨跑到开发人员那里告诉他们这个好消息，这会给你带来意想不到的结果。

讨论一些与工作无关的事情：作为一个测试人员经常和开发人员讨论一些与工作无关的事情，比如大家可以谈谈新闻，趣事，家庭...这样可以加强相互间的默契程度，许多统计表明，这样可以更好的提高软件工作质量。善于同领导沟通 测试人员往往是领导的眼和耳，领导根据测试人员的测试结果可以了解公司的产品质量，从而调整其他的工作。领导工作一般比较繁忙，所以作为一名优秀的测试人员要学会把测试结果进行总结，最好以图表的形势给领导看。掌握一些自动化测试工具 测试工作往往是比较繁琐，枯燥无味的工作，测试人员长期处于重复的手工工作，会降低测试效率，并且对于测试质量也往往是不利的；况且许多测试不使用测试工具是不可以进行的，比如性能测试，压力测试等等。目前市场上有许多测试工具供你使用，你可以根据自己的需要选择一些测试工具来辅助你的测试。但是要记住一点，不是说有了测试工具就不要人工测试了，测试工具不是万能的。善于学习的能力 软件测试技术随着时间的变化也在做一些提高和改进，作为一名优秀的测试人员要善于利用书籍，网站，论坛，交流等各种途径不断提高自己的软件测试水平。提高自己的表达能力 软件测试人员当发现软件中存在缺陷的时候，往往要书写缺陷报告，缺陷报告要写得详尽清楚，使开发人员能够尽快定位错误，修改错误，所以作为一名优秀的测试人员提高自己的写作能力是非常必要的。了解业务知识 更好的了

解你说测试软件的业务知识是非常重要的，对业务知识了解得越深入，越能够找出更深入，更关键，更隐蔽的软件错误。所以作为一名优秀的软件测试工程师，要多向该领域专家，同行学习，提高自己的业务知识水平。

1.1.170 测试计划要安排哪些方面？

1. 引言：目的、背景、范围、定义、参考资

2. 测试内容：测试功能清单

3. 测试规则：进入准则，暂停/退出准则、测试方法、测试手段、测试要点、测试工具

4. 测试环境：硬件环境、软件环境、特定测试环境要求

5. 项目任务：测试规划，测试设计，测试执行准备，测试执行，测试总结

6. 实施计划：工作量估计、人员需求及安排、进度安排、其它资源需求及安排。

7. 风险管理

1.1.171 为什么要有测试报告？一份日常的测试报告通常需要说明哪些内容？

1、概述，包括本次测试的目的，测试的背景介绍；

2、测试环境，包括测试软硬件环境及配置，以及测试环境的网络拓扑图；

3、测试的一些参考资料；

4、测试参与人员，以及投入的时间情况说明；

5、测试的进度情况，包括计划进度和实际进度；

6、测试情况介绍，包括测试的内容项说明。如功能测试具体的测试项，测试通过情况；性能测试的测试项，测试通过情况等；

7、缺陷的统计和分析，包括迭代次数，缺陷的分布情况，缺陷的覆盖情况，缺陷的发展趋势等；

8、本次测试的结论；

9、测试人员就本次测试的一些建议。

1.1.172 在您参与或负责的项目测试中,发生过哪些棘手的问题,最后是如何解决的?您在这个过程中做了什么?

例子：迭代开发后期，开始对整个系统从头回归一遍，这时候又发现了许多以前从未出现的 BUG。这个时期大家都很烦躁困惑，曾经运转良好的页面，突然出现存储问题；曾经更新正常的功能，突然无法更新；曾经显示正常的 Excel，突然显示错误… …这些都让人苦恼，当然，这些应该都是正常现象。测试人员在测试后期尤其需要提高警惕，不能漏过任何一个功能点，更不能忽略任何一次貌似无用的查询、翻页、按键。

最后，是大家一起进行的交付测试，人员包括了所有的编程人员及测试人员。这期间，除了对基本功能的回归测试外，还包括了并发测试及性能测试（这主要是编程人员在做），除此之外，我将过去提交修正过的所有 BUG 重新验证了一遍。在并发测试中，我们发现了很多之前单人测试难以发现的并发问题（包括多人一起提交，一起选择，一起修改等等），并发问题可以说层出不穷，甚至包括了同一台电脑打开两个页面分别进行修改的问题（由于我从一开始就是打开两个页面来测，一个为用户本人，一个为该用户代理人 delegator，所以有些问题在早期已经暴露），这是测试中的一个重点，也是比较严重的漏洞，需要在以后多加留意。

1.1.173 在测试工作中,您常使用的测试方法有哪些?它们都是在什么场景下使用的?

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.

1.1.174 什么是测试用例,设计测试用例时,您常用的设计方法有哪些?应如何设计才能保证测试用例的覆盖率?

测试用例就是 测试人员 用以测试被测软件的某个特性或特性组合的一组数据。这组数据可能是从用户处得来的实际的一组数据，也可能是测试人员专门设计出来的测试软件某些功能的一组数据。

等价类划分法、边界值分析法、错误推测法、判定表法、正交实验法。

要保证测试用例能够全面覆盖测试需求，要包含所有的情况。测试用例设计上划分为单功能测试

用例和测试场景设计，单功能测试覆盖的需求中的功能点，测试场景覆盖需求中的业务逻辑。

在设计测试用例的时候，可以使用多种测试用例设计方法。

●首先进行等价类划分，包括输入条件和输出条件的等价类划分，合理设置有效等价类和无效等价类，这是减少工作量和提高测试效率最有效的方法。

●必须使用边界值分析，经验表明，这种方法设计出的用例能发现很多程序错误。

●可以使用错误推测法追加一些测试用例，这需要依靠您的智慧和经验。

●对照程序逻辑检查已设计出的测试用例的逻辑覆盖度，如果没有达到覆盖标准应当再补充足够的测试用例。

●如果程序的功能说明中含有输入条件的组合情况，一开始就可选因果图和判定表驱动法。

●对于参数配置类的软件，要用正交试验法选择较少的组合方式达到最佳效果。

●对于业务流清晰的系统，可以利用场景法贯穿整个测试方案过程，在案例中综合使用各种测试方法。

当测试用例设计完成后，要组织测试用例的评审，这样可以吸取别人的意见，减少遗漏，补全测试用例。

1.1.175 黑盒测试主要是为了发现那几类错误?

1）基于规格说明的功能错误

2）基于规格说明的构件或系统行为错误

3）基于规格说明的性能错误

4）面向用户的使用错误

5）黑盒接口错误

1.1.176 测试工作的流程?缺陷状态有什么?设计测试用例有几种方法?

测试工程师的实际工作流程（以 P2P 中型版本为例，一个月一个版本）：

1 产品经理或者 SR 把需求书发下来给开发和测试

2 测试先看一遍，进行需求分析。测试组长编写测试计划，并且分配测试任务给测试人员（2 天时间）

（此时开发也在进行需求分析）

3 过了 2 天，产品经理再把测试和开发召集在一起，进行需求讲解（或者说需求评审），有问题可以直接问，如果发现需求有问题，也可以提出来，SR 回去会修改。（需求讲解时间 0.5 天）

4 讲完需求后，测试同事要进行测试场景的梳理和案例的编写了（xmind 和 Excel 就要用上了），一共 5 个工作日。（此时开发在编写代码）

5 之后就要进行案例评审了，评审时候有 SR、测试同事、开发同事，评审时候一般 SR、测试组长、对应模块的开发同事会提出一点意见，评审完之后，回去修改、补充一下案例。（案例评审 0.5 天）

6 修改完以后，有两种处理情况：

6.1 对大项目有时候要进行案例的第二次评审。

6.2 对小项目，在时间紧的时候，一般不会二审，但是要以邮件的形式把修改或者新增后的案例发出来，给领导看，并抄送给其他同事。（案例评审 0.5 天，修改案例 0.5 天，案例二审 0.5 天）

7 案例评审完就要开始测试了，一般测试环境开发搭建好（要说自己也会搭建，搭建流程背老师总结的）：

7.1 中型版本的测试一般分 2 轮：第一轮：5 天；第二轮：3 天；回归测试 2 天；（共 10 个工作日）。

8 回归测试完后，达到了上线标准，就会如期上线，一般当天晚上 12 点上线

1.2 需求分析

1.2.1 需求人员需要何时参加需求分析？

如果条件循序 原则上来说 是越早介入需求分析越好 因为测试人员对需求理解越深刻 对测试工作的开展越有利 可以尽早的确定测试思路 减少与开发人员的交互 减少对需求理解上的偏差原则上，测试人员对需求了解得越深入对测试工作越有利，所以最好一开始就应该参加需求分析工作。这样做可以带来如下好处：

1) 测试人员全程参与需求分析，对需求了解得很深入，减少了很多与开发人员的交互，节省了时间。测试人员参与前期开发讨论，直接掌握了不清晰的需求点。

2) 早期确定测试用例的编写思路，为测试打好基础

3) 可以获取一些测试数据，为测试用例设计提供帮助

4) 可以发现需求不合理的地方，降低了测试成本。

5) 测试人员主要的工作之一就是确认系统是否正确实现了需求。

1.2.2 如果需求一直在变化怎么办？

这是一个常见的令人头疼的问题。

1) 如果可能，尽早与承担该项目风险的人接触，以便了解需求会怎样改变，从而可以尽早地改变测试计划和策略。

2) 如果在对应用程序进行初始设计时多考虑一些适应性，那么以后在发生需求的改变时，就不需要再为改变做很多事情了。

3) 好的代码注释和好的文档有助于开发人员作出相应的改变。

4) 只要有可能，就应使用快速原型 (rapid prototyping)，以帮助用户确认他们的需求，从而减少变更。

5) 在项目的时间表中应当留出余量，以应付可能出现的变更。

6) 尽量把新的需求纳入应用软件的“下一版”，而把原始需求作为“第一版”。

7) 通过谈判，把易于实现的新的变更列入项目，而把难于实现的新需求列入该应用软件的以后的版本。

8) 要确保让客户和管理人员了解变更对进度表的影响、所带来的风险、以及因变更所引起的大量资金消耗。

9) 在应付改变时，应在为建立自动测试而作的努力和重新进行测试所做的努力之间取得平衡。

10) 在设计自动测试剧本时，试图使其有一些灵活性。

11) 在对应用软件进行自动测试时，要把注意力集中在看来不大会改变的部分。

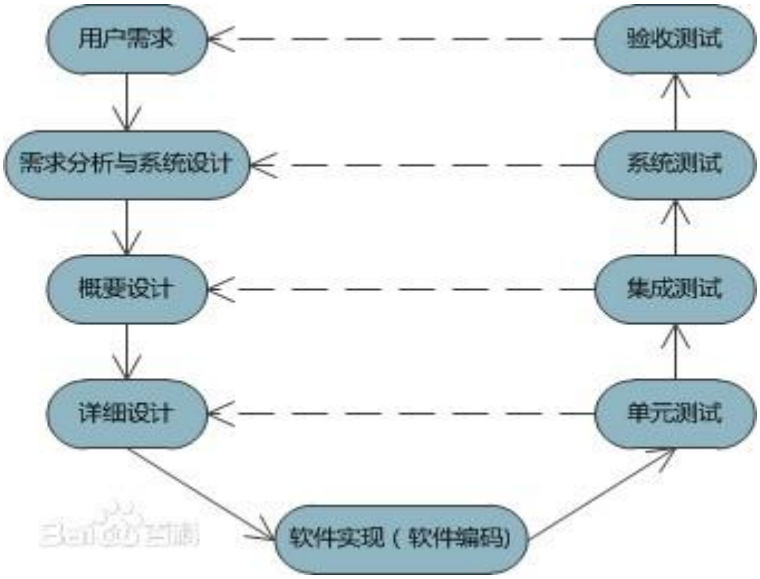
12) 对变更进行适当的风险分析，以减少回归测试的要求。

13) 在设计测试案例时要有一定的灵活性。做到这一点并不容易，所以要降低测试案例的详细程度，或者只建立高级的通用型的测试计划。

14) 少注意详细的测试计划和测试案例，要把重点放在专门的测试 (adhoc testing) 上。

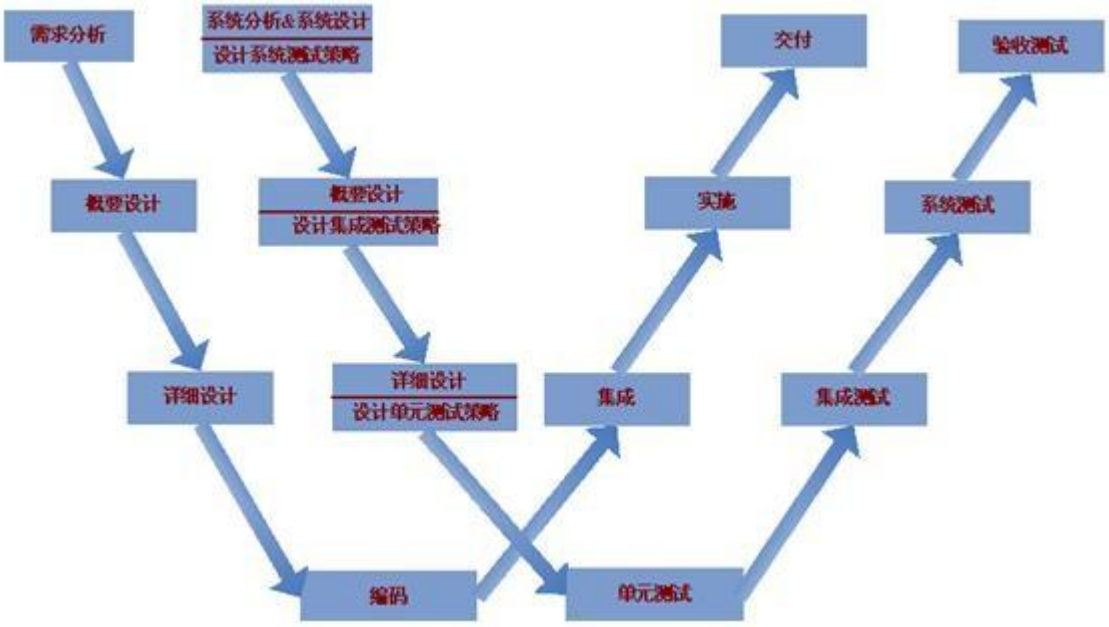
1.3 测试模型

1.3.1 常见测试模型有哪些？



特点：这是一种古老的瀑布模型，反映了实际和测试之间的关系

局限：仅仅把测试过程作为编码之后的一个阶段，忽视了测试对需求分析,系统设计的验证，如果前面设计错误，得一直到后期的验收测试才被发现，耗时耗力。



特点：测试与开发同时进行，在 V 模型的基础上，增加了在开发阶段的同步测试

局限：仍然不支持迭代，减少了一定错误发生率，但是需按照流水线进行设计、编码和测试

1.3.2 请根据”V”模型分别概述测试人员在软件的需求定义阶段、设计阶段、编码阶段、系统集成阶段的工作任务及其相应生成的文档？

系统统集成阶段的工作任务及其相应生成的文档？

需求定义阶段：根据项目需求提取测试需求 并形成测试需求文档，根据提取的测试需求和项目计划进行测试计划的拟定，测试计划文档

设计阶段：根据测试需求拟定测试方案并形成测试方案文档；根据测试方案制定测试用例，并形成测试用例文档

编码阶段：执行测试并完善测试用例文档

系统集成阶段：测试总结报告，阶段问题统计报告，测试问题报告

1.3.3 W 模型的描述？

软件测试主要内容是对软件正确性、完整性、安全性和质量确认及验证。为了验证这些，软件测试是与开发同步进行的，它们之间的关系同步进行一一对应，当开发进行需求分析、概要设计、详细设计、编码实现、模块集成、系统构建与实施、交付运行时，测试人员对应工作是需求测试、概要设计测试、详细设计测试、单元测试、集成测试、系统测试、验收测试。可以通过 W 模型[3]展示如图 2.3 所示：

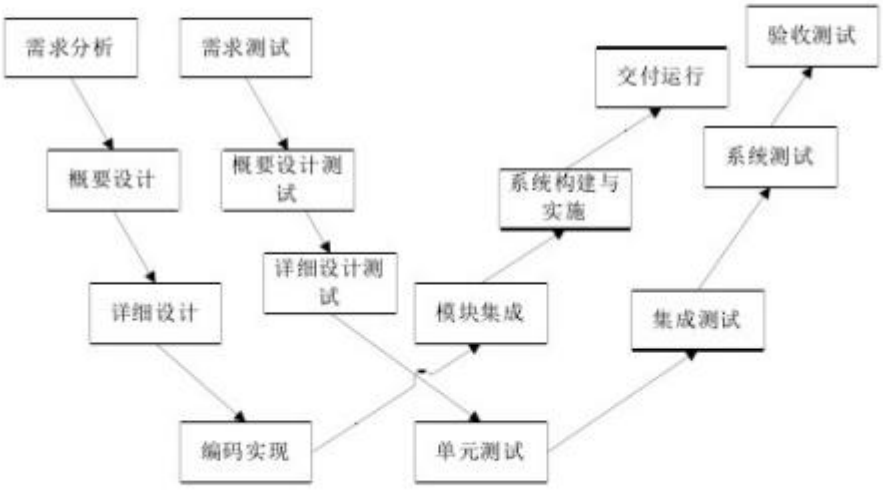
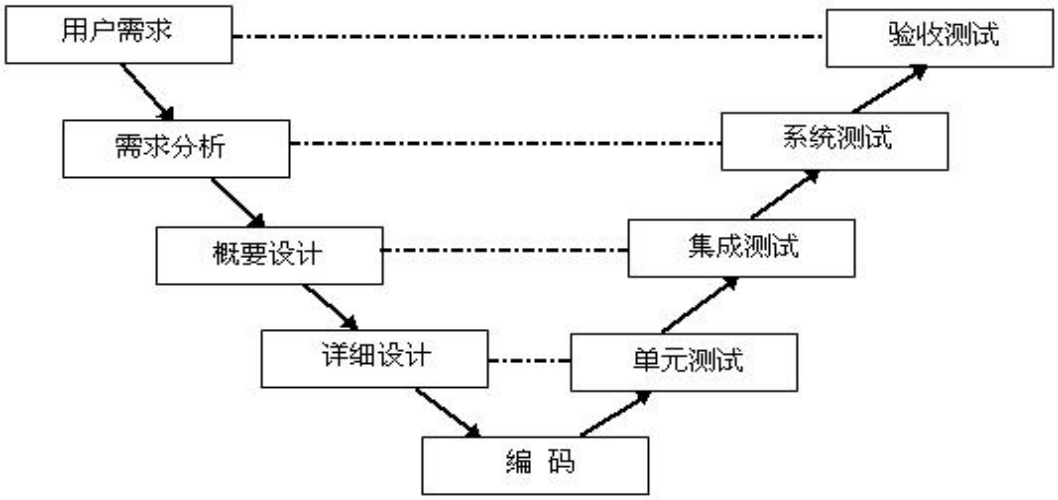


图 2.3 W 模型

图中显示 W 模型增加了软件开发各阶段中同步进行的验证和确认活动，测试的活动与软件开发整体是同步进行，测试的对象不仅仅是程序，还包括需求和设计，有利于尽早地全面的发现问题可降低软件开发和成本。

1.3.4画出软件测试的 V 模型图。



1.4 测试计划

1.4.1 测试计划工作的目的是什么？测试计划工作的内容都包括什么？其中哪些是最重要

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）

测试计划工作是对测试工作内容的一个有效的组织和规划，能保证测试工作有效的展开。测试计划工作包括测试目标，测试范围的定义，测试方法的选择，测试进度里程碑，测试资源的有效配置和管理。

测试计划工作也称为测试策略，主要描述测试工程的总体方法和目标，描述目前在进行那一阶段的测试（单元测试，集成测试，系统测试）以及每一阶段内进行的测试种类（功能测试，性能测试等）确定测试范围，生成测试数据等。

其中软件计划中的测试目标最重要，他的软件测试的所需要达成的最终结果。

1.4.2 测试计划编写的六要素？

why——为什么要进行这些测试

what—测试哪些方面，不同阶段的工作内容

when—测试不同阶段的起止时间

where—相应文档，缺陷的存放位置，测试环境等

who—项目有关人员组成，安排哪些测试人员进行测试

how—如何去做，使用哪些测试工具以及测试方法进行测试。

1.4.3 项目版本执行过程中，测试人员如何把控测试进度？

在项目的系统测试过程中，测试负责人要及时了解测试进度，跟踪 BUG 提交、修复及验证情况以及系统的拷机情况。

在开发初期阶段，测试组执行 BBFV 时，很多模块、功能点的开发完成进度和原计划会存在一定的偏差， 就需要测试负责人动态的刷新 WBS 计划，根据实际的开发进度调整测试计划。

在开发阶段，存在版本编译不出来导致无法测试，开发人员修复代码太随意导致版本稳定性反复，需求变更过大导致后端测试开发变更严重等现象，会导致测试工作无法正常进行。就需要测试负责人及时反馈出来，根据项目本身的特点进行对应的处理。

当测试进度出现延期时，要及时确认问题原因，如果是问题协查导致，则需及时与研发人员进行沟通协商，看问题是否必须在测试环境进行排查，若为必现问题可与研发协商要求其在自己环境进行排查，若必须占用测试环境，则需及时调整测试计划，若因此可能影响版本的发布，则应及时与 SE 确认。

若发现有较多 BUG 未解决，则应主动联系 SE 及研发人员召开 BUG 会确定问题的解决时间。若发现有较多 BUG 未验证，则应提醒项目组的测试人员及时进行验证，对于一些拷机或非必现的 BUG，建议测试人员在此 BUG 上现做拷机标记，连续拷机一周未再复现的做关闭处理，若再次复现则继续进行排查。

疑难问题的跟控：比较难复现的问题，怎么去尝试复现。比较难定位的问题，怎么驱动、反馈给 SE，协调开发人员定位问题。比较难处理的问题，怎么跟控反馈进度等

每天下班前需确认拷机内容，每天上班第一件事需确认拷机结果，只有这样才能保证拷机的效果，实现拷机的真正意义。

1.4.4 制定测试计划之前需要了解什么问题？

1.软件测试计划的目的是什么？是否所有人都知道？他们同意这个测试计划过程吗？

2.测试的是什么产品？是新程序还是维护升级的？是独立程序还是由多个小程序组成的？

3.产品的质量目标是什么？产品的功能需求和性能指标必须得到所有人的一致认可。

1.4.5 测试计划都包括哪些项？

测试计划的主体部分是应该包括：对时间的安排、人力物力的分配、总体的测试策略以及对风险的评估和相应的措施！还有项目的相关简介、测试范围、测试的参考文档和测试提交的文档、测试时间的安排、人力资源的分配、系统风险的评估和优先级的定义、缺陷严重级别标准以及在接下来测试工作中的编写测试用例和缺陷报告的模板！

1.4.6 怎样做好测试计划？

1.理解系统。从整个系统的高度了解被测系统必须满足的功能和非功能性需求。利用涉及整个系统的文档，形成对系统的整体了解。

2.及早介入。为了深入了解项目，测试人员应该在系统的开始阶段介入，可以增加对客户需求，客户问题，潜在风险，以及最重要的功能方面的理解

3.测试期望。程序员的期望是什么？客户的期望是什么？销售对测试的期望又是什么？测试目标必须是绝对的，以免说不清楚是否达到目标。

4.吸取教训。把以前工作中学习到的经验教训运用过来，对确定测试策略很有作用。

5.工作量大小。完成测试需要多少工作量？需要多少人员？

6.技术选择。系统会采取什么技术？系统会采用什么架构？这些信息有助于确定测试策略和测试工具。

7.时间表。系统开发和测试分配的时间有多长？截止日期是什么时候？

1.4.7 什么是测试资源

计划资源需求是确定测试策略必备条件的过程。在软件测试之前，要制定一个项目资源计划，包含每一个阶段的任务，所需要的资源，当发生类似到了使用期限或资源共享的事情时，要更新这个计划，在计划中，项目期间可能用到的任何资源都要考虑到，例

如：

1) 人员：人数，经验和专长，全职还是兼职。

2) 设备：计算机，测试硬件，测试工具。

3) 软件：应用程序，数据库程序和自定义工具。

4) 其它供应：软盘，电话，参考书，培训资料。

1.4.8 测试有哪些风险和问题

市场的压力

1) 测试时间不够

2) 测试资源的及时到位

3) 测试人员的技能需求

4) 开发进度的变化，需求的变更

5) 开发部门的版本控制

6) 短时间上线。这个是已经定好的，没有参考测试人员的意见。时间短往往不能得到充分的测试，测试策略必须根据可用的时间进行调整。尽快指出这样的问题非常重要，只有这样才能调整时间表，确定快速开发的风险并制定降低风险的策略。

7) 新的设计过程。引入新的设计过程会增加风险，新的设计过程包括新的工具和设计技术。如果采用新的技术，能否像我们预期的那样运转，都存在很大的风险

8) 复杂性。我们应该进行一些分析工作来确定哪个功能最复杂，哪个功能最容易出错，错误会对系统的哪些地方造成重大的影响。

9) 使用频率。软件最常用功能中隐藏的问题可能给用户造成严重的损失。

10) 不可测试的需求。不可测试的需求会对系统的成功造成巨大的威胁。如果测试组在需求阶段就验证了需求的可测试性，对需求进行了评审，那么此类问题会减少很多

1.5 测试策略

1.5.1 什么是“测试策略”？

测试策略描述测试工程的总体方法和目标 主要包括以下三个方面：

1 确定的测试技术和工具

2 制定测试启动 停止 完成标准

3 风险分析和应对方案

其目的 是为我们更好的写出高质量的用例 提供支撑

1.5.2 测试策略包括哪些？

测试策略包括:

1、要使用的测试技术和工具；

2、测试完成标准；

3、影响资源分配的特殊考虑例如测试与外部接口或者模拟物理损坏、安全性威胁。

1.5.3 系统测试的策略有哪些？

功能测试，性能测试，可靠性测试，负载测试，易用性测试，强度测试，安全测试， 配置测试，安装测试，卸载测试，文挡测试，故障恢复测试，界面测试，容量测试，兼容性测试，分布测试，可用性测试，

1.6 测试类型

1.6.1 请列出你所知道的软件测试种类，至少 5 项？

1.6.2 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？

黑盒测试：把测试对象当成一个黑盒子，测试人员完全不考虑逻辑结构和内部特性， 只依据程式的需求说明书来检查程序的功能是否满足它的功能说明。

白盒测试：把测试对象当成一个透明的盒子，允许测试人员利用程序内部逻辑结构及 相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。

单元测试：白盒测试的一种，对软件设计中的单元模块进行测试。

集成测试：在单元测试的基础上，对单元模块之间的连接和组装进行测试。系统测试：在所有都

考虑的情况下，对系统进行测试。

验收测试：第三方进行的确认软件满足需求的测试。

1.6.3 黑盒测试和白盒测试常用的测试方法有哪些，举个例子？

黑盒有等价类划分法，边界分析法，因果图法和错误猜测法。

白盒有逻辑覆盖法，循环测试路径选择，基本路径测试。

例子：在一次输入多个条件的完整性查询中。利用等价类划分法则和边界分析法则，首 先利用等价类划分法，可以一个或多个结果是 OK 的测试用例，然后确认多个 NG 的测试用例， 然后利用边界值分析法，可以对结果分别是 OK 和 NG 的测试用例进行扩展和补充。

1.6.4 简述黑盒测试和白盒测试的优缺点？

※ 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；
2. 与软件的内部实现无关；
3. 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；
4. 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；
5. 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；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；系统庞大时，测试开销会非常大。

1.6.5 在没有产品说明书和需求文档的情况下能够进行黑盒测试的设计吗？

能，可以通过其他工作内容去替代产品说明书和需求文档

根据客户的功能点整理测试需求追溯表

根据开发人员的 Software Specification List 整理功能测试点

开展项目跨部门讨论会，主要整理对功能点的理解和认识

测试人员整理用例需求疑问提交项目组或者产品

项目内部的用例品胜

邮件客户代表确认部分争议问题

项目的 Demo 和部分已经开发的系统参考同行业和竞争对手的类似产品

交叉模块之间的测试

咨询客户或相关者

1.6.6 单元测试的策略有哪些，主要内容有哪些？

逻辑覆盖，循环覆盖，同行评审，桌前检查，代码走查，代码评审，静态数据流分析

1.6.7 简述集成测试的过程

系统集成测试主要包括以下过程：

1. 构建的确认过程。
2. 补丁的确认过程。
3. 系统集成测试测试组提交过程。
4. 测试用例设计过程。
5. 测试代码编写过程。
6. Bug 的报告过程。
7. 每周/每两周的构建过程。
8. 点对点的测试过程。

9. 组内培训过程。

1.6.8 集成测试进入的准则？退出的准则？

集成测试由开发工程师完成

进入准则：集成完成、报告完成之后

退出准则：按照集成构件计划及增量集成策略完成了整个系统的集成测试

达到了测试计划中关于集成测试所规定的覆盖率的要求

集成工作版本满足设计定义的各项功能、性能要求

在集成测试中发现的错误已经得到修改，各级缺陷修复率达到标准

1.6.9 集成测试通常都有那些策略？

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)各个子功能组合起来，能否达到预期要求的父功能；

3)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；

4)全局数据结构是否有问题；

5)单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。

1.6.10设计系统测试计划需要参考哪些项目文挡？

软件测试计划，软件需求工件和迭代计划。系统测试计划的依据是:软件需求规格说明书

1.6.11系统测试计划是否需要同行审批，为什么

需要，系统测试计划属于项目阶段性关键文档，因此需要评审。

1.6.12Alpha 测试与 beta 的区别

Alpha 测试 在系统开发接近完成时对应用系统的测试；测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。

Beta 测试 当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。

Alpha 测试:用户在接近正式环境下的测试,开发人员在用户旁记录错误情况和使用中的问题.

Beta 测试:多个用户在实际使用环境下的测试,开发人员不在测试现场,用户通过发送报告的形式把发现的问题反馈给开发人员.

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测试。

β测试是由软件的多个用户在实际使用环境下进行的测试。这些用户返回有关错误信息给开发者。测试时，开发者通常不在测试现场。因而，β测试是在开发者无法控制的环境下进行的软件现场应用。

1.6.13系统测试阶段低级缺陷较多 怎么办？

公司有预测试这个流程 会在开展测试活动之前对主要功能点的正常流程做一个测试以判断这个版本是不是可测试版本 如果低级缺陷比较多 严重阻碍测试执行的话 我们会打回开发部 不执行测试

1.6.14系统测试的进入和退出准则？

进入：（产品集成完成，打成安装包，纳入配置库）需求确认完后

退出：

系统功能与用户需求说明书一致

测试计划规定的时间结束

所有确认缺陷都已修复

功能性测试用例通过率达到 100%

非功能性测试用例通过率达到 95%

1.6.15系统测试阶段低级缺陷较多怎么办？

公司有预测试这个流程 会在开展测试活动之前对主要功能点的正常流程做一个测试以判断这个版本是不是可测试版本 如果低级缺陷比较多 严重阻碍测试执行的话 我们会打回开发部 不执行测试

1.6.16系统测试包含哪些方面？

1.恢复测试、2.安全测试、3.强度测试、4.性能测试

1.6.17什么是验收测试？

验收测试的目的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。

1.6.18软件验收测试具体包括哪些测试？

正式验收测试，alpha 测试，beta 测试。

1.6.19.什么是功能测试？

功能测试是在规定的一段时间内运行软件系统的所有功能，以验证这个软件系统有无严重错误

1.6.202 请问功能测试和性能测试的区别是什么？(只总结了两个方面，有其他的自己补充)

①测试目的：

功能测试：检查实际软件的功能是否符合用户的需求，测功能是不是全部实现，某个实现是不是有 BUG。主要为了发现以下几类错误：A、是否有不正确或遗漏的功能？B、功能实现是否满足用户需求和系统设计的隐藏需求？ C、能否正确接收输入？能否正确输出结果？

性能测试：验证软件质量的三个质量特性，可靠性，正确性和效率。主要是测试产品的健壮性

②测试方式：

功能测试：按照系统需求说明书和测试用例，对产品的功能一步步进行测试。找出产品功能是否全部实现

性能测试：一般都使用性能工具对产品的健壮性进行评估。 通过创建场景和虚拟用户来模拟真是环境，进行压力测试和负载测试。

1.6.21兼容性测试

这类测试主要想验证软件产品在不同环境之间的兼容性 主要有操作系统和浏览器

1.6.22什么是易用性测试？

可使用性测试主要从使用的合理性和方便性等角度对软件系统进行检查，发现人为因素或使用上的问题。 要保证在足够详细的程度下，用户界面便于使用；对输入量可容错、响应时间和响应方式合理可行、输出信息有意义、正确并前后一致；出错信息能够引导用户去解决问题；软件文档全面、正规、确切。

1.6.23什么是文档测试

这种测试是检查用户文档(如用户手册)的清晰性和精确性。 用户文档中所使用的例子必须在测试中一一试过，确保叙述正确无误。

1.6.24怎么做好文档测试

仔细阅读，跟随每个步骤，检查每个图形，尝试每个示例。P142

检查文档的编写是否满足文档编写的目的

内容是否齐全，正确

内容是否完善

标记是否正确

1.6.25文档测试要注意什么？

文档的读者群、文档的术语、文档的正确性、文档的完整性、文档的一致性、文档的易用性、样例与示例、文档的语言

1.6.26什么是安全测试？

安全性测试是要检验在系统中已经存在的系统安全性、保密性措施是否发挥作用， 有无漏洞。

力图破坏系统的保护机构以进入系统的主要方法有以下几种： 正面攻击或从侧面、背面攻击系统中易受损坏的那些部分； 以系统输入为突破口，利用输入的容错性进行正面攻击

1.6.27什么时候适用自动化测试？

1) 可重复的、不知疲倦地运动,对于数据能进行精确的大批量的比较的;

2) 回归测试

3) 在机械化的执行和比较

1.6.28什么时候不宜使用自动化的情况

1) 周期短并且一次性的项目

2) 进度非常紧张的项目

3) 需求非常不稳定的项目

4) 界面尚未确定

5) 使用了很多第三方或自定义控件的项目

1.6.29什么是性能测试？

性能测试是要检查系统是否满足在需求说明书中规定的性能。特别是对于实时系统 或嵌入式系统。 性能测试常常需要与强度测试结合起来进行，并常常要求同时进行硬件和软件检测。 通常，对软件性能的检测表现在以下几个方面：响应时间、吞吐量、辅助存储区，例如缓冲区，工作区的大小等、处理精度，等等。

1.6.30您在从事性能测试工作时，是否使用过一些测试工具？如果有，请试述该工具的工作原 理，并以一个具体的工作中的例子描述该工具是如何在实际工作中应用的。

有使用过 LoadRunner,该工具能够录制测试人员的操作步骤，然后对这个操作步骤模拟出多个用户来播放出来。

1。Visural User Genertor 创建脚本，选择协议，录制操作，编辑操作。

2。中央控制器（Controller）调度虚拟用户。创建场景，选择脚本，建立虚拟用户，设计 shedual，设置 ip spoofer。

3。运行脚本。分析 shedual。

4。分析测试结果。

1.6.31您认为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？

性能测试工作的目的是检查系统是否满足在需求说明书中规定的性能，性能测试常常需要和强度测试结合起来，并常常要求同时进行软件和硬件的检测。性能测试主要的关注对象是响应时间，吞吐量，占用内存大小（辅助存储区），处理精度等。

1.6.32性能测试什么时候开始最合适

一般在功能测试的最后阶段执行 因为功能走通了 性能才有意义 总之性能测试要根据用户的实际性能指标来操作 是一个很重要的测试活动 要根据软件的属性以及它的实际情况来制定策略

1.6.33并发性能测试的目的主要体现在三个方面？

以真实的业务为依据，选择有代表性的、关键的业务操作设计测试案例，以评价系统的当前性能；当扩展应用程序的功能或者新的应用程序将要被部署时，负载测试会帮助确定系统是否还能够处理期望的用户负载，以预测系统的未来性能；通过模拟成百上千个用户，重复执行和运行测试，可以确认性能瓶颈并优化和调整应用，目的在于寻找到瓶颈问题。

1.7 测试流程

1.7.1 软件测试的基本流程有哪些？

需求分析、编写测试用例、评审测试用例、搭建环境、等待程序开发包、部署程序开发包、冒烟测试、执行具体的测试用例细节、Bug 跟踪处理回归测试、N 轮之后满足需求，测试结束

1.7.2 测试结束的标准是什么？

第一类标准：测试超过了预定时间，则停止测试。

第二类标准：执行了所有的测试用例，但并没有发现故障，则停止测试。

第三类标准：使用特定的测试用例设计方案作为判断测试停止的基础

第四类标准：正面指出停止测试的具体要求，即停止测试的标准可定义为查出某一预订数目的故障。

第五类标准：根据单位时间内查出故障的数量决定是否停止测试。

1.7.3 软件测试的原则是什么？

1) 应当把“尽早地和不断地进行软件测试”作为软件开发者的座右铭。

2) 测试用例应由测试输入数据和对应的预期输出结果这两部分组成。

3) 程序员应避免检查自己的程序。

4) 在设计测试用例时，应包括合理的输入条件和不合理的输入条件。

5) 软件测试的原则

6) 充分注意测试中的群集现象。 经验表明，测试后程序中残存的错误数目与该程序中已发现的错误数目成正比。

7) 严格执行测试计划，排除测试的随意性。

8) 应当对每一个测试结果做全面检查。

9) 妥善保存测试计划，测试用例，出错统计和最终分析报告，为维护提供方便。

1.8 用例设计

1.8.1 什么是测试用例，测试用例的基本要素？

测试用例是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求。

测试用例的基本元素： 测试索引，测试环境，测试输入，测试操作，预期结果，评价标准。

1.8.2 怎样写测试用例

在测试面试的过程中大多都会问到你是怎么来设计测试用例的刚开始的时候都会被问懵。

我们从两个方面来看怎么设计测试用例

（1）对于新手来说最简单的是从用户的角度来设计测试用例，即使之前没有接触过软件测试但是每个人的生活中都离不开”APP”。从用户的角度来讲，一个产品首先关注的当然是产品的功能，其次是兼容性和稳定性，最后是容错能力。翻译成测试用户就是，首先写功能测试用例然后写兼容性测试用例和稳定性测试用例，最后想想在测试过程中会遇到哪些异常，针对这些异常设计一些用例，来检验产

品的容错能力。

（2）常用的用例编写的方法有边界值法、等价类划分法、功能图法、因果图等常用的编写测试用例的方法。

1.8.3 描述测试用例设计的完整过程？

首先根据需求文档、概要设计、测试计划、测试方案细分出各功能模块的测试项

再根据各测试项，按照概要设计、详细设计以及测试方案中测试的覆盖率细分出测试子项

最后按照测试子项、根据测试用例的设计方法（因果图、边界值、等价类等的设计方法）书写测试用例。注意

1. 选用适合的用例管理工具（如 word，excel）
2. 用例一定要及时更新（补充新的想法，删除过时的需求）
3. 做好用例分级
4. 做好用例评审，写用例之前可以征询相关人员的意见，如果评审通过可以参考其执行测试，如果未通过，需要继续修改直到通过为止。
5. 可以考虑结对编写，这个是不错的主意
6. 要全面，包括功能、性能、兼容性、安全性、易用性、容错性等等
7. 注意把握适当的颗粒度

1.8.4 好的测试用例有哪些特点？

质量属性：

正确性：确保测试标题描述部分的内容正确性。

经济性：只为确定需要的目的设计相应的测试步骤。

可重复性：自我一致性，即不管谁执行此用例，结果一样。

适应性：既能适应短期需要，又能考虑长远需要。

可追踪性：用例能追踪到一个具体的需求。

自我清理性：单个用例不会影响整个测试环境，即用例执行完了可以恢复原有的测试环境。

结构化和可测试性

含有规范的测试标题和编号。

含有一个确定的测试某一个特定需求的目的。含

有关于测试方法的描述。

指定条件信息-环境、数据、预置的条件测试、安全入口等。

含有操作步骤和预期结果。

陈述任何辅助证据，例如截图报告并确保这些东西妥善保存。

确保测试环境的干净（即用例不会影响整个环境）。

描述时使用主动语气结构。

操作步骤不要超过 15 步。

确保单个用例测试执行时用时不超过 20 分钟。

自动化脚本用例添加必要的注释，比如目的、输入和期望结果。

如果可能，建议提供可选择性的预置条件测试。

用例之间的先后顺序是否跟业务流程一致，即用例在业务流程中的彼此顺序关系是否合理。

配置管理：

采用命名和编号规范归档。

保存为特定的格式，文件类型。

用例版本是否与当前被测试软件版本一致（对应）。

包含用例需要的相应测试对象，如特定数据库。

存档阅读。

存档时按角色控制访问方式当网络备份时存档。

离线归档。

1.8.5 测试用例制定的原则？

测试用例要包括欲测试的功能、应输入的数据和预期的输出结果。测试数据应该选用少量、高效的测试数据进行尽可能完备的测试；基本目标是：设计一组发现某个错误或某类错误的测试数据，测试用例应覆盖方面：

1、 正确性测试：输入用户实际数据以验证系统是满足需求规格说明书的要求；测试用 例中的测试点应首先保证要至少覆盖需求规格说明书中的各项功能，并且正常。

2、 容错性（健壮性）测试：程序能够接收正确数据输入并且产生正确（预期）的输出， 输入非法数据（非法类型、不符合要求的数据、溢出数据等），程序应能给出提示 并进行相应处理。把自己想象成一名对产品操作一点也不懂的客户，在进行任意操作。

3、 完整（安全）性测试：对未经授权的人使用软件系统或数据的企图，系统能够控制的程度，程序的数据处理能够保持外部信息（数据库或文件）的完整。

4、 接口间测试：测试各个模块相互间的协调和通信情况，数据输入输出的一致性和正确性。

5、 数据库测试：依据数据库设计规范对软件系统的数据库结构、数据表及其之间的数据调用关系进行测试。

6、 边界值分析法：确定边界情况（刚好等于、稍小于和稍大于和刚刚大于等价类边界值），针对我们的系统在测试过程中主要输入一些合法数据/非法数据，主要在边界值附近选取。

7、 压力测试：输入 10 条记录运行各个功能，输入 30 条记录运行，输入 50 条记录运行。。。进行测试。

8、等价划分：将所有可能的输入数据（有效的和无效的）划分成若干个等价类。

9、错误推测：主要是根据测试经验和直觉，参照以往的软件系统出现错误之处。

10、效率：完成预定的功能，系统的运行时间（主要是针对数据库而言）。

11、可理解（操作）性：理解和使用该系统的难易程度（界面友好性）。

12、可移植性：在不同操作系统及硬件配置情况下的运行性。

13、回归测试：按照测试用例将所有的测试点测试完毕，测试中发现的问题开发人员已经解决，进行下一轮的测试。

14、比较测试：将已经发版的类似产品或原有的老产品与测试的产品同时运行比较，或与已往的测试结果比较 。

1.8.6 测试用例是否纳入测试基线管理？测试用例发生变更的流程？测试用例如何进行标识？

是。测试用例没有变更流程

测试用例的标识为 ST-001 这种格式标识

1.8.7 什么时候编写测试用例？依据是什么？如何保证测试用例与需求的一

致性？需要同行评审吗？

在测试计划完成之后，按照计划进度编写测试用例。

依据是软件需求规格说明书

通过同行评审来对用例进行评审，需要同行评审

1.8.8 测试用例如何设计的？

在测试用例的设计之前首先要仔细阅读开发的详细设计文档，充分了解产品的详细功能，不清的地方与开发人员进行沟通，搞懂每个功能，尽量详细到输入框、按钮等小功能，功能点清楚之后按照功能模块分类进行用例编写。在具体的用例设计中会运用到等价类边界值等黑盒测试方法

1.8.9 如何保证用例覆盖到罕见缺陷？

(1)充足的设计时间

(2)充分的需求分析

(3)每一个功能点都有用例覆盖

(4)严格的评审流程

(5)保障输出都是有效的

(6)在测试执行过程中,会根据实际的项目情况,对用例做增加和修改

1.8.10什么时候编写测试用例？依据是什么？如何保证测试用例与需求的一致性？需要同行评审吗？

在测试计划完成之后，按照计划进度编写测试用例。

依据是软件需求规格说明书

通过同行评审来对用例进行评审，需要同行评审

1.8.11写测试用例时要注意什么问题

1、复用率：如果随着产品不停得升级，需要设计的详细些，追求一劳永逸；仅使用一两次，则没有必

要设计的过于详细；

2、项目进展：项目时间如果允许可以设计的详细些，反之则能执行即可；

3、使用对象：测试用例如果供多人使用，尤其让后参加测试的工程师来执行，则需要设计的详细些。

4、用例的冗余

5、操作步骤要细分简明，可执行

1.8.12如何在有限的情况下提高测试效率，保证产品的上线质量？

1、一个详细合理的详细的测试计划

2、测试尽早的介入项目，连接项目的业务需求，做好测试的前期准备

3、对测试项目前景充满信心，调整最佳心态，保持愉悦的工作心情

4、提高测试接受的标准，减少测试版本的送测次数

1.8.13如何降低漏测率

1、需求评审

2、梳理需求，尽早与开发人员、需求人员进行需求确认，统一不同角色对需求的认识

3、用例设计及评审

4、测试执行

5、bug 回归

6、发布前的功能回归

1.8.14测试用例的基本设计方法

1、等价类划分法

2、边界值分析法

3、错误推断法

4、因果图判定表法

5、正交实验法

6、流程法

7、场景法

1.8.15测试为什么要写测试用例

1、深入了解需求的过程

2、测试执行的指导

3、规划测试数据的准备

4、反应测试进度

5、举一反三发现隐藏缺陷

6、分析缺陷标准

1.9 缺陷 bug

1.9.1 什么是缺陷报告，缺陷报告的作用，缺陷报告的要点

(1) 缺陷报告是描述软件缺陷现象和重现步骤的集合。软件缺陷报告 Software Bug Report(SBR)或软件问题报告 software Problem Report(SPR)。

(2) 缺陷报告是软件测试人员的工作成果之一，体现软件测试的价值缺陷报告可以把软件存在的缺陷准确的描述出来，便于开发人员修正缺陷报告可以反映项目/产品当前的质量状态，便于项目整体进度和质量控制软件测试缺陷报告是软件测试的输出成果之一，可以衡量测试人员的工作能力。

(3) 标题(Title)简洁、准确、完整、反映缺陷本质、方便查询前缀+标题正文，标题正文采用结果和动作，或者现象和位置的方式表达；步骤(Steps)可复现、完整、简洁、准确按数字编号；实际结果(Actual results)准确、详细描述软件的现象和特征；期望结果(Expected results)准确、丰富、有理有据；平台(Platforms)准确；截图(Sereenshots)准确反映缺陷特征；注释(Notes)关于缺陷的辅助说明

1.9.2 缺陷报告的优先级别

最高优先级：立即修复，停止进一步测试

次高优先级：在产品发布之前必须修复

中等优先级：如果时间允许应该修复

最低优先级：可能会修复，但是也能发布

1.9.3 简单概述缺陷报告

现在缺陷报告一般不再使用纸质档文档编写，而是专用测试管理工具（如 TestDirector），这样便于缺陷管理。在这些工具中，每个缺陷作为一条记录输入指定的缺陷管理系统中。

1.9.4 缺陷报告包括哪些项？

缺陷报告包括：软件名称、版本号、功能模块、缺陷编号、对应的用例编号、编写时间、编写人、测试员、预期结果、实际结果、缺陷描述、严重级别、优先级别

1.9.5 软件测试缺陷报告的 5C 原则

Correct（准确）：每个组成部分的描述准确，不会引起误解；

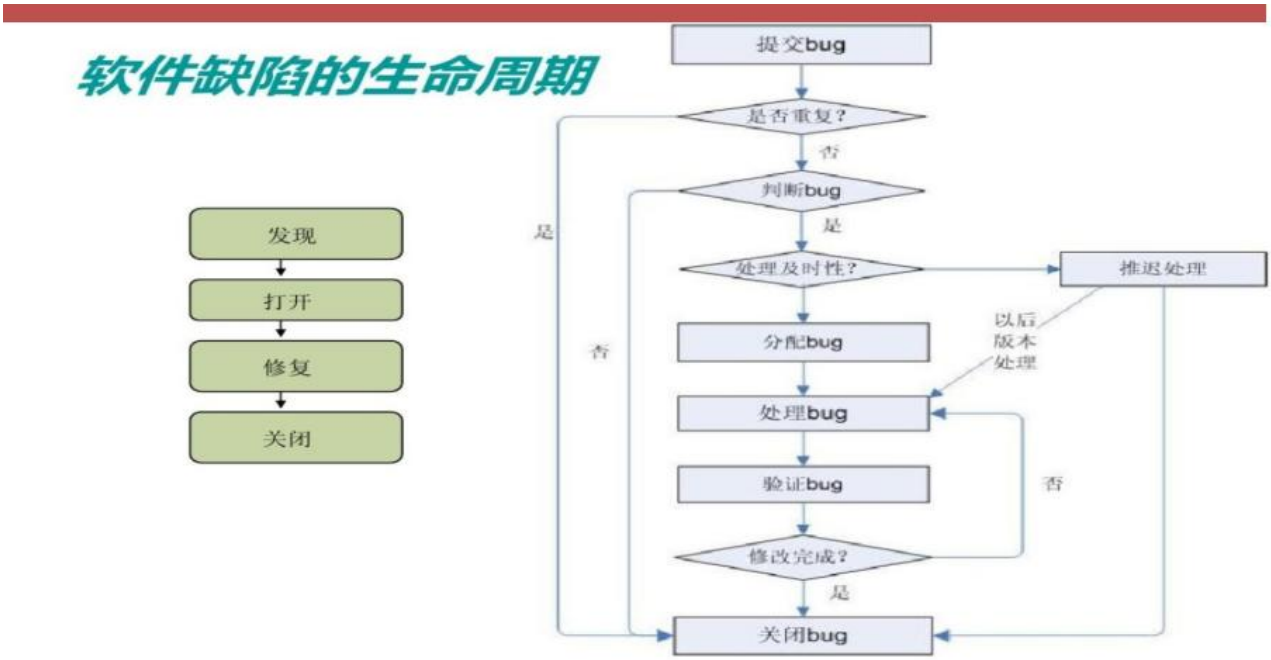
Clear（清晰）：每个组成部分的描述清晰，易于理解；

Concise（简洁）：只包含必不可少的信息，不包括任何多余的内容；

Complete（完整）：包含复现该缺陷的完整步骤和其他本质信息；

Consistent（一致）：按照一致的格式书写全部缺陷报告。

1.9.6 软件缺陷的生命周期？



测试人员提交新的 Bug 入库，错误状态为 New。 高级测试人员验证错误，如果确认是错误，分配给相应的开发人员，设置状态为 Open。如果不是错误，则拒绝，设置为 Declined(拒绝状态。 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，如没有解决置状态为 Reopen。

1.9.7 缺陷描述（报告单）中应该包括哪些内容？

缺陷的标题，简要描述。缺陷的类型。缺陷的详细步骤描述。缺陷的实际结果。期望结果。有的缺陷需要上传截图，日志信息。缺陷的等级。缺陷指派给开发同事。（开发主管）

1.9.8 如何提高缺陷的记录质量？

通用 UI 要统一、准确；尽量使用业界惯用的表达术语和表达方法；使用业界惯用的表达术语和表达方法，保证表达准确，体现专业化；每条缺陷报告只包括一个缺陷；不可重现的缺陷也要报告；明确指明缺陷类型；明确指明缺陷严重等级和优先等级；描述 (Description) ，简洁、准确，完整，揭示缺

陷实质，记录缺陷或缺陷出现的位置；短行之间使用自动数字序号，使用相同的字体、字号、行间距；每一个步骤尽量只记录一个操作；确认步骤完整，准确，简短；根据缺陷，可选择是否进行图象捕捉；检查拼写和语法缺陷；尽量使用短语和短句，避免复杂句型句式；缺陷描述内容。

1.9.9 如果一个缺陷被提交后，开发人员认为不是问题，怎么处理？

1. 首先，将问题提交到缺陷管理库里面进行备案。
2. 然后，要获取判断的依据和标准：

(1) 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；

(2) 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；

(3) 根据用户的一般使用习惯，来确认是否是缺陷；

(4) 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；

3. 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，不掺杂个人情绪。
4. 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。

1.9.10软件缺陷的原则

A、软件缺陷区别于软件 bug,它是在测试过程中出现的对系统有影响的,但是在设计中没有的或者对修改后的 bug 测试和开发人员有不同意见等

B、软件未达到产品说明书标明的功能。

C、软件出现了产品说明书指明不会出现的错误。

D、软件功能超出产品说明书指明范围。

E、软件未达到产品说明书虽未指出但应达到的目标。

F、软件测试员认为软件难以理解、不易使用、运行速度缓慢，或者最终用户认为不好。

1.9.11软件缺陷的特征。

1) “看不到”

2) ——软件的特殊性决定了缺陷不易看到

3) “看到但是抓不到”

4) ——发现了缺陷，但不易找到问题发生的原因所在

1.9.12软件缺陷产生的原因

1) 软件产品说明书（需求）——56%

2) 设计——27%

3) 编写代码——7%

4) 其他——10%

1.9.13什么是 Bug？

软件的 Bug 指的是软件中（包括程序和文档）不符合用户需求的问题。

常见的软件 Bug 分为以下三类：

（1）没有实现的功能

（2）完成了用户需求的功能，但是运行时会出现一些功能或性能上的问题

（3）实现了用户不需要的多余的功能

1.9.14缺陷处理流程

测试人员提交新的 Bug 入库，错误状态为 New。

高级测试人员验证错误，如果确认是错误，分配给相应的开发人员，设置状态为 Open。

如果不是错误，则拒绝，设置为 Declined(拒绝)状态。

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，如没有解决置状态为 Reopen。

1.9.15缺陷的等级划分

严重： 系统崩溃 数据丢失 数据毁坏

较严重：操作性失误 错误结果 遗漏功能

一般： 小问题 错别字 UI 布局 罕见故障

建议：不影响使用的瑕疵或更好的实现

1.9.16开发人员修复缺陷后，如何保证不影响其他功能？

重新执行用例、看是否出现错误结果。并对周围的一些相关功能点追加新的测试用例。

1.9.17状态为已修改的缺陷，实际没有修改怎么办？

加强项目质量管理，提高项目执行能力。如果测试人员发现了这样的问题，首先要弄清楚是什么原因导致这种情况，最终还是要督促开发人员，修改掉这些问题。如果是不能重现的问题或者是老版本中遗留下来的问题不能修改的 要做好标示。

1.9.18生产软件的最终目的是为了满足客户需求，我们以客户需求作为评判软件质量的标准，认为软件缺陷（ Software Bug ）的具体含义包括哪些几个因素？

1) 软件未达到客户需求的功能和性能；
2) 软件超出客户需求的范围；
3) 软件出现客户需求不能容忍的错误；
4) 软件的使用未能符合客户的习惯和工作环境。

1.9.19如何进行缺陷评估

评估软件质量的重要指标，通常评估模型假设缺陷的发现是呈泊松分布的；严格的缺陷评估要考察在测试过程中发现缺陷的间隔时间长短。评估要估计软件当前的可靠性并预测随着测试的继续进行，软件可靠性会怎样提高。

SQA Suite 提供四种形式进行缺陷评估：

1) 缺陷分布报告可以生成缺陷数量与缺陷属性的函数。如测试需求和状态。
2) 缺陷趋势报告可以看出缺陷增长和减少的趋势；
3) 缺陷年龄报告展示一个缺陷处于某种状态的时间长短
4) 测试结果进度报告展示测试过程在被测应用的几个版本中的执行结果以
5) 测试周期。

具体步骤

1) 回顾测试日记
2) 评估测试需求的覆盖率
3) 分析缺陷
4) 决定是否达到完成测试的标准，没有满足标准时
5) 再测试
6) 降低标准
7) 确定软件的一个满足标准的子集，看是否可以发布。

1.10 测试案例

1.10.1给你一个网站，你应该如何测试？

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：

功能性测试可以包括，但不限于以下几个方面：

链接测试。链接是否正确跳转，是否存在空页面和无效页面，是否有不正确的出错信息返回等。

提交功能的测试。

多媒体元素是否可以正确加载和显示。

多语言支持是否能够正确显示选择的语言等。

界面测试可以包括但不限于一下几个方面：

页面是否风格统一，美观

页面布局是否合理，重点内容和热点内容是否突出

控件是否正常使用

对于必须但为安装的空间，是否提供自动下载并安装的功能

文字检查

性能测试一般从以下两个方面考虑：

压力测试、负载测试、强度测试、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：

基本的登录功能的检查

是否存在溢出错误，导致系统崩溃或者权限泄露相关开发语言的常见安全性问题检查，例如 SQL 注入等。

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。合理的安排调整测试进度，提前获取测试所需的资源，建立管理体系（例如，需求变更、风险、配置、测试文档、缺陷报告、人力资源等内容）。

定期评审，对测试进行评估和总结，调整测试的内容。

（一个网站基本完工后，需要通过下面三步测试才可以交付）

a、制作者测试，包括美工测试页面、程序员测试功能。在做完后第一时间内有制作者本人进行测试。

b 全面测试 根据交工标准和客户要求，由专人进行全面测试

c 发布测试 网站发布到主服务器之后的测试，主要是防止环境不同导致的错误

1.10.2一个有广告的纸杯子，请设计测试用例？

测试项目：杯子

需求测试：查看杯子使用说明书界面测试：查看杯子外观

功能度：用水杯装水看漏不漏；水能不能被喝到安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

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

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等

易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

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

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

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

跌落测试: 杯子加包装(有填充物),在多高的情况摔下不破损

震动测试: 杯子加包装(有填充物),六面震动,检查产品是否能应对恶劣的铁路\公路\航空运输

基本功能测试（逻辑功能测试）。

（1）硬度：是否达到设计标准。

装载能力：在杯子内分别装入少量的、半杯的、满杯的，看其装载量是否达到设计标准。

装载种类：开水（是否产生异味）、温水、冷水、冰水、咖啡…

（2）界面测试（UI 测试）。

看其形状、大小设计是否适合人方便拿起。

外观是否吸引人（广告嘛），赏心悦目。

带广告的图案沾水受是否掉色、模糊。

（3）易用性测试。

看其形状、大小设计是否适合人方便拿起。

残疾人士用此杯去喝水的容程度。

杯子设计是否上大下小，在运输过程中可以套在一起有效利用空间，在使用时也容易拿开。

（4）稳定性测试（24 X 7 测试）。装入液体后记录其多少以后漏水。

（5）安全性测试。杯子所用的材料（包括纸基、涂层和广告颜料）是否符合食品卫生标准，在内外温度等环境因素下是否会与所盛各种饮料相反应，而产生对人体有害的物质。

（6）本地化测试。为国际化和本地化的需要，广告图案和文字是否在政治、宗教和文化方面具有广泛的适用性。

（7）对设计的改进建议。“如果是一次性杯子，能否标示已使用（比如变色）”和“杯子是否有使用者标贴（多人使用时防止混淆）”。

1.10.3一个身份证号码输入框，怎么设计用例？

校验身份证号规则的有效性（包括地址码、生日期码、顺序码和校验码

校验 15 位身份证号和 18 位身份正好都是可用的

校验末位是 X 的情况

校验不足 15 位、16-17 位和大于 18 位的情况

如果是必输项，校验不输入的时候会不会有正确的提示

如果不是必输项，则要校验不输入的时候流程能否正常进行

校验输入非数字的情况，是否会有正确提示信息（包括大小写字母、汉字、特殊字符和标点符号） 校验输入全角的数字的时候，系统是否会识别（这个得根据需求确定是否可以使用全角的数字）

1.10.4登录功能怎么设计测试用例？

具体需求：

有一个登录页面，有一个账号和一个密码输入框，一个提交按钮。此题的考察目的：

1、了解需求（测什么都是从了解需求开始）； 2、是否有设计 Test Case 的能力

3、是否熟悉各种测试方法；

4、是否有丰富的 Web 测试经验；

5、是否了解 Web 开发；

了解需求：

1、登录界面应该是弹出窗口式的，还是直接在网页里面；

2、账号长度和密码的强度（比如需要多少位、大小写敏感、特殊字符混搭等）；

3、界面美观是否有特殊要求？（即是否要进行 UI 测试）；

4、····

用例设计：

测试需求分析完成后，开始用例设计，主要可以从以下几个方面考虑： 功能测试(Function Test)

1、输入正确的账号和密码，点击提交按钮，验证是否能正确登录。（正常输入）

2、输入错误的账号或者密码, 验证登录会失败，并且提示相应的错误信息。（错误校验）

3、登录成功后能否跳转到正确的页面（低）

4、账号和密码，如果太短或者太长，应该怎么处理（安全性，密码太短时是否有提示）

5、账号和密码，中有特殊字符（比如空格），和其他非英文的情况（是否做了过滤）

6、记住账号的功能

7、登录失败后，不能记录密码的功能

8、账号和密码前后有空格的处理

9、密码是否加密显示（星号圆点等）

10、牵扯到验证码的，还要考虑文字是否扭曲过度导致辨认难度大，考虑颜色（色盲使用者），刷新或换一个按钮是否好用

11、登录页面中的注册、忘记密码，登出用另一帐号登录等链接是否正确

12、输入密码的时候，大写键盘开启的时候要有提示信息。

13、什么都不输入，点击提交按钮，看提示信息。（非空检查）

界面测试(UI Test)

1、布局是否合理，2 个 Testbox 和一个按钮是否对齐

2、Testbox 和按钮的长度，高度是否符合要求

3、界面的设计风格是否与 UI 的设计风格统一

4、界面中的文字简洁易懂，没有错别字。

性能测试(Performance Test)

1、打开登录页面，需要几秒

2 、输入正确的账号和密码后，登录成功跳转到新页面，不超过 5

秒安全性测试(Security Test)

1、登录成功后生成的 Cookie 是否有 HttpOnly(降低脚本盗取风险)

2、账号和密码是否通过加密的方式，发送给 Web 服务器

3、账号和密码的验证，应该是用服务器端验证，而不能单单是在客户端用 javaScript 验证

4、账号和密码的输入框，应该屏蔽 SQL 注入攻击

5、账号和密码的输入框，应该禁止输入脚本（防止 XSS 攻击）

6、错误登录的次数限制（防止暴力破解）

7、考虑是否支持多用户在同一机器上登录；

8、考虑一用户在多台机器上登录

可用性测试(Usability Test)

1、是否可以全用键盘操作，是否有快捷键

2、输入账号，密码后按回车，是否可以登录

3、输入框是否可以以 Tab 键切换兼容性测试（Compatibility Test）

1、主流的浏览器下能否显示正常已经功能正常（IE6~11, FireFox, Chrome, Safari 等 ）

2、不同的平台是否能正常工作，比如 Windows, Mac 3、移动设备上是否正常工作，比如 iPhone, Android

4、不同的分辨率

本地化测试 （Localization Test）

1、不同语言环境下，页面的显示是否正确.

软件辅助性测试 （Accessibility Test）

软件辅助功能测试是指测试软件是否向残疾用户提供足够的辅助功能 1、高对比度下能否显示正常（视力不好的人使用）

1.10.5 移动端和 web 端测试有什么区别

单纯从功能测试的层面上来讲的话，APP 测试、web 测试 在流程和功能测试上是没有区别的。根据两者载体不一样，则区别如下：

系统结构方面

web 项目，b/s 架构，基于浏览器的；web 测试只要更新了服务器端，客户端就会同步会更新。

app 项目，c/s 结构的，必须要有客户端；app 修改了服务端，则客户端用户所有核心版本都需要进行回归测试一遍。

性能方面

web 项目 需监测 响应时间、CPU、Memory

app 项目 除了监测 响应时间、CPU、Memory 外，还需监测 流量、电量等

兼容方面

（1）web 项目：

1.浏览器（火狐、谷歌、IE 等）

2.操作系统（Windows7、Windows10、Linux 等）

（2）app 项目：

1.设备系统:iOS（ipad、iphone）、Android（三星、华为、联想等） 、Windows（Win7、Win8）、OSX（Mac）

2.手机设备可根据 手机型号、分辨率不同相对于 Wed 项目，

APP 有专项测试

1.干扰测试：中断，来电，短信，关机，重启等

2.弱网络测试（模拟 2g、3g、4g，wifi 网络状态以及丢包情况）；网络切换测试（网络断开后重连、3g 切换到 4g/wifi 等）

3.安装、更新、卸载

安装：需考虑安装时的中断、弱网、安装后删除安装文件等情况卸载：需考虑 卸载后是否删除 app 相关的文件

更新：分强制更新、非强制更新、增量包更新、断点续传、弱网状态下更新

4.界面操作：关于手机端测试，需注意手势，横竖屏切换，多点触控，前后台切换

5.安全测试：安装包是否可反编译代码、安装包是否签名、权限设置，例如访问通讯录等

6.边界测试：可用存储空间少、没有 SD 卡/双 SD 卡、飞行模式、系统时间有误、第三方依赖（QQ、微信登录）等

7.权限测试：设置某个 App 是否可以获取该权限，例如是否可访问通讯录、相册、照相机等测试工具方面

自动化工具：APP 一般使用 Appium; Web 一般使用 Selenium

性能测试工具：APP 一般使用 JMeter; Web 一般使用 LR、JMeter

1.10.6 测试一个 C/S 客户端时，需要考虑的因素

客户端安装测试

客户端升级测试

客户端可维护性测试

（1）个体的客户端应用以“分离的”模式被测试——不考虑服务器和底层网络的运行；

（2）客户端软件和关联的服务器端应用被一起测试，但网络运行不被明显的考虑；

（3）完整的 C/S 体系结构，包括网络运行和性能被测试。

应用功能测试

客户端应用被独立地执行，以揭示在其运行中的错误。

服务器测试

测试服务器的协调和数据管理功能，也考虑服务器性能（整体反映时间和数据吞吐量）。数据库测试

测试服务器存储的数据的精确性和完整性，检查客户端应用提交的事务，以保证数据被正确地存储、更新和检索。

事务测试

创建一系列的测试以保证每类事务被按照需求处理。测试着重于处理的正确性，也关注性能问题。

网络通信测试

这些测试验证网络节点间的通信正常地发生，并且消息传递、事务和相关的网络交通无错的发生

1.10.7测试电梯，请详细描述

如果给你一台电梯，请问你如何测试它，分析如下：

1. 功能：上升、下降、停止、开门、关门、梯内电话、灯光、指示灯等；

2. 性能：速度、反应时间、关门时间等；

3. 压力：超载、尖锐物碰撞电梯壁等；

4. 安全：停电、报警装置、轿箱停靠位置、有人扒门时的情况等；

5. 可用性：按键高度、操作是否方便、舒适程度等；

6. UI：美观程度、光滑程度、形状、质感等；

7. 稳定性：长时间运行情况等；

8. 兼容性：不同电压是否可工作、不同类型电话是否可安装等。其实在简单分析的过程中，发现许多东西根本测试不全，比如电话、灯光、材质、调度程序、可维修性等，当发现在一个用例中无法说清楚时，这些应该拆分开来分别测试。可以告诉主考官，你需要模块化地测试电话、灯光等。再有在一起的组装测试。

下面是详细的测试点：

需求测试：查看电梯使用说明书、安全说明书等界面测试：查看电梯外观

功能测试：

1. 测试电梯能否实现正常的上升和下降功能。

2. 电梯的按钮是否都可以使用。

3. 电梯门的打开，关闭是否正常。

4. 报警装置是否可用。

5. 与其他电梯之间是否协作良好。

6. 通风状况如何。

7. 突然停电时的情况。

8. 上升途中的响应。

1）电梯本来在 1 楼，如果有人按 18 楼，那么电梯在上升到 5 楼的时候，有人按了 10 楼，这时候是否会在 10 楼先停下来

2）电梯下降到 10 层时显示满员，此时若 8 层有人等待电梯，是否在 8 层停。

9. 是否有手机信号

可靠性测试：

1.门关上的一刹那出现障碍物。

2.同时按关门和开门按钮。

3.点击当前楼层号码

4.多次点击同一楼层号码

5.同时按上键和下键

易用性：电梯的按钮的设计符合一般人的习惯吗

用户文档：使用手册是否对电梯的用法、限制、使用条件等有详细的描述

压力测试：1.看电梯的最大承重量，在负载过重时报警装置是否有提醒

稳定性测试：看垫底在最大负载下平行运行的最长时间

1.10.8对一只圆珠笔进行测试

1. 界面测试，无论我们做那类软件（嵌入式别提），只要给用户有看到的东西，从测试的角度，就要考虑界面测试，这个呢，现在针对微软的产品，某公司开发了一套界面检查表，我这里有一份，想要可以找我界面测试测什么，怎么测呢？针对这个问题我是这样回答的，印刷在产品上的图片，文字，这可能涉及不同的东西，有圆珠笔厂家的信息，也有针对不同用户的信息（譬如小孩子喜欢颜色搭配多一点的，而成人用稳重的产品等），可能涉及的还有人的审美观，你圆珠笔色彩搭配之类的

2. 功能测试，这是我们测试的重点，也是客户针对某家公司产品给出满意度的参考点，圆珠笔功能主要是书写，这里面涉及一个功用方面的焦点——书写的快慢程度，也就是流利不流利的问题（这涉及笔芯的材质问题）

针对这方面的测试，个人认为应从以下几点

a.材质问题，这涉及程序员和用户之间的关系，两者利益均有，程序员考虑成本问题，用户考虑污染问题，也就是说制作圆珠笔的材料与环境的问题，厂商考虑价格因素，用户考虑环境因素以及安全性因素

这就把安全性测试给说出来了，大的方面因为笔油材质的问题，和使用者之间的健康问题

有联系， 要测小的方面，笔油的速率，以及书写后是否马上可以涂抹，可否修改，这都涉及安全性的问题

b.性能问题，温度，湿度，气压对笔芯产生不同的影响

3. 安全性问题

测试不同的高度，笔身做自由落体损坏程度

4. 兼容性问题

不同的笔筒和笔芯之间的互相兼容

5. 强度测试

弹簧在不同的压力之下，承受变形的程度

在金山面试时候，考官特意问我针对笔芯那个米珠如何测试或者

1、界面测试

界面测试也就是对其外表先进行判断。

尺寸是否适合用户使用？用户需要的是什么样的尺寸，小孩和成年人使用的尺寸是有区别的；

色彩搭配是否合理？

形状是否美观？

是否方便携带和存放？

笔芯颜色是否与客户要求一致？

笔身印的 log 或者文字是否这么正确

2、**功能测试**

笔筒开合；

笔芯替换；

出墨快慢；

笔头出墨粗细；

是不是可操作性签字笔；

3、**性能测试**

笔芯的寿命；

笔墨的气味；

写过的字用纸水浸透后，笔墨是否会晕开

压力测试：笔尖在多大压力范围内可以正常写字，不能正常出墨，太重损坏笔尖或纸张；笔壳能在多大压力范围内正常使用？成人用力太重掰断笔壳，掉到地上易摔，能在纸上写出清晰的字

4、**性能测试**

握笔的地方纹路是否会硌手或太滑；

书写的流畅度；

写出的墨水多久能干；

高温和低温环境对笔芯出墨和笔壳的影响；

长时间不盖笔套，或笔盖盖多长时间不用，会不会对笔下次写字有影响

5、**安全测试**

笔墨是否有易燃性；

笔墨是否对皮肤有害；

笔杆折断，材质是否容易刮伤手；

误食笔芯是否会引起中毒（有小孩或者有人喜欢咬笔头）

6、**兼容性测试**

笔壳和笔芯是否能够很好的适应主流签字笔尺寸；

这个笔芯的笔尖如果损坏，换上其他的笔芯的笔尖是否能用；

这个笔芯的笔墨如果用完，换上其他笔芯的笔墨是否可以使用；

笔的笔墨如果在其他笔的笔墨上写字是否可以成功覆盖

7、其他测试

（1）比较测试

与其他品牌签字笔比较，优劣在哪些地方？

（2）场景测试

笔从高处摔到地上，笔尖是否会摔坏；

倒着写，是否可以写出很多字来；

扔到水里，笔墨会不会一直晕开；

笔在粗糙的纸上是否能写出字…

1.10.9游戏测试与软件测试的区别

游戏测试作为软件测试的一部分，它具备了软件测试所有的一切共同的特性：

1) 测试的目的是发现软件中存在的缺陷。

2) 测试都是需要测试人员按照产品行为描述来实施。产品行为描述可以是书面的规格说明书，需求文档，产品文件，或是用户手册，源代码，或是工作的可执行程序。

3) 每一种测试都需要产品运行于真实的或是模拟环境之下。

4) 每一种测试都要求以系统方法展示产品功能，以证明测试结果是否有效，以及发现其中出错的原因，从而让程序人员进行改进。

游戏世界测试，主要有以下几个特性：

1) 游戏情节的测试，主要指游戏世界中的任务系统的组成，有人也称为游戏世界的事件驱动，我喜欢称为游戏情感世界的测试。

2) 游戏世界的平衡测试，主要表现在经济平衡，能力平衡（包含技能，属性等等），保证游戏世界竞争公平。

3) 游戏文化的测试，比如整个游戏世界的风格，是中国文化主导，还是日韩风格等等，大到游戏整体，小到 NPC（游戏世界人物）对话，比如一个书生，他的对话就必需斯文，不可以用江湖语言 J。

1.10.10 想象一个登录框，包括 ID、密码、登录、取消，记住密码（复选框），

尽可能的写出你想到的测试点？

ID 测试要点：

字符类型（包括：数字、字母、汉字、特殊字符）

字符长度

默认值

空值

字符集

存在空格

复制、粘贴

密码测试要点：

密码长度（例如：密码不能少于 7 个字符，最长不能超过 20 个字符） 密码复杂度、强度（例如：密码必须包含特殊字符、数字字母大小写等等，长度是否长）

密码字符类型（例如：只允许输入数字、字母、特殊字符、下划线）

默认值

密码为空

字符集

存在空格

复制、粘贴

登录测试要点

① 用户名和密码都符合要求（格式上的要求）

② 用户名和密码都不符合要求（格式上的要求）

③ 用户名符合要求，密码不符合要求（格式上的要求）

④ 密码符合要求，用户名不符合要求（格式上的要求）

⑤ 用户名或密码为空

⑥ 数据库中不存在的用户名，不存在的密码

⑦ 数据库中存在的用户名，错误的密码

⑧ 数据库中不存在的用户名，存在的密码

⑨ 输入的数据前存在空格

⑩ 输入正确的用户名密码以后按[enter]是否能登陆

取消

鼠标左键点击“取消”按钮

鼠标左键双击“取消”按钮

鼠标右键点击“取消”按钮

鼠标右键双击“取消”按钮

鼠标指针移动到“取消”按钮”，按回车键 鼠标指针停留在“取消”按钮上

记住密码

（复选框）

点击选中“记住密码”按钮，重新登录

不选中“记住密码”按钮，重新输入密码登录

输入已存在 ID 和错误密码，勾选“记住密码”，点击登录 输入错误 ID 和正确密码，勾选“记住密码”，点击登录 输入正确 ID 和正确密码，勾选“记住密码”，点击登录 输入错误 ID 和错误密码，勾选“记住密码”，点击登录.

1、账号或密码登陆正确或错误时，进入后续页面或弹出反馈内容是否正确？2、各类浏览器的各个版本登陆测试兼容性，不同的浏览器有可能会错位显示。3、电脑和移动设备登陆测试。4、输入一段如'or'='or'之类代码作为账号或密码看是否能登陆。5、登录后，测试不操作后需多长时间会自动注销；退出后，刷新页面检查是否真的退出。

1.10.11 针对添加购物车这个测试点说一下你要怎么测试“添加购物车”

（增删改查的角度）

1 能否加入购物车，同一件商品能否再次添加到购物车。

2 购物车商品件数的上限限制（淘宝限制 100 件）

3 购物车是否可以正常移除商品，移除商品后，能否再添加回来。

4 添加的每种商品是否可以正常增减数量，数量大于 0

5 退出购物车，再去查询购物车，商品正常。

6 购物车的商品可以全选，取消全选，可以复选，选中的商品和数量可以正常下单。

7 商品添加到购物车以后，已下架。购物车会提示此宝贝已失效。

8 商品添加到购物车以后，降价了，购物车会有降价提示。

9 商品添加到购物车以后，库存不足了。

1.10.12 网上银行转账是怎么测的，设计一下测试用例。

回答思路：

宏观上可以从质量模型（万能公式）来考虑，重点需要测试转账的功能、性能与安全性。设计测试用例可以使用场景法为主，先列出转账的基本流和备选流。然后设计场景，最后根据场景设计数据。实际面试中需要举出具体的例子。

1 先检查界面。

2 再测试功能：

2.1 验证同行转账，跨行转账。

2.2 验证转账限额。

2.3 验证非法账户（挂失，冻结，锁定的账户）的转账。

3 再测试性能方面的。

2 Linux 基础

2.1 Linux 基础

2.1.1 说出 10 个以上的 Linux 命令

1、创建文件：touch

2、删除文件：rm(remove)

3、查看文件：cat

4、复制：cp(copy)

5、创建文件夹：mkdir(make directory)

6、剪切或者重命名：mv(move)

7、压缩解压缩：tar

8、查看：ls list

9、编辑：vi/vim

10、查看当前路径：pwd(Print Working Directory)

11、切换用户：su switch user

12、创建用户：useradd

13、删除用户：userdel

14、创建用户组：groupadd

15、删除用户组：groupdel

16、查找：find

17、修改权限：chmod(change mode)

18、查看进程：ps process

19、杀进程：kill

20、查看日志：tail

1、Linux 的每个命令，是一个或者多个英文单词的缩写。

2、拷贝：cp 文件 1 文件 2 cp -r 目录 1 目录 2

3、杀进程：kill -9 进程号

4、查看日志：tail -f catalina.out -n 300

查看命令的使用：

tail --help

man tail

2.1.2 在 RedHat 中，从 root 用户切到 userl 用户，一般用什么命令？

su

su user1 切换到 user1，但切换后的当前目录还是 root 访问的目录

su – user1 切换到 user1，并且当前目录切换到 user1 的根目录下（/home/user1/）

2.1.3 Linux 中，一般怎么隐藏文件？

文件名以一个.开头

2.1.4 在 Linux 系统中，一个文件的访问权限是 755，其含义是什么？

755 表示该文件所有者对该文件具有读、写、执行权限，该文件所有者所在组用户及其他用户对该文件具有读和执行权限。

2.1.5 如何查看 CPU 信息？

2.1.6 查看占用 CPU 使用率最高的进程？

ps -aux | sort -k3nr | head -K

2.1.7 如何查看一个文件的末尾 50 行？

查看/etc/profile 的前 10 行内容，应该是： # head -n 10 /etc/profile

查看/etc/profile 的最后 50 行内容，应该是： # tail -n 50 /etc/profile

2.1.8 如何过滤文件内容中包含”ERROR“的行？

grep "ERROR" file_name

cat file_name | grep "ERROR"

2.1.9 查看某端口号？

netstat -anp | grep port_number

2.1.10查看某进程号？

ps -ef | grep ps_name

ps -ef | grep ps_number

2.1.11grep 和 find 的区别？grep 都有哪些用法？

2.1.12查看 IP 地址？

ifconfig

2.1.13创建和删除一个多级目录？

mkdir -p ./a/b rm -rf ./a

2.1.14在当前用户家目录中查找 haha.txt 文件？

find ~/ -name haha.txt

2.1.15如何查询出 tomcat 的进程并杀掉这个进程，写出 linux 命令？

ps -ef | grep tomcat

kill -9 tomcat_port

2.1.16动态查看日志文件？

tail -f log_file

2.1.17查看系统硬盘空间的命令？

df -aTh

2.1.18查看当前机器 listen 的所有端口？

netstat -tlnp

2.1.19把一个文件夹打包压缩成.tar.gz 的命令，以及解压拆包.tar.gz 的命令？

tar zcvf xxx.tar.gz file

tar zxvf xxx.tar.gz

2.1.20Xshell 工具如果想要实现从服务器上传或者下载文件的话,可以在服务器上安装什么包？

lrzsz

2.1.21以/etc/passwd 的前五行内容为例，提取用户名？

cat /etc/passwd | head -n 5 | cut -d : -f 1

2.1.22在 linux 中 find 和 grep 的区别？

Linux 系统中 grep 命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。grep 全称是 Global Regular Expression Print，表示全局正则表达式版本，它的使用权限是所有用户。

linux 下 的 find：

功能：在目录结构中搜索文件，并执行指定的操作。此命令提供了相当多的查找条件，功能很强大。 语法：find 起始目录寻找条件操作说明：find 命令从指定的起始目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件并对之采取相关的操作。

简单点说说，grep 是查找匹配条件的行，find 是搜索匹配条件的文件。

2.1.23linux 查看文件用什么命令，查看进程用什么命令

查看文件内容的命令有 more less head tail cat

查看进程：ps -ef | grep 进程号

查看日志文件常用:less、view

2.1.24查看日志常用什么命令，主要查看什么内容

1 查看日志常用 less 命令或者 view 命令。

2 主要查看程序运行的记录，比如支付失败，后台就有报错信息打印到.log 日志文件中，就可以通过分析日志信息来初步定为问题。（补充：同时也去查询数据库，分析订单数据，查看支付状态等等）

PS:日志就是.log 的文本文件，和.txt 一样属于文本文件。vi 或者 vim 编辑器属于记事本软件，一般不会用来查看日志。

2.2 Linux 练习题

本套笔试题参考答案章节末尾

2.2.1 cron 后台常驻程序 (daemon) 用于：

A. 负责文件在网络中的共享

B. 管理打印子系统

C. 跟踪管理系统信息和错误

D. 管理系统日常任务的调度

2.2.2 下面哪个 Linux 命令可以一次显示一页内容？

A. pause

B. cat

C. more

D. grep

2.2.3 怎样了解您在当前目录下还有多大空间？

A. Use df

B. Use du /

C. Use du .

D. Use df .

2.2.4 怎样更改一个文件的权限设置？

A. attrib

B. chmod

C. change

D. file

2.2.5 下面哪个命令可以列出定义在以后特定时间运行一次的所有任务？

A. atq

B. cron

C. batch

D. at

2.2.6 在 bash 中，export 命令的作用是：

A. 在子 shell 中运行命令

B. 使在子 shell 中可以使用命令历史记录

C. 为其它应用程序设置环境变量

D. 提供 NFS 分区给网络中的其它系统使用

2.2.7 有一个备份程序 mybackup，需要在周一至周五下午 1 点和晚上 8 点各运行一次，下面哪条 crontab 的项可以完成这项工作？

A. 0 13,20 * * 1,5 mybackup

B. 0 13,20 * * 1,2,3,4,5 mybackup

C. * 13,20 * * 1,2,3,4,5 mybackup

D. 0 13,20 1,5 * * mybackup

2.2.8 如何从当前系统中卸载一个已装载的文件系统

A. umount

B. dismount

C. mount -u

D. 从 /etc/fstab 中删除这个文件系统项

2.2.9 哪一条命令用来装载所有在 /etc/fstab 中定义的文件系统？

A. amount

B. mount -a

C. fmount

D. mount -f

2.2.10运行一个脚本，用户不需要什么样的权限？

A. read

B. write

C. execute

D. browse on the directory

2.2.11下面哪条命令可以把 f1.txt 复制为 f2.txt?

A. cp f1.txt | f2.txt

B. cat f1.txt | f2.txt

C. cat f1.txt > f2.txt

D. copy f1.txt | f2.txt

2.2.12显示一个文件最后几行的命令是：

A. tac

B. tail

C. rear

D. last

2.2.13如何快速切换到用户 John 的主目录下？

A. cd @John

B. cd #John

C. cd &John

D. cd ~John

2.2.14如何在文件中查找显示所有以"*"打头的行？

A. find * file

B. wc -l * < file

C. grep -n * file

D. grep ‘^*’ file

2.2.15在 ps 命令中什么参数是用来显示所有用户的进程的？

A. a

B. b

C. u

D. x

2.2.16在一行结束位置加上什么符号，表示未结束，下一行继续？

A. /

B. \

C. ;

D. |

2.2.17命令 kill 9 的含义是：

A. kills the process whose PID is 9.

B. kills all processes belonging to UID 9.

C. sends SIGKILL to the process whose PID is 9.

D. sends SIGTERM to the process whose PID IS 9.

2.2.18如何删除一个非空子目录/tmp？

A. del /tmp/*

B. rm -rf /tmp

C. rm -Ra /tmp/*

D. rm -rf /tmp/*

2.2.19对所有用户的变量设置，应当放在哪个文件下？

A. /etc/bashrc

B. /etc/profile

C. ~/.bash_profile

D. /etc/skel/.bashrc

2.2.20在 Linux 系统中的脚本文件一般以什么开头？

A. $/bin/sh

B. #!/bin/sh

C. use /bin/sh

D. set shell=/bin/sh

2.2.21Linux 中，提供 TCP/IP 包过滤功能的软件叫什么？

A. rarp

B. route

C. iptables

D. filter

2.2.22在 vi 中退出不保存的命令是？

A. :q

B. :w

C. :wq

D. :q!

2.2.23使用什么命令检测基本网络连接？

A. ping

B. route

C. netstat

D. ifconfig

2.2.24下面哪个命令可以压缩部分文件：

A. tar -dzvf filename.tgz *

B. tar -tzvf filename.tgz *

C. tar -czvf filename.tgz *

D. tar -xzvf filename.tgz *

2.2.25对于 Apache 服务器，提供的子进程的缺省的用户是：

A. root

B. apached

C. httpd

D. nobody

2.2.26apache 的主配置文件是：

A. httpd.conf

B. httpd.cfg

C. access.cfg

D. apache.conf

2.2.27通过 Makefile 来安装已编译过的代码的命令是：

A. make

B. install

C. make depend

D. make install

2.2.28什么命令解压缩 tar 文件？

A. tar -czvf filename.tgz

B. tar -xzvf filename.tgz

C. tar -tzvf filename.tgz

D. tar -dzvf filename.tgz

2.2.29命令 netstat -a 停了很长时间没有响应，这可能是哪里的问题？

A. NFS.

B. DNS.

C. NIS.

D. routing.

2.2.30ping 使用的协议是：

A. TCP

B. UDP

C. SMB

D. ICMP

2.2.31下面哪个命令不是用来查看网络故障的？

A. ping

B. init

C. telnet

D. netstat

2.2.32TCP/IP 中，哪个协议是用来进行 IP 自动分配的？

A. ARP

B. NFS

C. DHCP

D. DNS

2.2.33Linux 参考答案：

01.D 02.C 03.C 04.B 05.A

06.C 07.B 08.A 09.B 10.B

11.C 12.B 13.D 14.D 15.A

16.B 17.D 18.B 19.B 20.B

21.C 22.D 23.A 24.C 25.D

26.A 27.D 28.B 29.B 30.D

31.B 32.C

3 MySQL 基础

3.1 基础知识

3.1.1 什么是数据库？

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库

3.1.2 什么是关系型数据库，主键，外键，索引分别是什么？

关系型数据库是由多张能互相联接的二维行列表格组成的数据库

主关键字(primary key)是表中的一个或多个字段，它的值用于唯一地标识表中的某一条记录

外键表示了两个关系之间的相关联系。以另一个关系的外键作主关键字的表被称为主表，具有此外键的表被称为主表的从表。外键又称作外关键字

在关系数据库中，索引是一种单独的、物理的对数据库表中一列或多列的值进行排序的一种存储结构，它是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单

3.1.3 写出表的增删改查 SQL 语法

表的创建：create table 表名 (列名 1 类型 约束，列 2 类型 约束…)

表的删除：drop table 表名

表的更改（结构的更改，不是记录的更新）：alter table 表名 add|drop 列名|约束名

插入记录：insert into 表名…values…

更新记录：update 表名 set 列名=值 where 条件

删除记录：delete from 表名 where 条件

3.1.4 SQL 的表连接方式有哪些？

SQL 中连接按结果集分为：内连接，外连接，交叉连接

内连接：inner join on，两表都满足的组合。内连接分为等值连接，不等连接，自然连接。

等值连接：两表中相同的列都会出现在结果集中。

自然连接：两表中具体相同列表的列会合并为同一列出现在结果集中。

外连接：分为左（外）连接，右（外）连接，全连接

左（外）连接：A left (outer) join B,以 A 表为基础，A 表的全部数据，B 表有的组合，没有的为。

右（外）连接：A right(outer) join B,以 B 表为基础，B 表的全部数据，A 表有的组合，没有的位 null。

全连接：A full (outer) join 两表相同的组合在一起，A 表有，B 表没有的数据（显示为 null），同样 B 表有，A 表没有的显示为 null。

交叉连接：cross join,就是笛卡尔乘积。

3.1.5 表的连接查询方式有哪些，有什么区别？

交叉连接即笛卡儿乘积，是指两个关系中所有元组的任意组合

使用内连接时，如果两个表的相关字段满足连接条件，就从这两个表中提取数据并组合成新的记录 自连接是一种特殊的内连接，它是指相互连接的表在物理上为同一张表，但可以在逻辑上分为两张表

外连接是只限制一张表中的数据必须满足连接条件，而另一张表中的数据可以不满足连接条件的连接方式

3.1.6 什么三范式？

1NF:表中的字段都是单一属性，不再可分。

2NF：在 1NF 的基础上，表中所有的非主属性都必须完全依赖于任意一组候选键，不能仅依赖于候选键中的某个属性。

3NF：在 2NF 的基础上，表中所有的属性都不依赖其他非主属性。

简单的说就是：1NF 表示每个属性不可分割，2NF 表示非主属性不存在对主键的部分依赖，3NF 表示不存在非主属性对主键的依赖传递。

3.1.7 SQL 的 select 语句完整的执行顺序？

1、from 子句组装来自不同数据源的数据；

2、where 子句基于指定的条件对记录行进行筛选； 3、group by 子句将数据划分为多个分组；

4、使用聚集函数进行计算；

5、使用 having 子句筛选分组； 6、计算所有的表达式；

7、select 的字段；

8、使用 order by 对结果集进行排序。

3.1.8 说一下 Mysql 数据库存储的原理？

储存过程是一个可编程的函数，它在数据库中创建并保存。它可以有 SQL 语句和一些特殊的控制结构组成。当希望在不同的应用程序或平台上执行相同的函数，或者封装特定功能时，存储过程是非常有用的。数据库中的存储过程可以看做是对编程中面向对象方法的模拟。它允许控制数据的访问方式。存储过程通常有以下优点：

1、存储过程能实现较快的执行速度

2、存储过程允许标准组件是编程。

3、存储过程可以用流程控制语句编写，有很强的灵活性，可以完成复杂的判断和较复杂的运算。

4、存储过程可被作为一种安全机制来充分利用。

5、存储过程能够减少网络流量

3.1.9 事务的特性？

1、原子性(Atomicity)：事务中的全部操作在数据库中是不可分割的，要么全部完成，要么均不执行。

2、一致性(Consistency)：几个并行执行的事务，其执行结果必须与按某一顺序串行执行的结果相一致。

3、隔离性(Isolation)：事务的执行不受其他事务的干扰，事务执行的中间结果对其他事务必须是透明的。

4、持久性(Durability)：对于任意已提交事务，系统必须保证该事务对数据库的改变不被丢失，即使数据库出现故障

3.1.10简述什么是存储过程和触发器？

存储过程：是数据库中的一个对象，Transact-SQL 语句的预编译集合，这些语句在一个名称下存储并作

为一个单元进行处理。（可以理解为 C 语言中的函数，有参数、返回值等函数特性）

触发器是一种特殊类型的存储过程，当使用下面的一种或多种数据修改操作在指定表中对数据进行修改时，触发器会生效：UPDATE、INSERT 或 DELETE。

3.1.11什么是数据库索引？

数据库索引，是数据库管理系统中一个排序的数据结构，以协助快速查询、更新数据库表中数据。索引的实现通常使用 B_TREE。B_TREE 索引加速了数据访问，因为存储引擎不会再去扫描整张表得到需要的数据；相反，它从根节点开始，根节点保存了子节点的指针，存储引擎会根据指针快速寻找数据。

3.1.12数据库怎么优化查询效率？

1、储存引擎选择：如果数据表需要事务处理，应该考虑使用 InnoDB，因为它完全符合 ACID 特性。如果不需要事务处理，使用默认存储引擎 MyISAM 是比较明智的

2、分表分库，主从。

3、对查询进行优化，要尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引

4、应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描

5、应尽量避免在 where 子句中使用!= 或<>操作符，否则将引擎放弃使用索引而进行全表扫描

6、应尽量避免在 where 子句中使用 or 来连接条件，如果一个字段有索引，一个字段没有索引，将导致引擎放弃使用索引而进行全表扫描

7、Update 语句，如果只更改 1、2 个字段，不要 Update 全部字段，否则频繁调用会引起明显的性能消耗，同时带来大量日志

8、对于多张大数据量（这里几百条就算大了）的表 JOIN，要先分页再 JOIN，否则逻辑读会很高，性能很差。

3.1.13你用的 Mysql 是哪个引擎，各引擎之间有什么区别？

主要 MyISAM 与 InnoDB 两个引擎，其主要区别如下：InnoDB 支持事务，MyISAM 不支持，这一

点是非常之重要。事务是一种高级的处理方式，如在一些列增删改中只要哪个出错还可以回滚还原，而 MyISAM 就不可以了；

MyISAM 适合查询以及插入为主的应用，InnoDB 适合频繁修改以及涉及到安全性较高的应用；

InnoDB 支持外键，MyISAM 不支持；

MyISAM 是默认引擎，InnoDB 需要指定； InnoDB 不支持 FULLTEXT 类型的索引；

InnoDB 中不保存表的行数，如 select count() from table 时，InnoDB；需要扫描一遍整个表来计算有多少行，但是 MyISAM 只要简单的读出保存好的行数即可。注意的是，当 count()语句包含 where 条件时 MyISAM 也需要扫描整个表；

对于自增长的字段，InnoDB 中必须包含只有该字段的索引，但是在 MyISAM 表中可以和其他字段一起建立联合索引；清空整个表时，InnoDB 是一行一行的删除，效率非常慢。MyISAM 则会重建表；

InnoDB 支持行锁（某些情况下还是锁整表，如 update table set a=1 where user like '%lee%'

3.1.14如何对查询命令进行优化？

a.应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索。

b.应尽量避免在 where 子句中对字段进行 null 值判断，避免使用!=或<>操作符，避免使用 or 连接条件，或在 where 子句中使用参数、对字段进行表达式或函数操作，否则会导致权标扫描

c.不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引。

d.使用索引字段作为条件时，如果该索引是复合索引，那么必须使用到该索引中的第一个字段作为条件时才能保证系统使用该索引，否则该索引将不会被使用。

e.很多时候可考虑用 exists 代替 in。

f.尽量使用数字型字段。

g.尽可能的使用 varchar/nvarchar 代替 char/nchar。

h.任何地方都不要使用 select from t ，用具体的字段列表代替“”，不要返回用不到的任何字段。

尽量使用表变量来代替临时表。

j. 避免频繁创建和删除临时表，以减少系统表资源的消耗。

k. 尽量避免使用游标，因为游标的效率较差。

l.在所有的存储过程和触发器的开始处设置 SET NOCOUNT ON ，在结束时设置 SETNOCOUNT OFF。

m.尽量避免大事务操作，提高系统并发能力。

n.尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理。

3.1.15数据库的优化？

1.优化索引、SQL 语句、分析慢查询；

2.设计表的时候严格根据数据库的设计范式来设计数据库；

3.使用缓存，把经常访问到的数据而且不需要经常变化的数据放在缓存中，能节约磁盘 IO

4.优化硬件；采用 SSD，使用磁盘队列技术(RAID0,RAID1,RDID5)等

5.采用 MySQL 内部自带的表分区技术，把数据分层不同的文件，能够提高磁盘的读取效率；

6.垂直分表；把一些不经常读的数据放在一张表里，节约磁盘 I/O；

7.主从分离读写；采用主从复制把数据库的读操作和写入操作分离开来；

8.分库分表分机器（数据量特别大），主要的原理就是数据路由；

9.选择合适的表引擎，参数上的优化

10.进行架构级别的缓存，静态化和分布式；

11.不采用全文索引；

12.采用更快的存储方式，例如 NoSQL 存储经常访问的数据。

3.1.16Sql 注入是如何产生的，如何防止？

程序开发过程中不注意规范书写 sql 语句和对特殊字符进行过滤，导致客户端可以通过全局变量 POST 和 GET 提交一些 sql 语句正常执行。产生 Sql 注入。下面是防止办法：

a.过滤掉一些常见的数据库操作关键字，或者通过系统函数来进行过滤。

b.在 PHP 配置文件中将 Register_globals=off;设置为关闭状态

c.SQL 语句书写的时候尽量不要省略小引号(tab 键上面那个)和单引号

d.提高数据库命名技巧，对于一些重要的字段根据程序的特点命名，取不易被猜到的

e.对于常用的方法加以封装，避免直接暴漏 SQL 语句

f.开启 PHP 安全模式：Safe_mode=on;

g.打开 magic_quotes_gpc 来防止 SQL 注入

h.控制错误信息：关闭错误提示信息，将错误信息写到系统日志。

i.使用 mysqli 或 pdo 预处理。

3.1.17NoSQL 和关系数据库的区别？

a.SQL 数据存在特定结构的表中；而 NoSQL 则更加灵活和可扩展，存储方式可以省是 JSON 文档、哈希表或者其他方式。

b.在 SQL 中，必须定义好表和字段结构后才能添加数据，例如定义表的主键(primary key)，索引(index),触发器(trigger),存储过程(stored procedure)等。表结构可以在被定义之后更新，但是如果有比较大的结构变更的话就会变得比较复杂。在 NoSQL 中，数据可以在任何时候任何地方添加，不需要先定义表。

c.SQL 中如果需要增加外部关联数据的话，规范化做法是在原表中增加一个外键，关联外部数据表。而在 NoSQL 中除了这种规范化的外部数据表做法以外，我们还能用如下的非规范化方式把外部数据直接放到原数据集中，以提高查询效率。缺点也比较明显，更新审核人数据的时候将会比较麻烦。

d.SQL 中可以使用 JOIN 表链接方式将多个关系数据表中的数据用一条简单的查询语句查询出来。NoSQL 暂未提供类似 JOIN 的查询方式对多个数据集中的数据做查询。所以大部分 NoSQL 使用非规范化的数据存储方式存储数据。

e.SQL 中不允许删除已经被使用的外部数据，而 NoSQL 中则没有这种强耦合的概念，可以随时删除任何数据。

f. SQL 中如果多张表数据需要同批次被更新，即如果其中一张表更新失败的话其他表也不能更新成功。这种场景可以通过事务来控制，可以在所有命令完成后再统一提交事务。而 NoSQL 中没有事务这个概念，每一个数据集的操作都是原子级的。

g.在相同水平的系统设计的前提下，因为 NoSQL 中省略了 JOIN 查询的消耗，故理论上性能上是优于 SQL 的 。

3.1.18MySQL 与 MongoDB 本质之间最基本的差别是什么

差别在多方面，例如：数据的表示、查询、关系、事务、模式的设计和定义、速度和性能。MongoDB 是由 C++语言编写的，是一个基于分布式文件存储的开源数据库系统。在高负载的情况下，添加更多的节点， 可以保证服务器性能。

MongoDB 旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。

MongoDB 是一个面向文档的数据库，目前由 10gen 开发并维护，它的功能丰富齐全，所以完全可以替代 MySQL。

与 MySQL 等关系型数据库相比，MongoDB 的优点如下：

①弱一致性，更能保证用户的访问速度。

②文档结构的存储方式，能够更便捷的获取数据。

③内置 GridFS，支持大容量的存储。

④内置 Sharding。

⑤第三方支持丰富。(这是与其他的 NoSQL 相比，MongoDB 也具有的优势)

⑥性能优越：

MongoDB 本身它还算比较年轻的一个产品，所以它的问题，就是成熟度肯定没有传统 MySQL 那么成熟稳定。所以在使用的时候：

尽量使用稳定版，不要在线上使用开发版，这是一个大原则；

另外一点，备份很重要，MongoDB 如果出现一些异常情况，备份一定是要能跟上。除了通过传统的复制的方式来做备份，离线备份也还是要有，不管你是用什么方式，都要有一个完整的离线备份。往往最后出现了特殊情况，它能帮助到你；另外，MongoDB 性能的一个关键点就是索引，索引是不是能有比较好的使用效率，索引是不是能够放在内存中，这样能够提升随机读写的性能。如果你的索引不能完全放在内存中， 一旦出现随机读写比较高的时候，它就会频繁地进行磁盘交换，这个时候，MongoDB 的性能就会急剧下降， 会出现波动。

另外，MongoDB 还有一个最大的缺点，就是它占用的空间很大，因为它属于典型空间换时间原则的类型。那么它的磁盘空间比普通数据库会浪费一些，而且到目前为止它还没有实现在线压缩功能，

在 MongoDB 中频繁的进行数据增删改时，如果记录变了，例如数据大小发生了变化，这时候容易产生一些数据碎片，出现碎片引发的结果，一个是索引会出现性能问题。

另外一个就是在一定的时间后，所占空间会莫名其妙地增大，所以要定期把数据库做修复，定期重新做索引，这样会提升 MongoDB 的稳定性和效率。在最新的版本里，它已经在实现在线压缩，估计应该在 2.0 版左右，应该能够实现在线压缩，可以在后台执行现在 repair DataBase 的一些操作。

如果那样，就解决了目前困扰我们的大问题。

3.1.19Mysql 数据库中怎么实现分页？

select * from table limit (start-1)*limit,limit;

其中 start 是页码，limit 是每页显示的条数。

3.1.20Mysql 数据库的操作?

修改表-修改字段，重命名版：

alter table 表名 change 原名新名类型及约束；

alter table students change birthday birth datetime not null; 修改表-修改字段，不重名版本：

alter table 表名 modify 列名类型和约束；alter table students modify birth date not null 全列插入：insert into 表名 values(...)

insert into students values(0,"郭靖", 1,"内蒙","2017-6"); 部分插入：值的顺序与给出的列顺序对应：

insert into students(name, birthday) values("黄蓉","2017-8"); 修改：update 表名 set 列 1=值 1，列 2=值 2.。。where

update students set gender=0, homwtown="古墓"， where id = 5; 备份：mysqldump -uroot -p 数据库名>python.sql,

恢复：mysql -uroot -p 数据库名< python.sql

3.1.21优化数据库？提高数据库的性能？

1.对语句的优化

①用程序中，保证在实现功能的基础上，尽量减少对数据库的访问次数；通过搜索参数，尽量减少对表的访问行数,最小化结果集，从而减轻网络负担；

②能够分开的操作尽量分开处理，提高每次的响应速度；在数据窗口使用 SQL 时，尽量把使用的索引放在选择的首列；算法的结构尽量简单；

③在查询时，不要过多地使用通配符如 SELECT * FROM T1 语句，要用到几列就选择几列如：SELECT COL1,COL2 FROM T1；

④在可能的情况下尽量限制尽量结果集行数如：SELECT TOP 300 COL1,COL2,COL3 FROMT1,因为某些情况下用户是不需要那么多的数据的。

⑤不要在应用中使用数据库游标，游标是非常有用的工具，但比使用常规的、面向集的 SQL 语句需要更大的开销；按照特定顺序提取数据的查找。

2.避免使用不兼容的数据类型

例如 float 和 int、char 和 varchar、binary 和 varbinary 是不兼容的。数据类型的不兼容可能使优化器无法执行一些本来可以进行的优化操作。例如:

SELECT name FROM employee WHERE salary ＞ 60000

在这条语句中,如 salary 字段是 money 型的,则优化器很难对其进行优化,因为 60000 是个整型数。我们应当在编程时将整型转化成为钱币型,而不要等到运行时转化。若在查询时强制转换，查询速度会明显减慢。

3.避免在 WHERE 子句中对字段进行函数或表达式操作。

若进行函数或表达式操作，将导致引擎放弃使用索引而进行全表扫描。

4.避免使用!=或＜＞、IS NULL 或 IS NOT NULL、IN ， NOT IN 等这样的操作符

5.尽量使用数字型字段

6.合理使用 EXISTS,NOT EXISTS 子句。

7.尽量避免在索引过的字符数据中，使用非打头字母搜索。

8.分利用连接条件

9.消除对大型表行数据的顺序存取

10.避免困难的正规表达式

11.使用视图加速查询

12.能够用 BETWEEN 的就不要用 IN

13.DISTINCT 的就不用 GROUP BY

14.部分利用索引

15.能用 UNION ALL 就不要用 UNION

16.不要写一些不做任何事的查询

17.尽量不要用 SELECT INTO 语句

18.必要时强制查询优化器使用某个索引

19.虽然 UPDATE、DELETE 语句的写法基本固定，但是还是对 UPDATE 语句给点建议：

a)尽量不要修改主键字段。

b)当修改 VARCHAR 型字段时，尽量使用相同长度内容的值代替。

c)尽量最小化对于含有 UPDATE 触发器的表的 UPDATE 操作。

d)避免 UPDATE 将要复制到其他数据库的列。

e)避免 UPDATE 建有很多索引的列。

f)避免 UPDATE 在 WHERE 子句条件中的列。

3.1.22什么是数据的完整性？

数据完整性指的是存储在数据库中的数据的一致性和准确性。

完整性分类：

（1）实体完整性：主键值必须唯一且非空。（主键约束）

（2） 引用完整性（也叫参照完整性）：外键要么为空，要么引用主表中存在的记录。（外键约束）。

（3）用户自定义完整性：针对某一具体关系数据库中的约束条件。

3.1.23存储过程和函数的区别?

相同点：存储过程和函数都是为了可重复的执行操作数据库的 sql 语句的集合。

1）存储过程和函数都是一次编译，就会被缓存起来，下次使用就直接命中已经编译好的 sql 语句，不需要重复使用。减少网络交互，减少网络访问流量。

不同点：标识符不同，函数的标识符是 function，存储过程是 proceduce。

1）函数中有返回值，且必须有返回值，而过程没有返回值，但是可以通过设置参数类型（in,out)来实现多个参数或者返回值。

2）存储函数使用 select 调用，存储过程需要使用 call 调用。

3）select 语句可以在存储过程中调用，但是除了 select..into 之外的 select 语句都不能在函数中使用。

4）通过 in out 参数，过程相关函数更加灵活，可以返回多个结果。

3.1.24怎么进行 SQL 的查询优化？

（1）从表连接的角度优化：尽量使用内连接，因为内连接是两表都满足的行的组合，而外连接是以其中一个表的全部为基准。

（2）尽量使用存储过程代替临时写 SQL 语句：因为存储过程是预先编译好的 SQL 语句的集合，这样可以减少编译时间。

（3）从索引的角度优化：对那些常用的查询字段简历索引，这样查询时值进行索引扫描，不读取数据块。

（4）还有一些常用的 select 优化技巧：

（5）A.只查询那些需要访问的字段，来代替 select*

B、将过滤记录越多的 where 语句向前移：在一个 SQL 语句中，如果一个 where 条件过滤的数据库记录越多，定位越准确，则该 where 条件越应该前移。

3.1.25索引的作用，聚集索引与非聚集索引的区别

索引是一个数据库对象，使用索引，可以是数据库程序无须对整个数据进行扫描，就可以在其中

找到目标数据，从而提高查找效率。索引的底层采用的是 B 树。

聚集索引：根据记录的 key 再表中排序数据行。

非聚集索引：独立于记录的结构，非聚集所以包含的 key，且每个键值项都有指向该简直的数据行的指针。

聚集索引与非聚集索引的区别：

（1）聚集索引的物理存储按索引排序，非聚集所以的物理存储不按索引排序。

（2） 聚集索引插入，更新数据的速度比非聚集索引慢，单查询速度更快。

（3）聚集索引的叶级结点保存的是时间的数据项，而非聚集结点的叶级结点保存的是指向数据项的指针。

（4）一个表只能有一个聚集索引（因为只有一种排序方式），但可以有多个非聚集索引。

3.2 查询练习

3.2.1 Student-Sourse-SC-Teacher 表关系如下：

Student（sid，Sname，Sage，Ssex）学生表

Course（cid，Cname，tid）课程表

SC（sid，cid，score）成绩表

Teacher（tid，Tname）教师表写出 sql 语句：

1. 查询课程“001“课程比”002“课程成绩高的所有学生的学号
2. 修改学号为 20131201 的语文成绩为 100‘
3. 插入一条名为“李四”的教师记录
4. 删除学习“叶平”老师课程的 sc 表记录

3.2.2 员工信息 A-员工亲属信息表 B 表关系如下：

员工信息表 A：员工标号（codecode，PK），员工姓名（codename），员工性别（codesex），联系

电话（codetel），备注（remarks）

员工亲属信息表 B：员工编码（codecode），亲属编码（recodecode，PK），亲属姓名（recodename），联系电话（recodetel），备注（remarks）

写出 sql 语句：

1. 向员工信息表中插入一条数据：（001，张三，男，010-62570007，北京市海淀区）
2. 查询出亲属数量大于 1 的员工编码，员工姓名，员工亲属数量有部分员工亲属信息重复录入（亲属编码不同，其他相同），出现这种情况的员工编码，重复的亲属编码，亲属姓名查询出来。

3.2.3 部门表 dept-雇员表 emp 表关系如下：

部门表 dept：部门标号（DEPTNO），部门名称（DNAME），所在位置（LOC）

雇员表 emp：员工标号（Empno），员工名称（Emname），员工工位（Job），经理（Mgr），雇佣日期（Hiredate），薪水（Sal），部门编号（Deptno）

写出 sql 语句：

1. 找出部门名称为 ACCOUNTING 的部门下的所有员工名称？
2. 找出部门名称为 SALES 的部门下每月需要发出的薪水总额？
3. 找出部门名称为 SALES 的部门的部门经理？
4. 找出部门名称为 RESEARCH 的部门下雇佣日期为 1980-12-17 的员工？

3.2.4 Student-coures-Studentcourse 表关系如下：

student(sno,sname,age,sdept) 学生表

course(cno,cname,teacher) 课程表

Studentcourse(sno,cno,grade)选课表写出 sql 语句：

1. 查询所有课程都及格的学生号和姓名
2. 查询平均分不及格的课程号和平均成绩
3. 找出各门课程的平均成绩，输出课程号和平均成绩
4. 找出没有选择 c2 课程的学生信息

3.2.5 看下图回答问题

A1

uid (用户id)	user (用户名)	post (帖子数)	hits (点击数)	data (日期)
1	小张	90	983	2012-03-24
2	小王	123	243	2014-06-21
3	小三	998	100	2014-06-21
4	小三	782	983	2014-06-03
5	小三	344	334	2014-07-02
...				

1. 请用一条 SQL 语句查询出发帖数大于 5 点击数由高到低排序？
2. 用一条 SQL 语句将“日期”为 21 日的记录帖子数全部设置为 0？
3. 用一条 SQL 筛选出 2014 年的记录且 post 大约 500 的数据，将其 hits 减少 50？

3. 2. 6 SQL 操作，有两张表，如下图所示

订单表：A

Order_id	User_id	Add_time
11701245001	10000	1498882474
11701245002	10001	1498882475

订单明细表：B

Id	Order_id	Goods_id	price
1	11701245001	1001	10
2	11701245001	1002	20
3	11701245002	1001	10

1. 用 SQL 查询购买过 goods_id 为 1001 的用户的 user_id
2. 用 SQL 查询 2017 年 7 月 1 号后（含 7 月 1 号）购买过 1001 这个商品的 user_id 和 oeder_id, goods_id 和 price
3. 用 SQL 查询出订单所含商品明细总金额）=50 的 order_id 和 user_id

3.2.7 下面是学生成绩表（score）结构说明

字段名称	字段解释	字段类型	字段长度	约束
sc_number	学号	字符	8	PK
sc_name	姓名	字符	50	Not null
sc_sex	性别	字符(男：1；女：0)	2	Not null
sc_courseid	课程号	字符	5	PK
sc_score	分数	数值	3	Not null
sc_ismakeup	当前考试是否为补考	字符(补考：1，非补考：0)	2	Not null

下面是课程表（course）说明

字段名称	字段解释	字段类型	字段长度	约束
co_id	课程号	字符	5	PK
co_name	课程名	字符	3	Not null
co_desc	课程介绍	字符	60	

1）请说明主键，外键的作用，以及建立索引的好处及坏处。

2）请写出课程表中建表 SQL 语句。

3）如果学号的前两位表示年级，要查找 98 级女生的姓名，请写出相应的 SQL 语句。

4）要查找所需要补考（小于 60 分）的学生姓名和这门课程的名称和成绩，请写出相应的 SQL 语句。

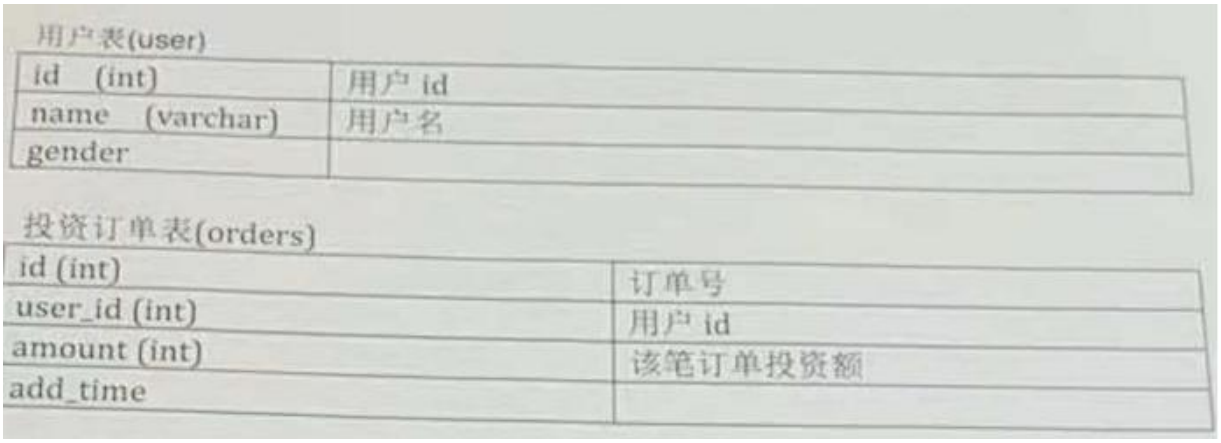
5）查询每个学生需要补考（小于 60 分）的课程的平均分，并以平均分排序。

6）（非必答题，请量力选做）请分析这两个表的设计，并谈谈这两个表设计中存在的问题，并根据你的理解给出优化的方案。

7）（非必答题，请量力选做）针对学生考试管理系统，如果要实现学生管理，课程管理，考试成绩管理的基本需求，请根据你对需求的理解，使用 UML 或者 E-R 图的方式给出一个简洁明了的数据库设计方案。

3.2.8 懒投资首页的懒人播报，统计了在懒投资平台的投资富豪榜，对应的库

表简化如下：



用户表(user)

id (int)	用户 id
name (varchar)	用户名
gender	

投资订单表(orders)

id (int)	订单号
user_id (int)	用户 id
amount (int)	该笔订单投资额
add_time	

富豪榜统计需求：

1）请给出，所有投资用户中，投资总额排名前 10 位的用户，按投资总额倒序排列，输出项如下，例如：

用户名，投资总额张 三 ，9000000 李四，8000000

……

2）给出富豪榜第一名的用户的单笔平均投资额

3.3 万年学生表经典面试题汇总

有 student 表和 score，创建表的语法如下，完成数据库题目

创建 student

```
CREATE or REPLACE TABLE   student (
    id   INT(10)   NOT NULL   UNIQUE   PRIMARY KEY   ,
    name   VARCHAR(20)   NOT NULL ,
    sex   VARCHAR(4)   ,
    birth   YEAR,
    department   VARCHAR(20) ,
    address   VARCHAR(50)
```

```
);
```

创建 score 表：

```
CREATE or REPLACE TABLE   score (
      id   INT(10)   NOT NULL   UNIQUE   PRIMARY KEY   AUTO_INCREMENT ,
      stu_id   INT(10)   NOT NULL ,
      c_name   VARCHAR(20) ,
      grade   INT(10)
);
```

3.3.1 为 student 表和 score 表增加记录

```
#向 student 表插入记录：
INSERT INTO student VALUES( 901,'张老大', '男',1985,'计算机系', '北京市海淀区');
INSERT INTO student VALUES( 902,'张老二', '男',1986,'中文系', '北京市昌平区');
INSERT INTO student VALUES( 903,'张三', '女',1990,'中文系', '湖南省永州市');
INSERT INTO student VALUES( 904,'李四', '男',1990,'英语系', '辽宁省阜新市');
INSERT INTO student VALUES( 905,'王五', '女',1991,'英语系', '福建省厦门市');
INSERT INTO student VALUES( 906,'王六', '男',1988,'计算机系', '湖南省衡阳市');

#向 score 表插入记录：
INSERT INTO score VALUES(NULL,901, '计算机',98);
INSERT INTO score VALUES(NULL,901, '英语', 80);
INSERT INTO score VALUES(NULL,902, '计算机',65);
INSERT INTO score VALUES(NULL,902, '中文',88);
INSERT INTO score VALUES(NULL,903, '中文',95);
INSERT INTO score VALUES(NULL,904, '计算机',70);
INSERT INTO score VALUES(NULL,904, '英语',92);
INSERT INTO score VALUES(NULL,905, '英语',94);
```

```
INSERT INTO score VALUES(NULL,906, '计算机',90);
INSERT INTO score VALUES(NULL,906, '英语',85);
```

3.3.2 查询 student 表的所有记录

```
SELECT * FROM student;
```

3.3.3 查询 student 表的第 2 条到 4 条记录

```
SELECT * FROM student LIMIT 1,3;
```

说明：

LIMIT 子句可以被用于强制 SELECT 语句返回指定的记录数。LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。初始记录行的偏移量是 0(而不是 1)

3.3.4 从 student 表查询所有学生的学号（id）、姓名（name）和院系（department）的信息

```
SELECT id,name,department FROM student;
```

3.3.5 从 student 表中查询计算机系和英语系的学生的信息

```
SELECT *
FROM student
 WHERE department IN ('计算机系','英语系');
```

3.3.6 从 student 表中查询年龄 18~22 岁的学生信息

```
SELECT id,name,sex,2018-birth AS age,
department,address
FROM student
WHERE 2018-birth BETWEEN  18 AND 22;
```

```
SELECT id,name,sex,2018-birth AS age,
department,address
FROM student
WHERE 2018-birth>=18 AND 2018-birth<=22;
```

3.3.7 从 student 表中查询每个院系有多少人

```
SELECT department, COUNT(id)
FROM student
GROUP BY department;
```

3.3.8 从 score 表中查询每个科目的最高分

```
SELECT c_name,MAX(grade)
FROM score
GROUP BY c_name;
```

3.3.9 计算每个考试科目的平均成绩

```
SELECT c_name,AVG(grade)
FROM score GROUP BY c_name;
```

3.3.10将计算机考试成绩按从高到低进行排序

```
SELECT stu_id, grade
FROM score
WHERE c_name= '计算机'
ORDER BY grade DESC;
```

desc 降序排列 esc：升序排列

3.3.11查询 student 表中学生的学号、姓名、年龄、院系和籍贯并且按照年龄

从小到大的顺序排列

```
select student.id,name,2017-birth,department,address
from student
where 2017-birthORDER BY 2017-birth
```

3.3.12查询 score 表中学生的学号、考试科目和成绩并且按照成绩从高到低的顺序排列。

```
select score.stu_id,c_name,grade
from score
ORDER BY grade DESC
```

数据检索-多表查询

3.3.13查询李四的考试科目（c_name）和考试成绩（grade）

```
SELECT c_name, grade
FROM score
WHERE
  stu_id=(SELECT id
            FROM student
            WHERE name= '李四' );
```

3.3.14用连接的方式查询所有学生的信息和考试信息

```
SELECT student.id,name,sex,birth,
department,address,c_name,grade
FROM student,score
WHERE student.id=score.stu_id;
```

作业：左连接右链接，内连接和外链接的区别。

3.3.15计算每个学生的总成绩

```
SELECT student.id,name,SUM(grade)
FROM student,score
WHERE student.id=score.stu_id
GROUP BY id;
```

3.3.16查询计算机成绩低于 95 的学生信息

```
SELECT * FROM student
WHERE id IN
(SELECT stu_id FROM score
WHERE c_name="计算机" and grade<95);
```

3.3.17查询同时参加计算机和英语考试的学生的信息

```
SELECT *   FROM student
WHERE
id =ANY( SELECT stu_id
                FROM score
                WHERE stu_id IN (
                        SELECT stu_id FROM
                        score WHERE c_name=   '计算机')
                        AND c_name= '英语' );


SELECT a.* FROM student a ,score b ,score c
WHERE
a.id=b.stu_id
AND b.c_name='计算机'
AND a.id=c.stu_id
```

```
AND c.c_name='英语';
```

3.3.18从 student 表和 score 表中查询出学生的学号，然后合并查询结果

```
SELECT id  FROM student
UNION
SELECT stu_id  FROM score;
```

3.3.19查询姓张或者姓王的同学的姓名、院系和考试科目及成绩

```
SELECT student.id, name,sex,birth,
department, address, c_name,grade
FROM student, score
WHERE
(name LIKE  '张%'  OR name LIKE  '王%')
AND
student.id=score.stu_id ;
```

3.3.20查询都是湖南的学生的姓名、年龄、院系和考试科目及成绩

```
SELECT student.id, name,sex,birth,department, address, c_name,grade
FROM student, score
WHERE address LIKE '湖南%'  AND
student.id=score.stu_id;
```

3.4 数据库企业真题

3.4.1 请编写 SQL 语句:

1）创建一张学生表，包含以下信息，学号，姓名，年龄，性别，家庭住址，联系电话

```
CREATE TABLE Student(
        s_id int(5),
```

```
s_name VARCHAR(20),
s_age INT(3),
s_sex char(2),
s_address VARCHAR(20),
s_phone char(11)
    )
```

2）修改学生表的结构，添加一列信息，学历

```
alter TABLE Student add s_ed VARCHAR(10);
```

3）修改学生表结构，删除一列信息，家庭住址

```
ALTER TABLE Student DROP COLUMN s_address;
```

4）向学生表添加如下信息： 学号 姓名 年龄 性别 联系电话 学历

1A22 男 123456 小学；2B21 男 119 中学；

3C23 男 110 高中；4D18 女 114 大学

```
INSERT into Student VALUES(1,'A',22,'男','123456','小学');
INSERT into Student VALUES(2,'B',21,'男','119','中学');
INSERT into Student VALUES(3,'C',23,'男','110','高中');
INSERT into Student VALUES(4,'B',18,'女','114','大学');
```

5)修改学生表的数据，将电话号码以 11 开头的学员的学历改为“大专”

```
UPDATE Student SET s_ed='大专' WHERE s_phone like '11%';
```

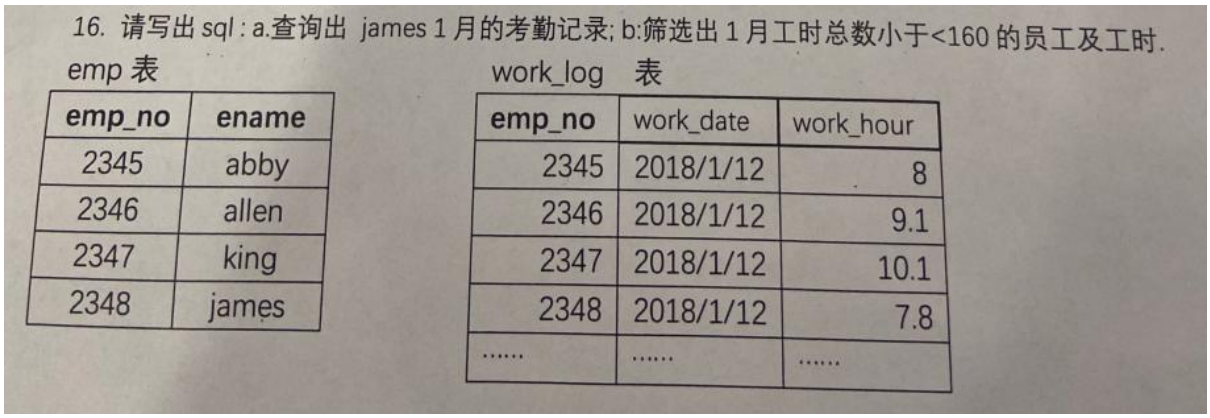
6)删除学生表的数据，姓名以 C 开头，性别为“男”的记录删除

```
DELETE FROM Student where s_sex='男' and s_name like 'C%';
```

7）查询学生表的数据，将所有年龄小于 22 岁的，学历为“大专”的，学生的姓名和学号示出来

```
SELECT s_name,s_id FROM Student WHERE s_sex < 22 and s_ed='大专';
```

3.4.2 写入如下 SQL 语句



16. 请写出 sql：a.查询出 james 1 月的考勤记录; b:筛选出 1 月工时总数小于<160 的员工及工时.

emp 表

emp_no	ename
2345	abby
2346	allen
2347	king
2348	james

work_log 表

emp_no	work_date	work_hour
2345	2018/1/12	8
2346	2018/1/12	9.1
2347	2018/1/12	10.1
2348	2018/1/12	7.8
.....	.....	.....

a：

```
select e.emp_no,e.ename,w.work_date,w.work_hour
from emp as e join work_log as w
where e.emp_no=w.emp_no
and e.ename ='james'
and month(w.work_date)=1;
```

b:

```
select e.emp_no,e.ename,sum(w.work_hour) as 工时
from emp as e join work_log as w
where e.emp_no=w.emp_no
and month(w.work_date)=1
group by e.emp_no
having sum(w.work_hour)<160；
```

3.4.3 编写 SQL

学生表(t_student)：

ID	NAME

课程分数表(t_score)：

ID	COURSE	SCORE

（1）查出班级中所有重名的学生

（2）按照课程总分数排名，找出排名前十的学生

（1） select ID,NAME from t_student group by NAME having count(1)>1;

（2） select t1.ID,t1.NAME from t_student as t1,t_score as t2

where t1.ID=t2.ID

group by t1.ID

order by sum(t2.SCORE) desc

limit 0,10;

此题考察分组、排序和分页能够做到上面这步已经是满分了，如果考虑总分相同属于同一名次，那么 SQL 可以这样写：

```
select t3.ID,t3.NAME from
(select t1.ID,t1.NAME,sum(t2.SCORE) as socre1 from t_student as t1,t_score as t2
where t1.ID=t2.ID
group by t1.ID) as t3
join (
select distinct sum(SCORE) as score2
from t_student as t1,t_score as t2
where t1.ID=t2.ID
group by t1.ID
order by sum(SCORE) desc
limit 0,10) as t4
```

where t3.score1 = t4.score2;

3.4.4 用一条 SQL 语句：查询出每门课都大于 80 分的学生姓名(表面：TestScores)

Name	Course	Score
张三	语文	81
张三	数学	75
李四	语文	76
李四	语文	76
王五	语文	81
王五	数学	100
王五	英语	90

参考答案：

方法一：使用 NOT IN 排除课程分数有小于等于 80 的同学

```
select distinct Name from TestScores where Name not in
(select Name from TestScores where Score <= 80);
```

方法二：使用 EXISTS 子查询排除有课程分数小于等于 80 的同学

```
select distinct Name from TestScores as t1
where not exists
(select * from TestScores as t2 where t2.Score<=80
  and t1.Name=t2.Name);
```

方法三：逆向思维，每门课都大于 80 分，表示所有课程最低分数也大于 80 分

```
select Name,min(Score)
from TestScores
```

```
group by Name
having min(Score) >80;
```

方法四：自表关联，某同学的所有课程数等于其大于 80 分的课程数，那么该同学的所有课程都大于 80 分

```
select t1.Name from
(select Name,count(1) as toatalNum from TestScores group by Name) as t1,
(select Name,count(1) as above80Num from TestScores where Score>=80 group by Name) as t2
where t1.Name=t2.Name and t1.toatalNum=t2.above80Num;
```

方法五：使用左连接过滤出课程分数没有小于或等于 80 分的同学

```
select distinct t1.Name from TestScores as t1
left join (select * from TestScores where Score <=80) as t2
on t1.Name = t2.Name
where t2.Name is null;
```

3.4.5 SQL 试题

学生表(学生 id，姓名，性别，分数)　student(s_id,name,sex,score)

班级表（班级 id,班级名称），class（c_id,c_name）

学生班级表（班级 id，学生 id），student_class（s_id,c_id）

（1） 查询一班得分在 80 分以上或者等于 60,61,62 的学生

（2） 查询所有班级的名称，和所有班中女生人数和女生的平均分

（1）

```
select * from student a ,class b,student_class c
    where a.s_id = c.s_id  and b.c_id = c.c_id
    and c_name = '一班'
    and (a.score > 80 or a.s_id in (60,61,62))
```

（2）

```
select c_name,count(*),avg(a.score)
    from student a ,class b,student_class c
```

```
where a.s_id = c.s_id  and b.c_id = c.c_id
and a.sex='女'
group by c_name;
```

IN 或 NOT IN 的语法为：SELECT * FROM table_name WHERE field [NOT] IN(data_list)。其中 data_list 可以是具体的数值，也可以是通过子查询得到的数据集，IN 相当于多个 OR 条件的叠加。

3.4.6 MySQL/Oracle 如何查询某一个表的前 10 行记录

查询 MySQL 表的前 10 条记录：

```
select * from  表名  limit 0,10;
select * from  表名  limit 10 offset 0;
select * from  表名  limit 10;
```

查询 Oracle 表的前 10 条记录：

```
select * from  表名 where rownum<=10;
```

3.4.7 SQL 面试题

已知如下表结构:

项目表 *prj*

字段名	中文名
prj_id	项目编号
prj_name	项目名称
prj_num	项目金额
per_id	项目经理(人员编码)

per_lev	项目经理等级

人员表 per:

字段名	中文名
per_id	人员编码
per_name	人员姓名
per_job	人员职级(初级、中级、高级)
dep_id	所属部门(部门编码)

部门表 dep:

字段名	中文名
dep_id	部门编码
dep_name	部门名称
dep_lend	部门经理

1. 请使用 SQL 语句查询出所有项目，项目经理的名称及其所属部门的名称

2. 请使用 SQL 语句将没有人员的部门从部门表中删除

3. 请使用 SQL 语句将项目表中的项目经理等级修改为该员工在人员表中的职级。并且将人员表中所有市场部的人员姓名后面加上"-商务"，如"张三-商务"。

参考答案：

```
select prj.Prj_name,per.Per_name,dep.Dep_id from prj
left join per on prj.Per_id = per.Per_id
left join dep on dep.Dep_name=per.Per_name;
```

3.4.8 数据库面试题

1. 进行多表联查时，应该注意什么

2. 如何对查询结果进行条件映射归集

3. 对某个表的值进行按天统计总条数、总金额数量，如何编写这个 SQL

参考答案：

1. 多表联查的关键在于找出表与表之间的关联字段，一般设计表的时候会根据具体业务和模块将数据拆分保存到多个表中间，在一个表中设计字段指代另一个表的字段，从而保证数据间的关系。另外多表联查要根据需要选择合适的连接查询方式，如内连接，左外连接，右外连接等

2. 条件查询相当于对结果集进行的过滤筛选。这时候我们会使用 where 字句，后面使用一个或者多个条件表达式，多个条件表达式之间使用 and 或者 or 进行连接

3. 加上表名为 table_name,日期字段为 create_time,金额字段为 amount，则 sql 如下

```
select date_format(create_time,'%Y%m%d') as 日期
count(1) as 总条数,sum(amount) as 总金额
from table_name
group by date_format(create_time,'%Y%m%d');
```

3.4.9 根据所学的 SQL 知识，写出如下相应的 SQL 语句，要求数据库返回的结果为：删除除了自动编号不同，其他都相同的学生冗余信息

学生表 student 如下：

自动编号	学号	姓名	课程编号	课程名称	分数
1	2005001	张三	0001	数学	69
2	2005002	李四	0001	数学	89
3·	2005001	张三	0001	数学	69

参考答案：

1. 思路：通过对除了自动编号之外的字段进行分组，得到某一条重复信息编号(如编号最大的或者最小的)，再删除每组内除了该条信息外的其它重复的信息

2. 分组获取重复信息最小的(或最大)的自动编号

select min(id) from student group by name,sex,sorce,subject

3. 删除较大自动编号重复信息，剩余每组自动编号最小的记录

delete from student where id not in(select min(id) from student group by name,sex,sorce,subject)

3.4.10SQL 题目：

表 1：student 表

学号 id	姓名 name	性别 sex	系别 dept
1	王鑫	男	计算机系
2	周洋	女·	计算技系
3	肖青萍	女	历史系
4	沈小勤	女	数学系
5	杨世才	男	数学系

表 2，Score 表

学号 id	课程名 classname	成绩 grade
1	C 语言基础	71
1	数据库	65
2	C 语言基础	53

2	数据库	79
3	哲学	77
3	史学概论	91
4	高等数学	75
4	统计学	60
5	高等数学	90
5	统计学	87

（1） 用连接方式查询所有学生的信息和考试成绩信息

（2） 计算每个学生的总成绩

参考答案：

（1） select * from student as s1,score as s2

where s1.id=s2.id;

（2） select s1.name as 姓名,sum(s2.grade) as 分数

from student as 1,score as s2

where s1.id=s2.id

group by s1.id;

3.4.11现有 emp 表，结构及数据如下

EMPNO 员工编号	ENAME 姓名	JOB 职位	MGR 所属上司	BIRTHDAY 出生日期	SAL 工资	DEPTNO 部门号
7369	SIMTH	CLERK	7902	17-Dec-80	800	20

7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	30
7566	JONES	MANAGER	7839	2-Apr-81	2975	20
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	30
7698	BLAKE	MANAGER	7839	1-May-81	2850	30

（1） 计算职位为 SALESMAN 的员工的工资总和

（2） 部门号为 20 的员工，工资从高到底排列

（3） 计算各个部门中，出生日期在 1981 你那 6 月 1 日之后的员工总数

参考答案：

（1） select sum(SAL) 工资总和 from emp where JOB="SELESMAN";

（2） select * from from emp where DEPTNO=20 order by SAL desc;

（3） select count(1) as 员工总数

from emp

where to_date(BIRTHDAY,'dd-mm-yy')>to_date('1981-06-01','dd-mm-yy')

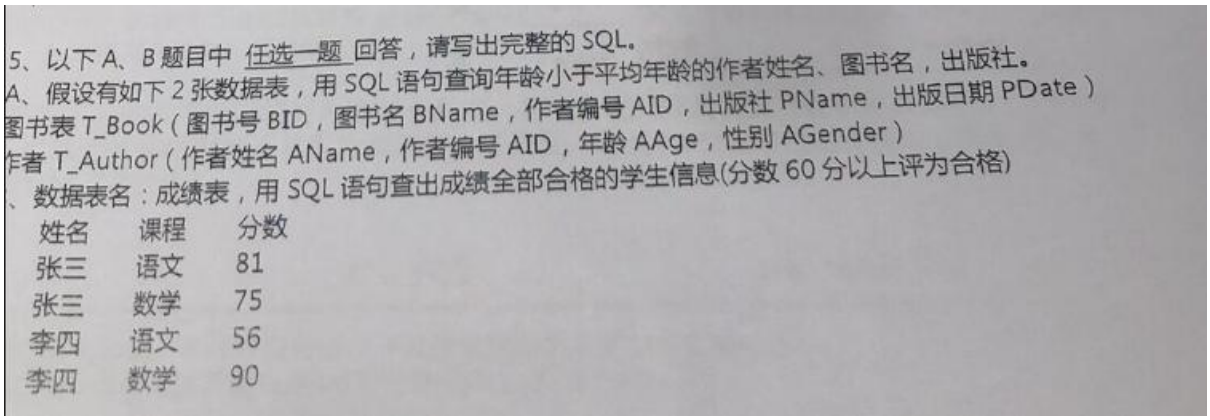
ORDER BY 是 SQL 中间的排序字句，语法为：field [ASC|DESC]...其中 ASC 表示升序，DESC 表示降序，ORDER BY 可以使用任意字段作为排序条件，可以指定多个字段进行排序，如：

SELECT * FROM 表名 ORDER BY 字段 ASC;

SELECT * FROM 表名 ORDER BY 字段 1 ASC,字段 2 DESC;

上面的 SQL 表示先以字段 1 升序，字段 1 相同，再以字段 2 降序进行排序

3.4.12SQL 编写题


5、以下 A、B 题目中 任选一题 回答，请写出完整的 SQL。
A、假设有如下 2 张数据表，用 SQL 语句查询年龄小于平均年龄的作者姓名、图书名，出版社。
图书表 T_Book（图书号 BID，图书名 BName，作者编号 AID，出版社 PName，出版日期 PDate）
作者 T_Author（作者姓名 AName，作者编号 AID，年龄 AAge，性别 AGender）
、数据表名：成绩表，用 SQL 语句查出成绩全部合格的学生信息(分数 60 分以上评为合格)

姓名	课程	分数
张三	语文	81
张三	数学	75
李四	语文	56
李四	数学	90

参考答案：

A：select a.AName,b.BName,b.PName

from T_Book as b join T_Author as a

where b.AID=a.AID

and age<(select avg(AAge) from T_Author);

B：select name

from t_score

group by name

having min(score) >60;

使用逆向思维，所有成绩都合格，表示所有课程最低分数大于 60 分。因此可以对学生按姓名进行分组，设置最低分数大于 60 分的分组条件，筛选出满足要求的学生

3.4.13根据表结构写出 1、2 题的 SQL：

数据库： --Student(S#,Sname,Sage,Ssex) --学生表

--Sourse(C#,Cname,T#) --课程表

--SC(S#,C#,score) --成绩表

--Teacher(T#,Tname) --教师表 --教师表

1、 查询每门课程被选修的学生数

2、 查询课程名称为"编程基础"，且分数不少于 90 分的学生姓名和分数

参考答案：

（1） select C#,count(S#) from sc group by c#;

（2） SELECT Student.Sname,Course.Cname,SC.score

FROM Student,Course,SC

WHERE Course.Cname="编程基础"

AND Student.S#=SC.S#

AND Course.C#=SC.C#

AND SC.score>90;

3.4.14编写 SQL

First Table Name：USER

Table Design：

NAME	TYPE
USER_ID	NUMBER(10,0)
USER_NAME	VARCHAR2(20 BYTE)
USER_BIR_DATE	DATE

* USER_ID is a primary key

TABLE Value

USER_ID	USER_NAME	USER_BIR_DATE

10001	TESTING	2020-01-01
10002	TESTING1	2020-01-02
10003	TESTING2	2020-01-03

Second Table Name：USER_CARD

TABLE Design:

NAME	TYPE
USER_CARD_ID	NUMBER(10,0)
USER_ID	NUMBER(10,0)
CARD_NUMBER	VARCHAR2(20 BYTE)

*USER_CARD_ID is a primary key

TABLE Value:

USER_CARD_ID	USER_ID	CARD_NUMBER
20001	10001	A12345678
20002	10002	A87654321
20003	100003	A24681357

(1) 查询出"USER"表中"USER_ID"为"0001"的数据

(2) 查询出"USER_CARD"表中"USER_ID"属于"USER"表中的数据

(3) 按"USER_ID"降序查找出"USER_CARD"表中"USER_ID"属于"USER"表的数据

(4) 查找出"USER_CARD"表中"USER_ID"属于"USER"表的第一条数据

(5) 按照表的字段类型及条
件"USER_ID"为"0004"，"USER_NAME"为"TESTING3"，"USER_BIR_DATE"为

"2020-01-04",插入到"USER"表中

参考答案：

（1） select * from user where user_id = '0001';

（2） select * from user_card a ,user b where a.user_id = b.user_id;

（3） select * from user_card a ,user b where a.user_id = b.user_id order by a.user_id desc;

（4） select * from user_card a ,user b where a.user_id = b.user_id limit 1;

（5） insert into user values('0004,'TESTING3','2020-01-04');

3.4.15现有以下三张表

第一张表为 cust，其表结构如下：

字段名	字段说明	是否为主键
Studentno	学号，数据类型为整型的	是
Name	学生名字，数据类型为字符串型的	否
Address	学生住址，数据类型为字符串型的	否
Telno	电话号码，数据类型为字符串型的	否
DepartID	系名对应 ID，数据类型为整型	否

第二张表为 mark，其表结构如下：

字段名	字段说明	是否为主键
studentno	学号，数据类型为整型的	是
english	英语成绩，数据类型为字符串型的	否

math	数学成绩，数据类型为字符串型的	否
computer	计算机成绩，数据类型为字符串型的	否

第三章表名为 department，其表结构如下：

DepartID	系名对应 ID，数据类型为整型的	是
DepartName	系名，数据类型为字符串型的	否

（1） 【2 分】请写出计算所有学生的英语平均成绩的 sql 语句

（2） 【5 分】现有五个学生，其学号假定分别为 11,22,33,44,55；请用一条 SQL 语句实现列出这个学生的数学成绩及其姓名、学生地址、电话号码；

（3） 【5 分】查询所有学生的姓名、计算机成绩，按照计算机成绩从高到低排序

（4） 【5 分】查询所有总成绩大于 240 分的学生学号、姓名、总成绩，按照总成绩从高到底排序

（5） 【8 分】将每个系总成绩第一名的学生查出来，查询结果包括系名、学号、姓名、总成绩；

参考答案：

1. select avg(english) from mark;

2. select Name 姓名,Address 地址,Telno 电话号码
from cust where Studentno in(11,22,33,44,55);

3. select c.Name 姓名,m.computer 计算器成绩
from cust c join mark m
on c.Studentno = m.Studentno
order by m.computer desc;

4. select c.Studentno 学号,c.Name 姓名,(m.english+m.math,m.computer) 总成绩

from cust c join mark m

on c.Studentno = m.Studentno

where 总成绩>240

order by 总成绩 desc

5. select t3.DepartName 系名,t1.Studentno 学号,t1.Name 姓名,

(t2.english+t2.math,t2.computer) 总成绩

from cust as t1,mark as t2

(select c.DepartID,c.DepartName

max(m.english+m.math,m.computer) totalScore

from cust c join mark m,department d

on c.Studentno = m.Studentno

and c.DepartID=m.DepartID

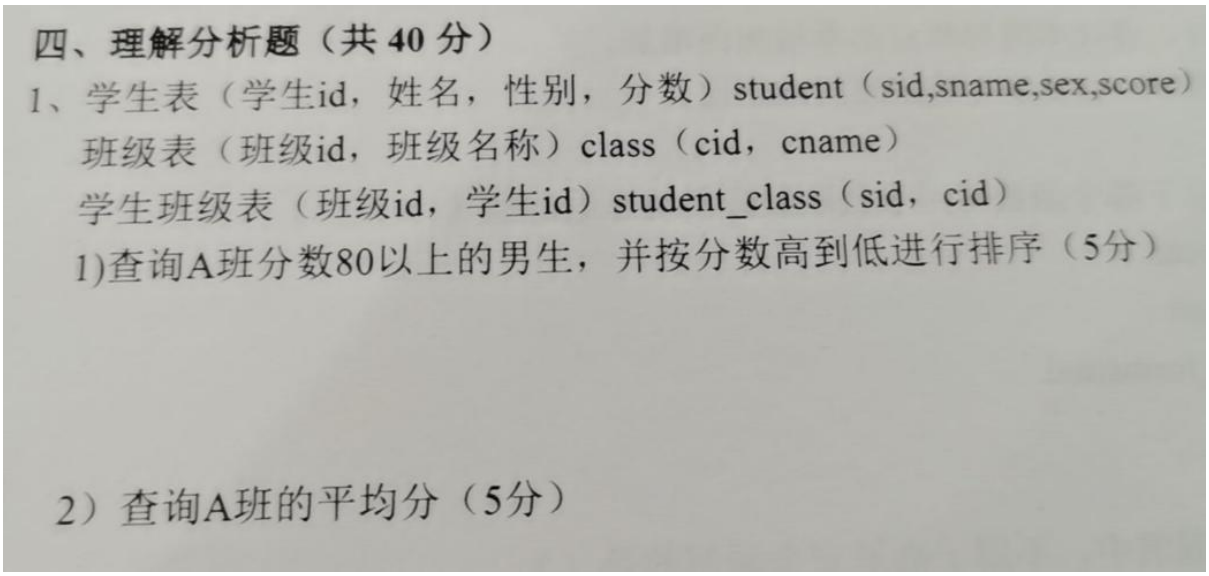
group by c.DepartID,c.DepartName) as t3

where t1.Studentno = t2.Studentno

and t1.DepartID=t2.DepartID

and (t2.english+t2.math,t2.computer) =t3.totalScore ;

3.4.16数据库面试题


四、理解分析题（共 40 分）

1、学生表（学生id，姓名，性别，分数）student（sid,sname,sex,score）

班级表（班级id，班级名称）class（cid，cname）

学生班级表（班级id，学生id）student_class（sid，cid）

1)查询A班分数80以上的男生，并按分数高到低进行排序（5分）

2）查询A班的平均分（5分）

参考答案：

1、 select t1.cid, t1.sid,t.sname,avg(t.score)

from student t, student_class t1

where t.sid = t1.sid

group by t1.cid

having avg(t.score) > 80

and t.sex = '男'

and t1.cid = 'A 班';

2、 select t1.cid, t1.sid,t.sname,avg(t.score)

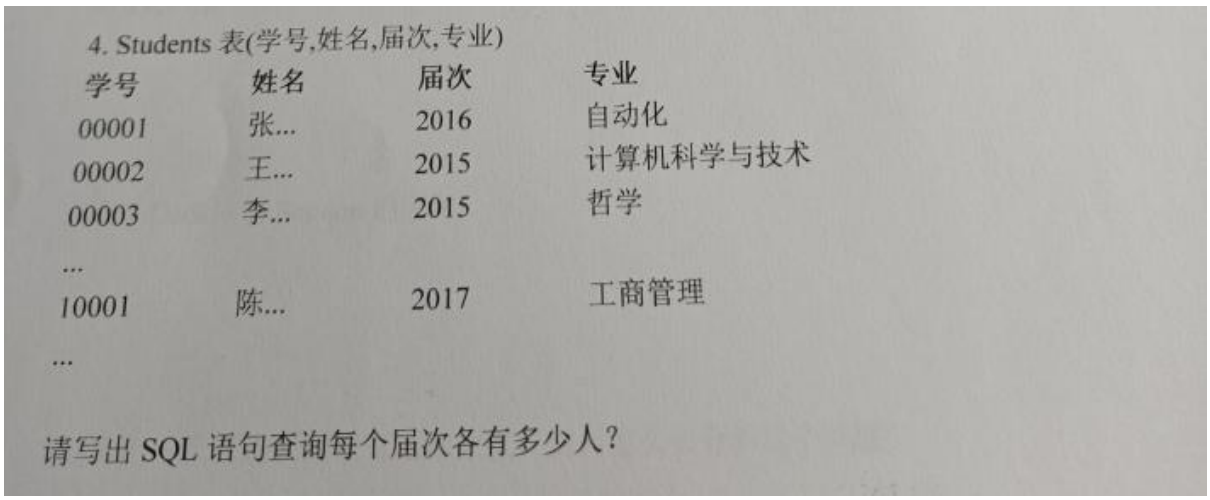
from student t, student_class t1

where t.sid = t1.sid

and t1.cid = 'A 班';

HAVING 一般和 GROUP BY 字句联合使用，筛选分组后的数据。与 WHERE 的区别是 where 字句在聚合前先筛选记录，作用在 group by 和 having 字句前，而 having 子句在聚合后对组记录进行筛选

3.4.17分组查询



4. Students 表(学号,姓名,届次,专业)

学号	姓名	届次	专业
00001	张...	2016	自动化
00002	王...	2015	计算机科学与技术
00003	李...	2015	哲学
...			
10001	陈...	2017	工商管理
...			

请写出 SQL 语句查询每个届次各有多少人?

参考答案：

select `届次`,count(1) as 人数 from Students group by `届次`;

解析：

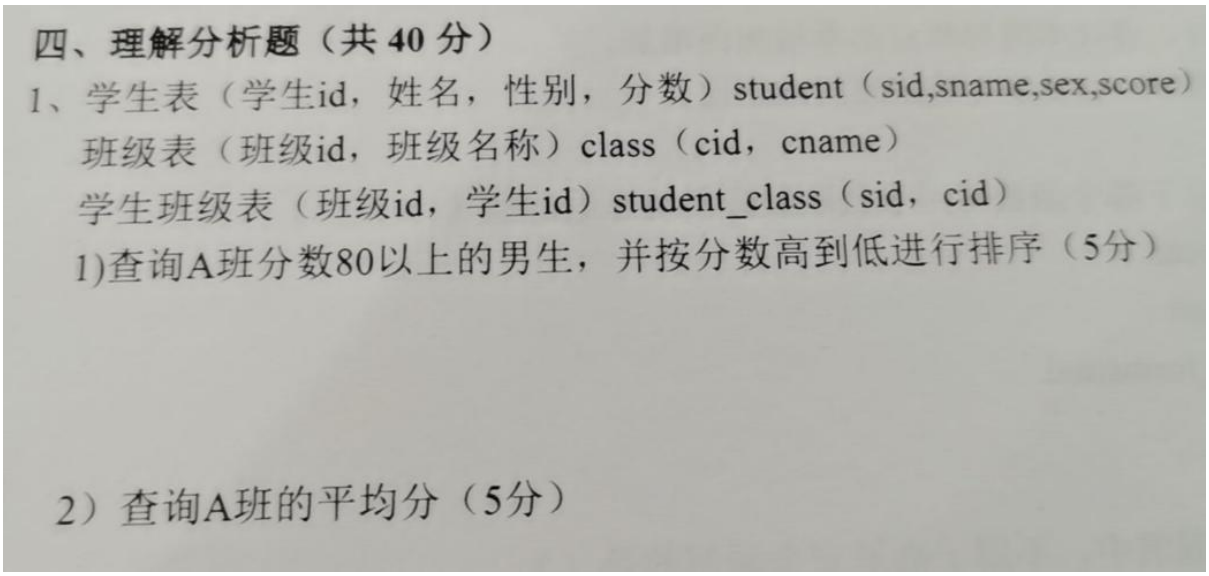
根据一列或多列对结果集进行分组的语法是：

GROUP BY 字段 1,字段 2

意思是字段 1、字段 2 相同的为一组。

在分组的列上我们可以使用聚合函数，常见的有：COUNT、SUM、AVG、MAX、MIN，如：SELECT AVG(字段) FROM S GROUP BY 字段;

3.4.18数据库面试题


四、理解分析题（共 40 分）

1、学生表（学生id，姓名，性别，分数）student（sid,sname,sex,score）

班级表（班级id，班级名称）class（cid，cname）

学生班级表（班级id，学生id）student_class（sid，cid）

1)查询A班分数80以上的男生，并按分数高到低进行排序（5分）

2）查询A班的平均分（5分）

参考答案：

1、 select t1.cid, t1.sid,t.sname,avg(t.score)

from student t, student_class t1

where t.sid = t1.sid

group by t1.cid

having avg(t.score) > 80

and t.sex = '男'

and t1.cid = 'A 班';

2、 select t1.cid, t1.sid,t.sname,avg(t.score)

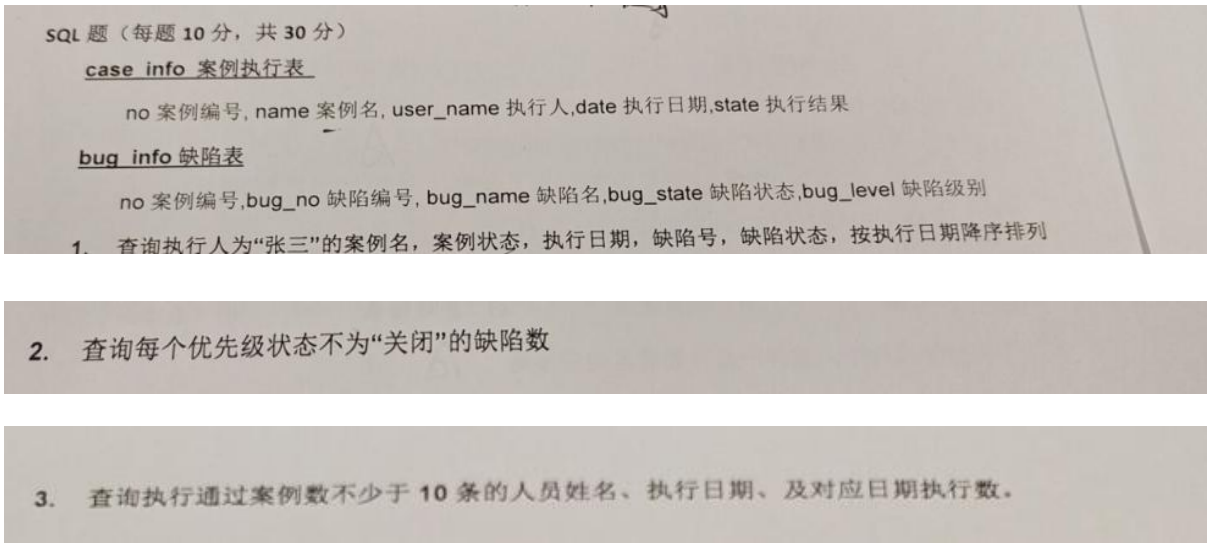
from student t, student_class t1

where t.sid = t1.sid

and t1.cid = 'A 班';

HAVING 一般和 GROUP BY 字句联合使用，筛选分组后的数据。与 WHERE 的区别是 where 字句在聚合前先筛选记录，作用在 group by 和 having 字句前，而 having 子句在聚合后对组记录进行筛选

3.4.19数据库面试真题



SQL 题（每题 10 分，共 30 分）

case_info 案例执行表

no 案例编号, name 案例名, user_name 执行人,date 执行日期,state 执行结果

bug_info 缺陷表

no 案例编号,bug_no 缺陷编号, bug_name 缺陷名,bug_state 缺陷状态,bug_level 缺陷级别

1. 查询执行人为“张三”的案例名，案例状态，执行日期，缺陷号，缺陷状态，按执行日期降序排列

2. 查询每个优先级状态不为“关闭”的缺陷数

3. 查询执行通过案例数不少于 10 条的人员姓名、执行日期、及对应日期执行数。

参考答案：

1、 select c.name,c.date,b.bug_no,b.bug_state,b.bug_level

from case_info as c join bug_info as b on c.no=b.no

where c.user_name='张三'

order by c.date desc;

2、 select bug_level,count(1) as '缺陷数'

from bug_info

where bug_level != '关闭'

group by bug_level;

3、 select t1.user_name,t1.date,count(1) as '当日执行数'

from case_info as t1

join (select user_name,count(1)

from case_info a

group by user_name,date) as t2 on t1.user_name=t2.user_name

where t1.user_name in(

select user_name

from case_info as c join bug_info as b on c.no=b.no

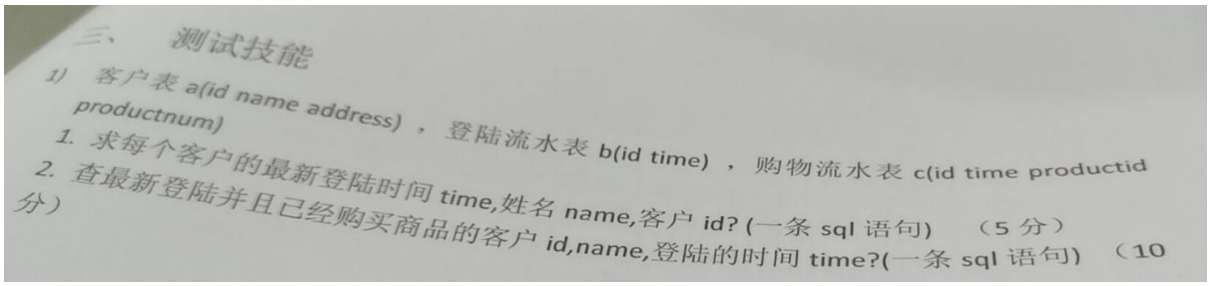
group by c.user_username

having count(1)>=10);

解析：

HAVING 一般和 GROUP BY 字句联合使用，筛选分组后的数据。与 WHERE 的区别是 where 字句在聚合前先筛选记录，作用在 group by 和 having 字句前，而 having 子句在聚合后对组记录进行筛选

3.4.20SQL 编写



1、 select max(b.time),a.name,a.id

from a join b on a.id=b.id

group by a.id;

2、 select a.id,a.name,max(b.time)

from c left join a on c.id=a.id

left join b on a.id=b.id

where a.id is not null

group by a.id;

解析：

常见的连接查询有如下几种：

内连接-INNER JOIN

内连接又称为等值连接，会显示左表及右表符合连接条件的记录

左外连接-LEFT JOIN

也称为左连接，显示左表的全部记录及右表符合连接条件的记录，右表不符合条件的显示 NULL。

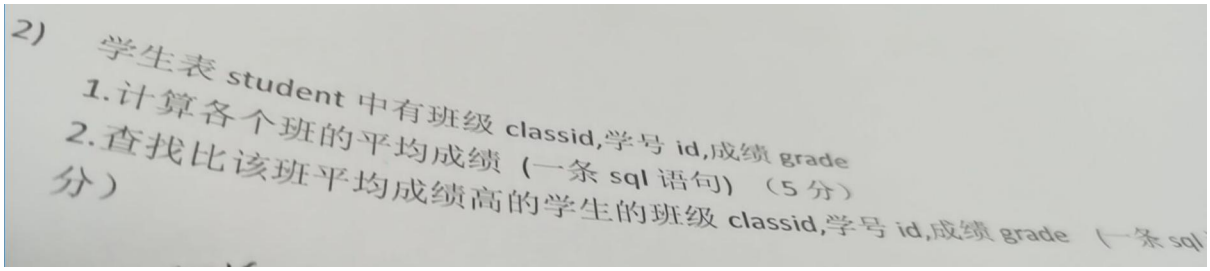
如：SELECT * FROM A LEFT JOIN B ON A.key=B.key;

右外连接-RIGHT JOIN

也称为右连接，显示右表的全部记录及左表符合连接条件的记录，左表不符合条件的显示 NULL，

如：SELECT * FROM A RIGHT JOIN B ON A.key=B.key;

3.4.21SQL 编写



参考答案：

1、 select avg(grade) from student group by classid;

2、 使用相关子查询

```
select * from student t1
where t1.grade >(
        select avg(grade) from student    t2
            where t1.classid=t2.classid);
```

或采用关联查询

```
select t1.classid,t1.id,t1.grade
```

```
from student t1
join (select t2.classid,avg(t2.grade) from student    t2
                group by t2.classid) as t3
        on t1.classid=t3.classid
        where t1.grade>t3.grade;
```

解析：

相关子查询依赖于外部查询的数据，外部查询每执行一次，子查询就执行一次。而非相关子查询则独立于外部查询，子查询只执行一次，执行完将结果传递给外部查询，如：

SELECT * FROM A WHERE A.id in(SELECT id FROM B);

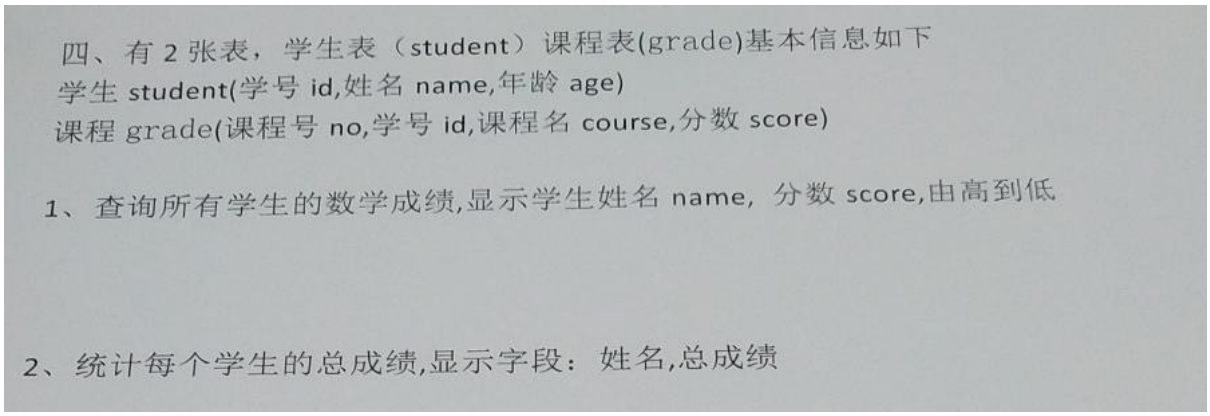
3.4.22现有一个表格，请编写 SQL 查询筛选出所有描述不是“无聊”且 id 为奇数的影片，结果按评分排序

id	movie	description	rating
1	复仇者联盟 4	还没看	9.1
2	正义联盟	非常好看	9.8
3	祖宗十八代	无聊	6.6
4	何以为家	2D，感动	8.9
5	金蝉脱壳 2	无聊	6.5

参考答案：

```
select * from table_name
where description!= '无聊'
and id%2=1
order by rating desc;
```

3.4.23万年学生表



参考答案：

1、 select s.name,g.score

from student s join grade g

on s.id=g.id

where g.course='数学'

order by g.score desc;

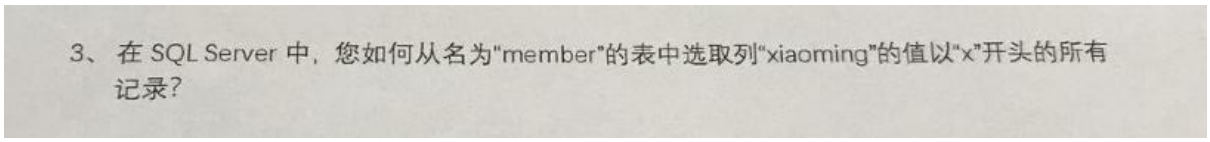
2、 select s.name,sum(g.score)

from student s join grade g

on s.id=g.id

group by s.id;

3.4.24模糊查询



参考答案：

select * from member where xiaoming like 'x%';

模糊查询的语法为：SELECT 字段 FROM 表 WHERE 某字段 LIKE 条件;

SQL 提供了多种模糊匹配方式，比较常用的有如下两种：

%：表示零个或多个字符，可以匹配任意类型和任意长度的字符，如

select * from member where xiaoming like 'x%';-- 以 x 开头

select * from member where xiaoming like '%x';-- 以 x 结尾

select * from member where xiaoming like '%x%'; -- 包含 x

_：表示任意单个字符，匹配单个任意字符，它常用来限制表达式的字符长度，如

select * from member where xiaoming like 'x_';-- 两个字符，以 x 开头

select * from member where xiaoming like '_x';-- 两个字符，以 x 结尾

select * from member where xiaoming like '_x_'; -- 三个字符，中间为 x

3.4.25数据库查询

4、在 SQL Server 中，您如何从名为“member”的表中筛选出手机号“phonenumber”出现重复的用户并排除空数据?

参考答案：

select * from member where phonenumber

in(select phonenumber from member

where phonenumber is not null

group by phonenumber

having count(1)>1)

使用首先使用 phonenumber is not null 排除空数据，再通过分组统计过滤出出现次数大于 1 次的 phonenumber 列表，利用 in 子查询获得这些手机号重复额用户信息

3.4.26万年学生表

有一张学生 student 表，存储是学生的基本信息(no 标识学号)，它有如下字段：

id	no	name	sex	age	province
1	2020001	张三	男	18	湖北
2	2020002	小王	女	22	山东
…					

grade 表(no 标识学号)，存储的是学生各科成绩，有如下字段：

id	no	class	grade
1	2020001	语文	88
2	2020002	英语	56
…			

请根据如下要求编写 sql

1、 查询全班语文成绩最高分数

2、 按照年龄对 student 表进行倒序排列

3、 查询班级中英语成绩及格的全部女生

参考答案：

1、 select max(`grade`) from grade where class='语文';

2、 select * from student order by age desc;

3、 select s.*

from student as s join grade as g

on s.no=g.no

where s.sex='女'

and g.grade >=60;

聚合函数、排序都是软件测试面试笔试题中间出现频率最高的考点之一。常见的聚合函数有 MAX（最大）、MIN（最小）、SUM（求和）、COUNT（计数）、AVG（平均）。

3.4.27数据库设计

已知教学数据库包含三个关系：学生关系 S（SND,SNAME,SA,SD）课程关系 C（CNO,CN,TNAME），选课关系 SC（SNO,CNO,G）其中，下划线的字段为该关系的码，SNO 代表学号，SNAME 代表学生姓名，SA 代表学生年龄，SD 代表学生所在系，CNO 代表课程号，CN 代表课程名，TNAME 代表任课老师姓名，G 代表成绩，请用 SQL 语句实现：

1、简历学生关系和选课关系，有完整约束的要定义完整性约束

```
 CREATE TABLE S(
SNO INT(10) not null PRIMARY KEY,-- 学号
SNAME VARCHAR(20) not null,-- 姓名
SA INT(3),-- 年龄
SD VARCHAR(10)-- 所在系
)
CREATE TABLE C(
CNO INT(10) not null PRIMARY KEY,-- 课程号
CN VARCHAR(20) UNIQUE , -- 课程名
TNAME INT(3) not null-- 任课老师
)
CREATE TABLE SC(
SNO INT(10),-- 学号
CNO VARCHAR(20), -- 课程号
G INT(3),-- 成绩
FOREIGN KEY (SNO) REFERENCES S(SNO),
FOREIGN KEY (CNO) REFERENCES C(CNO)
```

)

2、将下列学生信息插入学生关系中：李丹，18 岁，电信系，学号：20070206

INSERT into S(SNO,SNAME,SD) VALUES(20070206,'李丹','电信系');

3、找出选修了课程为“112002”的学生学号和姓名

SELECT S.SNO,S.SNAME FROM S S,SC C where S.SNO=C.SNO AND C.CNO='112002';

4、修改学号为“20070206”的学生所在的系为计算机

UPDATE S SET SD='计算机' WHERE SNO='20070206'

5、查询选修了数据库系统原理 这门课的学生的姓名和成绩

SELECT S.SNAME,C.G FROM S S,SC C where S.SNO=C.SNO;

4 Web 测试

4.1 web 测试

4.1.1 描述用浏览器访问 www.baidu.com 的过程

1. 先要解析出 baidu.com 对应的 ip 地址：

- 要先使用 arp 获取默认网关的 mac 地址
- 组织数据发送给默认网关(ip 还是 dns 服务器的 ip，但是 mac 地址是默认网关的 mac 地址)
- 默认网关拥有转发数据的能力，把数据转发给路由器
- 路由器根据自己的路由协议，来选择一个合适的较快的路径转发数据给目的网关
- 目的网关(dns 服务器所在的网关)，把数据转发给 dns 服务
- dns 服务器查询解析出 baidu.com 对应的 ip 地址，并原路返回请求这个域名的 client 得到了 baidu.com 对应的 ip 地址之后，会发送 tcp 的 3 次握手，进行连接
- 使用 http 协议发送请求数据给 web 服务器
- web 服务器收到数据请求之后，通过查询自己的服务器得到相应的结果，原路返回给浏览器
- 浏览器接收到数据之后通过浏览器自己的渲染功能来显示这个网页
- 浏览器关闭 tcp 连接，即 4 次挥手结束，完成整个访问过程

2. 了解的常用浏览器有哪些？

IE，Chrome，Safari，Firefox，Opera

4.1.2 以京东首页为例，设计用例框架。（注意框架设计逻辑，区域划分，专项测试等，不需要详细用例，需要查看 PC 可直接和辨识管提要求）

4.1.3 如何测试购买下单和退货流程

产品经理设计了单品优惠，组合优惠，订单优惠，优惠券优惠（优惠券优惠包含通用券，定向券，满减券，折扣券）和礼品卡，其中礼品卡上需要单独购买的。请问如何测试购买下单和退货流程，需要注意什么？（包含数据存储）

4.1.4 什么是 sql 注入，什么是跨站脚本，什么是跨站请求伪造？

SQL 注入攻击是注入攻击最常见的形式（此外还有 OS 注入攻击（Struts 2 的高危漏洞就是通过 OGNL 实施 OS 注入攻击导致的）），当服务器使用请求参数构造 SQL 语句时，恶意的 SQL 被嵌入到 SQL 中交给数据库执行。SQL 注入攻击需要攻击者对数据库结构有所了解才能进行，攻击者想要获得表结构有多种方式：

（1）如果使用开源系统搭建网站，数据库结构也是公开的（目前有很多现成的系统可以直接搭建论坛 ，电商网站，虽然方便快捷但是风险是必须要认真评估的）；

（2）错误回显（如果将服务器的错误信息直接显示在页面上，攻击者可以通过非法参数引发页面错误从而通过错误信息了解数据库结构，Web 应用应当设置友好的错误页，一方面符合最小惊讶原则，

一方面屏蔽掉可能给系统带来危险的错误回显信息）；

（3）盲注。防范 SQL 注入攻击也可以采用消毒的方式，通过正则表达式对请求参数进行验证，此外， 参数绑定也是很好的手段，这样恶意的 SQL 会被当做 SQL 的参数而不是命令被执行，JDBC 中的 PreparedStatement 就是支持参数绑定的语句对象，从性能和安全性上都明显优于 Statement。

XSS（Cross Site Script，跨站脚本攻击）是向网页中注入恶意脚本在用户浏览网页时在用户浏览器中执行恶意脚本的攻击方式。跨站脚本攻击分有两种形式：

反射型攻击（诱使用户点击一个嵌入恶意脚本的链接以达到攻击的目标，目前有很多攻击者利用论坛、微博发布含有恶意脚本的 URL 就属于这种方式）

持久型攻击（将恶意脚本提交到被攻击网站的数据库中，用户浏览网页时，恶意脚本从数据库中被加载到页面执行，QQ 邮箱的早期版本就曾经被利用作为持久型跨站脚本攻击的平台）。

CSRF 攻击（Cross Site Request Forgery，跨站请求伪造）是攻击者通过跨站请求，以合法的用户身份进行非法操作（如转账或发帖等）。CSRF 的原理是利用浏览器的 Cookie 或服务器的 Session，盗取用户身份，其原理如下图所示。防范 CSRF 的主要手段是识别请求者的身份，主要有以下几种方式：

（1）在表单中添加令牌（token）；

（2）验证码；

（3）检查请求头中的 Referer（前面提到防图片盗链接也是用的这种方式）。

令牌和验证都具有一次消费性的特征，因此在原理上一致的，但是验证码是一种糟糕的用户体验，不是必要的情况下不要轻易使用验证码，目前很多网站的做法是如果在短时间内多次提交一个表单未获得成功后才要求提供验证码，这样会获得较好的用户体验。

4.1.5 给你一个网站怎么开展测试？

1. 首先，查找需求说明、网站设计等相关文档，分析测试需求。
2. 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
3. 设计测试用例：

（1）功能性测试可以包括，但不限于以下几个方面：链接测试；链接是否正确跳转，是否存在空页面和无效页面，是否有不正确的出错信息返回等；提交功能的测试；多媒体元素是否可以正确加载和显示；多语言支持是否能够正确显示选择的语言等

（2）界面测试可以包括但不限于以下几个方面：页面是否风格统一，美观。页面布局是否合理，重点内容和热点内容是否突出。控件是否正常使用。对于必须但为安装的空间，是否提供自动下载并安装的功能。文字检查。

（3）性能测试一般从以下两个方面考虑：压力测试，负载测试，强度测试

（4）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。

（5）安全性测试：基本的登录功能的检查；是否存在溢出错误，导致系统崩溃或者权限泄露；相关开发语言的常见安全性问题检查，例如 SQL 注入等；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。

（6）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。

4. 开展测试，并记录缺陷。合理的安排调整测试进度，提前获取测试所需的资源，建立管理体系（例如， 需求变更、风险、配置、测试文档、缺陷报告、人力资源等内容）。

5. 定期评审，对测试进行评估和总结，调整测试的内容。

4.1.6 电商支付模块的测试如何展开？

支付流程里面就涉及到了第三方支付接口：

- **下单接口**

商户提交下单请求到第三方支付接口，第三方支付收单成功后返回下单成功结果给到商户系统。（下单接口的最终处理结果分为下单成功和下单失败，若未收到明确结果可调用单笔订单查询接口查询结果。）

- **支付接口**

调用该接口时指定支付参数，完成买家账户向商户账户的支付，采用页面跳转交互模式和后台通知交互模式。（结果分为两路返回：一路为前台在 return_url 页面跳转显示支付结果；一路为后台在

notify_url 收到支付结果通知后进行响应。）

- **退款接口**

调用第三方支付的支付请求接口返回付款成功后，在需要做退款处理时调用退款请求接口发起退款处理。（退款接口的最终处理结果分为退款成功和退款失败，若未收到明确结果可调用退款查询接口查询结果。）

- **单笔订单查询接口**

根据订单号查询单笔订单信息和状态。退款订单查询接口：调用第三方支付的退款接口返回后，在需要查询退款请求状态可调用退款订单查询接口查询退款订单的状态和订单信息。

测试过程中需要注意的主要测试点及异常场景：

- 首先要保证接口都能正常调用；
- 生成一笔订单，支付完成后，同步或异步重复回调，只有一次有效；
- 生成一笔订单，复制订单号和金额，再次生成一笔订单，用 fiddler 设置断点，用第一笔已完成的订单号和订单金额去替换现有的订单号和金额，无法完成支付；
- 生成一笔订单，跳转到第三方时修改金额，无法到账，或者如果是游戏充值游戏币的话，到账为篡改后的金额对应的游戏币；
- 异步通知屏蔽，同步有效，进行支付，同步能够正常到账；
- 同步设置无效，异步有效，进行支付，异步能够正常到账；
- 同步异步都设置无效，在第三方支付完成后，在重发机制时间范围内，设置异步有效，到下次通知时间点时，能够正常通知到账（补单机制的验证，如果商户收到第三方支付成功的通知后，要告知第三方支付收到了成功的通知，如果第三方支付收到商户应答不是 ok 或超时，第三方支付就会认为通知失败，会在规定的时间内持续调用 notify_url，一般有时间或次数的限制）；
- 针对支付订单在数据库中存储是否完整和正确进行校验（比如：第三方订单号--方便与第三方对账和问题排查、订单金额、订单状态等）；
- 如果是用户购买实物商品，用户发起退货，要保证退货流程正常，资金能正常返还，要考虑下并发情况的验证以确保安全性；
- 如果是用户购买虚拟商品，比如话费、油卡之类的商品，只有在发货失败的时候才能发起退货，注意验证；

4.1.7 如何开展兼容性测试？

（1）Web 兼容性测试

- 首先开展人工测试，测试工程师测试主流浏览器和常用操作系统测试主流程和主界面，看看主流程和主界面是否有问题，如果存在问题，那么记录下 bug 情况，以及浏览器型号和版本，以及操作系统，准确定位 bug 产生的原因，提交 bug，告知开发人员修改。所有的主流设备都需要进行测试， 只关注主流程和主界面，毕竟每个系统主流程和主界面不是很多，所以这个工作量还是可以承受的。
- 其次借助第三方测试工具，目前我觉得比较好用的第三方 Web 测试工具有 IEtester（离线）、SuperPreview（离线）和 Browsershots：browsershots.org（在线），一款可以测试 IE 的兼容， 一款可以测试主流浏览器的兼容，包括谷歌、火狐、Opera 等等。借助第三方测试工具，找到 bug 产生的位置，分析测试结果，告知程序员调整。

（2）APP 兼容性测试

- APP 的兼容性测试和 Web 测试类似，首先开展人工测试，测试工程师借助测试设备对主流程和主功能，主界面进行测试；收集所有的能收集到的不同型号的测试设备测试主流程和主界面，看看主流程和主界面是否有问题，如果存在问题，综合考虑设备的使用率等因素，看看是否需要调整，如果需要，那么记录下 bug 情况以及测试设备的型号和操作系统，准确定位 bug 产生的原因，提交 bug，告知开发人员修改。
- 其次借助第三方测试工具，对于 APP 的兼容性测试，推荐的是百度众测平台和云测平台，我经常使用的是云测平台，这两款测试工具里面包含了安卓和 iOS 的测试；测试很齐全，包括功能测试、深度兼容测试、性能测试、网络环境测试，还可以模拟海量用户测试，，还可以导入自己编写的测试用例进行功能测试，里面还包括测试专家的测试，当然了找专家是要花钱滴。基本进行兼容性测试是不需要花钱的；测试工程师把打包好的 apk 或者 IPA 文件，上传到测试平台，选择需要测试的设备型号，开始任务即可；等待一段时间，在等待的时间你是不需要盯着的，你可以做其他的工作。测试完成后会生成一份测试报告，可以查看错误页面和错误日志，如果需要调整，那么提交 bug， 告知程序员修改即可。

4.1.8 nginx,tomcat,apache 都是什么？

Nginx (engine x) 是一个高性能的 HTTP 和反向代理服务器，也是一个 IMAP/POP3/SMTP 服务器。

Apache HTTP Server 是一个模块化的服务器，源于 NCSAhttpd 服务器

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器，是开发和调试 JSP 程序的首选。

4.1.9 apache 和 nginx 的区别？

Nginx 相对 Apache 的优点：

轻量级，同样起 web 服务，比 apache 占用更少的内存及资源；

抗并发，nginx 处理请求是异步非阻塞的，支持更多的并发连接，而 apache 则是阻塞型的，在高并发下 nginx 能保持低资源低消耗高性能；

配置简洁；

高度模块化的设计，编写模块相对简单；

社区活跃。

Apache 相对 Nginx 的优点

- rewrite，比 nginx 的 rewrite 强大；
- 模块超多，基本想到的都可以找到； 少 bug ，nginx 的 bug 相对较多；
- 超稳定。

4.1.10Selenium 有哪些定位元素方法

1. id 定位
2. name 定位
3. class_name

4. tag_name
5. link_text
6. partial_link_text
7. Xpath
8. css

当定位一个元素是，如果存在 ID 属性值时，我们可以优先考虑 ID 定位，当没有 ID，有 name 属性值和 class 属性值时也可以采用 name 和 class_name 定位。当次能确定该元素的标签为页面唯一时，可以采用 tag_name 定位，一般来说采用 link_text 定位的方法比较少。如果 ID ，name，class_name 单个定位不了，可以采用 css 和 xpath 定位。

5 接口测试

5.1 接口测试

5.1.1 什么是接口

接口是指外部系统与系统之间以及内部各子系统之间的交互点。

包括外部接口、内部接口，内部接口又包括：上层服务与下层服务接口、同级接口。

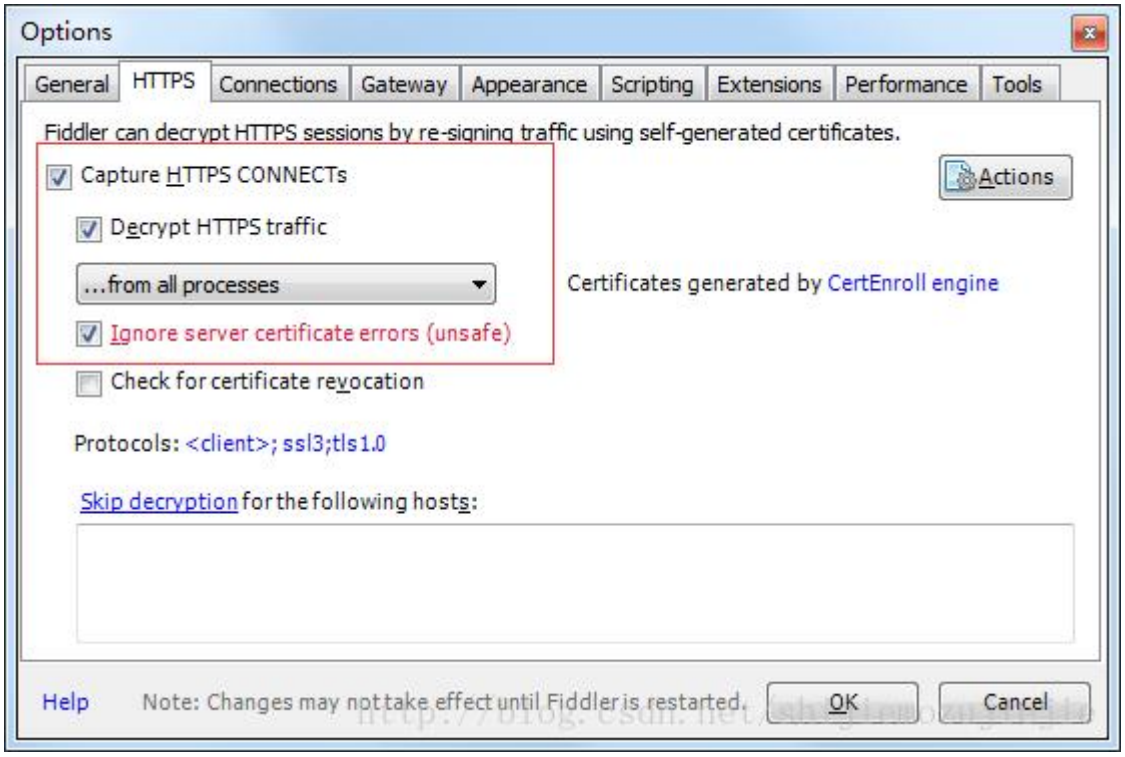
5.1.2 如果模块请求 http 改为了 https，测试方案应该如何制定，修改？

分别用 http 还有 https 登录试试。如果用 https 可以正常登录，地址栏显示一把锁头，那么这个网站是有部署 SSL 的。如果 http 和 https 都能够正常登录，进一步说明该网站没有设置强制 https 登录，或者说没有设置 http 链接自动跳转 https 链接；相反如果用 http 登录，结果跳转到 https 页面，说明网站部署了 SSL， 而且设置了 http 自动跳转 https。

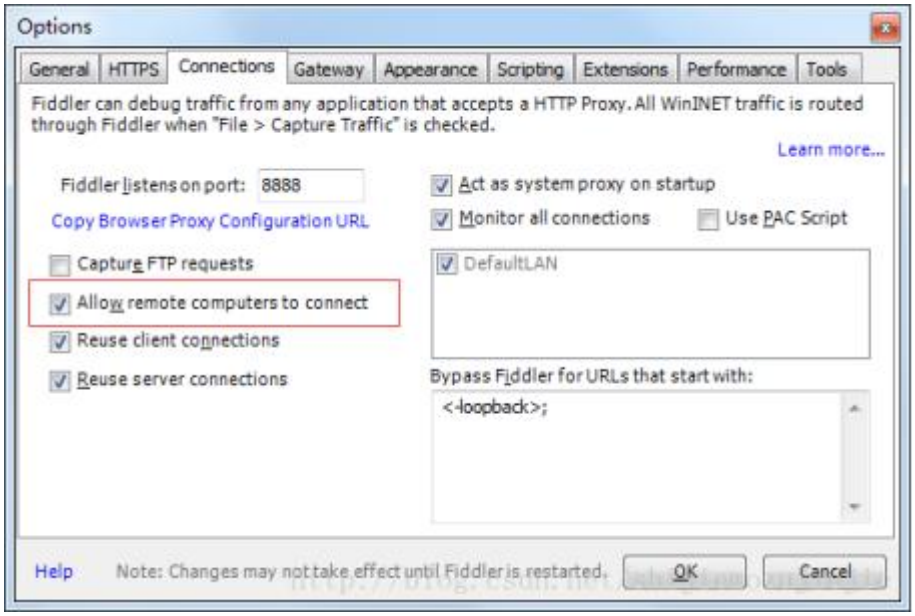
5.1.3 常用 HTTP 协议调试代理工具有什么？详细说明抓取 HTTPS 协议的设置过程？

Fiddler 是一个 http 协议调试代理工具

打开 Fiddler，进入 Tools-Options-HTTPS，配置允许抓取 HTTPS 连接和解析 HTTPS 流量然后选择要解释的来源，设置是否忽略服务证书错误（这些操作做完之后，在浏览器方位 IP:8888，安装证书就可以在浏览器抓取 HTTPS 协议了）

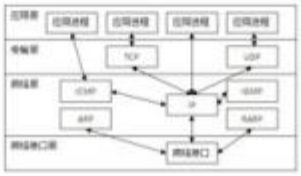


进入 Tools-Options-Connections，保证打开启抓取 HTTPS 连接，然后默认端口按需求是或否需要修改，然后点选允许远程计算机连接选项



5.1.4 描述 TCP/IP 协议的层次结构，以及每一层中重要协议

一、TCP/IP网络体系结构中，常见的接口层协议有：Ethernet 802.3、Token Ring 802.5、X.25、Frame relay、HDLC、PPP ATM等。1.网络层网络层包括：IP(Internet Protocol) 协议、ICMP(Internet Control Message Protocol) 、控制报文协议、ARP(Address Resolution Protocol) 地址转换协议、RARP(Reverse ARP)反向地址转换协议。 2.传输层传输层协议主要是：传输控制协议TCP(Transmission Control Protocol) 和用户数据报协议UDP(User Datagram protocol) 。3.应用层应用层协议主要包括如下几个：FTP、TELNET、DNS、SMTP、RIP、NFS、HTTP。



5.1.5 jmeter，一个接口的响应结果如下：

<!DOCTYPE html>

<html>

<head>

<title newBidId = “74956”encryptBidId=”713504275825”>小贷公司标管理</title>

请用正则表达式方法分别获取一下 74956 和 713504275825 这两个数值分别赋值给 A1 和 A2

5.1.6 接口产生的垃圾数据如何清理

造数据和数据清理，需要连数据库做增删改查的操作

测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

5.1.7 依赖第三方的接口如何处理

这个需要自己去搭建一个 mock 服务，模拟接口返回数据。如可以采用 moco，moco 是一个开源的框架，moco 服务搭建需要自己能够熟练掌握，面试会问你具体如何搭建 ，如何模拟返回的数据，是用的什么格式，如何请求的

5.1.8 测试的数据你放在哪?

测试数据到底该怎么放，这个是面试官最喜欢问的一个题了，似乎仁者见仁智者见智，没有标准的答案，有的人说放 excel，也有的说放.py 脚本，也有的说放 ini 配置文件，

还有放到 json，yaml 文件，txt 文件，甚至有的放数据库，五花八门，一百个做自动化的小伙伴有 100 个放的地方。

这里总结下测试的数据到底该怎么放？

首先测试的数据是分很多种的，有登录的账户数据，也有注册的账户数据，还有接口的参数，还有邮箱配置的数据等等等等，所以这个题不能一概而论给答死了。要不然就是给自己挖坑。

以下两个大忌不能回答：

测试的数据是不能写死到代码里面的，这个是原则问题，也是写代码的大忌（你要是回答写在代码里面，估计就是回去等通知了）

测试数据放到.py 的开头，这种其实很方便，对于少量的，固定不变的数据其实是可以放的，但是面试时候，千万不能这样说，面试官喜欢装逼的方法

测试数据存放总结：

1.对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）

2.对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成

3.对于一个接口有多组测试的参数，可以参数化，数据放 yaml,text,json,excel 都可以

4.对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理

5.对于邮箱配置的一些参数，可以用 ini 配置文件

6.对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据

7.对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到 py 脚本的开头，十年八年都不会变更的

总之不同的测试数据，可以用不同的文件管理

5.1.9 什么是数据驱动，如何参数化？

参数化和数据驱动的概念这个肯定要知道的，参数化的思想是代码用例写好了后，不需要改代码，只需维护测试数据就可以了，并且根据不同的测试数据生成多个用例

python 里面用 unittest 框架

```
import unittest
import ddt
# 测试数据
datas = [ {"user": "admin", "psw": "123", "result": "true"},
{"user": "admin1", "psw": "1234", "result": "true"},
{"user": "admin2", "psw": "1234", "result": "true"},
{"user": "admin3", "psw": "1234", "result": "true"},
{"user": "admin4", "psw": "1234", "result": "true"},
{"user": "admin5", "psw": "1234", "result": "true"},
{"user": "admin6", "psw": "1234", "result": "true"},
{"user": "admin7", "psw": "1234", "result": "true"},
{"user": "admin8", "psw": "1234", "result": "true"},
{"user": "admin9", "psw": "1234", "result": "true"},
{"user": "admin10", "psw": "1234", "result": "true"},
{"user": "admin11", "psw": "1234", "result": "true"}]
@ddt.ddt
class Test(unittest.TestCase):
@ddt.data(*datas)
def test_(self, d):
"""上海-悠悠：{0}"""
```

```
print("测试数据：%s" % d)
if __name__ == "__main__":
unittest.main()
```

unittest 框架还有一个 paramunittest 也可以实现

```
import unittest
import paramunittest
import time
# python3.6
@paramunittest.parametrized(
{"user": "admin", "psw": "123", "result": "true"},
{"user": "admin1", "psw": "1234", "result": "true"},
{"user": "admin2", "psw": "1234", "result": "true"},
{"user": "admin3", "psw": "1234", "result": "true"},
{"user": "admin4", "psw": "1234", "result": "true"},
{"user": "admin5", "psw": "1234", "result": "true"},
{"user": "admin6", "psw": "1234", "result": "true"},
{"user": "admin7", "psw": "1234", "result": "true"},
{"user": "admin8", "psw": "1234", "result": "true"},
{"user": "admin9", "psw": "1234", "result": "true"},
{"user": "admin10", "psw": "1234", "result": "true"},
{"user": "admin11", "psw": "1234", "result": "true"},
)
class TestDemo(unittest.TestCase):
def setParameters(self, user, psw, result):
'''这里注意了，user, psw, result 三个参数和前面定义的字典一一对应'''
self.user = user
self.user = psw
self.result = result
```

```
def testcase(self):
print("开始执行用例：----------")
time.sleep(0.5)
print("输入用户名：%s" % self.user)
print("输入密码：%s" % self.user)
print("期望结果：%s " % self.result)
time.sleep(0.5)
self.assertTrue(self.result == "true")
if __name__ == "__main__":
unittest.main(verbosity=2)
```

如果用的是 pytest 框架，也能实现参数化

```
# content of test_canshu1.py
# coding:utf-8
import pytest
@pytest.mark.parametrize("test_input,expected",
[ ("3+5", 8),
("2+4", 6),
("6 * 9", 42),
])
def test_eval(test_input, expected):
assert eval(test_input) == expected
if __name__ == "__main__":
pytest.main(["-s", "test_canshu1.py"])
```

pytest 里面还有一个更加强大的功能，获得多个参数化参数的所有组合，可以堆叠参数化装饰器

```
import pytest
@pytest.mark.parametrize("x", [0, 1])
@pytest.mark.parametrize("y", [2, 3])
```

```
def test_foo(x, y):
print("测试数据组合：x->%s, y->%s" % (x, y))
if __name__ == "__main__":
pytest.main(["-s", "test_canshu1.py"])
```

5.1.10下个接口请求参数依赖上个接口的返回数据

这个很容易，不同的接口封装成不同的函数或方法，需要的数据 return 出来，用一个中间变量 a 去接受，后面的接口传 a 就可以了

5.1.11依赖于登录的接口如何处理

登录接口依赖 token 的，可以先登录后，token 存到一个 yaml 或者 json，或者 ini 的配置文件里面，后面所有的请求去拿这个数据就可以全局使用了

如果是 cookies 的参数，可以用 session 自动关联

s=requests.session()

后面请求用 s.get()和 s.post()就可以自动关联 cookies 了

5.1.12接口测试的步骤有哪些？

1）发送接口请求

2）测试接口获取返回值

3）断言：判断实际结果是否符合预期

5.1.13接口测试中依赖登录状态的接口如何测试？

依赖登最状态的接口，本质上是在每次发送请求时需要带上存储有账户有效信息的 Session 或 Cookie 才能发送成功，在构建 POST 请求时添加必要的 Session 或 Cookie

5.1.14依赖于第三方数据的接口如何进行测试？

可以利用一些 MOCK 工具（如：JSON Server、Easy Mock）来模拟第三方的数据返回，最大限度的降低对第三方数据接口的依赖

5.1.15解释什么是 SOAP？

SOAP 代表简单对象访问控制，它是一种基于 XML 的协议，用于在计算机之间交换信息。

5.1.16解释什么是 REST API？

这是开发人员执行请求并接收响应的一组功能。在 REST API 中，通过 HTTP 协议进行交互

REST - 代表状态转移，它正快速成为 API 创建的标准。

5.1.17API 测试发现的 Bug 类型是什么？

缺少或重复的功能

无法正常处理错误条件

可靠性

安全

未使用的标志

未实现错误

错误处理不一致

性能

多线程问题

错误不正确

5.1.18我们测试的接口属于哪一类？

服务器接口（基于 HTTP 协议的接口）,大多数人常说的接口测试，通常是 B/S 架构，由客户端（浏览器）调用，或模拟客户端（浏览器）调用服务器提供的请求接口，由服务器完成处理并返回一个应答的过程。例如：Webservice 接口，http 接口，jms 接口，hessian 接口。

5.1.19Cookie 保存在哪里？

如果设置了过期时间，Cookie 保存在硬盘中。

如果没有设置过期时间，Cookie 保存在内存中。

5.1.20HTTP 有哪些请求方法？

HTTP 共有如下 7 种请求方式，每种都可以发送 Header 和 Body：

GET

POST

PUT

DELETE

OPTIONS

HEAD

PATCH

5.1.21接口自动化测试的流程？

基本的接口功能自动化测试流程为：需求分析-->用例设计-->脚本开发-->测试执行-->结果分析

5.1.22接口测试用例的编写要点有哪些？

1）必填字段：请求参数必填项、可选项

2）合法性：输入输出合法、非法参数

3）边界：请求参数边界值等

4）容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理

5）响应数据校验：断言、数据提取传递到下一级接口...

6）逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况

7）性能：对接口模拟并发测试，逐步加压，分析瓶颈点

8）安全性：构造恶意的字符请求，如：SQL 注入、XSS、敏感信息、业务逻辑（如：跳过某些关键步骤；未经验证操纵敏感数据）

* 测试每个参数类型不合法的情况(类型不合法容易遗漏 NULL 型)

* 测试每个参数取值范围不合法的情况

* 测试参数为空的情况

* 测试参数前后台定义的一致性

* 测试每个参数的上下限(这里容易出致命的 BUG，如果程序处理不当，可能导致崩溃)

* 如果两个请求有严格的先后顺序，需要测试调转顺序的情况

5.1.23提到 UI 级别测试和 API 测试之间的关键区别？

UI（用户界面）是指测试图形界面，如用户如何与应用程序交互，测试应用程序元素，如字体，图像，布局等。UI 测试基本上侧重于应用程序的外观和感觉。

而 API 可以实现两个独立的软件系统之间的通信。实现 API 的软件系统包含可由另一软件系统执行的功能或子例程

5.1.24HTTPS 的工作原理

我们都知道 HTTPS 能够加密信息，以免敏感信息被第三方获取，所以很多银行网站或电子邮箱等等安全级别较高的服务都会采用 HTTPS 协议。

客户端在使用 HTTPS 方式与 Web 服务器通信时有以下几个步骤，如图所示。

（1）客户使用 https 的 URL 访问 Web 服务器，要求与 Web 服务器建立 SSL 连接。

（2）Web 服务器收到客户端请求后，会将网站的证书信息（证书中包含公钥）传送一份给客户端。

（3）客户端的浏览器与 Web 服务器开始协商 SSL 连接的安全等级，也就是信息加密的等级。

（4）客户端的浏览器根据双方同意的安全等级，建立会话密钥，然后利用网站的公钥将会话密钥加密，并传送给网站。

（5）Web 服务器利用自己的私钥解密出会话密钥。

（6）Web 服务器利用会话密钥加密与客户端之间的通信。

5.1.25HTTPS 有哪些优点？

尽管 HTTPS 并非绝对安全，掌握根证书的机构、掌握加密算法的组织同样可以进行中间人形式的攻击，但 HTTPS 仍是现行架构下最安全的解决方案，主要有以下几个好处：

（1）使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；

（2）HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全，可防止数据在传输过程中不被窃取、改变，确保数据的完整性。

（3）HTTPS 是现行架构下最安全的解决方案，虽然不是绝对安全，但它大幅增加了中间人攻击的成本。

（4）谷歌曾在 2014 年 8 月份调整搜索引擎算法，并称“比起同等 HTTP 网站，采用 HTTPS 加密的网站在搜索结果中的排名将会更高”。

5.1.26HTTPS 的缺点

虽然说 HTTPS 有很大的优势，但其相对来说，还是存在不足之处的：

（1）HTTPS 协议握手阶段比较费时，会使页面的加载时间延长近 50%，增加 10%到 20%的耗电；

（2）HTTPS 连接缓存不如 HTTP 高效，会增加数据开销和功耗，甚至已有的安全措施也会因此而受到影响；

（3）SSL 证书需要钱，功能越强大的证书费用越高，个人网站、小网站没有必要一般不会用。

（4）SSL 证书通常需要绑定 IP，不能在同一 IP 上绑定多个域名，IPv4 资源不可能支撑这个消耗。

（5）HTTPS 协议的加密范围也比较有限，在黑客攻击、拒绝服务攻击、服务器劫持等方面几乎起不到什么作用。最关键的，SSL 证书的信用链体系并不安全，特别是在某些国家可以控制 CA 根证书的情况下，中间人攻击一样可行。

5.1.27HTTPS 和 HTTP 的区别主要如下：

1、https 协议需要到 ca 申请证书，一般免费证书较少，因而需要一定费用。

2、http 是超文本传输协议，信息是明文传输，https 则是具有安全性的 ssl 加密传输协议。

3、http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。

4、http 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 http 协议安全。

5.1.28POST 和 GET 有什么区别？

GET 在浏览器回退时是无害的，而 POST 会再次提交请求。

GET 产生的 URL 地址可以被保存为书签，而 POST 不可以。

GET 请求会被浏览器主动 cache，而 POST 不会，除非手动设置。

GET 请求只能进行 url 编码，而 POST 支持多种编码方式。

GET 请求参数会被完整保留在浏览器历史记录里，而 POST 中的参数不会被保留。

GET 请求在 URL 中传送的参数是有长度限制的，而 POST 没有。

GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。

GET 参数通过 URL 传递，POST 放在请求 Body 中。

GET 产生一个 TCP 数据包，POST 产生两个 TCP 数据包。

GET 请求的 URL 传参有长度限制，而 POST 请求没有长度限制

GET 请求的参数只能是 ASCII 码，所以中文需要 URL 编码，而 POST 请求传参没有这个限制；

GET 产生一个 TCP 数据包；POST 产生两个 TCP 数据包

5.1.29Session 与 Cookie 有什么区别?

保存位置。SESSION 数据保存在服务器端，Cookie 数据保存在客户端浏览器

保存方式。SESSION 默认被存在在服务器的一个文件里，可以手动设置放在文件、数据库、或内存中；Cookie 默认保存在客户端内存中，如果设置了过期时间就保存在硬盘中。

依赖关系。SESSION 依赖 Cookie 来识别 session_id，如果浏览器禁用了 Cookie，SESSION 也会失效，此时可以通过 url 传递 session_id。

安全性。因为 SESSION 数据保存在服务端，所以 SESSION 安全性比 Cookie 高。

尺寸大小。SESSION 基本上没有大小限制，COOKIE 保存的内容比较小，具体由浏览器决定。

服务器性能。SESSION 对服务器的压力会更大一些，而 Cookie 放在客户端，所以对服务器基本没影响

5.1.30TCP 和 UDP 有什么区别

TCP 面向连接（如打电话要先拨号建立连接）;UDP 是无连接的，即发送数据之前不需要建立连接

TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP 尽最大努力交付，即不保证可靠交付

TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流;UDP 是面向报文的 UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）

每一条 TCP 连接只能是点到点的;UDP 支持一对一，一对多，多对一和多对多的交互通信

TCP 首部开销 20 字节;UDP 的首部开销小，只有 8 个字节

TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道

5.1.31什么是 TCP/IP？

TCP/IP，也就是互联网协议套件（英语：Internet Protocol Suite，缩写 IPS）是一个网络通信模型，以及一整个网络传输协议家族，为网际网络的基础通信架构。

因为该协议家族的两个核心协议：TCP（传输控制协议）和 IP（网际协议）为该家族中最早通过的标准，所以它常被通称为 TCP/IP 协议族，简称 TCP/IP。

由于在网络通讯协议普遍采用分层的结构，当多个层次的协议共同工作时，类似计算机科学中的堆栈，因此又被称为 TCP/IP 协议栈。

TCP/IP 提供点对点的链接机制，将数据应该如何封装、定址、传输、路由以及在目的地如何接收，都加以标准化。

它将软件通信过程抽象化为四个抽象层，采取协议堆栈的方式，分别实现出不同通信协议。

协议族下的各种协议，依其功能不同，被分别归属到这四个层次结构之中，常被视为是简化的七层 OSI 模型。

5.1.32在 API 测试中测试的常用协议是什么？

HTTP

JMS

REST

SOAP

UDDI

5.1.33cookie 有什么作用？

cookie 可以解决 http 的无状态的问题，与服务器进行交互，作为 http 规范存在。它具有极高的简便性、可扩展性和可用性，也可以通过加密和 SSL 技术来提高其安全性。因此推荐使用 cookie 作为标识而不是身份验证的工具。

其中 cookie 的作用就是为了解决 HTTP 协议无状态的缺陷所作出的努

力

5.1.34Cookie 测试的测试点

1.禁止使用 Cookie

设置浏览器禁止使用 Cookie，访问网页后，检查存放 Cookie 文件中未生成相关文件；

2.Cookie 存储路径

按照操作系统和浏览器对 Cookie 存放路径的设置，检查存放路径是否与设置一致；

3.Cookie 过期检查

按照 Cookie 过期时间，检查存放文件该 Cookie 是否被自动删除

4.检查浏览器中 Cookie 选项

通过不同浏览器，设置是否接受 Cookie 文件，如同意接受 Cookie，检查存放路径中是否存在 Cookie 文件

5.浏览器删除 Cookie

通过浏览器的设置，删除 Cookie 文件

6.Cookie 加密

提交敏感信息时，数据应加密

7.Cookie 保存信息

验证 Cookie 能正常工作

8.篡改 Cookie

修改 Cookie 内容，查看系统功能是否出现异常，或数据错乱

9.Cookie 的兼容性

使用不同类型，或同一类型不同版本的浏览器，检查 cookie 文件的兼容性

10.刷新操作对 cookie 的影响

进行刷新操作后，是否重新生成 cookie 文件或是对 cookie 文件进行修改

11.检查 cookie 内容存储是否完整正确

若 cookie 进行了加密，先对 cookie 文件内容进行解密，然后检查是否按照设计要求存储了相关所有的 cookie 记录信息。

12.对应硬盘存储空间没有空闲时，是否能进行 cookie 内容的有效存储

13.多次做相同的操作或设置，检查是否更新或添加了新的 cookie 文件

按照设计要求进行判断

14.如果使用 cookie 来统计次数，则要检测是否统计正确

例如通过用户登录次数进行统计,访问网站，每次都需要输入用户名和密码,分第一次登陆还是非第一次登录

5.1.35cookie 的缺点

（1） 大小和数目受限制。浏览器对一个域 cookie 的条目数有上限要求，且每个 cookie 的大小不得超过 4kb。

（2）存在安全性问题，易被人拦截。

（3）需要指定域，不可以跨域

（4）浪费带宽，因为我每次请求一个新的页面，cookie 都会被自动发送过去。

（5）有的移动端浏览器不支持 cookie 或浏览器禁用 cookie

（6）有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

5.1.36cookie 与 session 的区别

1、cookie 数据存放在客户的浏览器上，session 数据放在服务器上。

2、cookie 不是很安全，别人可以分析存放在本地的 cookie 并进行 cookie 欺骗考虑到安全应当使用 session。

3、session 会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能考虑到减轻服务器性能方面，应当使用 cookie。

4、单个 cookie 保存的数据不能超过 4K，很多浏览器都限制一个站点最多保存 20 个 cookie。

5、所以个人建议：

（1）将登陆信息等重要信息存放为 session

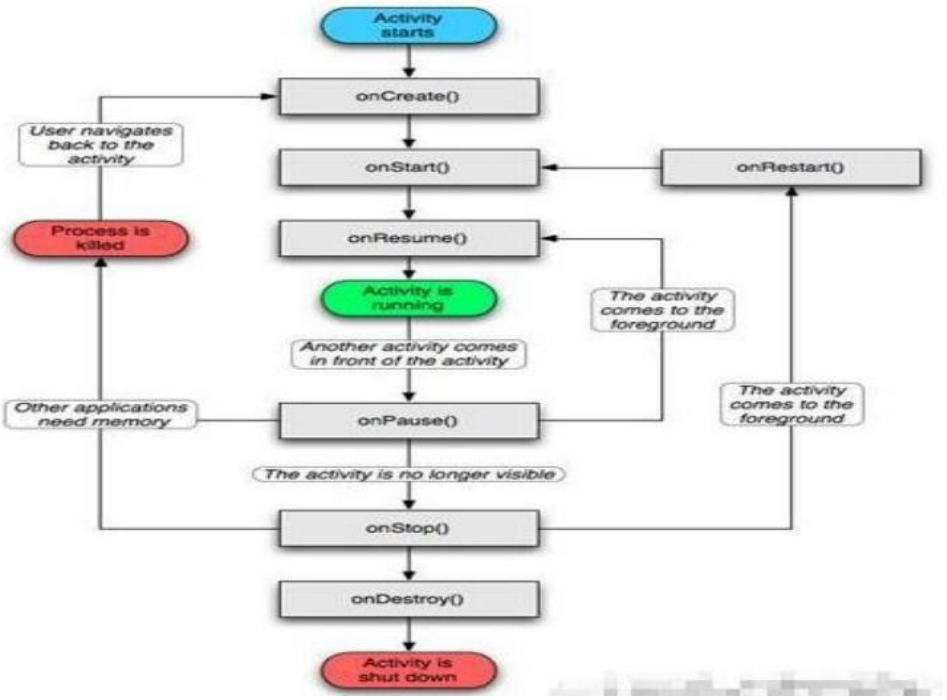
（2）其他信息如果需要保留，可以放在 cookie 中

6 App 测试

6.1 APP 测试

6.1.1 什么是 Android 四大组件？

Android 的四大组件包括：Activity、Service、BroadcasReceiver、ContentProvider



6.1.2 当点击 APP 图标启动程序，说明将要发生那些过程？

1.点击桌面 app 图标，Launcher 进程采用 Binder IPC 向 system_server 进程发起 startActivity 请求；

2.system_server 进程收到请求后，向 zygote 进程发送创建进程的请求（zygote 进程是 Android 系统的第一个进程，zygote 意为受精卵，所有进程都是由它孵化而来） 3.zygote 进程 fork 出新的子进程，即 App 进程；

4.App 进程，通过 Binder IPC 向 system_server 进程发起 attachApplication 请求

5.system_server 进程收到请求后，进行一系列的准备工作，通过 Binder IPC 向 App 进程发送

scheduleLaunchActivity 请求

6.App 的 binder 线程（ Application Thread ） 在收到请求后， 通过 handler 向主线程发送 LAUNCH_ACTIVITY 消息

7.主线程收到 Message 后，通过发射机制创建目标 Activity，并回调 Activity.onCreate()方法

6.1.3 APP 测试的内容主要包括哪些，如何开展？

功能测试：

1. 业务逻辑正确性测试：依据：产品文档->测试用例编写

兼容性测试：

1.系统版本：Android:官方版本,定制版本;IOS：官方提供版本

2.分辨率：720 * 1280 1080* 1920

3.网络情况:2g 3g 4g 5g Wi-Fi

异常测试

1.热启动应用:应用在后台长时间待机;应用在后台待机过程中，手机重启

2.网络切换和中断恢复:网络切换;中断恢复：

3.电话信息中断恢复 升级，安装

卸载测试

1.升级测试：临近版本升级(1.0->1.1);跨版本(1.0->....->2.2)

2.安装测试：首次安装;覆盖安装(同版本，不同版本覆盖);卸载后安装

3.卸载测试：首次卸载;卸载安装后在卸载

健壮性测试

1.手机资源消耗：cpu，内存

2.流量消耗：图片，数据，视频

3.电量测试

4.崩溃恢复

6.1.4 Android 的兼容性测试都考虑哪些内容？

品牌机型兼容：根据市场占有率、发布时间等指标对主流、最新机型进行重点兼容

ROM 兼容：需兼容原生的 ROM（2.1、2.2、2.3、4.0、4.1、4.2）；第三方 ROM（小米、百度易、点心、魅族、阿里云……）

屏幕兼容：需兼容 HVGA、VGA、WVGA、FWVGA、720p、1080p 屏幕分辨率，并考虑不同 PPI 的情况

软件兼容：安全类软件（百度手机管家、360 优化大师、360 安全卫士、QQ 手机管家、安卓优化大师、网秦、LBE），输入法软件（系统自带、Sogou、百度）

版本兼容：服务器端需要兼容产品早期版本所需的 API 接口

网络兼容：WiFi、3 大运营商的 2G,3G,4G 网络，需区分 WAP 和 NET 接入

6.1.5 针对 App 的安装功能，写出测试点？

- **安装**

1.正常安装测试，检查是否安装成功。

2.APP 版本覆盖测试。例如：先安装一个 1.0 版本的 APP,再安装一个高版本(1.1 版本)的 APP，检查是否被覆盖。

3.回退版本测试。例如：先装一个 2.0 版本的 APP,再安装一个 1.0 版本的 APP,正常情况下版本是可以回退的。

4.安装时内存不足，弹出提示。

5.根据安装手册操作，是否正确安装。

6.安装过程中的意外情况（强行断电、断网、来电话了、查看信息）等等，检查会发生的情况。

7.通过‘同步软件’，检查安装时是否同步安装了一些文件。

8.在不同型号、系统、屏幕大小、分辨率上的手机进行安装。

9.安装时是否识别有 SD 卡，并默认安装到 sd 卡中。

10.安装完成后，能否正常启动应用程序。

11.安装完成后，重启手机能否正常启动应用程序。

12.安装完成后，是否对其他应用程序造成影响。

13.安装完成后，能否添加快捷方式。

14.安装完成后，杀毒软件是否会对其当做病毒处理。

15.多进程进行安装，是否安装成功。

16.在安装过程中，所有的提示信息必须是英文或者中文，提示信息中不能出现代码、符号、乱码等。

17.安装之后，是否自动启动程序。

18.是否支持第三方安装。

19.在安装中点击取消。

- **卸载**

1.用自己的卸载程序进行卸载，检查是否卸载干净。

2.用第三方工具，检查是否卸载干净。

3.在卸载过程中，点击取消按钮，看是否正常退出卸载程序，检查软件是否还能继续正常使用。

4.卸载过程中，出现意外（比如手机关机，没电，查看信息，接打电话），程序是否还能运行。

5.在卸载过程中，突然重启设备，再次访问程序，是否还能运行。

6.在没用使用程序时，删除目录文件，看程序是否能运行。

7.在使用过程中，直接删除目录文件，程序是否还能运行。

8.不同系统、硬件环境、网络环境下进行卸载。

9.卸载成功后，是否对其他程序有影响。

10.卸载后再次安装，是否正常使用。

11.在卸载过程中，所有的提示信息必须是英文或者中文，提示信息中不能出现代码、符号、乱码等。

- **更新**

1.当客户端有新版本时，提示更新。

2.非强制更新，可以取消更新，旧版本正常使用，下次使用软件时，仍然会出现更新提示。

3.强制更新，强制更新而用户没有更新时，退出客户端，下次启动，依然提示更新。

4.不卸载更新，检查是否可以更新。

5.不卸载更新，检查资源同名文件如图片等是否更新成最新版本。

6.非 wifi 网络下，提示是否更新，取消就加入待下载，wifi 下自动更新。

6.1.6 常用的 ADB 命令？

- adb --help / adb :看见帮助信息
- adb start-server:启动 adb 服务
- adb kill-server:关闭 adb 服务
- adb devices:查看手机设备号
- adb shell getprop ro.build.version.release:获取系统版本
- adb push 电脑手机
- adb pull 手机电脑
- adb logcat | grep(unix) 包 名
- adb logcat | findstr(win) 包 名
- adb shell :进入 shell 命令行，可以操作 Linux 命令
- adb shell dumpsys window windows | grep mFocusedApp:获取包名 启动名(win：adb shell dumpsys window windows | findstr mFocusedApp)
- adb install 路径/apk 文件:安装 apk 到手机上
- adb uninstall 包名:卸载 app 从手机上
- adb shell am start -W 包名/启动名:app 启动时间

6.1.7 在查看 logcat 命令日志时候怎么内容保存到本地文件？

输出重定向：logcat >> log_file_name

6.1.8 App 崩溃（闪退），可能是什么原因导致的？

缓存垃圾过多：由于安卓系统的特性,如果长时间不清理垃圾文件.会导致越来越卡.也会出现闪退情况.

运行的程序过多,导致内存不足

应用版本兼容问题：如果应用版本太低，会导致不兼容，造成闪退。此外，有些新版本在调试中，也会造成应用闪退。解决方法：如果是版本太旧，更新为新版本即可；如果是新版本闪退，可能是应用在改版调试，可卸载后安装旧版。

检查 APP 中访问网络的地方，组件中的 ImageView 是否可以正常的下载并显示到 app 页面上。检查 APP 的 sdk 和手机的系统是否兼容。

在一些特定情况下的闪退,比如播放视频,在 Android5.0 升级到 Android6.0 的时候,有些系统 API 老版本有,新版本没有,到时回去对象的时候失败,报空,系统就会出现闪退问题.

6.1.9 如何测试监测 app 的内存使用、CPU 消耗、流量使用情况？

adb shell top

Android 应用性能测试通常包括：启动时间、内存、CPU、耗电量、流量、流畅度等根据手机的使用应用频度和强度不同，可将应用使用强度分为如下几种状态：

空闲状态：指启动应用后，不做任何操作或切换到后台运行的情况称为空闲状态，该情况为应用对内存的消耗是最小的。

中强度状态：该情况用户使用应用的强度和时间长短不确定，相对来说使用时长偏长。

高强度状态：该种情况为应用内高频率的使用，用户很少达到，跑 monkey 时可认为高强度状态， 该种情况常用来测试应用内存泄漏的情况测试时，可根据用户的操作习惯模拟应用使用频率和强度等级。使用 adb 命令，手机连接电脑开启 USB 调试模式，进入 adbshell。

（1）查看 CPU 占用率

使用命令 top -m 10 -s cpu（-t 显示进程名称，-s 按指定行排序，-n 在退出前刷新几次，-d 刷新间隔，-m 显示最大数量），如下图：

```
User 31%, System 25%, IOW 0%, IRQ 0%
User 262 + Nice 14 + Sys 225 + Idle 382 + IOW 5 + IRQ 0 + SIRQ 2 = 890

  PID PR CPU% S  #THR     VSS     RSS PCY UID      Name
30522  2  32% R    89 1240904K 115244K  fg u0_a177  com.baidu.tieba
  255  0   7% R    19 149396K   4256K  fg system   /system/bin/surfaceflin
  788  1   2% S   113 1457960K 115224K  fg system   system_server
10054  0   1% R     1   4520K   1500K  fg shell    top
25607  0   1% S   128 1276172K  77504K  bg u0_a127  com.tencent.mobileqq
  290  5   0% S     5  25052K    560K  fg system   /system/bin/aal
10099  2   0% S     1      0K      0K  fg root     kworker/u16:5
 8442  2   0% S     1      0K      0K  fg root     kworker/u16:2
23515  0   0% S    56 1344228K  41224K  bg u0_a130  com.mfashiongallery.em
 7535  1   0% S     1      0K      0K  fg root     kworker/u16:7
```

参数含义：

- PID：progressidentification，应用程序 ID
- S: 进程的状态，其中 S 表示休眠，R 表示正在运行，Z 表示僵死状态，N 表示该进程优先值是负数。
- #THR：程序当前所用的线程数
- VSS：VirtualSet Size 虚拟耗用内存（包含共享库占用的内存）
- RSS： ResidentSet Size 实际使用物理内存（包含共享库占用的内存）
- UID：UserIdentification，用户身份 ID
- Name:应用程序名称

在测试过程中，QA 需要关注对应包的 cpu 占用率，反复进行某个操作，cpu 占用过高且一直无法释放，此时可能存在风险。如果你想筛选出你自己的应用的话可以用下面命令 top -d 3| grep packageName

```
130|shell@hermes:/ $ top -d 3| grep com.baidu.tieba
30522  1   0% S    79 1190024K  93004K  fg u0_a177  com.baidu.tieba
30595  1   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  2  16% R    79 1190024K  93456K  fg u0_a177  com.baidu.tieba
30595  2   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  2  18% S    79 1190352K  93744K  fg u0_a177  com.baidu.tieba
30595  2   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  0  18% R    79 1190352K  93908K  fg u0_a177  com.baidu.tieba
30595  2   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  0  20% S    79 1190352K  94260K  fg u0_a177  com.baidu.tieba
30595  0   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  1   8% S    79 1190352K  94260K  fg u0_a177  com.baidu.tieba
30595  2   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
30522  0   0% S    79 1190352K  94260K  fg u0_a177  com.baidu.tieba
30595  2   0% S    29 1018196K  43296K  fg u0_a177  com.baidu.tieba:remote
```

（1）查看内存使用情况

（2）dumpsys meminfo <package_name>或 dumpsys meminfo <package_id>

```
255|shell@hermes:/ $ dumpsys meminfo 30522
Applications Memory Usage (kB):
Uptime: 31122761 Realtime: 160602067

** MEMINFO in pid 30522 [com.baidu.tieba] **
                   Pss  Private  Private  Swapped     Heap     Heap     Heap
                 Total    Dirty    Clean    Dirty     Size    Alloc     Free
                ------   ------   ------   ------   ------   ------   ------
  Native Heap     9908     9856        0     1896    18572    12792     1507
  Dalvik Heap    29494    29236        0    16132    46560    45625      935
 Dalvik Other     2260     2188        0        0
        Stack      700      700        0        0
       Ashmem      128      124        0        0
    Other dev        5        0        4        0
     .so mmap     5395      432     3872     1680
     .apk mmap     250        0      160        0
     .ttf mmap     919        0      836        0
     .dex mmap    6680        0     6032        0
     code mmap    3712        0     1224        0
    image mmap    3714     1752      152      192
    Other mmap     434        8      392        0
      Graphics   12488    12488        0        0
         GL      14419    14419        0        0
       Unknown    7628     7600        0       56
         TOTAL   98134    78803    12672    19956    65132    58417     2442

 Objects
               Views:      233         ViewRootImpl:        1
         AppContexts:        9           Activities:        1
              Assets:        7       AssetManagers:        7
        Local Binders:       38       Proxy Binders:       36
     Death Recipients:        3
      OpenSSL Sockets:        0
```

参数含义：

- Naitve Heap Size: 从 mallinfo usmblks 获得，代表最大总共分配空间
- Native Heap Alloc: 从 mallinfo uorblks 获得，总共分配空间
- Native Heap Free: 从 mallinfo fordblks 获得，代表总共剩余空间
- Native Heap Size 约等于 Native Heap Alloc + Native Heap Free
- mallinfo 是一个 C 库， mallinfo 函数提供了各种各样的通过 C 的 malloc（）函数分配的内存的统计信息。
- Dalvik Heap Size:从 Runtime totalMemory()获得，Dalvik Heap 总共的内存大小。
- Dalvik Heap Alloc: Runtime totalMemory()-freeMemory() ，Dalvik Heap 分配的内存大小。
- Dalvik Heap Free:从 Runtime freeMemory()获得，Dalvik Heap 剩余的内存大小。
- Dalvik Heap Size 约等于 Dalvik HeapAlloc + Dalvik Heap Free 重点关注如下几个字段：

- Native/Dalvik 的 Heap 信息中的 alloc ：具体在上面的第一行和第二行，它分别给出的是 JNI 层和 Java 层的内存分配情况，如果发现这个值一直增长，则代表程序可能出现了内存泄漏。

Total 的 PSS 信息：这个值就是你的应用真正占据的内存大小，通过这个信息，你可以轻松判别手机中哪些程序占内存比较大了。

6.1.10弱网测试怎么测

弱网环境测试主要依赖于弱网环境的模拟。环境搭建方式一般有两种：软件方式和硬件方式。软件方式的成本低，主要就是通过模拟网络参数来配置弱网环境，通常来讲可以达到测试目的.一般可通过热点共享设置。在各类网络软件中，主要就是对带宽、丢包、延时等进行模拟弱网环境。如果要求更接近弱网环境， 比如现在很多的专项测试，会更倾向于通过硬件方式来协助测试，但这种方式相对会麻烦很多，一般会由网维协助搭建。当然，对于有些无法模拟的情况，只能靠人工移动到例如电梯、地铁等信号比较弱的地方。

6.1.11“//*[contains(@text,”登录”)]”是什么意思

定位第一个元素 text 属性包含登录的元素

6.1.12Appium 都有哪些启动方式

1.客户端启动

2.命令行启动

7 管理工具

7.1 管理工具

7.1.1 简述常用的 Bug 管理或者用例管理工具,并且描述其中一个工作流程?

常用：testlink，QC，mantis，禅道，TAPD，JIRA 。

TAPD：产品创建(需求，计划，模块)-->项目创建（PM 排期、任务分解）-->研发(编码、单元 测试等)--> 测试(测试计划，用例，执行，bug，报告等)。

7.1.2 禅道和 qc 的区别?

同为缺陷管理工具。

QC

作为缺陷管理工具，QC 在缺陷管理方面，做的相对完善。提 bug 页面：填写内容可以根据测试需求， 不断修改添加新的字段；以我上一家公司为例，在提 bug 过程中，有一下几个必填项：Bug 状态（new、fixed、closed 等）、发现人员、缺陷发现阶段(测试阶段、上现阶段等)、缺陷来源（测试人员给出的 bug 定位）、Bug 分类（功能、性能等问题）、测试阶段（单元测试、集成测试、系统测试等）、归属需求、缺陷回归次数、优先级、分配给，这些必填项再加上 bug 标题和操作描述、上传附件，使很多疑问都变得清晰。

缺陷查看页面：可以根据自己需要选择要呈现的字段，相对人性化可操作，每个显示的字段都可以进行筛选，使研发人员很快能定位到属于自己的 bug，再根据 bug 状态、优先级进行筛选，使未完结的 bug 能有序并无遗漏地完成修改；页面还有注释功能，研发人员能写出针对本问题的各种感想，方便完善而又人性化。

禅道（开源版）

禅道涉及面非常广，但是在缺陷管理这方面，与老牌的 QC 还是略逊一筹。提 bug 页面：页面是非常清晰整洁的 web 页面，但是需要填写的字段，并没有完全覆盖开发和测试人员的全部需求。页面字段：产品模块（对应 QC 中的项目）、所属项目（对应 QC 中的需求）、影响版本（bug 所属版本？）、当前指派（修改 bug 的人员）、bug 标题、重现步骤、相关需求（页面标注了这个字段，但是什么也没有显示， 并且没有可填写的位置）、相关任务、类型/严重。

8 Python 基础

8.1 Python 基础

8.1.1 斐波那契数列求 N?

```
N=int(input("N:"))
fib=[0 for x in range(N+1)]
fib[0]=0
fib[1]=1
for i in range(2,N+1):
    fib[i]=fib[i-1]+fib[i-2]
print fib[N]
```

8.1.2 字符串反序输出？

print(a_str[::-1])

8.1.3 判断回文？

astr[::-1] == a_str

8.1.4 统计 python 源代码文件中代码行数，去除注释，空行，进行输出？

```
#-*-coding:utf-8-*-

import os

if __name__=='__main__':
    codeline=0
    expline=0
    blankline=0

    filename=raw_input('Please input the  file name:')

    fi=open(filename)
    while fi.tell()!=os.path.getsize(filename):
        temp=fi.readline()
        if temp.startswith('#'):
            expline+=1
        elif temp=='/n':
            blankline+=1
        elif temp.startswith('"""'):
            expline+=1
            while True:
                temp=fi.readline()
                expline+=1
                if temp.endswith('"""/n'):
                    break
        else:
            codeline+=1
    else:
        print 'the codeline is:'+str(codeline)
        print 'the expline is:'+str(expline)
        print 'the blank line is:'+str(blankline)
```

8.1.5 python 调用 cmd 并返回结果？

python 的 OS 模块。

- OS 模块调用 CMD 命令有两种方式：os.popen(),os.system(). 都是用当前进程来调用。
- os.system 是无法获取返回值的。 当运行结束后接着往下面执行程序。 用法如：OS.system("ipconfig").
- OS.popen 带返回值的，如何获取返回值。如
- p=os.popen(cmd)
- print p.read().得到的是个字符串。
- 这两个都是用当前进程来调用，也就是说它们都是阻塞式的。管道 subprocess 模块。

- 运行原理会在当前进程下面产生子进程。
- sub=subprocess.Popen(cmd,shell=True,stdout=subprocess.PIPE)
- sub.wait()
- print sub.read()

8.1.6 冒泡排序

```
def bsort(alist):
    l = len(alist)
    for i in range(l-1):
        flag = 1
        for j in range(l-i-1):
            if alist[j] > alist[j+1]:
                alist[j],alist[j+1] = (alist[j+1],alist[j])
                flag = 0
        if flag == 1:
            return alist
```

8.1.7 1,2,3,4 这 4 个数字，能组成多少个互不相同的且无重复的三位数，都是多少？

```
1  i = 0
2  for x in range(1,5):
3      for y in range(1,5):
4          for z in range(1,5):
5                  if (x!=y) and (y!=z) and (z!=x):
6                      i += 1
7                      if i%4:
8                          print("%d%d%d" % (x, y, z), end=" | ")
9                      else:
10                         print("%d%d%d" % (x, y, z))
```

8.1.8 4.给定一个整数 N，和一个 0-9 的数 K，要求返回 0-N 中数字 K 出现的次数

```
def digitCounts(self, k, n):
  count = 0
  for i in range(n+1):
    if i == 0 and i == k:
      count += 1
    while( i // 10 >= 0 and i != 0):
      j = i % 10
      if j == k:
        count  += 1
          i = i //10
  return count
```

8.1.9 请用 python 打印出 10000 以内的对称数（对称数特点：数字左右对称，如：1,2,11,121,1221 等）

8.1.10判断 101-200 之间有多少个素数，并输出所有的素数

```
from math import sqrt
#定义一个是否素数函数，如果n等于1，则返回false
def is_prime(n):
    if n == 1:
        return False
    for i in range(2,int(sqrt(n))+1):
        if n%i == 0:
            return False
    return True
```

8.1.11 一个输入三角形的函数，输入后输出是否能组成三角形，三角形类型，请用等价类划分法设计测试用例

```
a, b, c = map(int, input().split())
if a < +c and b < a + c and c < a + b:
    if a == b == c:
        print('等边三角形')
    elif a == b or a == c or b == c:
        if a * a + b * b == c * c or a * a + c * c == b * b or b * b + c * c == a * a:
            print('等腰直角三角形')
        else:
            print('等腰三角形')
    elif a * a + b * b == c * c or a * a + c * c == b * b or b * b + c * c == a * a:
        print('直角三角形')
    else:
     print('普通三角形')
else:
    print('无法构成三角形')
```

8.1.12编程题

8.1.12.1 请编写一个完整出程序，实现如下功能：从键盘输入数字 n，程序自动计算 n！，并输出。（注

1:n！=1*2*3*……*n，注 2:请使用递归实现）（可以使用任何开发语言，最好使用 JAVA）

8.1.12.2 如果现在有一台刚安装了 WinXP 的计算机，请简单说明如何能够让以上程序得以运行。

8.1.12.3 写代码将如下数据从小到大排序，语言不限。（不可以直接使用 sort（）等排序方法） 234，82，5，10，86，90

8.1.12.4 如何使用 Python 发送一封邮件？

8.1.12.5 Linux 下如何查看 ip 地址，如何用 Python 或 TCL 删除当前文件夹下所有文件以及目录？

8.1.12.6 给 x 变量赋值为 abccaefs，并统计 x 变量中单词出现的次数（java 或 Python 任选一种语言编写）

8.2 输入与输出

8.2.1 代码中要修改不可变数据会出现什么问题？抛出什么异常？

代码不会正常运行，抛出 TypeError 异常。

3.代码中要修改不可变数据会出现什么问题？抛出什么异常？

代码不会正常运行，抛出 TypeError 异常。

8.2.2 print 调用 Python 中底层的什么方法?

print 方法默认调用 sys.stdout.write 方法，即往控制台打印字符串。

8.2.3 简述你对 input()函数的理解?

在 Python3 中，input()获取用户输入，不论用户输入的是什么，获取到的都是字符串类型的。在 Python2 中有 raw_input()和 input(), raw_input()和 Python3 中的 input()作用是一样的， input()输入的是什么数据类型的，获取到的就是什么数据类型的。

8.2.4 python 两层列表怎么提取第二层的元素

```
aa = [[(55736,)], [(55739,)], [(55740,), (55801,)], [(55748,)], [(55783,), (55786,),
(55787,), (55788,)],[(55817,), (55821,)], [(55818,)]]



def getelement(aa):
    for elem in aa:
        if type(elem) == type([]):
            for element in getelement(elem):
                yield element
        else:
            yield elem
    for elem in getelement(aa):


print(elem)(55736, )(55739,)(55740,)(55801,)(55748,)(55783,)(55786,)(55787,)(55788,
)(55817,)(55821,)(55818,)
```

8.3 条件与循环

8.3.1 阅读下面的代码，写出 A0，A1 至 An 的最终值？

1.	A0 = dict(zip(('a', 'b', 'c', 'd', 'e'), (1, 2, 3, 4, 5)))
2.	A1 = range(10)
3.	A2 = [i for i in A1 if i in A0]
4.	A3 = [A0[s] for s in A0]
5.	A4 = [i for i in A1 if i in A3]
6.	A5 = {i:i*i for i in A1}
7.	A6 = [[i, i*i] for i in A1]

答案：

1.	A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4}
2.	A1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
3.	A2 = []
4.	A3 = [1, 3, 2, 5, 4]
1.	A4 = [1, 2, 3, 4, 5]
2.	A5 = {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
3.	A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36],[7, 49],[8, 64] [9,81]]

8.3.2 range 和 xrange 的区别？

两者用法相同，不同的是 range 返回的结果是一个列表，而 xrange 的结果是一个生成器，前者是直接开辟一块内存空间来保存列表，后者是边循环边使用，只有使用时才会开辟内存空间，所以当列表很长时，使用 xrange 性能要比 range 好。

8.3.3 考虑以下 Python 代码，如果运行结束，命令行中的运行结果是什么？

```
l = []
for i in range(10):
    l.append({'num':i})
print(l)
```

结果： [{'num': 0}, {'num': 1}, {'num': 2}, {'num': 3}, {'num': 4}, {'num': 5}, {'num': 6}, {'num': 7}, {'num': 8}, {'num': 9}]

8.3.4 在考虑以下代码，运行结束后的结果是什么？

```
l = []
a = {'num':0}
for i in range(10):
    a['num'] = i
    l.append(a)
print(l)
```

[{'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}, {'num': 9}]

8.4 字典

8.4.1 什么是字典

dict:字典，字典是一组键(key)和值(value)的组合，通过键(key)进行查找，没有顺序， 使用大括号”{}”;

应用场景：

dict，使用键和值进行关联的数据;

8.4.2 现有字典 d={‘a’:24，‘g’:52，‘i’:12，‘k’:33}请按字典中的 value 值进行排序？

sorted(d.items()， key = lambda x:x[1]) 。

8.4.3 说一下字典和 json 的区别？

字典是一种数据结构，json 是一种数据的表现形式，字典的 key 值只要是能 hash 的就行，json 的必须是字符串。

8.4.4 什么是可变、不可变类型？

可变不可变指的是内存中的值是否可以被改变，不可变类型指的是对象所在内存块里面的值不可以改变，有数值、字符串、元组；可变类型则是可以改变，主要有列表、字典。

8.4.5 存入字典里的数据有没有先后排序？

存入的数据不会自动排序，可以使用 sort 函数对字典进行排序。

8.4.6 字典推导式？

d = {key: value for (key, value) in iterable}

8.4.7 现有字典 d={ ‘a’ :24，’ g’ :52，’ l’ :12，’ k’ :33}请按字 典中的 value 值进行排序？

sorted(d.items()， key = lambda x:x[1])

8.5 字符串

8.5.1 什么是 Python 字符串

str:字符串是 Python 中最常用的数据类型。我们可以使用引号('或")来创建字符串。

8.5.2 如何理解 Python 中字符串中的¥字符？

有三种不同的含义：

1、转义字符

2、路径名中用来连接路径名

3、编写太长代码手动软换行。

8.5.3 请反转字符串“aStr”？

print('aStr'[::-1])

8.5.4 请按 alist 中元素的 age 由大到小排序

```
alist = [{'name': 'a', 'age': 20}, {'name': 'b', 'age': 30}, {'name': 'c', 'age': 25}]

def sort_by_age(list1):

    return sorted(alist, key = lambda x: x['age'], reverse = True)


sort_by_age(alist)
```

8.6 列表

8.6.1 什么是 Python 中的 list

list:是 Python 中使用最频繁的数据类型，在其他语言中通常叫做数组，通过索引进行查找，使用方括号”[]”, 列表是有序的集合。应用场景：定义列表使用 [ ] 定义，数据之间使用 “，”分割。

列表的索引从 0 开始：索引就是数据在列表中的位置编号，索引又可以被称为下标。

【注意】：从列表中取值时,如果超出索引范围,程序会产生异常。IndexError: list index out of range

8.6.2 列表增加

列表名.insert(index, 数据)：在指定位置插入数据(位置前有空元素会补位)。# 往列表 name_list 下标为 0 的地方插入数据

In [3]: name_list.insert(0, "Sasuke") In [4]: name_list

Out[4]: ['Sasuke', 'zhangsan', 'lisi', 'wangwu', 'zhaoliu']

现有的列表下标是 0-4，如果我们要在下标是 6 的地方插入数据，那个会自动插入到下标为 5 的地方， 也就是# 插入到最后

In [5]: name_list.insert(6, "Tom") In [6]: name_list

Out[6]: ['Sasuke', 'zhangsan', 'lisi', 'wangwu', 'zhaoliu', 'Tom']

列表名.append(数据)：在列表的末尾追加数据(最常用的方法)。

In [7]: name_list.append("Python")

In [8]: name_list

Out[8]: ['Sasuke', 'zhangsan', 'lisi', 'wangwu', 'zhaoliu', 'Tom', 'Python'] 列表.extend(Iterable)：将可迭代对象中的元素追加到列表。

有两个列表 a 和 b a.extend(b) 会将 b 的元素追加到列表 In [10]: a = [11, 22, 33]

In [11]: b = [44, 55, 66]

In [12]: a.extend(b)

In [13]: a

Out[13]: [11, 22, 33, 44, 55, 66]# 有列表 c 和 字符串 c In [14]: c = ['j', 'a', 'v', 'a']

In [15]: d = "python" In [16]: c.extend(d)

In [17]: c

Out[17]: ['j', 'a', 'v', 'a', 'p', 'y', 't', 'h', 'o', 'n']

8.6.3 取值和修改取值：列表名[index] ：根据下标来取值。

In [19]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu"]

In [20]: name_list[0]

Out[20]: 'zhangsan'

In [21]: name_list[3]

Out[21]: 'zhaoliu'

修改：列表名[index] = 数据：修改指定索引的数据。

In [22]: name_list[0] = "Sasuke"

In [23]: name_list

Out[23]: ['Sasuke', 'lisi', 'wangwu', 'zhaoliu']

8.6.4 删除 del 列表名[index]：删除指定索引的数据。

In [25]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu"]# 删除索引是 1 的数据

In [26]: del name_list[1]

In [27]: name_list

Out[27]: ['zhangsan', 'wangwu', 'zhaoliu']

8.6.5 列表名.remove(数据)：删除第一个出现的指定数据。

In [30]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu", "lisi"]# 删除 第一次出现的 lisi 的数据

In [31]: name_list.remove("lisi")

In [32]: name_list

Out[32]: ['zhangsan', 'wangwu', 'zhaoliu', 'lisi']

8.6.6 列表名.pop()：删除末尾的数据,返回值：返回被删除的元素。

In [33]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu"]# 删除最后一个元素

In [34]: name_list.pop()

Out[34]: 'zhaoliu' In [35]: name_list

Out[35]: ['zhangsan', 'lisi', 'wangwu']

8.6.7 列表名.pop(index)：删除指定索引的数据，返回被删除的元素。

In [36]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu"]# 删除索引为 1 的数据 lisi

In [37]: name_list.pop(1)

Out[37]: 'lisi'

In [38]: name_list

Out[38]: ['zhangsan', 'wangwu', 'zhaoliu']

8.6.8 列表名.clear()：清空整个列表的元素。

In [40]: name_list = ["zhangsan", "lisi", "wangwu", "zhaoliu"]

In [41]: name_list.clear()

In [42]: name_list Out[42]: []

8.6.9 排序列表名.sort()：升序排序 从小到大。

In [43]: a = [33, 44, 22, 66, 11]

In [44]: a.sort()

In [45]: a

Out[45]: [11, 22, 33, 44, 66]

8.6.10列表名.sort(reverse=True)：降序排序 从大到小。

In [46]: a = [33, 44, 22, 66, 11]

In [47]: a.sort(reverse=True)

In [48]: a

Out[48]: [66, 44, 33, 22, 11]

8.6.11列表名.reverse()：列表逆序、反转。

In [50]: a = [11, 22, 33, 44, 55]

In [51]: a.reverse()

In [52]: a

Out[52]: [55, 44, 33, 22, 11]

8.6.12len(列表名)：得到列表的长度。

In [53]: a = [11, 22, 33, 44, 55]

In [54]: len(a)

Out[54]: 5

8.6.13列表名.count(数据)：数据在列表中出现的次数。

In [56]: a = [11, 22, 11, 33, 11]

In [57]: a.count(11)

Out[57]: 3

8.6.14列表名.index(数据)：数据在列表中首次出现时的索引，没有查到会报错。

In [59]: a = [11, 22, 33, 44, 22]

In [60]: a.index(22)

Out[60]: 1

8.6.15if 数据 in 列表： 判断列表中是否包含某元素。

a = [11, 22, 33, 44 ,55]

if 33 in a:

print("找到了 ")

8.6.16循环遍历

使用 while 循环：

```
a = [11, 22, 33, 44, 55]
i = 0
while i < len(a):
    print(a[i])
    i += 1
```

使用 for 循环：

a = [11, 22, 33, 44, 55]

for i in a:

print(i)

8.6.17写一个列表生成式，产生一个公差为 11 的等差数列

```
print([x*11 for x in range(10)])
```

8.6.18给定两个列表，怎么找出他们相同的元素和不同的元素?

```
list1 = [1 , 2 , 3]
.
list2 = [3 , 4 , 5]
.
set1 = set(list1)
.
set2 = set(list2)
.
print(set1&set2)
.
print(set1^set2)
.
```

8.6.19请写出一段 Python 代码实现删除一个 list 里面的重复元素?

比较容易记忆的是用内置的 set：

```
l1 = ['b'，'c'，'d'，'b'，'c'，'a'，'a']
l2 = list(set(l1))
print l2
```

如果想要保持他们原来的排序：用 list 类的 sort 方法：

```
l1 = ['b'，'c'，'d'，'b'，'c'，'a'，'a']
l2 = list(set(l1))
l2.sort(key=l1.index)
print l2
```

也可以这样写：

```
l1 = ['b'，'c'，'d'，'b'，'c'，'a'，'a']
l2 = sorted(set(l1)，key=l1.index)
print l2
```

也可以用遍历：

```
l1 = ['b', 'c', 'd', 'b', 'c', 'a', 'a']
l2 = []
for i in l1:
if not i in l2:
l2.append(i)
print l2
```

8.6.20给定两个 list A ,B，请用找出 A ,B 中相同的元素，A ,B 中不同的元素

A、B 中相同元素：print(set(A)&set(B))

A、B 中不同元素：print(set(A)^set(B))

新的默认列表只在函数被定义的那一刻创建一次。当 extendList 被没有指定特定参数 list 调用时，这组 list 的值

随后将被使用。这是因为带有默认参数的表达式在函数被定义的时候被计算，不是在调用的时候被计算。

8.7 元组

tuple:元组，元组将多样的对象集合到一起，不能修改，通过索引进行查找，使用括号”()”；应用场景：把一些数据当做一个整体去使用，不能修改

8.8 集合

8.8.1 什么是集合

set:set 集合，在 Python 中的书写方式的{}，集合与之前列表、元组类似，可以存储多个数据，但是这些数据是不重复的。集合对象还支持 union(联合), intersection(交), difference(差)和 ysmmetric_difference(对称差集) 等数学运算.

8.8.2 快速去除列表中的重复元素

In [4]: a = [11,22,33,33,44,22,55]

In [5]: set(a)

Out[5]: {11, 22, 33, 44, 55}

8.8.3 交集：共有的部分

In [7]: a = {11,22,33,44,55}

In [8]: b = {22,44,55,66,77}

In [9]: a&b

Out[9]: {22, 44, 55}

8.8.4 并集：总共的部分

In [11]: a = {11,22,33,44,55}

In [12]: b = {22,44,55,66,77}

In [13]: a | b

Out[13]: {11, 22, 33, 44, 55, 66, 77}

8.8.5 差集：另一个集合中没有的部分

In [15]: a = {11,22,33,44,55}

In [16]: b = {22,44,55,66,77}

In [17]: b - a

Out[17]: {66, 77}

8.8.6 对称差集(在 a 或 b 中，但不会同时出现在二者中)

In [19]: a = {11,22,33,44,55}

In [20]: b = {22,44,55,66,77}

In [21]: a ^ b

Out[21]: {11, 33, 66, 77}

8.9 文件操作

8.9.1 4G 内存怎么读取一个 5G 的数据？(2018-3-30-lxy)

方法一：

可以通过生成器，分多次读取，每次读取数量相对少的数据（比如 500MB）进行处理，处理结束后在读取后面的 500MB 的数据。

方法二：

可以通过 linux 命令 split 切割成小文件，然后再对数据进行处理，此方法效率比较高。可以按照

行数切割， 可以按照文件大小切割。

8.9.2 现在要处理一个大小为 10G 的文件，但是内存只有 4G，如果在只修改 get_lines 函数而其他代码保持不变的情况下，应该如何实现？需要考虑的问题都有哪些？

```
def get_lines():
    l = []
    with open('file.txt', 'rb') as f:
        data = f.readlines(60000)
        l.append(data)
    yield l
```

要考虑到的问题有：内存只有 4G 无法一次性读入 10G 的文件，需要分批读入。分批读入数据要记录每次读入数据的位置。分批每次读入数据的大小，太小就会在读取操作上花费过多时间。

8.9.3 read、readline 和 readlines 的区别?

read:读取整个文件。

readline：读取下一行，使用生成器方法。

readlines：读取整个文件到一个迭代器以供我们遍历。

8.10 函数

8.10.1Python 函数调用的时候参数的传递方式是值传递还是引用传递？

Python 的参数传递有：位置参数、默认参数、可变参数、关键字参数。

函数的传值到底是值传递还是引用传递，要分情况：

不可变参数用值传递：像整数和字符串这样的不可变对象，是通过拷贝进行传递的，因为你无论如何都不可能在原处改变

不可变对象

可变参数是引用传递的：

比如像列表，字典这样的对象是通过引用传递、和 C 语言里面的用指针传递数组很相似，可变对象能在函数内部改变。

8.10.2对缺省参数的理解 ？

缺省参数指在调用函数的时候没有传入参数的情况下，调用默认的参数，在调用函数的同时赋值时，所传入的参数会替代默认参数。

*args 是不定长参数，他可以表示输入参数是不确定的，可以是任意多个。

**kwargs 是关键字参数，赋值的时候是以键 = 值的方式，参数是可以任意多对在定义函数的时候不确定会有多少参数会传入时，就可以使用两个参数。

8.10.3为什么函数名字可以当做参数用?

Python 中一切皆对象，函数名是函数在内存中的空间，也是一个对象。

8.10.4Python 中 pass 语句的作用是什么？

在编写代码时只写框架思路，具体实现还未编写就可以用 pass 进行占位，使程序不报错，不会进行任何操作。

8.11 内建函数

8.11.1map 函数和 reduce 函数？

①从参数方面来讲：

map()包含两个参数，第一个参数是一个函数，第二个是序列（列表 或元组）。其中，函数（即 map 的第一个参数位置的函数）可以接收一个或多个参数。

reduce()第一个参数是函数，第二个是序列（列表或元组）。但是，其函数必须接收两个参数。

②从对传进去的数值作用来讲：

map()是将传入的函数依次作用到序列的每个元素，每个元素都是独自被函数“作用”一次 。reduce()是将传人的函数作用在序列的第一个元素得到结果后，把这个结果继续与下一个元素作用（累积计算）。

8.11.2递归函数停止的条件？

递归的终止条件一般定义在递归函数内部，在递归调用前要做一个条件判断，根据判断的结果选择是继续调用自身，还是 return;返回终止递归。

终止的条件：

判断递归的次数是否达到某一限定值

判断运算的结果是否达到某个范围等，根据设计的目的来选择

8.11.3回调函数，如何通信的?

回调函数是把函数的指针(地址)作为参数传递给另一个函数，将整个函数当作一个对象，赋值给调用的函数。

8.11.4Python 主要的内置数据类型都有哪些？ print dir(‘a ’) 的输出？

内建类型：布尔类型、数字、字符串、列表、元组、字典、集合；输出字符串‘a’的内建方法；

8.11.5print(list(map(lambda x: x * x, [y for y in range(3)])))的输出？

[0， 1， 4]

8.12 Lambda

8.12.1什么是 lambda 函数？ 有什么好处？

lambda 函数是一个可以接收任意多个参数(包括可选参数)并且返回单个表达式值的函数

lambda 函数比较轻便，即用即仍，很适合需要完成一项功能，但是此功能只在此一处使用，连名字都很随意的情况下；

匿名函数，一般用来给 filter， map 这样的函数式编程服务;

作为回调函数，传递给某些应用，比如消息处理

8.12.2什么是 lambda 函数？它有什么好处？写一个匿名函数求两个数的和？

lambda 函数是匿名函数；使用 lambda 函数能创建小型匿名函数。这种函数得名于省略了用 def 声明函数的标准步骤；

```
f = lambda x, y: x + y

print(f(2017, 2018))
```

8.13 面向对象

8.13.1结构化程序设计和面向对象程序设计各自的特点及优缺点是什么？

结构化程序设计思想采用了模块分解与功能抽象和自顶向下、分而治之的方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。它的重点在于把功能进行分解。但是由于在实际开发过程当中需求会经常发生变化，因此，它不能很好的适应需求

变化的开发过程。结构化程序设计是面向过程的。

面向对象程序设计以需求当中的数据作为中心，来进行设计，具有良好的代码重用性。

封装性：也叫数据隐藏，用户无需知道内部工作流程，只要知道接口和操作就可以的，C++中一般用类来实现封装。

继承性： 一种支持重用的思想，在现有的类型派生出新的子类，例如新型电视机在原有型号的电视机上增加若干中功能而得到，新型电视机是原有电视机的派生，继承了原有电视机的属性，并增加了新的功能。

多态性：指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。

动态联编：指一个计算机程序自身彼此关联的过程，按照联编所进行的阶段不同，可分为两种不同的联编方法：静态联编和动态联编。

8.13.2Python 中的可变对象和不可变对象？

不可变对象，该对象所指向的内存中的值不能被改变。当改变某个变量时候，由于其所指的值不能被改变， 相当于把原来的值复制一份后再改变，这会开辟一个新的地址，变量再指向这个新的地址。

可变对象，该对象所指向的内存中的值可以被改变。变量（准确的说是引用）改变后，实际上是其所指的值直接发生改变，并没有发生复制行为，也没有开辟新的出地址，通俗点说就是原地改变。

Python 中，数值类型（int 和 float）、字符串 str、元组 tuple 都是不可变类型。而列表 list、字典 dict、集合 set 是可变类型。

8.13.3Python 中 is 和==的区别？

is 判断的是 a 对象是否就是 b 对象，是通过 id 来判断的。

==判断的是 a 对象的值是否和 b 对象的值相等，是通过 value 来判断的。

8.13.4Python 的魔法方法？

魔法方法就是可以给你的类增加魔力的特殊方法，如果你的对象实现 （重载）了这些方法中的某一个，那么这个

方法就会在特殊的情况下被 Python 所调用，你可以定义自己想要的行为，而这一切都是自动发生的。 它们经常是两个下划线包围来命名的（比如 init ， lt ），Python 的魔法方法是非常强大的，所以了解其使用方法也变得尤为重要！ init 构造器，当一个实例被创建的时候初始化的方法。但是它并 不是实例化调用的第一个方法。

__new__ 才是实例化对象调用的第一个方法，它只取下 cls 参数，并把 其他参数传给 init 。 new 很少使用，但是也有它适合的场景，尤其 是当类继承自一个像元组或者字符串这样不经常改变的类型的时候。

___call 允许一个类的实例像函数一样被调用 。

____getitem__定义获取容器中指定元素的行为，相当于 self[key] 。

___getattr____定义当用户试图访问一个不存在属性的时候的行为 。

____setattr___定义当一个属性被设置的时候的行为 。

____getattribute___定义当一个属性被访问的时候的行为 。

8.13.5面向对象中怎么实现只读属性?

1、将对象私有化，通过共有方法提供一个读取数据的接口。class person:

```
class person:
    def  init  (self,x):
        self._age = 10;


def age(self):
    return self._age;
    t = person(22)
```

```
    # t. age = 100
    print(t.age())
```

2 、 最 好 的 方 法 class MyCls(object):

```
class MyCls(object):
    __weight = 50

    @property #以访问属性的方式来访问 weight 方法
    def weight(self):
        return self.__weight

if __name__ == ' main ':
    obj = MyCls()
    print(obj.weight)
    obj.weight = 12
```

8.13.6谈谈你对面向对象的理解？

面向对象是相对于面向过程而言的。面向过程语言是一种基于功能分析的、以算法为中心的程序设计方法；而面向对象是一种基于结构分析的、以数据为中心的程序设计思想。在面向对象语言中有一个有很重要东西，叫做类。面向对象有三大特性：封装、继承、多态。

8. 14 正则表达式

8.14.1Python 里 match 与 search 的 区 别 ？

match()函数只检测 RE 是不是在 string 的开始位置匹配， search()会扫描整个 string 查找匹配；

也就是说 match()只有在 0 位置匹配成功的话才有返回，如果不是开始位置匹配成功的话，match()就返回 none。

8.14.2Python 字符串查找和替换？

re.findall(r’目的字符串’，’原有字符串’) #查询

re.findall(r'cast'，'itcast.cn')[0]

re.sub(r‘要替换原字符’，’要替换新字符’，’原始字符串’)

re.sub(r'cast'，'heima'，'itcast.cn')

8.14.3用 Python 匹配 HTML g tag 的时候，<.*> 和 <.*?> 有什么区别？

<.*>是贪婪匹配，会从第一个“<”开始匹配，直到最后一个“>”中间所有的字符都会匹配到，中间可能会包含“<>”。

<.*?>是非贪婪匹配，从第一个“<”开始往后，遇到第一个“>”结束匹配，这中间的字符串都会匹配到，但是不会有“<>”。

8.14.4请写出下列正则关键字的含义？

语法	说明	表达式实例	完整匹配的字符串
字符			
一般字符	匹配自身	abc	abc
.	匹配任意除换行符"\n"外的字符。 在 DOTALL 模式中也能匹配换行符。	a.c	abc
	转义字符,使后一个字符改变原来的意思。		

\	如果字符串中有字符*需要匹配,可以使用*或者字符集[*]	a\.c a\\c	a. c a\c
[...]	字符集(字符类)。对应的位置可以是字符集中任意字符。字符集中的字符可以逐个列出,也可以给出范围, 如[abc]或[a-c]。第一个字符如果是^ 则表示取反,如[^abc]表示不是abc 的其他字符。 所有的特殊字符在字符集中都失去其原有的特殊含义。在字符集中如果要使用]、 -或^,可以在前面加上反斜杠,或把]、-放 在第一个字符,把^ 放在非第一个字符。	a[bcd]e	abe ace ade
预定义字符集(可以写在字符集[...]中)			
\d	数字:[0-9]	a\dc	a1c
\D	非数字:[^\d]	a\Dc	abc
\s	空白字符:[<空格>\t\r\n\f\v]	a\sc	ac
\S	非空白字符:[^\s]	a\Sc	abc

\w	单词字符:[A-Za-z0-9_]	a\wc	abc
\W	非单词字符:[^\W]	a\Wc	ac
数量词(用在字符或(...)之后)			
*	匹配前一个字符 0 或无限次。	abc*	ab abccc
+	匹配前一个字符 1 次或无限次	abc+	abc abccc
?	匹配前一个字符 0 次或 1 次。	abc?	Ab abc
{m}	匹配前一个字符 m 次	ab{2}c	abbc

8.15 异常

8.15.1在 except 中 return 后还会不会执行 finally 中的代码？怎么抛出自定义异常？

会继续处理 finally 中的代码；用 raise 方法可以抛出自定义异常。

8.15.2介绍一下 except 的作用和用法？

except: #捕获所有异常 except: <异常名>: #捕获指定异常

except:<异常名 1, 异常名 2>: 捕获异常 1 或者异常 2

except:<异常名>,<数据>:捕获指定异常及其附加的数据

except:<异常名 1,异常名 2>:<数据>:捕获异常名 1 或者异常名 2,及附加的数据

8.16 模块和包

8.16.1常用的 Python 标准库都有哪些？

os 操作系统，time 时间，random 随机，pymysql 连接数据库，threading 线程，multiprocessing 进程，queue 队列。第三方库：

django 和 flask 也是第三方库，requests，virtualenv，selenium，scrapy，xadmin，celery， re，hashlib，md5。常用的科学计算库（如 Numpy，Scipy，Pandas)。

8.16.2赋值、浅拷贝和深拷贝的区别？

一、赋值在 Python 中，对象的赋值就是简单的对象引用，这点和 C++不同，如下所示：

```
a = [1,2,"hello",['python', 'C++']]
b = a
```

在上述情况下，a 和 b 是一样的，他们指向同一片内存，b 不过是 a 的别名，是引用。

我们可以使用 b is a 去判断，返回 True，表明他们地址相同，内容相同，也可以使用 id()函数来查看两个列表的地址是否相同。

赋值操作(包括对象作为参数、返回值)不会开辟新的内存空间，它只是复制了对象的引用。也就是说除了 b 这个名字之外，没有其他的内存开销。修改了 a，也就影响了 b，同理，修改了 b，也就影响了 a。

二 、 浅 拷 贝 (shallow copy)

浅拷贝会创建新对象，其内容非原对象本身的引用，而是原对象内第一层对象的引用。

浅拷贝有三种形式:切片操作、工厂函数、copy 模块中的 copy 函数。

比如上述的列表 a；

切片操作：b = a[:] 或者 b = [x for x in a]；工厂函数：b = list(a)；

copy 函数：b = copy.copy(a)；

浅拷贝产生的列表 b 不再是列表 a 了，使用 is 判断可以发现他们不是同一个对象，使用 id 查看，他们也不指向同一片内存空间。但是当我们使用 id(x) for x in a 和 id(x) for x in b 来查看 a 和 b 中元素的地址时，可以看到二者包含的元素的地址是相同的。

在这种情况下，列表 a 和 b 是不同的对象，修改列表 b 理论上不会影响到列表 a。

但是要注意的是，浅拷贝之所以称之为浅拷贝，是它仅仅只拷贝了一层，在列表 a 中有一个嵌套的 list，如果我们修改了它，情况就不一样了。

比如：a[3].append('java')。查看列表 b，会发现列表 b 也发生了变化，这是因为，我们修改了嵌套的 list，

修改外层元素，会修改它的引用，让它们指向别的位置，修改嵌套列表中的元素，列表的地址并未发生变化，指向的都是用一个位置。

三、 深拷贝(deep copy)

深拷贝只有一种形式，copy 模块中的 deepcopy()函数。

深拷贝和浅拷贝对应，深拷贝拷贝了对象的所有元素，包括多层嵌套的元素。因此，它的时间和空间开销要高。

同样的对列表 a，如果使用 b = copy.deepcopy(a)，再修改列表 b 将不会影响到列表 a，即使嵌

套的列表具有更深的层次，也不会产生任何影响，因为深拷贝拷贝出来的对象根本就是一个全新的对象，不再与原来的对象有任何的关联。

四、 拷贝的注意点？

对于非容器类型，如数字、字符，以及其他的“原子”类型，没有拷贝一说，产生的都是原对象的引用。如果元组变量值包含原子类型对象，即使采用了深拷贝，也只能得到浅拷贝。

8.16.3 init 和 new 的区别？

init 在对象创建后，对对象进行初始化。

new 是在对象创建之前创建一个对象，并将该对象返回给 init。

8.16.4Python 里面如何生成随机数？

在 Python 中用于生成随机数的模块是 random，在使用前需要 import.

如下例子可以酌情列举：

random.random()：生成一个 0-1 之间的随机浮点数；

random.uniform(a, b)：生成[a,b]之间的浮点数；

random.randint(a, b)：生成[a,b]之间的整数；

random.randrange(a, b, step)：在指定的集合[a,b)中，以 step 为基数随机取一个数；

random.choice(sequence)：从特定序列中随机取一个元素，这里的序列可以是字符串，列表，元组等。

8.16.5输入某年某月某日，判断这一天是这一年的第几天？(可以用 Python 标准库)

```
import datetime



def dayofyear():
    year = input("请输入年份: ")
    month = input("请输入月份: ")
```

```
    day = input("请输入天: ")
    date1 = datetime.date(year=int(year), month=int(month), day=int(day))
    date2 = datetime.date(year=int(year), month=1, day=1)
    return (date1 - date2 + 1).days
```

8.16.6打乱一个排好序的 list 对象 alist？

```
import random
random.shuffle(alist)
```

8.16.7说明一下 os.path 和 sys.path 分别代表什么？

os.path 主要是用于对系统路径文件的操作。sys.path 主要是对 Python 解释器的系统环境参数的操作

（动态的改变 Python 解释器搜索路径）。

8.16.8Python 中的 os 模块常见方法？

- os.remove()删除文件
- os.rename()重命名文件
- os.walk()生成目录树下的所有文件名
- os.chdir()改变目录
- os.mkdir/makedirs 创建目录/多层目录
- os.rmdir/removedirs 删除目录/多层目录
- os.listdir()列出指定目录的文件
- os.getcwd()取得当前工作目录
- os.chmod()改变目录权限
- os.path.basename()去掉目录路径，返回文件名

- os.path.dirname()去掉文件名，返回目录路径
- os.path.join()将分离的各部分组合成一个路径名
- os.path.split()返回（dirname(),basename())元组
- os.path.splitext()(返回 filename,extension)元组
- os.path.getatime\ctime\mtime 分别返回最近访问、创建、修改时间
- os.path.getsize()返回文件大小 os.path.exists()是否存在
- os.path.isabs()是否为绝对路径
- os.path.isdir()是否为目录
- os.path.isfile()是否为文件

8.16.9Python 的 sys 模块常用方法？

- sys.argv 命令行参数 List，第一个元素是程序本身路径
- sys.modules.keys() 返回所有已经导入的模块列表
- sys.exc_info() 获取当前正在处理的异常类,exc_type、exc_value、exc_traceback 当前处理的异常详细信息
- sys.exit(n) 退出程序，正常退出时 exit(0)
- sys.hexversion 获取 Python 解释程序的版本值，16 进制格式如：0x020403F0
- sys.version 获取 Python 解释程序的版本信息
- sys.maxint 最 大 的 Int 值
- sys.maxunicode 最 大 的 Unicode 值
- sys.modules 返回系统导入的模块字段，key 是模块名，value 是模块
- sys.path 返回模块的搜索路径，初始化时使用 PYTHONPATH 环境变量的值
- sys.platform 返回操作系统平台名称
- sys.stdout 标准输出
- sys.stdin 标准输入
- sys.stderr 错误输出

- sys.exc_clear() 用来清除当前线程所出现的当前的或最近的错误信息

- sys.exec_prefix 返回平台独立的 python 文件安装的位置
- sys.byteorder 本地字节规则的指示器，big-endian 平台的值是'big',little-endian 平台的值是'little'
- sys.copyright 记录 python 版权相关的东西
- sys.api_version 解 释 器 的 C 的 API 版 本
- sys.version_info 元组则提供一个更简单的方法来使你的程序具备 Python 版本要求功能

8.16.10 模块和包是什么

在 Python 中，模块是搭建程序的一种方式。每一个 Python 代码文件都是一个模块，并可以引用其他的模块，比如对象和属性。

一个包含许多 Python 代码的文件夹是一个包。一个包可以包含模块和子文件夹。

8.17 Python 特性

8.17.1Python 是强语言类型还是弱语言类型？

Python 是强类型的动态脚本语言。

强类型：不允许不同类型相加。

动态：不使用显示数据类型声明，且确定一个变量的类型是在第一次给它赋值的时候。

脚本语言：一般也是解释型语言，运行代码只需要一个解释器，不需要编译。

8.17.2谈一下什么是解释性语言，什么是编译性语言?

计算机不能直接理解高级语言，只能直接理解机器语言，所以必须要把高级语言翻译成机器语言，计算机才能执行高级语言编写的程序。

解释性语言在运行程序的时候才会进行翻译。

编译型语言写的程序在执行之前，需要一个专门的编译过程，把程序编译成机器语言（可执行文件）。

8.17.3Python 中有日志吗?怎么使用?

有日志。

Python 自带 logging 模块，调用 logging.basicConfig()方法，配置需要的日志等级和相应的参数，Python 解释器会按照配置的参数生成相应的日志。

8.17.4Python 是如何进行类型转换的？

内建函数封装了各种转换函数，可以使用目标类型关键字强制类型转换，进制之间的转换可以用 int(‘str’，base=’n’)将特定进制的字符串转换为十进制，再用相应的进制转换函数将十进制转换 为目标进制。

可以使用内置函数直接转换的有：

list---->tuple tuple(list)

tuple---->list list(tuple)

8.17.5工具安装问题

windows 环 境

Python2 无法安装 mysqlclient。Python3 无法安装 MySQL-python、 flup、functools32、Gooey、Pywin32、 webencodings。 matplotlib 在 python3 环境中安装报错：The following required packages can not be

built:freetype, png。需要手动下载安装源码包安装解决。

scipy 在 Python3 环境中安装报错，numpy.distutils.system_info.NotFoundError，需要自己手

工下载对应的安装包，依赖 numpy,pandas 必须严格根据 python 版本、操作系统、64 位与否。运行 matplotlib 后发现基础包 numpy+mkl 安装失败，需要自己下载，国内暂无下载源

centos 环境下

Python2 无法安装 mysql-python 和 mysqlclient 包，报错：EnvironmentError: mysql_config not found，解决方案是安装 mysql-devel 包解决。使用 matplotlib 报错：no module named _tkinter，安装 Tkinter、tk-devel、tc-devel 解决。 pywin32 也无法在 centos 环境下安装。

8.17.6关于 Python 程序的运行方面，有什么手段能提升性能？

使用多进程，充分利用机器的多核性能

对于性能影响较大的部分代码，可以使用 C 或 C++编写

对于 IO 阻塞造成的性能影响，可以使用 IO 多路复用来解决

尽量使用 Python 的内建函数

尽量使用局部变量

8.17.7Python 中的作用域？

Python 中，一个变量的作用域总是由在代码中被赋值的地方所决定。当 Python 遇到一个变量的话它会按照这的顺序进行搜索：

本地作用域(Local)--->当前作用域被嵌入的本地作用域(Enclosing locals)--->全局/模块作用域

(Global)--->内置作用域(Built-in)。

8.17.8什么是 Python？

- Python 是一种编程语言，它有对象、模块、线程、异常处理和自动内存管理，可以加入其他语言的对比。
- Python 是一种解释型语言，Python 在代码运行之前不需要解释。
- Python 是动态类型语言，在声明变量时，不需要说明变量的类型。
- Python 适合面向对象的编程，因为它支持通过组合与继承的方式定义类。
- 在 Python 语言中，函数是第一类对象。
- Python 代码编写快，但是运行速度比编译型语言通常要慢。
- Python 用途广泛，常被用走"胶水语言"，可帮助其他语言和组件改善运行状况。
- 使用 Python，程序员可以专注于算法和数据结构的设计，而不用处理底层的细节。

8.17.9什么是 Python 的命名空间？

在 Python 中，所有的名字都存在于一个空间中，它们在该空间中存在和被操作——这就是命名空间。它就好像一个盒子，每一个变量名字都对应装着一个对象。当查询变量的时候，会从该盒子里面寻找相应的对象。

8.17.10 你所遵循的代码规范是什么？请举例说明其要求？

PEP8 规范。

1.变量

常量：大写加下划线 USER_CONSTANT。

私有变量 ：小写和一个前导下划线 _private_value。

Python 中不存在私有变量一说，若是遇到需要保护的变量，使用小写和一个前导下划线。但这只是程序员之间的一个约定，用于警告说明这是一个私有变量，外部类不要去访问它。但实际上，外部类还是可以访问到这个变量。

内置变量: 小写，两个前导下划线和两个后置下划线 class 两个前导下划线会导致变量在解释期间被更名。这是为了避免内置变量和其他变量产生冲突。用户定义的变量要严格避免这种风格。以免导致混乱。

2.函数和方法

总体而言应该使用，小写和下划线。但有些比较老的库使用的是混合大小写，即首单词小写，之后每个单词第一个字母大写，其余小写。但现在，小写和下划线已成为规范。

私有方法 ：小写和一个前导下划线

这里和私有变量一样，并不是真正的私有访问权限。同时也应该注意一般函数不要使用两个前导下划线 (当遇到两个前导下划线时，Python 的名称改编特性将发挥作用)。

特殊方法 ：小写和两个前导下划线，两个后置下划线这种风格只应用于特殊函数，比如操作符重载等。

函数参数 ：小写和下划线，缺省值等号两边无空格

3.类

类总是使用驼峰格式命名，即所有单词首字母大写其余字母小写。类名应该简明，精确，并足以从中理解类所完成的工作。常见的一个方法是使用表示其类型或者特性的后缀，例如:

SQLEngine，MimeTypes 对于基类而言，可以使用一个 Base 或者 Abstract 前缀 BaseCookie，AbstractGroup

4.模块和包

除特殊模块 init 之外，模块名称都使用不带下划线的小写字母。

若是它们实现一个协议，那么通常使用 lib 为后缀，例如:

```
import smtplib

import os

import sys
```

5.关于参数

不要用断言来实现静态类型检测。断言可以用于检查参数，但不应仅仅是进行静态类型检测。

Python 是动态类型语言，静态类型检测违背了其设计思想。断言应该用于避免函数不被毫无意义的调用。

不要滥用 *args 和 **kwargs。*args 和 **kwargs 参数可能会破坏函数的健壮性。它们使签名变得模糊，而且代码常常开始在不应该的地方构建小的参数解析器。

6.其他

使用 has 或 is 前缀命名布尔元素 is_connect = True

has_member = False

用复数形式命名序列

members = ['user_1', 'user_2']

用显式名称命名字典

person_address = {'user_1':'10 road WD', 'user_2' : '20 street huafu'}

避免通用名称

诸如 list, dict, sequence 或者 element 这样的名称应该避免。避免现有名称诸如 os, sys 这种系统已经存在的名称应该避免。

7.一些数字

一行列数 ：PEP 8 规定为 79 列。根据自己的情况，比如不要超过满屏时编辑器的显示列数。

一个函数 ： 不要超过 30 行代码, 即可显示在一个屏幕类，可以不使用垂直游标即可看到整个函数。一个类 ： 不要超过 200 行代码，不要有超过 10 个方法。一个模块 不要超过 500 行。

8.验证脚本可以安装一个 pep8 脚本用于验证你的代码风格是否符合 PEP8。

8.18 Python2 与 Python3 的 区 别

核心类差异

1.Python3 对 Unicode 字符的原生支持。

Python2 中使用 ASCII 码作为默认编码方式导致 string 有两种类型 str 和 unicode，Python3 只支持 unicode 的 string。Python2 和 Python3 字节和字符对应关系为：

2.Python3 采用的是绝对路径的方式进行 import。

Python2 中相对路径的 import 会导致标准库导入变得困难（想象一下，同一目录下有 file.py，如何同时导入这个文件和标准库 file）。Python3 中这一点将被修改，如果还需要导入同一目录的文件必须使用绝对路径，否则只能使用相关导入的方式来进行导入。

3.Python2 中存在老式类和新式类的区别，Python3 统一采用新式类。新式类声明要求继承 object，必须用新式类应用多重继承。

4.缩进严格程度

Python3 使用更加严格的缩进。Python2 的缩进机制中，1 个 tab 和 8 个 space 是等价的，所以在缩进中可以同时允许 tab 和 space 在代码中共存。这种等价机制会导致部分 IDE 使用存在问题。Python3 中 1 个 tab 只能找另外一个 tab 替代，因此 tab 和 space 共存会导致报错：TabError: inconsistent use of tabs and spaces in indentation.

废弃类差异

1.print 语句被 Python3 废弃，统一使用 print 函数

2.exec 语句被 python3 废弃，统一使用 exec 函数

3.execfile 语句被 Python3 废弃，推荐使用 exec(open("./filename").read())

4.不相等操作符"<>"被 Python3 废弃，统一使用"!="

5.long 整数类型被 Python3 废弃，统一使用 int

6.xrange 函数被 Python3 废弃，统一使用 range，Python3 中 range 的机制也进行修改并提高了大数据集生成效率

7.Python3 中这些方法再不再返回 list 对象：dictionary 关联的 keys()、values()、items()，zip()，map()，filter()，但是可以通过 list 强行转换：

8. mydict={"a":1,"b":2,"c":3}

9.mydict.keys() #<built-in method keys of dict object at 0x000000000040B4C8>

10.list(mydict.keys()) #['a', 'c', 'b']

11.迭代器 iterator 的 next()函数被 Python3 废弃，统一使用 next(iterator)

12.raw_input 函数被 Python3 废弃，统一使用 input 函数

13.字典变量的 has_key 函数被 Python 废弃，统一使用 in 关键词

14.file 函数被 Python3 废弃，统一使用 open 来处理文件，可以通过 io.IOBase 检查文件类型

15.apply 函数被 Python3 废弃

16.异常 StandardError 被 Python3 废弃，统一使用 Exception

修改类差异

1.浮点数除法操作符“/”和“//”的区别

“ / ”：

Python2：若为两个整形数进行运算，结果为整形，但若两个数中有一个为浮点数，则结果为浮点数；

Python3:为真除法，运算结果不再根据参加运算的数的类型。

“//”：

Python2：返回小于除法运算结果的最大整数；从类型上讲，与"/"运算符返回类型逻辑一致。

Python3：和 Python2 运算结果一样。

2.异常抛出和捕捉机制区别

Python2

```
raise IOError, "file error" #抛出异常
except NameError, err:  #捕捉异常
```

Python3

```
raise IOError("file error") #抛出异常
except NameError as err: #捕捉异常
```

3.for 循环中变量值区别

Python2

for 循环会修改外部相同名称变量的值 i = 1

```
print （'comprehension: ', [i for i in range(5)]）
print （'after: i =', i） #i=4
```

Python3

```
for 循环不会修改外部相同名称变量的值
i = 1
print （'comprehension: ', [i for i in range(5)]）
print （'after: i =', i ） #i=1
```

4.round 函数返回值区别

Python2，round 函数返回 float 类型值

isinstance(round(15.5),int) #True

Python3，round 函数返回 int 类型值

isinstance(round(15.5),float) #True

5.比较操作符区别

Python2 中任意两个对象都可以比较

11 < 'test' #True

Python3 中只有同一数据类型的对象可以比较

11 < 'test' # TypeError: unorderable types: int() < str()

第三方工具包差异

我们在 pip 官方下载源 pypi 搜索 Python2.7 和 Python3.5 的第三方工具包数可以发现，Python2.7 版本对应的第三方工具类目数量是 28523,Python3.5 版本的数量是 12457，这两个版本在第三方工具包支持数量差距相当大。

我们从数据分析的应用角度列举了常见实用的第三方工具包（如下表），并分析这些工具包在 Python2.7 和 Python3.5 的支持情况：

	工具名	用途
数据收集	scrapy	网页采集，爬虫
数据收集	scrapy-redis	分布式爬虫
数据收集	selenium	web 测试，仿真浏览器
数据处理	beautifulsoup	网页解释库，提供 lxml 的支持
数据处理	lxml	xml 解释库
数据处理	xlrd	excel 文件读取

数据处理	xlwt	excel 文件写入
数据处理	xlutils	excel 文件简单格式修改
数据处理	pywin32	excel 文件的读取写入及复杂格式定制
数据处理	Python-docx	Word 文件的读取写入
数据分析	numpy	基于矩阵的数学计算库
数据分析	pandas	基于表格的统计分析库

分类	工具名	用途
数据分析	scipy	科学计算库，支持高阶抽象和复杂模型
数据分析	statsmodels	统计建模和计量经济学工具包
数据分析	scikit-learn	机器学习工具库
数据分析	gensim	自然语言处理工具库
数据分析	jieba	中文分词工具库
数据存储	MySQL-python	mysql 的读写接口库
数据存储	mysqlclient	mysql 的读写接口库
数据存储	SQLAlchemy	数据库的 ORM 封装
数据存储	pymssql	sql server 读写接口库
数据存储	redis	redis 的读写接口
数据存储	PyMongo	mongodb 的读写接口
数据呈现	matplotlib	流行的数据可视化库
数据呈现	seaborn	美观的数据可是湖库，基于 matplotlib
工具辅助	jupyter	基于 web 的 python IDE，常用于数据分析
工具辅助	chardet	字符检查工具

工具辅助	chardet	字符检查工具
工具辅助	ConfigParser	配置文件读写支持
工具辅助	requests	HTTP 库，用于网络访问

9 Selenium 相关

9.1 Selenium 基础

9.1.1 什么是 Selenium?

Selenium 就是一套专门用于自动化 Web 浏览器的工具。而已！你用这个东西来做什么完全取决于你。主要是用于自动化 Web 应用程序进行测试，但肯定不仅限于此。无聊的基于 Web 的管理任务也可以（也应该！）也是自动化的。

Selenium 有一些最大的浏览器供应商的支持，他们已经采取（或正在采取）步骤使 Selenium 成为其浏览器的本地部分。 它也是无数其他浏览器自动化工具，API 和框架的核心技术。

最新的 Selenium 版本已经是 3.0（2016 年 10 月 13 日正式 release），但是因为是新技术，Selenium3.0 的使用范围还不太广泛。变动的范围也不是很大，主要是更倾向于 Webdriver，而更多的摒弃了 RC。

9.1.2 什么是 Selenium Webdriver

Webdriver (Selenium2)是一种用于 Web 应用程序的自动测试工具，它提供了一套友好的 API，与 Selenium 1(Selenium-RC)相比，Selenium 2 的 API 更容易理解和使用，其可读性和可维护性也大大提高。Webdriver 完全就是一套类库，不依赖于任何测试框架，除了必要的浏览器驱动，不需要启动其他进程或安装其他程序，也不必像 Selenium 1 那样需要先启动服务。

另外，二者所采用的技术方案也不同。Selenium 1 是在浏览器中运行 JavaScript 来进行测试，而 Selenium 2 则是通过原生浏览器支持或者浏览器扩展直接控制浏览器。

Selenium 2 针对各个浏览器而开发的，它取代了嵌入到被测 Web 应用中的 JavaScript。与浏览器的紧密集成，支持创建更高级的测试，避免了 JavaScript 安全模型的限制。除了来自浏览器厂商的支持，Selenium 2 还利用操作系统级的调用模拟用户输入。

9.1.3 S 什么是 elenium IDE？

Selenium IDE 是 Selenium 脚本的集成开发环境。 它被实现为 Firefox 扩展(插件)，并允许您记录，编辑和调试测试。

9.1.4 2.常用自动化测试工具机器运行原理，写出一段元素查找的代码？

webdriver 原理：

- 每个 Selenium 命令，这里指的是所谓的基础操作，例如，点击、输入等，都会创建一条 HTTP 请求， 发送给 Browser WebDriver
- Browser WebDriver 使用一个 HTTP Server 监听和接收 HTTP 请求
- HTTP Server 根据协议规则定义这些 Selenium 命令对应的浏览器具体操作
- 浏览器执行这些操作
- 浏览器将执行状态返回给 HTTP Server
- HTTP Server 再将这些状态信息返回给自动化脚本

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome()
driver.get("http://www.python.org")
assert "Python" in driver.title
elem = driver.find_element_by_name("q")
elem.send_keys("pycon")
elem.send_keys(Keys.RETURN)
assert "No result found." not in driver.page_source
driver.close()
```

9.1.5 如何开展自动化测试框架的构建？

我们公司的自动化测试框架主要是有页面库，数据驱动，测试脚本，测试报告，持续集成这几个部分组成的。

页面对象库对自动化包括工具（selenium，appium）API 的二次封装，还有使用二次封装后的自动化工具类实现的页面元素封装（Page Object）然后会给封装好的页面设置一个统一入口类。这些之中会有一

个页面元素文件专门存放元素的定位方法。

数据驱动部分主要是测试脚本中使用的数据文件（excel,yaml,txt）以及读取方法类，如果数据涉及到数据库，也会把对应的数据读取方法封装到这个部分。

测试脚本主要是通过 pytest 测试框架进行编写的，选择其的原因主要有其支持 assert 语句断言，适合复杂的功能测试，执行过程中可以自定义用例执行顺序和跳过以及预期，支持重复执行，还可兼容 unittest 编写的测试用例，最重要的是支持参数化和方便持续集成工具集成。

测试报告主要是通过 pytest 自动生成的 Allure 报告，其可读性可生动的数据表图比 pytest 报告更能反应测试结果，也可以集成与 Jenkins 中。

持续集成方面主要是通过 Jenkins 进行实现的，目的在于测试脚本的无人值守执行以及自动生成测试报告，方便测试人员能够省出时间进行更多的功能测试和探索性测试。（通过设置几个 git,gitlab，mailer,allure， 等功能插件，配置 Allure 报告，默认邮件发送设置。用例脚本主要存放在 gitlab 用例库中，设置好轮询策略之后，配置报告发送的目标邮箱，就可以实现持续集成实践中的测试环节）

9.1.6 4.如何设计自动化测试用例:

编写测试脚本之前要编写测试用例，而且测试用例不能直接使用手工测试的用例。

自动化的测试用例是一个完整的场景。用户登录系统到用户退出。

用例之验证一个功能点。不用试图登陆后验证所有的功能在退出

测试用例尽量只做正向的逻辑验证。

用例之间不要产生关联，相互独立，也要高内聚，低耦合

测试用例关注的是功能逻辑的实现，字段无关

测试用例的上下文必须有一定的顺序性，前置条件清晰

检查点的设置要侧重，全面，灵活

测试用例对数据的操作要进行还原

测试用例必须是可回归的

用例选择遵循成本始终，构建场景，目的冒烟回归，繁琐功能，主体流程

用例转型遵循前置配置，抛异常，步骤验证，高内聚，关门归原

9.1.7 webdriver 如何开启和退出一个浏览器？

开启：dr = webdriver.浏览器类型() 关闭：dr.quit()

9.1.8 什么是自动化测试框架?

测试自动化框架是设置特定产品的自动化规则的集成系统。该系统集成了功能库，测试数据源，对象详细信息和各种可重复使用的模块。这些组件用作需要组装以代表业务流程的小型构建块。该框架为测试自动化提供了基础，并简化了自动化工作。

也是为自动化软件测试提供支持的假设框架，概念和工具的主要优点是维护成本低。如果任何测试用例发生变化，那么只需要更新测试用例文件，驱动程序脚本和启动脚本将保持不变。理想情况下，如果应用程序发生更改，则无需更新脚本。

选择正确的框架/脚本技术有助于降低成本。与测试脚本相关的成本是由于开发和维护工作。测试自动化期间使用的脚本的方法对成本有影响。

通常使用各种框架/脚本技术：

线性（程序代码，可能由使用记录和播放的工具生成）

结构化（使用控制结构 - 通常是“if-else”，“switch”，“for”，“while”条件/语句）

数据驱动（数据存储在数据库，电子表格或其他机制中，比如 xml）

关键字驱动

行为驱动

混合（使用上述两种或更多种模式）

自动化测试框架主要负责：

定义表达期望的格式

创建一个挂钩或驱动被测应用程序的机制

执行测试

报告结果

9.1.9 Selenium 是什么，流行的版本有哪些？

Selenium 是基于 Web 的最流行的 UI 自动化测试工具。它提供了一组支持多种平台的公开 API（例如 Linux，Windows，Mac OS X 等）。此外，像 Google Chrome，Mozilla Firefox，Internet Explorer 和 Safari 等所有现代浏览器都可以用来运行 Selenium 测试。它也涵盖了 Android 平台，其中 Appium 是实现 Selenium Webdriver 界面的工具，用于移动自动化。

值得注意的是，除了许多后来的小型版本之外，硒还有三个主要版本：

Selenium 1.0 或 Selenium RC，于 2004 年初发布，提供了一个使用服务器与浏览器交换命令和响应的 API 集。

Selenium 2.0 或 Selenium Webdriver，在 2011 年中推出，并在 Selenium 功能中引入了一系列重大改进。这些新的 API 完全取代了服务器组件，并与目标浏览器本地交互。

Selenium 3.0，这个版本是在 2016 年末发布的大版本。它带来的主要变化是引入 Webdriver API 的 W3C 规范，用于浏览器自动化。也就是说，每个主要的浏览器都会有自己的 Webdriver API 来实现功能。

9.1.10你如何从命令行启动 Selenium RC？

```
// 简单的启动 Selenium RC 的方法是
java -jar selenium-server.jar
// 在浏览器中运行一套 Selenese 脚本
java -jar selenium-server.jar -htmlSuite
```

9.1.11在我的机器端口 4444 不是免费的。我怎样才能使用另一个端口？

```
//你可以在运行 selenium 服务器时指定端口为 -
Java -jar selenium-server.jar -port 5555
```

9.1.12什么是 Selenium Server，它与 Selenium Hub 有什么不同？

Selenium Server 是使用单个服务器作为测试节点的一个独立的应用程序。Selenium hub 代理一个或多个 Selenium 的节点实例。一个 hub 和多个 node 被称为 Selenium grid。运行 SeleniumServer 与在同一主机上用一个 hub 和单个节点创建 de Selenium grid 类似。

9.1.13你如何从 Selenium 连接到数据库？

Selenium 是一个 Web UI 自动化工具。它不提供任何 API 来建立数据库连接。这取决于你使用 Selenium 进行自动化的编程语言。在下面的例子中，我们假设正在使用 Java。

一个 Connection 对象表示与数据库的连接。当我们使用连接方法连接到一个数据库时，我们创建了一个连接对象，它代表了与数据库的连接。单个数据库可能有一个连接或多个连接，还可能有多个连接到不同的数据库上。

我们可以使用 Connection 对象来做以下事情：

创建用于执行 SQL 语句的 Statement，PreparedStatement 和 CallableStatement 对象。

可以帮助我们提交或回滚一个 JDBC 事务。

如果你想知道连接到的数据库或数据源信息，Connection 对象通过使用 DatabaseMetaData 就可以收集有关数据库或数据源的信息。

可以帮助我们关闭数据源。Connection.isClosed() 方法只有在调用了 Connection.close()时才返回 true 。此方法用于关闭所有连接。

首先我们需要通过使用 DriverManager.getConnection()方法，建立与数据库的连接。这个方法接受一个包含 URL 的字符串。DriverManager 类尝试查找可以连接到由字符串 URL 表示的数据库的驱动程序。每当调用 getConnection()方法时，DriverManager 类都会检查可以连接到 URL 中指定的数据库的所有已注册的 Driver 类的列表。

语法：

```
String url = "jdbc: odbc: makeConnection";
Connection con = DriverManager.getConnection(url, "userID", "password");
```

9.1.14你如何验证多个页面上存在的一个对象？

可以使用下面的 Selenium 命令来检查：

```
assertTrue(selenium.isElementPresent(locator));
```

9.1.15XPath 中使用单斜杠和双斜杠有什么区别？

如果 XPath 是从文档节点开始，它将允许创建“绝对”路径表达式。

例如 “/ html / body / p”匹配所有的段落元素。

如果 XPath 在文档中的任意位置开始进行选择匹配，那么它将允许创建“相对”路径表达式。

例如 “// p”匹配所有的段落元素。

9.1.16如何编写 Selenium IDE / RC 的用户扩展？

用户扩展（UX）存储在 Selenium IDE 或 Selenium RC 用来激活扩展的单独文件中。它包含用 JavaScript 编写的函数定义。

因为 Selenium 的核心是用 JavaScript 开发的，所以要符合原语言的标准规则来创建扩展。要创建一个扩展，我们必须用下面的设计格式来编写函数。

```
// 样例
Selenium.prototype.doFunctionName = function(){
}
```

函数名称前面的“do”告诉 Selenium 这个函数可以被调用为一个步骤命令，而不是作为内部函数或私有函数被调用。

9.1.17如何在页面加载成功后验证元素的存在？

它可以通过下面的代码行来实现。

只需一点时间（以秒为单位）来检查元素，如下所示：

```
public void waitForElementPresent(String element, int timeout) throws Exception {
    for (int second = 0;; second++) {
        if (second >= timeout)
            fail("Timeout. Unable to find the Specified element" + element);
            try {
                if (selenium.isElementPresent(element))
                break;
            } catch (Exception e) {
        }
        Thread.sleep(1000);
```

```
    }
}
```

9.1.18你对 Selenium Grid 有什么了解？它提供了什么功能？

Selenium Grid 是一款利用现有计算基础架构大幅加速 Web 应用程序功能测试的工具。允许测试者轻松地在多台机器上并行运行多个测试，并且可以在异构环境中运行。

基于优秀的 Selenium Web 测试工具，Selenium Grid 允许测试者并行运行多个 Selenium Remote Control 实例。更好的是，它集成显示所有 Selenium 远程控制，所以不必担心实际的基础设施。Selenium Grid 将运行 Selenium 测试套件所需的时间，缩短到 Selenium 实例的单个实例运行时间的一小点。

9.1.19如何从你的 Java Class 启动 Selenium 服务器？

```
try {
    seleniumServer = new SeleniumServer();
    seleniumServer.start();
} catch (Exception e) {
    e.printStackTrace();
}
```

9.1.20Selenium 中有哪些验证点？

Selenium 主要有三种验证点：

检查页面标题

检查某些文字

检查某些元素（文本框，下拉菜单，表等）

-

9.1.21什么是 XPath？什么时候应该在 Selenium 中使用 XPath？

XPath 是一种在 HTML / XML 文档中定位的方法，可用于识别网页中的元素。如果没有与页面上的元素相关联的名称/ ID，或者名称/ ID 的一部分是常量，则必须使用 XPath。

绝对路径用 -/ 单斜杠

相对路径用 -// 双斜杠

ID，类，名称也可以用于 XPath ：

//input[@name='q']

//input[@id='lst-ib']

//input[@class=' lst']

如果 id / name / class 的一部分是常量 ：

//input[contains(@id,'lst-ib')

9.1.22什么是 Selenium 的 CSS 定位器策略？用例子来解释。

CSS 位置策略可以与 Selenium 一起使用来定位元素，它使用 CSS 定位方法，其中 -

绝对路径用 - （空格符号）

相对路径用 ->表示

ID，类，名称也可以用于 XPath：

css=input[name='q']

css=input[id='lst-ib'] or input#lst-ib

css=input[class='lst'] or input.lst

如果 id / name / class 只有一部分是常量：

css=input[id*='lst-ib')]

使用内部文本的元素位置策略：

css = a:contains('log out')

9.1.23当有很多定位器时，如 ID、名称、XPath、CSS 定位器，我应该使用哪一个？

如果有唯一的名称或标识符可用，那么应该使用它们来代替 XPath 和 CSS 定位器。如果没有，那么 CSS 定位器应该被优先考虑，因为在大多数现代浏览器中，它们的评估速度比 XPath 更快。

9.1.24在 Selenium 中处理多个弹出窗口的机制是什么？

可以使用命令 getWindowHandles()来处理多个弹出窗口。

然后将所有窗口名称存储到 Set 变量中并将其转换为数组。

接下来，通过使用数组索引，导航到特定的窗口。

driver.switchTo().window(ArrayIndex);

9.1.25你如何处理使用 Selenium 的 Ajax 控件？

来看一个例子。假如一个文本框是一个 Ajax 控件，当我们输入一些文本时，它会显示自动建议的值。处理这样的控件，需要在文本框中输入值之后，捕获字符串中的所有建议值；然后，分割字符串，取值就好了。

9.1.26Selenium Webdriver 优于 Selenium RC 的优点是什么？

Selenium RC 的架构相当复杂，WebDriver 的架构比 Selenium RC 简单些。

Selenium RC 比较慢，因为它使用了另外一个名为 Selenium Core 的 JavaScript 程序。相反，WebDriver 比 Selenium RC 更快，因为它直接与浏览器对话，并使用浏览器自己的引擎来进行控制。

像其他 JavaScript 代码一样，Selenium Core 可以访问禁用的元素。Webdriver 以更现实的方式与页面元素进行交互。

Selenium RC 的 API 集已经有所改进，但是仍有经常让人困惑的冗余部分。WebDriver API 更简单，不包含任何冗余或混淆的命令。

Selenium RC 无法支持无头 HtmlUnit 浏览器。它需要一个真正的、可见的浏览器来操作。Web Driver 可以支持无头 HtmlUnit 浏览器。

Selenium RC 内置了测试结果生成器，并自动生成测试结果的 HTML 文件。Web 驱动程序没有自动生成测试结果文件的内置命令。

-

9.1.27“GET”和“NAVIGATE”方法的主要区别是什么？

Get 方法能获得一个页面进行加载、或获取页面源代码、或获取文本，就这三。而 Navigate 将通过刷新，回退，前进的方式导航。

例如 -如果我们想要前进，并做一些功能，并返回到主页。

这可以通过调用< navigate()>方法来实现。

driver.get() 方法会等到整个页面被加载后才可以，而 driver.navigate()只是重定向到该网页，并不会等待。

9.1.28隐式等待与显式等待有什么不同？

隐式等待是设置的全局等待，分为 1、页面加载超时等待 ；2、页面元素加载超时；3、异步脚本超时。如果是页面元素超时，设置等待时间，是对页面中的所有元素设置加载时间。隐式等待是其实可以理解成在规定的时间范围内，浏览器在不停的刷新页面，直到找到相关元素或者时间结束。

显式等待只是用于特定搜索的一个计时器。它的可扩展性更强，你可以设置它来等待任何条件。通常情况下，可以使用一些预先构建的条件来等待元素变得可点击，可见，不可见等，或者只是编写适合需求的条件。

9.1.29你将如何处理 Selenium WebDriver 中的警报/弹出窗口？

有两种类型的警报通常被引用。

基于 Windows 的警报弹出窗口

基于 Web 的警报弹出窗口

基于 Web 的警报弹出窗口。

WebDriver 为用户提供了一种使用 Alert 界面处理这些弹出窗口的非常有效的方法。

void dismiss() - 一旦出现弹出窗口，dismiss()方法就会点击“Cancel”按钮。

void accept() - 只要弹出窗口出现，accept()方法就会点击“Ok”按钮。

String getText() - getText()方法返回警告框中显示的文本。

void sendKeys(String stringToSend) - sendKeys()方法将指定的字符串模式输入到警告框中。

基于 Windows 的警报弹出窗口。

处理基于 windows 的弹出窗口总是有点棘手，因为我们知道 Selenium 是一个自动化测试工具，它只支持 Web 应用程序测试，也就是说，它不支持基于 Windows 的应用程序，窗口警报就是其中之一。

Robot class 是基于 Java 的实用程序，它模拟键盘和鼠标操作，并可以有效地用于处理基于 windows 的弹出与键盘事件的帮助。

KeyPress 和 KkeyRelease 方法可以分别模拟用户按下和释放键盘上某个键的操作。

9.1.30如何解决 IE 中的 SSL 认证问题？

// 打开浏览器后添加下面的命令

driver.navigate().to(“javascript:document.getElementById(‘overridelink’).click()”);

9.1.31Selenium WebDriver 中的可用定位器是什么？

ID,

Name,名称

CSS,

XPath,

Class name,

TagName,

LinkText, 链接文本

Partial Link Text.部分链接文本

9.1.32如何处理 WebDriver 中的 AJAX 控件？

AJAX 代表异步 JavaScript 和 XML。它不依赖于创建有效的 XML 所需的打开和关闭标签的额外开销。大部分时间 WebDriver 自动处理 Ajax 控件和调用。如果不能处理的话，可以按照下面的方式来处理。

//Waiting for Ajax Control

WebElement AjaxElement = (new WebDriverWait(driver,

10)).until(ExpectedConditions.presenceOfElementLocated(By.("")));

9.1.33大致分类和比较 TDD/BDD 和 DDD 框架？

你可能听说过所有的这些缩写词。在这里会简要地解释它们，以及它们在系统测试生命周期中如

何发挥作用的。

TDD - 测试驱动开发。

也被称为测试驱动设计，是一个软件开发的方法，在源代码上重复进行单元测试。写测试、看它失败、然后重构。这个概念是，先编写测试，然后来检查我们写的代码是否正常工作。每次测试后，重构完成，然后再次执行相同或类似的测试。该过程需要重复多次，直到每个单元在功能上按预期工作。TDD 是由 XP 引入的。

BDD - 行为驱动开发。

行为驱动的开发将 TDD 的一般技术和原理与领域驱动设计的思想相结合。其目的是帮助人们设计系统（即开发人员）确定合适的测试来编写测试 - 即反映利益相关者所期望行为的测试。

DDD 域驱动的开发。

DDD 将业务领域概念映射到软件工件中。DDD 框架提供以下好处：

帮助团队在业务和 IT 利益相关者之间建立一个共同的模型

该模型是模块化的，可扩展的，易于维护，该设计反映了一种商业模式。

它提高了业务领域对象的可重用性和可测试性。

9.1.34什么是数据驱动框架？它与关键字驱动框架有什么不同？

数据驱动框架

在这个框架中，测试用例逻辑驻留在测试脚本中。测试数据被分离并保存在测试脚本之外。测试数据是从外部文件（Excel 文件）中读取的，并被加载到测试脚本中的变量中。变量用于输入值和验证值。

关键字驱动

关键字/表驱动框架需要开发数据表和关键字。它们独立于执行它们的测试自动化工具。可以使用或不使用应用程序来设计测试。在关键字驱动的测试中，被测试的应用程序的功能记录在一个表格中，以及每个测试的分步说明。

9.1.35解释使用 TestNG 而不是 JUnit 框架的好处？

TestNG 相较于 Junit 的优势：

在 JUnit 中，我们必须声明@BeforeClass 和@AfterClass，这是 JUnit 中的一个约束，而在 TestNG 中

没有像这样的约束。

TestNG 提供了更多的 setUp / tearDown 级别。1.@ Before/AfterSuite 2.@Before/AfterTest 3.@Before/AfterGroup

TestNG 中不需要扩展任何类。

TestNG 中没有方法名称约束，就像 JUnit 一样。

在 TestNG 中，我们可以告诉测试一个方法依赖于另一个方法，而在 JUnit 中这是不可能的。

测试用例的分组在 TestNG 中可用，而 JUnit 中则不可用。执行可以基于组完成。例如，如果你已经定义了许多案例，并通过将 2 个组分别定义为“离职“与”回归”隔离。如果你只是想执行“理智”的情况，那就告诉 TestNG 执行“理智”。TestNG 将自动执行属于“离职”组的案例。

另外，TestNG 支持并行测试用例执行。

9.1.36与@Test 注释相关的 TestNG 参数的目的是什么？

在 TestNG 中，参数是修改注释功能的关键字。

9.1.37可以使用 TestNG 运行一组测试用例吗？

是的，TestNG 框架支持在测试组的帮助下执行多个测试用例。

它提供了以下选项来运行特定组中的测试用例。

如果想基于回归测试或冒烟测试等其中一个组来执行测试用例，那么：

@Test(groups = {“regression-tests”, “smoke-tests”})

9.1.38WebDriver 哪个实现是最快的，为什么？

WebDriver 的最快的实现是 HTMLUnitDriver。

原因是 HTMLUnitDriver 不会在浏览器中执行测试。相反，它使用简单的 HTTP 请求 - 响应机制来运行测试用例。

这种方法比需要启动浏览器来测试执行的方式要快得多。

9.1.39是否可以在 Selenium 2.0 中使用 Selenium RC API？

是的，可以用 Selenium 2.0 来模拟 Selenium 1.0 API（即 RC）。但并不是所有的 Selenium 1.0 方法

都支持。

为了达到这个目的，需要从 WebDriver 获取 Selenium 实例并使用 Selenium 方法。

在 Selenium 2.0 中模拟 Selenium 1.0 时，方法执行速度也可能会变慢。

9.1.40可以在 Java，Dot Net 或 Ruby 中使用 Selenium Grid 吗？

使用 Java，可以利用 TestNG 的并行测试功能来驱动 Selenium Grid 测试。

使用.Net，可以使用“Gallio”并行执行测试。

使用 Ruby，可以使用“DeepTest”来分发测试。

10 性能测试

10.1 性能测试基础

10.1.1性能测试有哪些分类

1.负载测试

在这里，负载测试指的是最常见的验证一般性能需求而进行的性能测试，在上面我们提到了用户最常见的性能需求就是“既要马儿跑，又要马儿少吃草”。因此负载测试主要是考察软件系统在既定负载下的性能表现。我们对负载测试可以有如下理解：

（1）负载测试是站在用户的角度去观察在一定条件下软件系统的性能表现。

（2）负载测试的预期结果是用户的性能需求得到满足。此指标一般体现为响应时间、交易容量、并发容量、资源使用率等。

2.压力测试

压力测试是为了考察系统在**条件下的表现，**条件可以是超负荷的交易量和并发用户数。注意，这个** 条件并不一定是用户的性能需求，可能要远远高于用户的性能需求。可以这样理解，压力测试和负载测试不同的是，压力测试的预期结果就是系统出现问题，而我们要考察的是系统处理问题的方式。比如说，我们期待一个系统在面临压力的情况下能够保持稳定，处理速度可以变慢，但不能系统崩溃。因此，压力测试是能让我们识别系统的弱点和在极限负载下程序将如何运行。

例子：负载测试关心的是用户规则和需求，压力测试关心的是软件系统本身。

3．并发测试

验证系统的并发处理能力。一般是和服务器端建立大量的并发连接，通过客户端的响应时间和服务器端的性能监测情况来判断系统是否达到了既定的并发能力指标。负载测试往往就会使用并发来创造负载，之所以把并发测试单独提出来，是因为并发测试往往涉及服务器的并发容量，以及多进程/多线程

协调同步可能带来的问题。这是要特别注意，必须测试的。

4．基准测试

当软件系统中增加一个新的模块的时候，需要做基准测试，以判断新模块对整个软件系统的性能影响。按照基准测试的方法，需要打开/关闭新模块至少各做一次测试。关闭模块之前的系统各个性能指标记下来作为基准（Benchmark），然后与打开模块状态下的系统性能指标作比较，以判断模块对系统性能的影响。

5.稳定性测试

“路遥知马力”，在这里我们要说的是和性能测试有关的稳定性测试，即测试系统在一定负载下运行长时间后是否会发生问题。软件系统的有些问题是不能一下子就暴露出来的，或者说是需要时间积累才能达到能够度量的程度。为什么会需要这样的测试呢？因为有些软件的问题只有在运行一天或一个星期甚至更长的时间才会暴露。这种问题一般是程序占用资源却不能及时释放而引起的。比如，内存泄漏问题就是经过一段时间积累才会慢慢变得显著，在运行初期却很难检测出来；还有客户端和服务器在负载运行一段时间后，建立了大量的连接通路，却不能有效地复用或及时释放。

6.可恢复测试

测试系统能否快速地从错误状态中恢复到正常状态。比如，在一个配有负载均衡的系统中，主机承受了压力无法正常工作后，备份机是否能够快速地接管负载。可恢复测试通常结合压力测试一起来做。

10.1.2你认为性能测试的目的是什么？做好性能测试的工作的关键是什么？

性能测试工作的目的是检查系统是否满足在需求说明书中规定的性能，性能测试常常需要和强度测试结合起来，并常常要求同时进行软件和硬件的检测。

性能测试主要的关注对象是响应时间，吞吐量，占用内存大小（辅助存储区），处理精度等。

10.1.3服务端性能分析都从哪些角度来进行？

从维度上划分，性能指标主要分为两大类，分别是业务性能指标和系统资源性能指标。业务性能指标可以直观地反映被测系统的实际性能状况，常用的指标项有：

1.并发用户数

2.事务吞吐率（TPS/RPS）

3.事务平均响应时间

4.事务成功率

系统资源性能指标，主要是反映整个系统环境的硬件资源使用情况，常用的指标包括：

1.服务器：CPU 利用率、处理器队列长度、内存利用率、内存交换页面数、磁盘 IO 状态、网卡带宽使用情况等；

2.数据库：数据库连接数、数据库读写响应时长、数据库读写吞吐量等；

3.网络：网络吞吐量、网络带宽、网络缓冲池大小；

4.缓存（Redis）：静态资源缓存命中率、动态数据缓存命中率、缓存吞吐量等；

5.测试设备（压力发生器）：CPU 利用率、处理器队列长度、内存利用率、内存交换页面数、磁盘 IO 状态、网卡带宽使用情况等。

如何理解压力测试，负载测试以及性能测试？

10.1.4如何理解压力测试，负载测试以及性能测试？

性能测试（Performance Test）：通常收集所有和测试有关的所有性能，被不同人在不同场合下进行使用。

压力测试 stress test：是在一定的『负荷条件』下，长时间连续运行系统给系统性能造成的影响。负载测试 Load test：在一定的『工作负荷』下，给系统造成的负荷及系统响应的时间。

5.编写一个 http 接口性能测试方案，测试过程的关注点有哪些，流程等？

一、准备工作

1、系统基础功能验证

性能测试在什么阶段适合实施？切入点很重要！一般而言，只有在系统基础功能测试验证完成、系统趋于稳定的情况下，才会进行性能测试，否则性能测试是无意义的。

2、测试团队组建

根据该项目的具体情况，组建一个几人的性能测试 team，其中 DBA 是必不可少的，然后需要一至几名系统开发人员（对应前端、后台等），还有性能测试设计和分析人员、脚本开发和执行人员；在正式开始工作之前，应该对脚本开发和执行人员进行一些培训，或者应该由具有相关经验的人员担任。

3、工具的选择

综合系统设计、工具成本、测试团队的技能来考虑，选择合适的测试工具，最起码应该满足一下几点：支持对 web（这里以 web 系统为例）系统的性能测试，支持 http 和 https 协议；工具运行在 Windows 平台上； 支持对 webserver、前端、数据库的性能计数器进行监控；

4、预先的业务场景分析

为了对系统性能建立直观上的认识和分析，应对系统较重要和常用的业务场景模块进行分析，针对性的进行分析，以对接下来的测试计划设计进行准备。

二、测试计划

测试计划阶段最重要的是分析用户场景，确定系统性能目标。

1、性能测试领域分析

根据对项目背景，业务的了解，确定本次性能测试要解决的问题点；是测试系统能否满足实际运

行时的需要，还是目前的系统在哪些方面制约系统性能的表现，或者，哪些系统因素导致系统无法跟上业务发展？

确定测试领域，然后具体问题具体分析。

2、用户场景剖析和业务建模

根据对系统业务、用户活跃时间、访问频率、场景交互等各方面的分析，整理一个业务场景表，当然其中最好对用户操作场景、步骤进行详细的描述，为测试脚本开发提供依据。

3、确定性能目标

前面已经确定了本次性能测试的应用领域，接下来就是针对具体的领域关注点，确定性能目标（指标）； 其中需要和其他业务部门进行沟通协商，以及结合当前系统的响应时间等数据，确定最终我们需要达到的响应时间和系统资源使用率等目标；比如：登录请求到登录成功的页面响应时间不能超过 2 秒；报表审核提交的页面响应时间不能超过 5 秒；文件的上传、下载页面响应时间不超过 8 秒；服务器的 CPU 平均使用率小于 70%，内存使用率小于 75%；各个业务系统的响应时间和服务器资源使用情况在不同测试环境下，各指标随负载变化的情况等；

4、制定测试计划的实施时间

预设本次性能测试各子模块的起止时间，产出，参与人员等等。

三、测试脚本设计与开发

性能测试中，测试脚本设计与开发占据了很大的时间比重。

1、测试环境设计

本次性能测试的目标是需要验证系统在实际运行环境中的性能外，还需要考虑到不同的硬件配置是否会是制约系统性能的重要因素！因此在测试环境中，需要部署多个不同的测试环境，在不同的硬件配置上检查应用系统的性能，并对不同配置下系统的测试结果进行分析，得出最优结果（最适合当前系统的配置）。

这里所说的配置大概是如下几类：数据库服务器;应用服务器;负载模拟器;软件运行环境，平台。

测试环境测试数据，可以根据系统的运行预期来确定，比如需要测试的业务场景，数据多久执行一次备份转移，该业务场景涉及哪些表，每次操作数据怎样写入，写入几条，需要多少的测试数据来使得测试环境的数据保持一致性等等。可以在首次测试数据生成时，将其导出到本地保存，在每次测试开

始前导入数据， 保持一致性。

2、测试场景设计

通过和业务部门沟通以及以往用户操作习惯，确定用户操作习惯模式，以及不同的场景用户数量，操作次数，确定测试指标，以及性能监控等。

3、测试用例设计

确认测试场景后，在系统已有的操作描述上，进一步完善为可映射为脚本的测试用例描述，用例大概内容如下：

用例编号：查询表单_xxx_x1（命名以业务操作场景为主，简洁易懂即可） 用例条件：用户已登录、具有对应权限等

操作步骤：系统业务场景描述

4、脚本和辅助工具的开发及使用

按照用例描述，可利用工具进行录制，然后在录制的脚本中进行修改；比如参数化、关联、检查点等等， 最后的结果使得测试脚本可用，能达到测试要求即可；建议尽量自己写脚本来实现业务操作场景，这样对个人技能提升较大；一句话：能写就绝不录制！！！

四、测试执行与管理

在这个阶段，只需要按照之前已经设计好的业务场景、环境和测试用例脚本，部署环境，执行测试并记录结果即可。

1、建立测试环境

按照之前已经设计好的测试环境，部署对应的环境，由运维或开发人员进行部署，检查，并仔细调整，

同时保持测试环境的干净和稳定，不受外来因素影响。

2、执行测试脚本

这一点比较简单，在已部署好的测试环境中，按照业务场景和编号，按顺序执行我们已经设计好的测试脚本。

3、测试结果记录

根据测试采用的工具不同，结果的记录也有不同的形式；现在大多的性能测试工具都提供比较完整的界面图形化的测试结果，当然，对于服务器的资源使用等情况，可以利用一些计数器或第三方监控工具来对其进行记录，执行完测试后，对结果进行整理分析。

五、测试分析

1、测试环境的系统性能分析

根据我们之前记录得到的测试结果（图表、曲线等），经过计算，与预定的性能指标进行对比，确定是否达到了我们需要的结果；如未达到，查看具体的瓶颈点，然后根据瓶颈点的具体数据，进行具体情况具体分析（影响性能的因素很多，这一点，可以根据经验和数据表现来判断分析）。

2、硬件设备对系统性能表现的影响分析

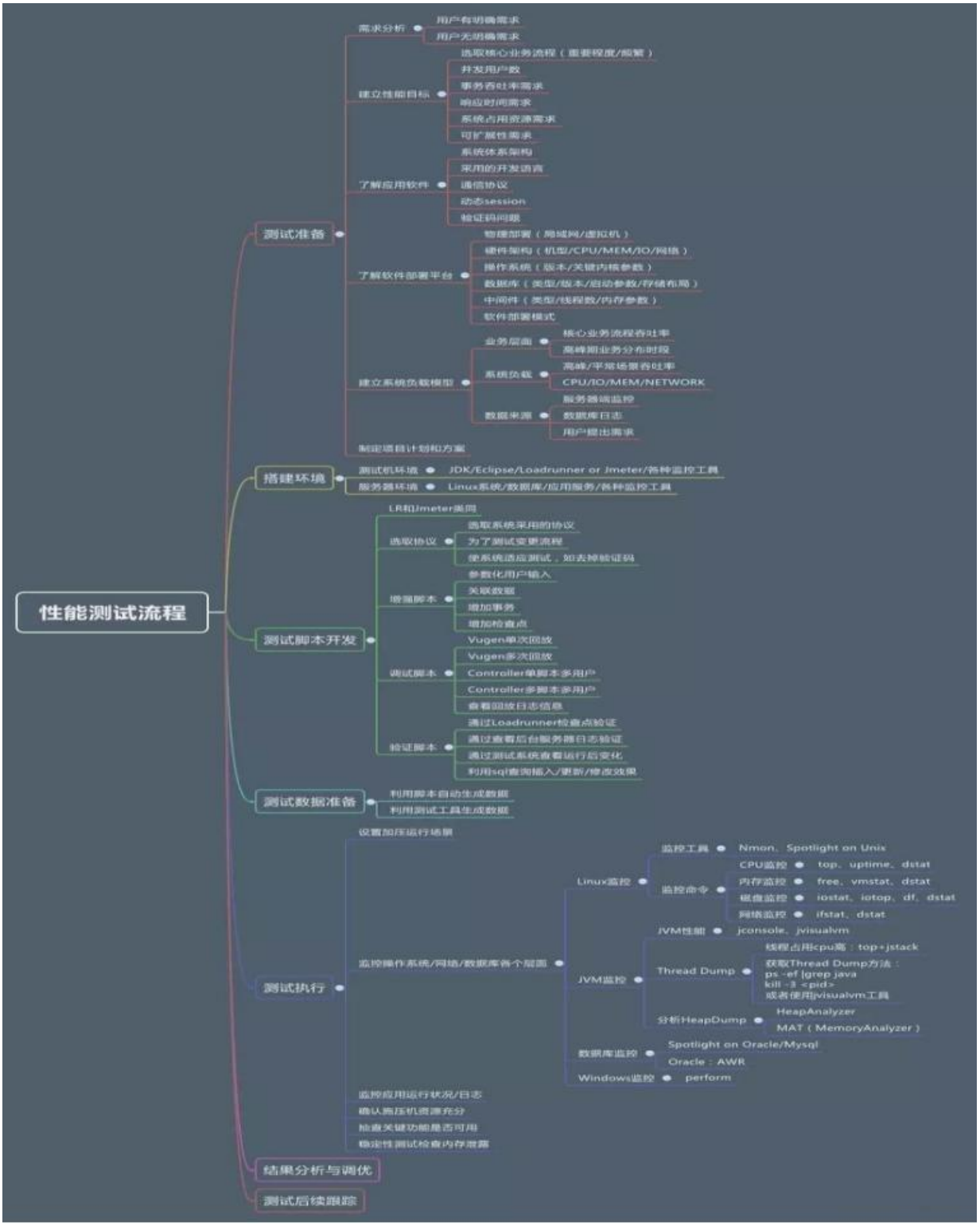
由于之前设计了几个不同的测试环境，故可以根据不同测试环境的硬件资源使用状况图进行分析，确定瓶颈是在数据库服务器、应用服务器抑或其他方面，然后针对性的进行优化等操作。

3、其他影响因素分析

影响系统性能的因素很多，可以从用户能感受到的场景分析，哪里比较慢，哪里速度尚可，这里可以根据 2\5\8 原则对其进行分析；至于其他诸如网络带宽、操作动作、存储池、线程实现、服务器处理机制等一系列的影响因素，具体问题具体分析，这里就不一一表述了。

4、测试中发现的问题

在性能测试执行过程中，可能会发现某些功能上的不足或存在的缺陷，以及需要优化的地方，这也是执行多次测试的优点。



10.1.5如何判断是否有内存泄漏及关注的指标？

10.1.6描述软件产生内存泄露的原因以及检查方式。（可以结合一种开发语言进行描述）

内存泄露的原因，主要是由于开发过程当中申请了计算机资源（例如对象、内存等），但是使用资源完成以后没有及时释放资源导致的。例如在 C 语言当中使用了 malloc 申请了内存，但是未使用 free 来释放内存。

10.1.7简述什么是值传递，什么是地址传递，两者区别是什么？

值传递主调函数传递给被调函数的是值的拷贝，不是原值；地址传递主调函数传递给被调函数的是值的地址。区别是值传递被调函数中的操作不改变主调函数的值，而地址传递则不同。

10.1.8什么是系统瓶颈？

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。

11 LordRunner 相关

11.1 LordRunner 相关

11.1.11.LoadRunner 的工作原理是什么？

LoadRunner 工作原理：

LoadRunner 通过模拟上千万用户实施并发负载，实时性能监控的系统行为和性能方式来确认和查找问题。

1、VuGen 发生器：捕捉用户的业务流，并最终将其录制成一个脚本：

（1）选择相应的一种协议；

（2）在客户端模拟用户使用过程中的业务流程，并录制成一个脚本；

（3）编辑脚本和设置 Run-Time Settings 项；

（4）编译脚本生成一个没有错误的可运行的脚本。

2、控制器（Controller）：

（1）设计场景，包括手动场景设计和目标场景设计两种方式；

（2）场景监控，可以实时监控脚本的运行的情况。可以通过添加计数器来监控 Windows 资源、应用服务器和数据库使用情况。

场景设计的目的是设计出一个最接近用户实际使用的场景，场景设计越接近用户使用的实际情况，测试出来的数据就越接近真实值。

3、负载发生器（Load Generators）

模拟用户对服务器提交请求。通常，在性能测试过程中会将控制器和负载发生器分开；当使用多台负载发生器时，一定要保证负载均衡（指在进行性能测试的过程中，保证每台负载发生器均匀地对服务器进行施压）。

4、分析器（Analysis）：主要用于对测试结果进行分析。

11.1.2LoadRunner 分哪三部分？

用户动作设计；

场景设计；

测试数据分析；

11.1.3LoadRunner 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。

11.1.4什么是并发？在 lordrunner 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。

11.1.5LoadRunner 脚本如何录制和编写？

打开 loadrunner 的 Virtual User Generator，新建脚本

在弹出框中选择 Web（HTTP/HTML）协议，然后点击创建按钮

弹出 start Recording 窗口，选择对应的录制类型（Internet Applications），选择浏览器（这里我们选择 IE），选择需要测试的 web 地址，选择浏览器安装地址。点击 ok

自动打开 IE 浏览器，进入相对应地址，在页面上方显示一个录制工具条。此时我们发给服务器的所有请求都会被记录在脚本中。输入用户名和密码，在点击登录前插入事务，输入事务名称，点击 ok

然后点击登录按钮，待登录成功，显示出成功页面后，点击结束事务，再点击 ok。然后点击工具条上的停止按钮。结束录制，回到脚本中。这时候需要等待会，待自动生成脚本。

生成的脚本含有刚才录制的信息，点击菜单栏，回放按钮。回放如果有红色，是报错信息，没有红色， 如下图，说明运行成功。

还可点击“View”菜单栏的“Test Results”进行查看。显示 passed 即为成功。脚本便可使用。

11.1.6LoadRunner 中的 Think Time 有什么作用？

用户在执行连续操作之间等待的时间称为“思考时间”，它是决定对服务器施压大小的因素之一。设置思考时间，是为了更真实的模拟用户。Vuser 使用 Lr_think_time 函数来模拟用户思考时间。录制 Vuser 脚本时， VuGen 将录制实际思考时间，并插入到 Vuser 脚本中响应的 Lr_think_time 语句。可以编辑录制的 Lr_think_time 语句，并向 Vuser 脚本手动添加更多 Lr_think_time 语句。

可以通过选择【插入】>【步骤】>【思考时间】来插入思考时间步骤。当录制 Java Vuser 脚本时，不会在 Vuser 脚本中生成 Lr_think_time 语句。

可以使用【Run-time Settings】，更改执行 Vuser 脚本时 Lr_think_time 语句的运行方式。

11.1.74.在搜索引擎中输入汉字就可以解析到对应的域名，请问如何用 LoadRunner 进行测试？

a)建立测试计划，确定测试标准和测试范围

b)设计典型场景的测试用例，覆盖常用业务流程和不常用的业务流程等 c)根据测试用例，开发自动测试脚本和场景：

i.录制测试脚本

1.新建一个脚本（Web/HTML 协议）

2.点击录制按钮，在弹出的对话框的 URL 中输入”about:blank”。

3.在打开的浏览器中进行正常操作流程后，结束录制。

4.调试脚本并保存。可能要注意到字符集的关联。

ii.设置测试场景

1.针对性能设置测试场景，主要判断在正常情况下，系统的平均事务响应时间是否达标

2.针对压力负载设置测试场景，主要判断在长时间处于满负荷或者超出系统承载能力的条件下，系统是否会崩溃。

iii.执行测试，获取测试结果，分析测试结果

11.1.85.一台客户端有三百个客户与三百个客户端有三百个客户对服务器施压，有什么区别？

300 个用户在一个客户端上，会占用客户机更多的资源，而影响测试的结果。线程之间可能发生干扰，而产生一些异常。

300 个用户在一个客户端上，需要更大的带宽。

IP 地址的问题，可能需要使用 IP Spoof 来绕过服务器对于单一 IP 地址最大连接数的限制。

所有用户在一个客户端上，不必考虑分布式管理的问题；而用户分布在不同的客户端上，需要考虑使用控制器来整体调配不同客户机上的用户。同时，还需要给予相应的权限配置和防火墙设置。

11.1.9客户交付一个性能测试项目，请阐述你的实施流程。

测试设计阶段：

1)了解被测系统的性能需求，定义测试目标和范围；

2)了解系统的技术信息，如系统架构等；

3)确定测试方案、进度安排，并制定测试计划,场景设置方案,及需要收集的测试数据；

4）同相关人员协商讨论测试方案；

5）准备数据收集模板；不同项目的性能测试，需要收集的数据不同；针对性的制定一个模板，更符合需要；

测试环境准备：

1）技术准备；选择性能测试工具；测试方案中涉及到的技术问题；测试数据的收集方案实现；如：如何监控系统资源等；

2）搭建测试环境；

3）创建初始数据；如虚拟用户使用的账号等；

测试执行阶段：

1）录制脚本；

2）调试脚本；

3）执行场景；

4）收集测试数据，并简单整理；

测试分析阶段：

1）分析测试数据；

提交测试报告。

11.1.10 解释 5 个常用的性能指标的名称与具体含义。

- 并发：所有用户在同一时刻对系统执行操作，一般指做同一件事情或操作。
- 在线：所有用户在一段时间内对系统执行操作。
- 请求响应时间

  从 client 端发出请求到得到响应的整个时间；

  包括：client 端响应时间+网络响应时间+Server 端响应时间。
- 事务请求响应时间

  完成相应事务所用的时间；这个是性能测试中重点关注的指标。
- TPS（Transaction Per Second）

  每秒钟系统能够处理的交易或事务的数量。它是衡量系统处理能力的重要指标。TPS 是

LoadRunner 中重要的性能参数指标。

- 点击率（Hit Per Second）

  每秒发送的 HTTP 请求的数量；点击率越大对 Server 的压力越大

- 资源利用率

  对不同资源的使用程度，如 CPU，I/O,内存，……

11.1.11 写出 5 个 Loadrunner 中常用函数，并对其中 2 个举例说明用法。

字符串复制

strcpy(str,"Hello ") ;

字符串连接

strcat(str,"World !");

lr_message("str: %s",str);

sprintf(s, "%s love %s.", "I", “ocean"); //产生：“I love ocean. "

变量转为参数,将变量 str 的值存到参数 Param 中

lr_save_string(str,"Param");

参数复制

lr_save_string(lr_eval_string("{Param}"),"Param_1");

参数转为变量

strcpy(str1,lr_eval_string("{Param_1}"));

11.1.12 简述 LoadRunner 的工作原理？

Loadrunner 会自动监控指定的 URL 或应用程序所发出的请求及服务器返回的响应，它做为一个第三方（Agent）监视客户端与服务器端的所有对话，然后把这些对话记录下来，生成脚本，再次运行时模拟客户端发出的请求，捕获服务器端的响应。

11.1.13 什么是集合点？设置集合点有什么意义？LoadRunner 中设置集合点的函数是哪个？

集合点：是一个并发访问的点，例如在测试计划中，可能会要求系统能够承受 1000 人同时提交数据，在 LoadRunner 中可以通过在提交数据操作前面加入集合点，这样当虚拟用户运行到提交数据的集合点时，LoadRunner 就会检查同时有多少用户运行到集合点，如果不到 1000 人，LoadRunner 就会命令已经到集合点的用户在此等待，当在集合点等待的用户达到 1000 人时，LoadRunner 命令 1000 人同时去提交数据，并发访问的目的。

注意：集合点经常和事务结合起来使用，常放在事务的前面，集合点只能插入到 Action 部分，vuser_init 和 vuser_end 中不能插入集合点。集合点函数如下：lr_rendezvous(“SubmitData”)

11.1.14 HTML-based script 与 URL-based script 的脚本有什么区别？

使用“HTML-based script”的模式录制脚本，VuGen 为用户的每个 HTML 操作生成单独的步骤，这种脚本看上去比较直观；使用“URL-based script”模式录制脚本时，VuGen 可以捕获所有作为用户操作结果而发送到服务器的 HTTP 请求，然后为用户的每个请求分别生成对应方法。

通常，基于浏览器的 Web 应用会使用“HTML-based script”模式来录制脚本；而没有基于浏览器的 Web 应用、Web 应用中包含了与服务器进行交互的 Java Applet、基于浏览器的应用中包含了向服务器进行通信的 JavaScript/VBScript 代码、基于浏览器的应用中使用了 HTTPS 安全协议，这时使用“URL-based script”模式进行录制。

11.1.15 如何设置 LaodRunner 才能让集合点只对一半的用户生效？

在 Loadrunner 中,进入 Controller->Scenario->Rendezvous ...->Policy，系统弹出 Policy 对话框。在该对话框中可以设定集合点执行的策略。

第一项：表示当所有用户数的 X%到达集合点时，开始释放等待的用户并继续执行场景。

第二项：表示当前正在运行用户数的 X%到达集合点时，开始释放等待的用户并继续执行场景。

第三项：表示当 X 个用户到达集合点时，开始释放等待的用户并继续执行场景。

其中还有一项 Timeout between Vusers，就 30 秒来说，当第一个用户到达集合点后，再等待 30 秒，如果在 30 秒内到达的用户数达到指定的数量，就开始继续执行场景。如果在 30 秒内还没有达到指定的用户数量，就不再等待，开始释放等待的用户并继续执行场景。

由上可知，将第一项用户数设置成 50%可。

11.1.16 LoadRunner 的 Controller 组件中 Pacing 参数的作用是什么？

设置 LoadRunner 中事务（trsaction）在每次循环中的间隔时间，系统可以利用该间隔时间进行相应事务的结束收尾工作的处理。加大 Pacing 参数，可使系统压力减小。

11.1.17 LoadRunner 中如何监控 Windows 资源？

进入被监视 windows 系统，开启以下二个服务：Remote Procedure Call(RPC)和 Remote Registry Service；在 controller 中，Windows Resources 窗口中右击鼠标选择 Add Measurements，添加被监控 windows 的 IP 地址，选择所属系统，然后选择需要监控的指标就可以开始监控。

11.1.18 如果让 QALoad 模拟 LoadRunner 中只对关注的性能点进行迭代测试，你有什么好方法？

可以将 QAload 脚本中关注的事务点写成一个循环，如果进行长时间的综合场景测试，则可将其写成一个永真循环，如 while (1) {事务点}，则对关注的性能点进行循环，而脚本其他代码不会进行循环。

11.1.19 什么是负载测试？

负载测试是通过逐步增加系统负载，测试系统性能的变化，并最终确定在满足性能指标的情况下，

系统所能承受的最大负载量的测试，例如，访问一个页面的响应时间规定不超过 1 秒，负载测试就是测试在响应时间为 1 秒时，系统所能承受的最大并发访问用户的数量。

11.1.20 什么是性能测试？

性能测试：指在一定的约束条件下（指定的软件、硬件、网络环境等），确定系统所能承受的最大负载压力。

11.1.21 说明负载测试过程？

第一步：计划测试。在这里，我们需开发一个明确定义的测试计划，以确保该测试方案能完成负载测试目标。第二步：创建虚拟用户。创建的脚本需要包含单个虚拟用户需要执行的操作、多个虚拟用户作为一个整体要执行的操作、以及能够作为事务来度量的操作。第三步：创建场景。一个场景描述了在一个测试会话中发生的事件。它包含了当场景中运行时的机器、脚本和虚拟用户。我们使用 LoadRunner 中的 Controller 创建场景。我们可以创建手动场景也可以创建基于目标的场景。在手动场景中，我们定义虚拟用户的数量、负载生成器、被分配到每个脚本中虚拟用户的百分比。对于 web 测试，我们创建基于目标的场景，其中目标即测试过程中要达成的性能目标。LoadRunner 会由此自动为我们创建一个场景。第四步：运行场景。我们通过配置多个虚拟用户同时执行任务来模拟对服务器加压。在测试之前，我们设置场景的配置和计划安排。我们可以运行整个场景、一组虚拟用户或单个虚拟用户。第五步：监控场景。我们使用 LoadRunner 联机运行来监测场景执行、事务系统资源，Web 资源，Web 服务器资源，Web 应用服务器资源，数据库服务器资源，网络延迟，流媒体资源，防火墙服务器资源，ERP 服务器资源，Java 性能监视器。第六步：分析测试结果。在场景执行时，LoadRunner 记录了应用软件在不同负载下的性能。我们可以使用 LoadRunner 的图表和报告来分析应用软件的性能。

11.1.22 我们什么时候做负载和性能测试？

我们一旦完成界面（GUI）测试，我们就可以执行负载测试。现代的软件系统架构庞大而复杂的。而单用户测试主要是侧重于系统组件的功能和用户界面的测试，应用测试侧重于整个系统的性能和可靠性。

例如，一个典型的应用测试场景，描绘了 1000 个用户，同时登录到系统。这就产生了问题，如系统的响应时间是多少，它会崩溃么；是否兼容不同的应用软件和系统平台；它是否可以支撑成千上万的用户等，这时我们就需要做负载和性能测试。

11.1.23 什么是 LoadRunner 的组件？

LoadRunner 的组件有虚拟用户生成器，控制器、代理程序、LoadRunner 的分析器和监控器，LoadRunner 在线帮助

11.1.24 你用 LoadRunner 的哪个组件录制脚本？

虚拟用户生成器（VuGen）可以用来录制脚本。它通过多种应用程序类型和通讯协议来确保你开发一个 Vuser 脚本。

11.1.25 在多用户模式下你用 LoadRunnner 的哪个组件来回放脚本？

虚拟用户生成器（VuGen）可以用来录制脚本。它通过多种应用程序类型和通讯协议来确保你开发一个 Vuser 脚本。

11.1.26 在多用户模式下你用 LoadRunnner 的哪个组件来回放脚本？

Controller 组件可以用来在多用户模式下回放脚本。这个过程可以在一组虚拟用户以组的形式执行脚本的场景中运行时完成。

11.1.27 什么是场景

场景定义了发生在每个测试会话中的事件。例如，一个场景定义和控制了要加载的虚拟用户的数量，要被执行的动作，虚拟用户模拟压力时的机器。

11.1.28 解释 Web Vuser 脚本的录制模式

我们使用 VuGen 通过记录一个用户在客户端执行典型的业务流程来开发一个 Vuser 脚本。VuGen 通过记录客户端和服务器之间的交互来创建脚本。例如，基于 Web 的应用程序，VuGen 监测客户端直到数据库，跟踪所有发送出的请求，再从数据库服务器接收。我们使用 VuGen：监测应用程序和服务器的交互；使形成函数调用；插入生成的函数调用到一个 Vuser 脚本中。

11.1.29 为什么创建参数?

参数就像脚本中的变量。他们被用来改变对服务器的输入值来模拟真正的用户。每次当脚本运行时，不同的数据设置被发送到服务器。为了更精确的在 Controller 中测试，从而更好的模拟使用模型；一个脚本在系统上模拟很多不同的用户。

11.1.30 什么是关联？解释自动关联和手动关联的区别

关联是用来获取每次脚本运行时由嵌套请求产生的唯一的数据。关联提供值以避免产生重复值的错误，并且优化代码（避免嵌套请求）。自动关联是我们自己为关联设置的一定规则。它可以是应用服务器特定的。这里的值被通过规则创建的数据取代。在手动关联里，我们要关联的值被浏览和创建关联以用来关联。

11.1.31 什么是关联？解释自动关联和手动关联的区别，你在哪里设置自动关联的选项

从 web 自动关联的观点来说，可以在录制选项中和关联表单中设置。这里我们可以使整个脚本关联，选择在线信息或者离线的动作，在这里我们可以定义关联的规则。为数据库自动关联可以使用显示输出窗口和为了关联浏览，找出关联查询的图表，选择我们想关联的查询值来完成。如果我们知道要被关联的特殊值，我们只用为这个值创建关联，指定如何创建这个值

11.1.32 什么函数可以捕捉到 web Vuser 脚本的动态值？

Web_reg_save_param 函数保存动态的数据信息到一个参数中。

11.1.33 什么时候你在虚拟用户产生器中禁用日志，什么时候选择标准日志和扩展日志？

一旦我们调试脚本来验证它的功能，我们可以只记录错误。当我们在场景中添加一个脚本，日志记录将自动被禁用。标准日志选项：当您选择标准日志，它在脚本执行调试时创建一个标准的功能和发送信息的日志。大负荷的测试场景时，禁用此选项。当你复制一个脚本到一个场景，日志自动禁用扩展日志选项：选择扩展日志创建一个扩展的日志，包括警告和其他信息。禁用此选项为大负荷的测试场景。当你复制一个脚本的场景，将被自动禁用日志记录。我们可以指定附加信息，用扩展日志选项添加到扩展日志中。

11.1.34 你如何调试 LoadRunner 的脚本？

VuGen 包含两个选项帮助调试 VuGen 脚本—通过逐步命令和断点运行某一步。选项对话框中的 Debug 设置，使我们决定在场景运行期间进行跟踪的程度。Debug 信息被写到输出窗口，我们可以使用 lr_set_debug_message 函数手动设置你的脚本中的信息类。如果我们希望得到关于一小部分脚本的调试信息时，这就有意义了。

11.1.35 你怎么写 LR 中用户自定义的函数？写几个你以前项目中的函数？

在我们创建用户自定义的函数之前我们需要创建外部的库函数。我们添加这个库到 VuGen bin 目录下。一旦库被添加，然户我们分配给用户自定义的函数作为参数。函数应该有以下格式：__declspec（dllexport）char*<函数名称>（char*，char *）。用户自定义函数如下：GetVersion，GetCurrentTime，GetPltform，这些都是我之前的项目中用到的用户自定义的函数。

11.1.36 在 run-time setting 里你可以设置哪些改变？

Run Time Setting 中我们所做的是：a）Pacing—它有迭代次数。b）Log—在此，我们有已禁用的标准日志。c）扩展 Think Time—在 Think Time 里有两个选项，忽略 think time 和回放 think time。d)General—在 General 选项卡中，我们可以设置 vusers 作为进程或者多线程，将每一步作为一个事务。

11.1.37 你在哪里设置 Vuser 测试时迭代？

我们在 VuGen 中的 Run Time Setting 中设置迭代。这个导航是 Run Time Setting，Pacing 选项卡，设置迭代次数。

11.1.38 你如何在负载下执行功能测试？

负载下的功能可以通过同时运行多个 Vuser 来测试。通过增加一定数量的 Vuser，我们可以判断服务器可以承受多少负载。

11.1.39 什么是 Ramp up？你如何设置？

这个选项被用来逐步增加 Vuser 的数量/服务器上的负载。初始值设置，间隔的等待值可以被指定。要设置 Ramp up，进入“场景调度选项”

11.1.40 Vuser 作为线程运行的优势是什么？

VuGen 提供使用多线程的设施。这使每个产生器中运行更多的 Vuser。如果 Vuser 作为进程运行，相同的驱动程序为每个 Vuser 加载到内存，从而占用大量的内存。这限制了在单发生器中运行的 Vuser 的数量。如果 Vuser 作为线程运行，为给定数量的 Vuser（如 100），只有一个驱动程序的实例被加载到内存。每个线程共享父驱动程序的内存，从而使每个产生器运行更多的 Vuser。

11.1.41 如果你想停止执行出错的脚本，怎么做？

Lr_abort 函数中止执行 Vuser 脚本。它指示 Vuser 停止执行 Action 部分，执行 vuser_end 部分，并结束执行。这个函数是有用的，当你需要手工停止一个脚本的执行，作为一个指定错误条件下的结果。当你使用这个函数停止一个脚本，Vuser 被分配成停止状态。为让它生效，我们必须首先在 Run—Time Setting 中取消选择“Continue on error”

11.1.42 响应时间和吞吐量间的关系是什么？

吞吐量图表显示的是 Vuser 每秒从服务器收到的以字节为单位的数据量。当我们比较它和事务响应时间，我们会发现，若吞吐量下降，响应时间也会下降。同样，高峰时期的吞吐量和最高的响应时间大约在同一个时间。

11.1.43 你如何识别性能瓶颈？

性能瓶颈可以使用监控器监测。这些监控器可能是应用服务器监控器、web 服务器监控器和网络监控器。它们帮助找出在场景中导致响应时间增加有问题的区域。通常所做的测试指标是性能响应时间、吞吐量，点击率，网络延迟图等

11.1.44 如果 web 服务器、数据库服务器、网络都一切正常，那么哪里可能有问题？

问题可能是系统本身，应用程序服务器或为应用程序编写的代码。

11.1.45 你如何找出 web 服务器相关的问题？

利用 web 资源监控器，我们可以发现 web 服务器的性能。使用这些监测器我们可以分析发生在场景中的 web 服务器的吞吐量，每秒点击数、以及每秒 HTTP 响应数，每秒下载的网页数量。

11.1.46 你是怎么找到数据库中的相关问题？

监视运行“数据库”监测器和在“数据资源图”的帮助下，我们可以发现数据库中的相关问题。例如您可以在 Controller 运行前指定您想要的资源来监控，然后你可以看到数据库中的相关问题

11.1.47 覆盖图和关联图之间的区别是什么？

覆盖图：它覆盖两个图表的内容，使共用一个 X 轴。合并后的图形左 Y 轴显示当前图形中的值，右 Y 轴显示合并图的 Y 轴的值。关联图：绘制相互对立的两个图表的 Y 轴。活动图的 Y 轴，成为合并后的图的 X 轴。合并图的 Y 轴成为合并后的图的 Y 轴

11.1.48 你是怎么计划负载的？标准是什么？

计划负载测试，以决定用户数量，我们将使用的机器型号，在哪里运行。它是基于两个重要文件，任务分配图和事务状况。任务分配图，给我们提供用户数量为一个特定的业务信息和负载时间。从这个图决定使用高峰期和不工作时段。事务配置文件，为我们提供了有关交事务方面的情况，包括事务的名称和我们决定的它们关于场景的优先等级。

11.1.49 vuser_init 动作包含什么？

Vuser_init 动作包含登录到服务器的程序

11.1.50 vuser_end 动作包含什么？

Vuser_end 部分包含注销的程序。

11.1.51 什么是 Think Time？你如何改变这个阈值？

Think Time 是一个真实的用户动作之间的等待时间。例如：当一个用户从服务器接收数据时，用户可以在响应前等待几秒钟来检查响应数据。此延迟就是 Think Time。更改这个阈值：阈值是一个标准水平，思考时间低于阈值水平时将被忽略，Vugen 不会生成思考时间语句。默认值是 5 秒。我们可以在 Vugen

录制选项中改变 Think Time 阈值。

11.1.52 简述使用 Loadrunner 的步骤

制定性能测试计划—>开发测试脚本—>设计测试场景—>执行测试场景—>监控测试场景—>分析测试结果

11.1.53 什么是集合点？设置集合点有什么意义？Loadrunner 中设置集合点的函数是哪个？

在性能测试过程中，需要模拟大量用户在同一时刻，访问系统并同时操作某一任务，可以通过配置集合点来实现，多个用户同时进行某操作；

集合点可以在服务器上创建密集的用户负载，使 LoadRunner 能够测试服务器在负载状态下的性能。

设置集合点函数：lr_rendezvous(“Meeting”)； // Meeting 是集合点名称

11.1.54 你如何在负载测试模式下执行功能测试？

在负载测试模式下，可以通过同时运行数个虚拟用户，通过增加虚拟用户数，确定服务器在多大的负载量下，仍然可以正常运行，我一般进行核心功能操作，验证核心功能运行是否正常。

11.1.55 什么是逐步递增？你如何来设置？

虚拟用户数随着负载时间逐渐增加，可以帮助确定系统响应时间减慢的准确时间点。

可以在“加压”选项卡中进行设置：如下图所示，将设置更改为：“每 30 秒启动 2 个 Vuser”

11.1.56 响应时间和吞吐量之间的关系是什么？

当系统吞吐量未达到系统处理极限时，系统性能不会衰减，交易平均响应时间一般也不会递增，当系统达到吞吐量极限时，客户端交易会在请求队列中排队等待，等待的时间会记录在响应时间中，故交易平均响应时间一般会递增。

11.1.57 说明一下如何在 LR 中配置系统计数器？

以 windows 资源监控为例，可右键点“添加度量”，输入系统 IP、选择平台类型，确定即可，详细参加 LR 自带操作手册^_^。
对于监控不同类型的操作系统，需要做一些准备工作，可参见监控操作系统资源部分。

11.1.58 在 LoadRunner 中为什么要设置思考时间和 pacing

录制时记录的是客户端和服务端的交互，如果要精确模拟 用户的行为，那么客户操作客户端时花费了很多时间要怎么模拟呢?

录入 填写提交的内容，从列表中下拉搜索选择特定的值等，这时 LOADRUNNER 不会记录用户 的客户端操作，而是记录了用户这段时间，成为思考时间(Think-time)，因为用户的这些客户端操作不会影响服务端，只是让服务器端在这段时间内没有请求而已。，所以加入思考时间就能模拟出熟练的或者生疏的用户操作，接近实际对于服务端的压力。

Vuser 思考时间模拟实际用户在不同操作之间等待的时间。例如，当用户收到来自服务器的数据时，可能要等待几秒钟查看数据，然后再做出响应。这种延迟就称为“思考时间”。VuGen 使用 lr_think_time 函数将思考时间值录制到 Vuser 脚本中。以下录制的函 数指明用户等待了 8 秒钟才执行下一个操作：

```
lr_think_time(8);
```

当您运行了 Vuser 脚本并且 Vuser 遇到了上述 lr_think_time 语句时，默认情况下，Vuser 将等待 8 秒钟后再执行下一个操作。可以使用思考时间运行时设置来影响运行脚本时 Vuser 使用录制思考时间的方式。

11.1.59 如何理解 TPS?

TPS 主要还是体现服务器对当前录制的事务的处理速度快慢。TPS 高并不代表性能好。

TPS 是 Transactions Per Second 的 缩 写，也就是事务数/ 秒。它是软件测试结果的测量单位。一个事务是指一个客户机向服

务器发送请求然后服务器做出反应的过程。客户机在发送请求时开始计时，收到服务器响应后结

束计时，以此来计算使用的时

间和完成的事务个数，最终利用这些信息来估计得分。客户机使用加权协函数平均方法来计算客户机的得分，试软件就是利用

客户机的这些信息使用加权协函 数平均方法来计算服务器端的整体 TPS 得分。

11.1.60 loadrunner 中的设置线程和进程的区别

loadrunner 中，在进行运行设置中有一项选择，是按进程运行 Vuser 或按线程运行 Vuser?下面进行分别来讲：

1.按进程运行 Vuser：Controller 将使用驱动程序 mdrv 运行 Vuser。如果按进程方式运行每个 Vuser，则对于每个 Vuser 实例，都将启动一个 mdrv 进程。如果设置了 10 个 Vuser，则在任务管理器中出现 10 个 mdrv 进程。多个 mdrv 进程肯定会占用大量内存及其他系统资源，这就限制了可以在任一负载生成器上运行的 Vuser 的数量。

2.按线程运行 Vuser:及设置了 10 个 Vuser，其只会调用一个驱动程序 mdrv.而每个 Vuser 都按线程运行，这些线程 Vuser 将共享父进程的内存段。这就节省了大量内存控件，从而可以在一个负载生成器上运行更多的 Vuser。

任何选择都是有两面性的。选择线程方式运行 Vuser 会带来一些安全问题。因为线程的资源是从进程资源中分配出来的，因此同一个进程中的多个线程会有共享的内存空间，这样可能会引起多个线程的同步问题，调度不好，就会出问题，不如 A 线程要用的资源就必须等待 B 线程释放，而 B 也在等待其他资源释放才能继续。这就会出现这样的问题：同一个测试场景，用线程并发就会超时失败或报错，而用进程并发就没错。

虽然会有区别，但两种方式的运行都会给服务端造成的压力是一样的。

11.1.61 HTML-Based scrīpt 和 URL-Based scrīpt 录制的区别？

基于浏览器的应用程序推荐使用 HTML-Based scrīpt。

不是基于浏览器的应用程序推荐使用 URL-Based scrīpt。

如果基于浏览器的应用程序中包含了 Java scrīpt 并且该脚本 向服务器产生了请求，比如 DataGrid 的分

页按钮等，也要使用 URL-Based scrīpt 方式录制。

基于浏览器的应用程序中使用了 HTTPS 安全协议，使用 URL-Based scrīpt 方式录制。

录制过程中不要使用浏览器的“后退”功能，LoadRunner 对其支持不太好。

11.1.62 本次通过 loadRunner 录制 SQL Server 介绍一下如何测试一个 sql 语句或存储过程的执行性能。

主要分如下几个步骤完成：

第一步、测试准备

第二步、配置 ODBC 数据源

第三步、录制 SQL 语句在 Sql Server 查询分析器中的运行过程

第四步、优化录制脚本，设置事务

第五步、改变查询数量级查看 SQL 语句的性能

第六步、在 controller 中运行脚本

11.1.63 LoadRunner 如何创建脚本？

1. 启动 VuGen:选择需要新建的协议脚本，可以创建单协议，或是多协议脚本
2. 点击 Start Record 按钮，输入程序地址，开始进行录制
3. 使用 VuGen 进行录制：创建的每个 Vuser 脚本都至少包含三部分：vuser_init、一个或多个 Actions 及 vuser_end。录制期间，可以选择脚本中 VuGen 要插入已录制函数的部分。运行多次迭代的 Vuser 脚本时，只有脚本的 Actions 部分重复，而 vuser_init 和 vuser_end 部分将不重复

11.1.64 LoadRunner 如何设置 Recording Options 选项？（以单协议 http/html 为例）

1.菜单 tools->Recording Options 进入录制的设置窗体

2.Recording 标签页:选用哪种录制方式

3.Browser 标签页：浏览器的选择

4.Recording Proxy 标签页：浏览器上的代理设置

5.Advanced 标签页：可以设置录制时的 think time，支持的字符集标准等

6.Correlation 标签页：手工设置关联，通过关联可在测试执行过程中保存动态值。使用这些设置可以配置 VuGen 在录制过程中执行的自动关联的程度。

11.1.65 LoadRunner 如何选择协议？

LoadRunner 属于应用在客户端的测试工具，在客户端模拟大量并发用户去访问服务器，从而达到给服务器施加压力的目的。所以说 LoadRunner 模拟的就是客户端，其脚本代表的是客户端用户所进行的业务操作，即只要脚本能表示用户的业务操作就可以。

1.LR 支持多种协议，请大家一定要注意，这个地方协议指的是你的 Client 端通过什么协议访问的 Server，Client 一般是面向最终使用者的，Server 是第一层 Server 端，因为现在的体系架构中经常 Server 层也分多个层次，什么应用层，什么数据层等等，LR 只管 Client 如何访问第一层 Server.

2.特别要注意某些应用，例如一个 Web 系统，这个系统是通过 ActiveX 控件来访问后台的，IE 只是一个容器，而 ActiveX 控件访问后台是通过 COM/DCOM 协议的，这种情况就不能使用 Web 协议，否则你什么也录制不到，所以，LR 工程师一定要了解应用程序的架构和使用的技术。 3. 象 HTTPS，一般来讲一定要选择多协议，但在选择具体协议的时候一定只选 Web 协议，这时候才能作那个端口映射。

n 通常协议选择

1.对于常见的 B/S 系统，选择 Web(Http/Html)

2.测一个 C/S 系统，根据 C/S 结构所用到的后台数据库来选择不同的协议，如果后台数据库是 sybase，则采用 sybaseCTlib 协议，如果是 SQL server,则使用 MS SQL server 的协议，至于 oracle 数据库系统，当然就使用 Oracle 2-tier 协议。

3.对于没有数据库的 C/S（ftp,smtp）这些可以选择 Windwos Sockets 协议。

4.至于其他的 ERP，EJB（需要 ejbdetector.jar），选择相应的协议即可.

5. 一般可以使用 Java vuser 协议录制由 java 编写的 C/S 模式的软件, ,当其他协议都没有用时,只能使用 winsocket 协议

11.1.66 Loadrunner 支持哪些常用协议？

Web(HTTP/HTML)

Sockets

.net 协议

web services

常用数据库协议（ODBC，ORACLE，SQLSERVER 等）

邮件(SMTP、pop3)

其它协议

11.1.67 性能测试的类型都有哪些？

负载测试(Load Test)

通过逐步增加系统负载，测试系统性能的变化，并最终确定在满足性能指标的情况下，系统所能承受的最大负载量的测试。

压力测试(Stress Test)

通过逐步增加系统负载，测试系统性能的变化，并最终确定在什么负载条件下系统性能处于失效状态，并以此来获得系统能够提供的最大服务级别的测试。

压力测试是一种特定类型的负载测试。

疲劳强度测试

通常是采用系统稳定运行情况下能够支持的最大并发用户数或者日常运行用户数，持续执行一段时间业务，通过综合分析交易执行指标和资源监控指标来确定系统处理最大工作量强度性能的过程。

疲劳强度测试可以反映出系统的性能问题，例如内存泄漏等。

大容量测试(Volume Test)

对特定存储、传输、统计、查询业务的测试。

11.1.68 Loadrunner 常用的分析点都有哪些？

Vusers：

提供了生产负载的虚拟用户运行状态的相关信息，可以帮助我们了解负载生成的结果。

Rendezvous（负载过程中集合点下的虚拟用户）：

当设置集合点后会生成相关数据，反映了随着时间的推移各个时间点上并发用户的数目，方便我们了解并发用户的变化情况。

Errors（错误统计）：

通过错误信息可以了解错误产生的时间和错误类型，方便定位产生错误的原因。

Errors per Second（每秒错误）：

了解在每个时间点上错误产生的数目，数值越小越好。通过统计数据可以了解错误随负载的变化情况，定为何时系统在负载下开始不稳定甚至出错。

Average Transaction Response Time（平均事务响应时间）：

反映随着时间的变化事务响应时间的变化情况，时间越小说明处理的速度越快。如果和用户负载生成图合并，就可以发现用户负载增加对系统事务响应时间的影响规律。

Transactions per Second（每秒事务）：

TPS 吞吐量，反映了系统在同一时间内能处理事务的最大能力，这个数据越高，说明系统处理能力越强。

Transactions Summary（事务概要说明）

统计事物的 Pass 数和 Fail 数，了解负载的事务完成情况。通过的事务数越多，说明系统的处理能力越强；失败的事务数越小说明系统越可靠。

Transaction performance Summary(事务性能概要)：

事务的平均时间、最大时间、最小时间柱状图，方便分析事务响应时间的情况。柱状图的落差越小说明响应时间的波动小，如果落差很大，说明系统不够稳定。

Transaction Response Time Under Load（用户负载下事务响应时间）：

负载用户增长的过程中响应时间的变化情况，该图的线条越平稳，说明系统越稳定。

Transactions Response time(事务响应时间百分比)：

不同百分比下的事务响应时间范围，可以了解有多少比例的事物发生在某个时间内，也可以发现响应时间的分布规律，数据越平稳说明响应时间变化越小。

Transaction Response Time（各时间段上的事务数）：

每个时间段上的事务个数，响应时间较小的分类下的是无数越多越好。

Hits per Second（每秒点击）：

当前负载重对系统所产生的点击量记录，每一次点击相当于对服务器发出了一次请求，数据越大越好。

Throughput（吞吐量）：

系统负载下所使用的带宽，该数据越小说明系统的带宽依赖就越小，通过这个数据可以确定是不是网络出现了瓶颈。

HTTP Responses per Second（每秒 HTTP 响应）：

每秒服务器返回各种状态的数目，一般和每秒点击量相同。点击量是客户端发出的请求数，而 HTTP 响应数是服务器返回的响应数。如果服务器的响应数小于点击量，那么说明服务器无法应答超出负载的连接请求。

Connections per Second（每秒连接）：

统计终端的连接和新建的连接数，方便了解每秒对服务器产生连接的数量。同时连接数越多，说明服务器的连接池越大，当连接数随着负载上升而停止时，说明系统的连接池已满，通常这时候服务器会返回 504 错误。需要修改服务器的最大连接来解决该问题。

LoadRunner 不执行检查方法怎么解决？

在录制 Web 协议脚本中添加了检查方法 Web_find，但是在脚本回放的过程中并没有执行。

错误现象：在脚本中插入函数 Web_find，在脚本中设置文本以及图像的检查点，但是在回放过程中并没有对设置的检查点进行检查，即 Web_find 失效。

错误分析：由于检查功能会消耗一定的资源，因此 LoadRunner 默认关闭了对文本以及图像的检查，所以在设置检查点后，需要开启检查功能。

解决办法：打开运行环境设置对话框进行设置，在“Run-time Settings”的“Internet Protocol”选项里的“Perference”中勾选“Check”下的“Enable Image and text check”选项。

11.1.69 并发用户数是什么？跟在线用户数什么关系？

并发主要是针对服务器而言，是否并发的关键是看用户操作是否对服务器产生了影响。因此，并发用户数量的正确理解为：在同一时刻与服务器进行了交互的在线用户数量，这种交互既可以是单向的传输数据，也可以是双向的传送数据。

并发用户数是指系统运行期间同一时刻进行业务操作的用户数量。

该数量取决于用户操作习惯、业务操作间隔和单笔交易的响应时间。

使用频率较低的应用系统并发用户数一般为在线用户数的 5%左右。

使用频率较高的应用系统并发用户数一般为主线用户数的 10%左右

11.1.70 LoadRunner 请求无法找到如何解决？

在录制 Web 协议脚本回放脚本的过程中，会出现请求无法找到的现象，而导致脚本运行停止。

错误现象：Action.c(41): Error -27979: Requested form. not found [MsgId: MERR-27979]

Action.c(41): web_submit_form. highest severity level was “ERROR”,0 body bytes, 0 header bytes [MsgId: MMSG-27178]”

这时在 tree view 中看不到此组件的相关 URL。

错误分析：所选择的录制脚本模式不正确，通常情况下，基于浏览器的 Web 应用会使用“HTML-based script”模式来录制脚本；而没有基于浏览器的 Web 应用、Web 应用中包含了与服务器进行交互的 Java Applet、基于浏览器的应用中包含了向服务器进行通信的 JavaScript/VBScript 代码、基于浏览器的应用中使用 HTTPS 安全协议，这时则使用“URL-based script”模式进行录制。

解决办法：打开录制选项配置对话框进行设置，在“Recording Options”的“Internet Protocol”选项里的“Recording”中选择“Recording Level”为“HTML-based script”，单击“HTML Advanced”，选择“Script. Type”为“A script. containing explicit”。然后再选择使用“URL-based script”模式来录制脚本。

11.1.71 LoadRunner HTTP 服务器状态代码都有哪些？如何解决？

在录制 Web 协议脚本回放脚本的过程中，会出现 HTTP 服务器状态代码，例如常见的页面-404 错误提示、-500 错误提示。

错误现象 1：-404 Not Found 服务器没有找到与请求 URI 相符的资源，但还可以继续运行直到结束。

错误分析：此处与请求 URI 相符的资源在录制脚本时已经被提交过一次，回放时不可再重复提交同样的资源，而需要更改提交资源的内容，每次回放一次脚本都要改变提交的数据，保证模拟实际环境，造成一定的负载压力。

解决办法：在出现错误的位置进行脚本关联，在必要时插入相应的函数。

错误现象 2：-500 Internal Server Error 服务器内部错误，脚本运行停止。

错误分析：服务器碰到了意外情况，使其无法继续回应请求。

解决办法：出现此错误是致命的，说明问题很严重，需要从问题的出现位置进行检查，此时需要此程序的开发人员配合来解决，而且产生的原因根据实际情况来定，测试人员无法单独解决问题，而且应该尽快解决，以便于后面的测试

11.1.72 HTTP 的超时有哪三种？

HTTP-request connect timeout、HTTP-request receive timeout、step download timeout

11.1.73 在什么地方设置 HTTP 页面 filter?

在 runtime_settings 中 download filter 里面进行设置。

11.1.74 如何设置可以让一个虚拟 IP 对应到一个 Vuser?

11.1.75 ≈什么是 contentcheck?如何来用？

ContentCheck 的设置是为了让 VuGen 检测何种页面为错误页面。如果被测的 Web 应用没有使用自定义的错误页面，那么这里不用作更改；如果被测的 Web 应用使用了自定义的错误页面，那么这里需要定义，以便让 VuGen 在运行过程中检测，服务器返回的页面是否包含预定义的字符串，进而判断该页面是否为错误页面。如果是，VuGen 就停止运行，指示运行失败。

使用方法：点击在 runtime settings 中点击“contentcheck”，然后新建立一个符合要求的应用程序和规则，设定需要查找的文本和前缀后缀即可使用。

11.1.76 network 中的 speed simulation 是模拟的什么带宽？

模拟用户访问速度的带宽。

11.1.77 生成 WEB 性能图有什么意义？大概描述即可。

可以很直观的看到，在负载下系统的运行情况以及各种资源的使用情况，可以对系统的性能瓶颈定位、性

能调优等起到想要的辅助作用。

11.1.78 WAN emulation 是模拟什么的？

可以很直观的看到，在负载下系统的运行情况以及各种资源的使用情况，可以对系统的性能瓶颈定位、性

能调优等起到想要的辅助作用。

11.1.79 树视图和脚本视图各有什么优点？

Tree View 的好处是使用户更方便地修改脚本，Tree View 支持拖拽，用户可以把任意一个节点拖拽到他想

要的地方，从而达到修改脚本的目的。用户可以右键单击节点，进行修改/删除当前函数参数属性，增加函

数等操作，通过 Tree View 能够增加 LoadRunner 提供的部分常用通用函数和协议相关函数。

Script View 适合一些高级用户，在 Script View 中能够看到一行行的 API 函数，通过 Script View 向脚本中增加一些其他 API 函数，对会编程的高手来说很方便

11.1.80 LR 中的 API 分为几类？

A：通用的ＡＰＩ：，就是跟具体的协议无关，在任何协议的脚本里都能用的；

B：针对协议的：像 lrs 前缀是 winsock 的；lrd 的是针对 database;

C：自定义的：这个范围就比较广了；

12 计算机网络

12.1 计算机网络基础

12.1.1什么是局域网和广域网

一、局域网

局域网（Local Area Network），简称 LAN，是指在某一区域内由多台计算机互联成的计算机组。“某一区域”指的是同一办公室、同一建筑物、同一公司和同一学校等，一般是方圆几千米以内。局域网可以实现文件管理、应用软件共享、打印机共享、扫描仪共享、工作组内的日程安排、电子邮件和传真通信服务等功能。

局域网是封闭型的，可以由办公室内的两台计算机组成，也可以由一个公司内的上千台计算机组成。

二、广域网

广域网（Wide Area Network），简称 WAN，是一种跨越大的、地域性的计算机网络的集合。通常跨越省、市，甚至一个国家。广域网包括大大小小不同的子网，子网可以是局域网，也可以是小型的广域网。

12.1.2DNS 是什么,它是如何工作的?

域名解析服务。用于将域名解析为 IP，或反和将 IP 解析为域名。客户机可指定 DNS 服务器来解析，或用本机 hosts 文件进行解析。Windows 下配置 DNS 服务器在《搭建 Windows 测试环境》中有。

12.1.3描述 TCP/IP 协议的层次结构，以及每一层中重要协议。

参考答案：（可以回答五层结构）

TCP/IP	协议
应用层/Application	HTTP、SMTP、FTP
传输层/Transport	TCP、UDP
网络层/Network	IP
链路层/Link	ARP、RARP

12.1.4请简述 ip 地址,网关,子网掩码的含义.

IP 地址是 TCP/IP 网络中的主机(或称为节 点)的惟一地址。IP 地址是网络层的逻辑地址

缺省网关(Default Gateway)是指缺省的路由器。只有在不同子网之间通信时，才需要配置缺省网关的 IP 地址

子网(Subnet)是在 TCP/IP 网络上，用路由器连接的网段，子网掩码(Subnet Mask)用来确定 IP 地址中的网络地址部分。其格式与 IP 地址相同，也是一组 32 位的二进制数。

12.1.5简述子网掩码的用途。

子网掩码主要用来判断两个 IP 地址是否处在同一个局域网当中；子网掩码是由连续的 2 进制 1 组成的。子网掩码和 IP 地址进行按位与运算后，结果一致，表示处于一个局域网当中，如果不一致，表示不再一个局域网当中，需要寻找路由。

12.1.6一台计算机的 IP 是 192.168.10.71 子网掩码 255.255.255.64 与 192.168.10.201 是同一局域网吗?

子网掩码不对。不可能出现 255.255.255.64 的子网掩码。 另外，也不能说成“同一局域网”，局域网是针对物理的拓扑结构而言。事实上，我们研究的是否在同一子网的一些 IP，往往都是同一个局域网内。

12.1.7请简述 DNS、活动目录、域的概念。

DNS：域名服务，作用是将网络域名解析成 IP 地址；

活动目录：微软提供的目录服务的一种，它存储有关网络上的对象信息，并使管理员和用户更方便的查找和使用这类信息；

域：网络系统的一个安全边界，在一个域当中，计算机和用户共享一些列的安全信息。

12.1.810M 兆宽带是什么意思？理论下载速度是多少？

首先我们要搞懂其中的区别，运营商说的 10M，完整的单位应该是 10Mbps（bps：比特率），而我们讲的下载速度单位是 MB，虽然都念兆，但是不一样的。

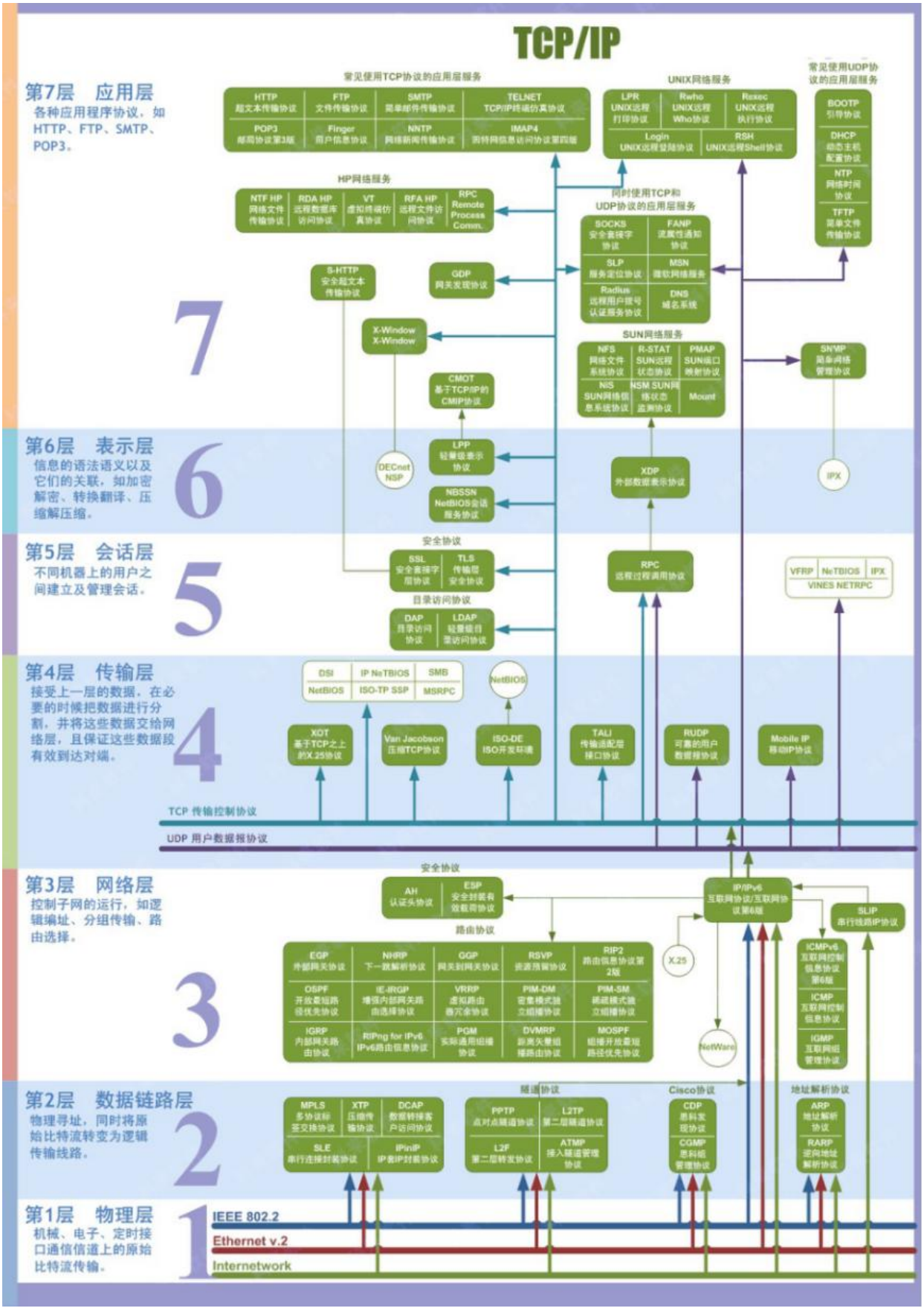
它们之间的换算关系是：1MB=8×1Mbps，换个方式看：1Mbps÷8=128KB，也就是说，运营商称的 10M 宽带，实际速度是 10Mbps÷8=1280KB，约 1.25MB。

12.1.9什么是 IP 地址？

IP 地址是指互联网协议地址（英语：Internet Protocol Address，又译为网际协议地址），是 IP Address 的缩写。IP 地址是 IP 协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异。

网络类别	最大网络数	IP地址范围	最大主机数	私有IP地址范围
A	126 (2^7-2)	1.0.0.0--126.255.255.255	2^24-2	10.0.0.0--10.255.255.255
B	16384(2 ^14)	128.0.0.0--191.255.255.255	2^16-2	172.16.0.0--172.31.255.255
C	2097152 (2^21)	192.0.0.0--223.255.255.255	2^8-2	192.168.0.0--192.168.255.255

12.1.10 OSI 七层网络模型的划分？


TCP/IP
第7层 应用层
各种应用程序协议，如HTTP、FTP、SMTP、POP3。
第6层 表示层
信息的语法语义以及它们的关联，如加密解密、转换翻译、压缩解压缩。
第5层 会话层
不同机器上的用户之间建立及管理会话。
第4层 传输层
接受上一层的数据，在必要的时候把数据进行分割，并将这些数据交给网络层，且保证这些数据段有效到达对端。
第3层 网络层
控制子网的运行，如逻辑编址、分组传输、路由选择。
第2层 数据链路层
物理寻址，同时将原始比特流转变为逻辑传输线路。
第1层 物理层
机械、电子、定时接口通信信道上的原始比特流传输。
常见使用TCP协议的应用层服务
HTTP
FTP
SMTP
TELNET
POP3
Finger
NNTP
IMAP4
UNIX网络服务
LPR
Rwho
Rexec
Login
RSH
常见使用UDP协议的应用层服务
BOOTP
DHCP
NTP
TFTP
HP网络服务
NTF HP
RDA HP
VT
RFA HP
RPC Remote Process Comm.
同时使用TCP和UDP协议的应用层服务
SOCKS
FANP
SLP
MSN
Radius
DNS
S-HTTP
GDP
X-Window
X-Window
SUN网络服务
NFS
R-STAT
PMAP
NIS
NSM SUN网
Mount
SNMP
CMOT
基于TCP/IP的CMIP协议
LPP
DECnet NSP
NBSSN
NetBIOS会话服务协议
XDP
IPX
安全协议
SSL
TLS
RPC
VFRP
NeTBIOS
IPX
VINES NETRPC
目录访问协议
DAP
LDAP
DSI
IP NeTBIOS
SMB
NetBIOS
ISO-TP SSP
MSRPC
NetBIOS
XOT
Van Jacobson
ISO-DE
TALI
RUDP
Mobile IP
TCP 传输控制协议
UDP 用户数据报协议
安全协议
AH
ESP
IP/IPv6
SLIP
路由协议
EGP
NHRP
GGP
RSVP
RIP2
X.25
OSPF
IGRP
VRRP
PIM-DM
PIM-SM
RIPng for IPv6
PGM
DVMRP
MOSPF
NetWare
ICMPv6
ICMP
IGMP
隧道协议
Cisco协议
地址解析协议
MPLS
XTP
DCAP
PPTP
L2TP
L2F
ATMP
CDP
CGMP
ARP
RARP
SLE
IPinIP
IEEE 802.2
Ethernet v.2
Internetwork

12.1.11 TCP 和 UDP 有什么不同？

TCP：

优点：可靠 稳定

TCP 的可靠体现在 TCP 在传输数据之前，会有三次握手来建立连接，而且在数据传递时，有确认. 窗口. 重传. 拥塞控制机制，在数据传完之后，还会断开来连接用来节约系统资源。

缺点：慢，效率低，占用系统资源高，易被攻击

在传递数据之前要先建立连接，这会消耗时间，而且在数据传递时，确认机制. 重传机制. 拥塞机制等都会消耗大量时间，而且要在每台设备上维护所有的传输连接。然而，每个连接都会占用系统的 CPU，内存等硬件资源。因为 TCP 有确认机制. 三次握手机制，这些也导致 TCP 容易被利用，实现 DOS. DDOS. CC 等攻击。

UDP：

优点：快，比 TCP 稍安全

UDP 没有 TCP 拥有的各种机制，是一种无状态的传输协议，所以传输数据非常快，没有 TCP 的这些机制，被攻击利用的机会就少一些，但是也无法避免被攻击。

缺点：不可靠，不稳定

因为没有 TCP 的这些机制，UDP 在传输数据时，如果网络质量不好，就会很容易丢包，造成数据的缺失。

适用场景：

TCP：当对网络质量有要求时，比如 HTTP，HTTPS，FTP 等传输文件的协议；POP，SMTP 等邮件传输的协议

UDP：对网络通讯质量要求不高时，要求网络通讯速度要快的场

景

12.1.12 HTTP 属于哪一层的协议？

HTTP 协议属于应用层协议

12.1.13 HTTP 和 HTTPS 的区别？

安全性上的区别:HTTPS：HTTP 协议的安全加强版，通过在 HTTP 上建立加密层，对传输数据进行加密。主要作用可以分为两种：一种是建立一个信息安全通道，来保证数据传输的安全；另一种就是确认网站的真实性。

表现形式：HTTPS 站点会在地址栏上显示一把绿色小锁，表明这是加密过的安全网站，如果采用了全球认证的顶级 EV SSL 证书的话，其地址栏会以绿色高亮显示，方便用户辨认。

SEO：在 2015 年之前百度是无法收录 HTTPS 页面的，不过自从 2015 年 5 月份百度搜索全站 HTTPS 加密后，就已经可以收录 HTTPS 了。谷歌则是从 2014 年起便开始收录 HTTPS 页面，并且 HTTPS 页面权重比 HTTP 页面更高。从 SEO 的角度来说，HTTPS 和 HTTP 区别不大，甚至 HTTPS 效果更好。

技术层面：如果要说 HTTPS 和 HTTP 的区别，最关键的还是在技术层面。比如 HTTP 标准端口是 80， 而 HTTPS 标准端口是 443；HTTP 无需证书，HTTPS 需要 CA 机构颁发的 SSL 证书；HTTP 工作于应用层， HTTPS 工作于传输层。

12.1.14 cookies 和 session 的区别？

cookies:是针对每一个网站的信息，每一个网站只对应一个，其它网站不能访问，这个文件是保存在客户端的，每次你打相应网站，浏览器会查找这个网站的 cookies，如果有就会将这个文件起发送出去。cookies 文件的内容大致包函这些信息如用户名，密码，设置等。

session： 是针对每一个用户的，只有客户机访问，程序就会为这个客户新增一个 session。session 里主要保存的是用户的登录信息，操作信息等。这个 session 在用户访问结束后会被自动消失（如果

超时也会）。

12.1.15 HTTP 的 get 请求和 post 请求的区别？

（1）在客户端，Get 方式在通过 URL 提交数据，数据在 URL 中可以看到；POST 方式，数据放置在 HTML HEADER 内提交。

（2）GET 方式提交的数据最多只能有 1024 字节，而 POST 则没有此限制。

（3）安全性问题。正如在（1）中提到，使用 Get 的时候，参数会显示在地址栏上，而 Post 不会。所以，如果这些数据是中文数据而且是非敏感数据，那么使用 get；如果用户输入的数据不是中文字符而且包含敏感数据，那么还是使用 post 为好。

（4）安全的和幂等的。所谓安全的意味着该操作用于获取信息而非修改信息。幂等的意味着对同一 URL 的多个请求应该返回同样的结果。

12.1.16 HTTP1.0 和 HTTP1.1 有什么区别

HTTP 协议老的标准是 HTTP/1.0，目前最通用的标准是 HTTP/1.1。

在同一个 tcp 的连接中可以传送多个 HTTP 请求和响应.

多个请求和响应可以重叠，多个请求和响应可以同时进行. 更加多的请求头和响应头(比如 HTTP1.0 没有 host 的字段).

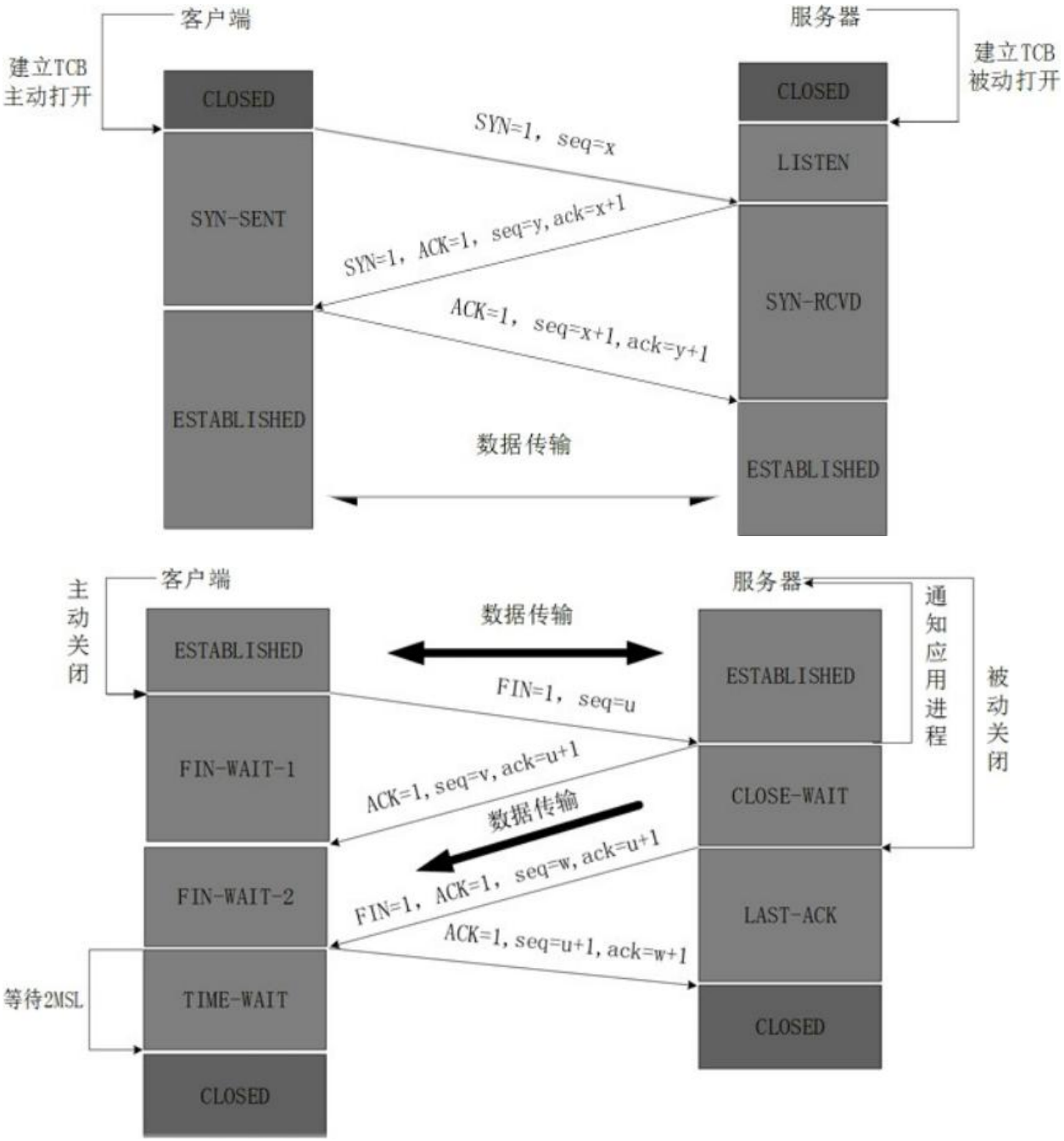
它们最大的区别：

在 HTTP/1.0 中,大多实现为每个请求/响应交换使用新的连接。HTTP 1.0 规定浏览器与服务器只保持短暂的连接，浏览器的每次请求都需要与服务器建立一个 TCP 连接，服务器完成请求处理后立即断开 TCP 连接，服务器不跟踪每个客户也不记录过去的请求。

在 HTTP/1.1 中,一个连接可用于一次或多次请求/响应交换,尽管连接可能由于各种原因被关闭。HTTP1.1 支持持久连接，在一个 TCP 连接上可以传送多个 HTTP 请求和响应，减少了建立和关闭连接的消耗和延迟。一个包含有许多图像的网页文件的多个请求和应答可以在一个连接中传输，但每个单

独的网页文件的请求和应答仍然需要使用各自的连接。HTTP 1.1 还允许客户端不用等待上一次请求结果返回，就可以发出下一次请求，但服务器端必须按照接收到客户端请求的先后顺序依次回送响应结果，以保证客户端能够区分出每次请求的响应内容，这样也显著地减少了整个下载过程所需要的时间。

12.1.17 TCP 的连接建立过程，以及断开过程？



12.1.18 客户端使用 DHCP 获取 IP 的过程？

发现阶段：即 DHCP 客户端寻找 DHCP 服务器的阶段。

提供阶段：即 DHCP 服务器提供 IP 地址的阶段。

选择阶段：即 DHCP 客户端选择某台 DHCP 服务器提供的 IP 地址的阶段。

确认阶段：即 DHCP 服务器确认所提供的 IP 地址的阶段。

12.1.19 写出某个网段的网络地址和广播地址？

算法：

子网掩码与 IP 地址进行位与运算，得处网络地址

网络地址 | (~子网掩码)，得出广播地址 （|：位或运算； ~：按位取反）

例如:IP 地址 10.145.129.20,掩码 255.255.248.0,网络地址和广播地址怎么计算？

网络地址 10.145.128.0 广播地址 10.145.135.255

解答：

IP 转换成二进制：00001010 10010001 10000001 00010010

掩码转换成二进制：11111111 11111111 11111000 00000000

IP 与掩码相与得网络地址（全 1 为 1，见 0 为 0）：00001010 10010001 10000000 00000000

网络地址转换成十进制为：10,145,128,0

看你的掩码把后 24 位的前 13 为划成了子网，后 11 为划成了主机，故：

广播地址则要把网络地址的主机位全换 1，得：

成

00001010,10010001,10000111,1111111

广播地址转换成十进制为：10,145,135,255

首先由 ip 地址结合子网掩码可以看出的是这是一个由 A 类地址，“借用”13 位的主机位而得到的子网，所以很轻易地得到 网络地址是：10.145.128.0，也即：00001010.10010001.10000 000.00000000（前 21（8+13）位是网络位，后 11 位是主机位）。至于广播地址，网络位+全为 1 的主机位，即得：00001010.10010001.10000111.11111111，10 进制表达方式就是 10.145.135.255

12.1.20 什么是 VPN 都有什么类型？

VPN 的隧道协议主要有三种，PPTP、L2TP 和 IPSec，其中 PPTP 和 L2TP 协议工作在 OSI 模型的第二层，又称为二层隧道协议；IPSec 是第三层隧道协议。

12.1.21 B/S 和 C/S 的区别

b/s 代表浏览器和服务器架构；c/s 代表客户端和服务器架构

网络环境不同（c/s 建立在专用的局域网上，b/s 建立在广域网上）

安全要求不同（c/s 必须安装客户端，安全度较高；b/s 安全度较低）

系统维护不同（c/s 升级困难，需要重新安装最新客户端；b/s 无缝升级）

对系统要求不同（c/s 对系统要求较高；b/s 对系统要求较低）

12.1.22 21、TCP/UDP 有哪些区别？

参考答案：TCP-有连接,所以握手过程会消耗资源,过程为可靠连接,不会丢失数据,适合大数据量交换

UDP-非可靠连接,会丢包,没有校验,速度快,无须握手过程

	TCP	UDP
是否连接	面向连接	面向非连接
传输可靠性	可靠的	不可靠的

应用场合	传输大量数据	少量数据
速度	慢	快

12.1.23 22、ISO 模型？HUB、tch、Router 是 ISO 的第几层设备？

参考答案：从底向上：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层

HUB：1 层（物理层）；Switch：2 层（数据链路层）；Router：3 层（网络层）

12.1.24 线程和进程的区别

进程——资源分配的最小单位，线程——程序执行的最小单位。线程进程的区别体现在几个方面：

第一：因为进程拥有独立的堆栈空间和数据段，所以每当启动一个新的进程必须分配给它独立的地址空间，建立众多的数据表来维护它的代码段、堆栈段和数据段，这对于多进程来说十分“奢侈”，系统开销比较大， 而线程不一样，线程拥有独立的堆栈空间，但是共享数据段，它们彼此之间使用相同的地址空间，共享大部分数据，比进程更节俭，开销比较小，切换速度也比进程快，效率高，但是正由于进程之间独立的特点，使得进程安全性比较高，也因为进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。一个线程死掉就等于整个进程死掉。

第二：体现在通信机制上面，正因为进程之间互不干扰，相互独立，进程的通信机制相对很复杂，譬如管道，信号，消息队列，共享内存，套接字等通信机制，而线程由于共享数据段所以通信机制很方便。。

属于同一个进程的所有线程共享该进程的所有资源，包括文件描述符。而不同过的进程相互独立。

线程又称为轻量级进程，进程有进程控制块，线程有线程控制块；

线程必定也只能属于一个进程，而进程可以拥有多个线程而且至少拥有一个线程；

第四：体现在程序结构上，举一个简明易懂的列子：当我们使用进程的时候，我们不自主的使用 if else 嵌套来判断 pid，使得程序结构繁琐，但是当我们使用线程的时候，基本上可以甩掉它，当然程序内部执行功能单元需要使用的时候还是要使用，所以线程对程序结构的改善有很大帮助。

12.1.25 常用的响应码

HTTP 响应码，也称 http 状态码(HTTP Status Code)，反映了 web 服务器处理 HTTP 请求状态，每一个响应码都代表了一种服务端反馈的响应状态，标识了本次请求是否成功。我们应该了解常见的响应码代表的状态，通过响应码能够对错误进行排查和定位，这是一个测试的必备技能~ HTTP 响应码通常分为五大类：

1XX——信息类（Information），表示收到 http 请求，正在进行下一步处理，通常是一种瞬间的响应状态

2XX——成功类（Successful），表示用户请求被正确接收、理解和处理

200（OK）：请求成功。一般用于 GET 与 POST 请求

201（Created）：已创建。成功请求并创建了新的资源 202（Accepted）：

3XX——重定向类（Redirection），表示没有请求成功，必须采取进一步的动作

301(Moved Permanently):资源被永久移动。请求的资源已被永久的移动到新 URI，返回信息会包括新的 URI，浏览器会自动定向到新 URI。今后任何新的请求都应使用新的 URI

302（Found）：资源临时移动。资源只是临时被移动，客户端应继续使用原有 URI

304：用其他策略获取资源

4XX——客户端错误（Client Error），表示客户端提交的请求包含语法错误或不能正确执行

400（Bad Requests）：客户端请求的地址不存在或者包含不支持的参数

401（Unauthorized）：未授权，或认证失败。对于需要登录的网页，服务器可能返回此响应

403（Forbidden）：没权限。服务器收到请求，但拒绝提供服务

404（Not Found）：请求的资源不存在。遇到 404 首先检查请求 url 是否正确

5XX——服务端错误（Server Error），表示服务器不能正确执行一个正确的请求（客户端请求的方法及参数是正确的，服务端不能正确执行，如网络超时、服务僵死，可以查看服务端日志再进一步解决）

500（Internal Server Error）：服务器内部错误，无法完成请求

503（Service Unavailable）：由于超载或系统维护（一般是访问人数过多），服务器无法处理客户端的请求 ，通常这只是暂时状态

12.1.26 手工修改 Tomcat 端口，在那个文件里？

Tomcat 服务器的 conf 目录下的主配置文件 server.xml

13 组成原理

13.1 组成原理

13.1.1 计算机基本组成

A.冯·诺依曼计算机的特点（机器以运算器为中心）

1.计算机由运算器、存储器、控制器、输入设备和输出设备五大部件组成

2.指令(程序)和数据以二进制不加区别地存储在存储器中

3.程序自动运行

B.现代计算机由三大部分组成（已经转化为以存储器为中心）

1.CPU(Central Processing Unit) 中央处理器，核心部件为 ALU(Arithmetic Logic Unit,算术逻辑单元)和 CU(Control Unit,控制单元)

2.I/O 设备(受 CU 控制)

3.主存储器(Main Memory,MM)，分为 RAM(随机存储器)和 ROM(只读存储器)

13.1.2 一条指令在 CPU 的执行过程

1.Ad(Address) 形式地址

2.DR(Data Register) 数据寄存器

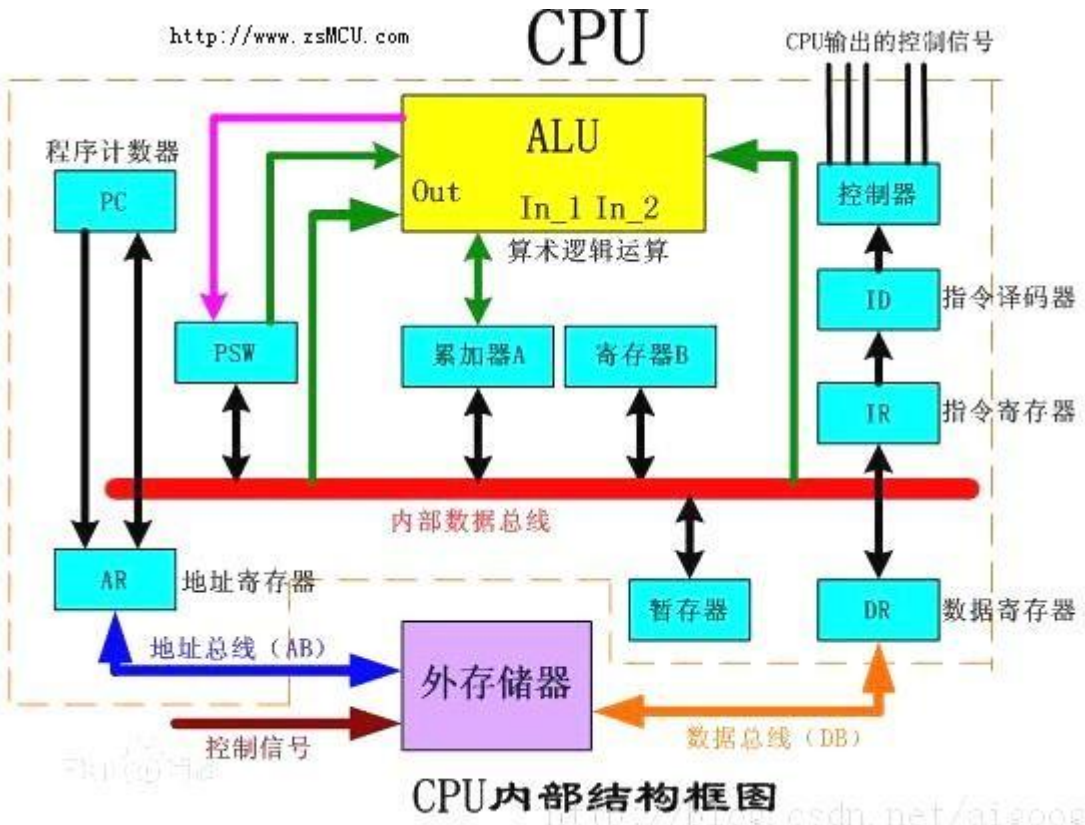
3.AR(Address Register) 地址寄存器(MAR)

4.IR(Instruction Register) 指令寄存器

5.BR(Buffer Register) 缓冲寄存器(MBR)

6.ID(Instruction Decoder) 指令译码器

7.PC(ProgramCounter) 程序计数器



过程详述：

几乎所有的冯·诺伊曼型计算机的 CPU，其工作都可以分为 5 个阶段：

取指令

指令译码

执行指令

访存取数

结果写回

1.取指令阶段

取指令（Instruction Fetch，IF）阶段是将一条指令从主存中取到指令寄存器的过程。程序计数器 PC 中的数值，用来指示当前指令在主存中的位置。当一条指令被取出后，

PC 中的数值将根据指令字长度而自动递增：若为单字长指令，则(PC)+1àPC；若为双字长指令，则(PC)+2àPC，依此类推。

//PC -> AR -> Memory

//Memory -> IR

2.指令译码阶段

取出指令后，计算机立即进入指令译码（Instruction Decode，ID）阶段。

在指令译码阶段，指令译码器按照预定的指令格式，对取回的指令进行拆分和解释，识别区分出不同的指令类别以及各种获取操作数的方法。

在组合逻辑控制的计算机中，指令译码器对不同的指令操作码产生不同的控制电位，以形成不同的微操作序列；在微程序控制的计算机中，指令译码器用指令操作码来找到执行该指令的微程序的入口， 并从此入口开始执行。

// { 1.Ad

// Memory -> IR -> ID -> { 2.PC 变化

// { 3.CU（Control Unit）

3.访存取数阶段

根据指令需要，有可能要访问主存，读取操作数，这样就进入了访存取数（Memory，MEM）阶段。此阶段的任务是：根据指令地址码，得到操作数在主存中的地址，并从主存中读取该操作数用于运算。

//Ad -> AR -> AD -> Memory

4.执行指令阶段

在取指令和指令译码阶段之后，接着进入执行指令（Execute，EX）阶段。

此阶段的任务是完成指令所规定的各种操作，具体实现指令的功能。为此，CPU 的不同部分被连接起来，以执行所需的操作。

例如，如果要求完成一个加法运算，算术逻辑单元 ALU 将被连接到一组输入和一组输出，输入端提供需要相加的数值，输出端将含有最后的运算结果。

//Memory -> DR -> ALU

5.结果写回阶段

作为最后一个阶段，结果写回（Writeback，WB）阶段把执行指令阶段的运行结果数据“写回”到某种存储形式：结果数据经常被写到 CPU 的内部寄存器中，以便被后续的指令快速地存取；在有些情况下， 结果数据也可被写入相对较慢、但较廉价且容量较大的主存。许多指令还会改变程序状态字寄存器中标志位的状态，这些标志位标识着不同的操作结果，可被用来影响程序的动作。

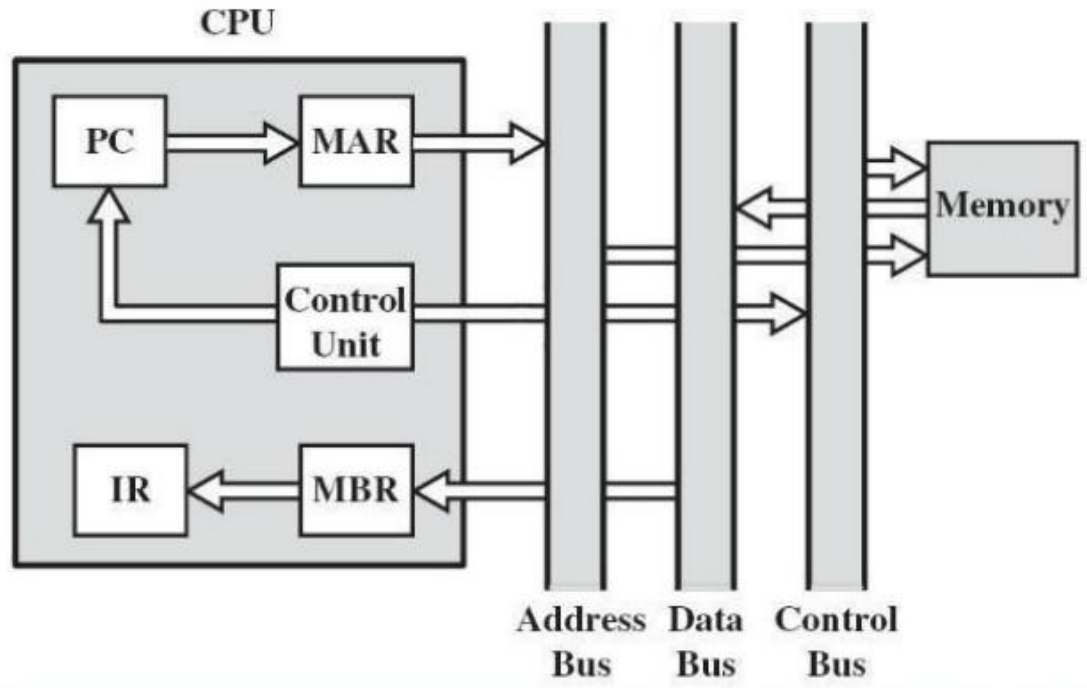
//DR -> Memory

6.循环阶段

在指令执行完毕、结果数据写回之后，若无意外事件（如结果溢出等）发生，计算机就接着从程序计数器 PC 中取得下一条指令地址，开始新一轮的循环，下一个指令周期将顺序取出下一条指令。

//重复 1~5

//遇 hlt(holt on)停止



指令的执行过程

13.1.33.计算机的逻辑部件

1.若逻辑电路的输出状态仅和当时的输入状态有关，而与过去的输入状态无关，则称这种逻辑电路为组合逻辑电路

常见的组合逻辑电路有三态门、异或门、加法器、算术逻辑单元、译码器

2. 若逻辑电路的输出状态不但与当时的输入状态有关,而且还与电路在此前的输入状态有关，则称这种逻辑电路为时序逻辑电路。时序电路内必须有能存储信息的记忆元件即触发器。触发器是构成时序电路的基础。

3. “阵列”是逻辑元件在硅芯片上以阵列形式排列。常见阵列逻辑电路有:只读存储器 ROM、可编程序逻辑阵列 PLA、可编程序阵列逻辑 PAL、通用阵列逻辑 GAL 等

13.1.4说出 4 种以上常用的操作系统及其主要的应用范围（微软的操作系统除外）。

参考答案：

Linux（Red Hat、SUSE、Debian、Trubo Linux）：主要用于搭建各类服务器

MAC OS：苹果机的用于图像处理

Unix（AIX：IBM 服务器的专用操作系统；

Solaris：Sun 操作系统；FreeBSD、NetBSD）

操作系统，

13.1.5Windows 操作系统中 PATH 环境变量的作用是什么？

PATH 是 Windows 操作系统环境变量，PATH 作用是用户在命令行窗口执行一个命令，则在 PATH 变量设置的目录下依次寻找该命令或对应的执行文件，若找到，则执行，若没有找到，则命令行窗口返回无效命令。

13.1.6目前流行的操作的系统有哪些？请举例说明安装操作系统的注意事项？

MS Windows 系列：win 98、windows 2000 系列、win XP、win 2003 Server、win Vista 等等。

UNIX 类：SVRx、FreeBSD、OpenBSD、NetBSD、Solaris、各种 Linux 等等。Mac OS……

多重引导时，一般先安装 win 操作系统，从低版本到高，再安装 Linux

14 数据结构与算法

14.1 数据结构与算法

14.1.1冒泡排序

冒泡思想：通过无序区中相邻记录的关键字间的比较和位置的交换，使关键字最小的记录像气泡一样逐渐向上漂至水面。整个算法是从最下面的记录开始，对每两个相邻的关键字进行比较，把关键字较小的记录放到关键字较大的记录的上面，经过一趟排序后，关键字最小的记录到达最上面，接着再在剩下的记录中找关键字次小的记录，把它放在第二个位置上，依次类推，一直到所有记录有序为止

```
def bsort(alist):
    l = len(alist)
    for i in range(l-1):
        flag = 1
        for j in range(l-i-1):
            if alist[j] > alist[j+1]:
                alist[j],alist[j+1] = (alist[j+1],alist[j])
                flag = 0
        if flag == 1:
            return alist
```

14.1.2插入排序

插入排序的基本操作就是将一个数据插入到已经排好序的有序数据中，从而得到一个新的、个数加一的有序数据，算法适用于少量数据的排序，时间复杂度为 O(n^2)。是稳定的排序方法。插入算法把要排序的数组分成两部分：第一部分包含了这个数组的所有元素，但将最后一个元素除外（让数组多一个空间才有插入的位置），而第二部分就只包含这一个元素（即待插入元素）。在第一部分排序完成后，再将这个最后元素插入到已排好序的第一部分中。

```
def insert_sort(lists):  # 插入排序
    count = len(lists)
    for i in range(1, count):
        key = lists[i]
        j = i - 1
        while j >= 0:
            if lists[j] > key:
                lists[j + 1] = lists[j]
                lists[j] = key
        j -= 1

    return lists
```

14.1.3希尔排序

希尔排序(Shell Sort)是插入排序的一种。也称缩小增量排序，是直接插入排序算法的一种更高效的改进版本。希尔排序是非稳定排序算法。该方法因 DL.Shell 于 1959 年提出而得名。 希尔排序是把记录按下标的一定增量分组，对每组使用直接插入排序算法排序；随着增量逐渐减少，每组包含的关键词越来越多，当增量减至 1 时，整个文件恰被分成一组，算法便终止。

```
def shell_sort(lists):
    # 希尔排序
    count = len(lists)
    step = 2
    group = count / step
    while group > 0:
        for i in range(0, group):
            j = i + group
```

```
        while j < count:
            k = j - group
            key = lists[j]
            while k >= 0:
                if lists[k] > key:
                    lists[k + group] = lists[k]
                    lists[k] = key
                k -= group
            j += group
        group /= step


    return lists
```

14.1.4直接选择排序

基本思想：第 1 趟，在待排序记录 r1 ~ r[n]中选出最小的记录，将它与 r1 交换；第 2 趟，在待排序记录 r2 ~ r[n]中选出最小的记录，将它与 r2 交换；以此类推，第 i 趟在待排序记录 r[i] ~ r[n]中选出最小的记录，将它与 r[i]交换，使有序序列不断增长直到全部排序完毕。

```
def select_sort(lists):


# 选择排序
    count = len(lists)
    for i in range(0, count):
        min = i
        for j in range(i + 1, count):
            if lists[min] > lists[j]:
```

```
            min = j
        lists[min], lists[i] = lists[i], lists[min]


    return lists
```

14.1.5堆排序

堆排序(Heapsort)是指利用堆积树（堆）这种数据结构所设计的一种排序算法，它是选择排序的一种。可以利用数组的特点快速定位指定索引的元素。堆分为大根堆和小根堆，是完全二叉树。大根堆的要求是每个节点的值都不大于其父节点的值，即 A[PARENT[i]] >= A[i]。在数组的非降序排序中，需要使用的就是大根堆， 因为根据大根堆的要求可知，最大的值一定在堆顶。

```
def adjust_heap(lists, i, size):
    lchild = 2 * i + 1
    rchild = 2 * i + 2
    max = i
    if i < size / 2:
        if lchild < size and lists[lchild] > lists[max]:
            max = lchild
        if rchild < size and lists[rchild] > lists[max]:
            max = rchild
        if max != i:
            lists[max], lists[i] = lists[i], lists[max]
            adjust_heap(lists, max, size)



def build_heap(lists, size):
    for i in range(0, (size / 2))[::-1]:
```

```
        adjust_heap(lists, i, size)


def heap_sort(lists):
    size = len(lists)
    build_heap(lists, size)


    for i in range(0, size)[::-1]:
        lists[0], lists[i] = lists[i], lists[0]
        adjust_heap(lists, 0, i)
```

14.1.6归并排序

归并排序是建立在归并操作上的一种有效的排序算法,该算法是采用分治法（Divide and Conquer）的一个非常典型的应用。将已有序的子序列合并，得到完全有序的序列；即先使每个子序列有序，再使子序列段间有序。若将两个有序表合并成一个有序表，称为二路归并。

归并过程为：比较 a[i]和 a[j]的大小，若 a[i]≤a[j]，则将第一个有序表中的元素 a[i]复制到 r[k]中，并令 i 和 k 分别加上 1；否则将第二个有序表中的元素 a[j]复制到 r[k]中，并令 j 和 k 分别加上 1，如此循环下去，直到其中一个有序表取完，然后再将另一个有序表中剩余的元素复制到 r 中从下标 k 到下标 t 的单元。归并排序的算法我们通常用递归实现，先把待排序区间[s,t]以中点二分，接着把左边子区间排序，再把右边子区间排序， 最后把左区间和右区间用一次归并操作合并成有序的区间[s,t]。

```
def merge(left, right):
    i, j = 0, 0
    result = []
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
```

```
            i += 1
        else:

            result.append(right[j])
            j += 1
    result += left[i:]
    result += right[j:]
    return result


def merge_sort(lists):
    # 归并排序
    if len(lists) <= 1:
        return lists

    num = len(lists) / 2
    left = merge_sort(lists[:num])
    right = merge_sort(lists[num:])
    return merge(left, right)
```

14.1.7基数排序

基数排序（radix sort）属于“分配式排序”（distribution sort），又称“桶子法”（bucket sort）或 bin sort， 顾名思义，它是透过键值的部份资讯，将要排序的元素分配至某些“桶”中，藉以达到排序的作用，基数排序法是属于稳定性的排序，其时间复杂度为 O (nlog(rm，其中 r 为所采取的基数，而 m 为堆数，在某些时候， 基数排序法的效率高于其它的稳定性排序法。

```
import math



def radix_sort(lists, radix=10):
    k = int(math.ceil(math.log(max(lists), radix)))
    bucket = [[] for i in range(radix)]
    for i in range(1, k + 1):
        for j in lists:
            bucket[j / (radix ** (i - 1)) % (radix ** i)].append(j)
        del lists[:]
        for z in bucket:
            lists += z
            del z[:]
    return lists
```

15 逻辑题

15.1 逻辑题

15.1.1烧一根不均匀的绳，从头烧到尾总共需要 1 个小时。现在有若干条材质相同的绳子,问如何用烧绳的方法来计时一个小时十五分钟呢？

第一步：A 绳从两头烧，同时 B 绳只烧一头。30 分钟后，A 烧完了。

第二步：A 烧完，同时 B 绳另一头也点燃，开始两头烧，烧完是 15 分钟。

第三步：再取一根 C 绳从两头烧，烧完 30 分钟。

三步加起来就是 1 小时 15 分钟。

15.1.2你有一桶果冻，其中有黄色、绿色、红色三种，闭上眼睛抓取同种颜色的两个。抓取多少个就可以确定你肯定有两个同一颜色的果冻？

根据抽屉原理，4 个（只有三个抽屉，最多第四个有重合）

1、第一次就抓取了两个一样颜色.

2、第一次抓取的两个颜色不同.那就再抓两个,要么这两个相同,要么有至少一个与第一次有相同。

15.1.3如果你有无穷多的水，一个 3 公升的提捅，一个 5 公升的提捅，两只提捅形状上下都不均匀，问你如何才能准确称出 4 公升的水？

用 5 升桶满桶，倒入 3 升桶中，倒满后大桶里剩 2 升。

把 3 升桶倒空，把那 2 升倒入 3 升桶中。

用 5 升桶满桶再向 3 升里倒，倒入一升就满，大桶里剩下的是 4 升。

15.1.4一个岔路口分别通向诚实国和说谎国。来了两个人，已知一个是诚实国的，另一个是说谎国的。诚实国永远说实话，说谎国永远说谎话。现在你要去说谎国，但不知道应该走哪条路，需要问这两个人。请问应该怎

么问？

问其中一人：另外一个人会说哪一条路是通往诚实国的？回答者所指的那条路必然是通往说谎国的。

15.1.512 个球一个天平，现知道只有一个和其它的重量不同，问怎样称才能用三次就找到那个球呢？（注意此题并未说明那个球的重量是轻是重，所以需要仔细考虑）

12 个球：第一次：4，4 如果平了：那么剩下的球中取 3 放左边,取 3 个好球放右边，称：如果左边重 ，那么取两个球称一下，哪个重哪个是次品，平的话第三个重，是次品，轻的话同理如果 平了，那么剩下一个次品，还可根据需要称出次品比正品轻或者重。如果不平：那么不妨 设左边重右边轻，为了便于说明，将左边 4 颗称为重球，右边 4 颗称为轻球，剩下 4 颗称为好球取重球 2 颗，轻球 2 颗放在左侧，右侧放 3 颗好球和一颗轻球。如果左边重，称那两颗重球，重的一个次品，平的话右边轻球次品。如果右边重，称左边两颗轻球，轻的一个次品。如果平，称剩下两颗重球，重的一个次品，平的话剩下那颗轻球次品。13 个球：第一次：4，4，如果平了 剩 5 颗球用上面的方法仍旧能找出次品，只是不能知 次品是重是轻，如果不平，同上。

15.1.6在一天的 24 小时之中，时钟的时针、分针和秒针完全重合在一起的时候有几次？都分别是什么时间？你怎样算出来的？

23 次，因为分针要转 24 圈，时针才能转 1 圈，而分针和时针重合两次之间的间隔显然>1 小时，它们有 23 次重合机会，每次重合中秒针有一次重合机会，所以是 23 次重合时间可以对照手表求出，也可列方程求出。

15.1.7已知：每个飞机只有一个油箱，飞机之间可以相互加油（注意是相互，没有加油机）一箱油可供一架飞机绕地球飞半圈，问题：为使至少一架飞机绕地球一圈回到起飞时的飞机场，至少需要出动几架飞机？（所有

飞机从同一机场起飞，而且必须安全返回机场，不允许中途降落，中间没有飞机场）

3 架飞机 5 架次，飞法：ABC 3 架同时起飞，1/8 处，C 给 AB 加满油，C 返航，1/4 处，B 给 A 加满油，B 返航，A 到达 1/2 处，C 从机场往另一方向起飞，3/4 处，C 同已经空油箱的 A 平分剩余油量，同时 B 从机场起飞，AC 到 7/8 处同 B 平分剩余油量，刚好 3 架飞机同时返航。所以是 3 架飞机 5 架次。

15.1.8一间囚房里面关押着两个犯人。每天监狱都会为这间囚房提供一罐汤，让这两个犯人自己分。起初，这两个人经常会发生争执，因为他们总是有人认为对方的汤比自己的多。后来他们找到了一个两全其美的办法：一个人分汤，让另一个人先选。于是争端就这么解决了。可是，现在这间囚房里又加进来一个新犯人，现在是三个人来分汤。必须寻找一个新的方法来维持他们之间的和平。该怎么办呢？按：心理问题，不是逻辑问题

先让甲分汤，分好后由乙和丙按任意顺序给自己挑汤，剩余一碗留给甲。这样乙和丙两人的总和肯定是他们两人可拿到的最大。然后将他们两人的汤混合之后再按两人的方法再次分汤。

15.1.9一张长方形的桌面上放 n 个一样大小的圆形硬币。这些硬币中可能有一些不完全在桌面内，也可能有一些彼此重叠；当再多放一个硬币而它的圆心在桌面内时，新放的硬币便必定与原先某些硬币重叠。请证明整个桌面可以用 4n 个硬币完全覆盖。

要想让新放的硬币不与原先的硬币重叠，两个硬币的圆心距必须大于直径。也就是说，对于桌面上任意一点，到最近的圆心的距离都小于 2，所以，整个桌面可以用 n 个半径为 2 的硬币覆盖。

把桌面和硬币的尺度都缩小一倍，那么，长、宽各是原桌面一半的小桌面，就可以用 n 个半径为 1 的硬币覆盖。那么，把原来的桌子分割成相等的 4 块小桌子，那么每块小桌子都可以用 n 个半径为 1 的硬币覆盖，因此，整个桌面就可以用 4n 个半径为 1 的硬币覆盖。

15.1.10 有五间房屋排成一列 所有房屋的外表颜色都不一样 所有的屋主来自不同的国家 所有的屋主都养不同的宠物；喝不同的饮料；抽不同的香烟 提示：

- 英国人住在红色房屋里
- 瑞典人养了一只狗
- 丹麦人喝茶
- 绿色的房子在白色的房子的左边
- 绿色房屋的屋主喝咖啡
- 抽 pall mall 香烟的屋主养鸟
- 黄色屋主抽 dunhill 位于最中间的屋主喝牛奶
- 挪威人住在第一间房屋里
- 抽 blend 的人住在养猫人家的隔壁
- 养马的屋主在抽 dunhill 的人家的隔壁
- 抽 blue master 的屋主喝啤酒
- 德国人抽 prince 挪威人住在蓝色房子隔壁
- 只喝开水的人家住在抽 blend 的隔壁

问：谁养鱼？

1.填写表格

2）过程：

（1）位于最中间的屋主喝牛奶：可以得出第三间房子的主人喝的饮料是牛奶。

（2）挪威人住在第一间房屋里：可以得出第一间房子的主人国籍是挪威人。

（3）挪威人住在蓝色房子隔壁：可以得出第二间房子的主人房子的颜色是蓝色。

（4）绿色的房子在白色的房子的左边；绿色房屋的屋主喝咖啡：由于绿色房子和白色房子是连在一起的，所以现在可以选择的房子颜色是 3、4、5 号，绿色房子和白色房子在这三间房子里面；而绿色房子在白色房子的左边，因此，若 3 号是绿色，4 号就是白色，若 4 号是绿色，5 号是白色，由于绿色房子的屋主喝咖啡，因此，绿色房子不可能是 3 号，因此，4 号是绿色，5 号是白色；第四间房子的主人喝的饮料是咖啡。

（5）英国人住在红色房屋里：1 号房子是挪威人，因此 1 号排除，2、4、5 号房子均有颜色，因此，3 号房子是红色的，国籍是英国人。

（6）黄色屋主抽 Dunhill：剩余的 1 号房子的颜色是黄色，房主抽的是 Dunhill。

（7）养马的屋主在抽 Dunhill 的人家的隔壁：抽 Dunhill 是 1 号，因此 2 号养马。

（8）抽 Blue Master 的屋主喝啤酒：现在饮料和香烟都没有确定的是 2 号和 5 号；假设：若 5 号是，5 号房子主人和啤酒，抽 Blue Master。

（9）在（8）假设成立的前提下，丹麦人喝茶：国籍和饮料都没有确定的只有 2 号，因此，2 号房主的国籍是丹麦人，喝的是茶。

（10）在（8）假设成立的前提下，德国人抽 Prince：国籍没有定的是 4 号和 5 号，而 5 号抽 Blue Master，因此，4 号房主是德国人，抽 Prince。

（11）在（8）假设成立的前提下，瑞典人养了一只狗：只剩下 5 号，因此，5 号房主国籍是瑞典人，养狗。

（12）在（8）假设成立的前提下，抽 Pall Mall 香烟的屋主养鸟：香烟和宠物都没有确定的只有 3 号，因此，3 号房主抽 Pall Mall，养鸟。

（13）在（8）假设成立的前提下，抽 Blend 的人住在养猫人家的隔壁：只剩下 2 号，因此，2 号房主抽 Blend，1 号房主养猫。

（14）在（8）假设成立的前提下，只喝开水的人家住在抽 Blend 的隔壁：只剩下 1 号，1 号房主喝的饮料是开水。

（15）最后剩一个就是养鱼。

（16）因此，（8）的假设成立。

15.1.11 如果 29 只青蛙在 29 分钟里捕捉到了 29 只苍蝇，那么，要在 87 分

钟内捉到 87 只苍蝇，需要多少只青蛙才行？

平均 1 只青蛙抓一只苍蝇要 29 分钟，那么 87 分钟内每只青蛙可以抓 3 只苍蝇，87 只苍蝇只需要 87/3=29 只青蛙

15.1.12 一个人花 8 块钱买了一只鸡，9 块钱卖掉了。然后他觉得不划算，花 10 块钱又买回来了，11 块钱卖给了另外一个人，请问他赚了多少钱？

2 元钱

15.1.13 A、B、C、D、E 五名学生有可能参加计算机竞赛，根据下列条件判断哪些人参加了竞赛？

1）A 参加时，B 也参加；

2）B 和 C 只有一个人参加；

3）C 和 D 或者都参加，或者都不参加；

4）D 和 E 中至少有一个人参加；

5）如果 E 参加，那么 A 和 D 也都参加。

根据条件有可能参加竞赛的人是：C 和 D。因为 C 和 D 参加符合条件（3），

由条件（2）知：B 不参加，

因为 B 不参加，

所以 由条件（1）知：A 不参加，

因为 A 不参加，D 参加，

所以 由条件（5）知：E 不参加。

所以 有可能参加的是：C 和 D。

15.1.14 一天张三的店里来了一位顾客，挑了 25 元的货。顾客拿出 100 元，

张三没零钱找不开，就到隔壁店里把这 100 元换成零钱，回来给顾客找了 75 元的零钱。过一会，李四回来找张三，说刚才的钱是假钱，张三马上给李四换了真钱，请问张三赔了多少？

75 块钱跟 25 元的货

15.1.15 如果 20 分钟前离上午 9 点钟的分数钟，等于现在离上午 12 点的分钟数的 3 倍，那么现在离上午 12 点还有多少分钟？）

（十一点 20 分）40 分钟

16 人力资源

16.1 人力资源

16.1.1你的测试职业发展是什么？你自认为做测试的优势在哪里？

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。

优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我有信心在工作中得以发挥。

16.1.2你找工作时，最重要的考虑因素为何？

工作的性质和内容是否能让我发挥所长，并不断成长。

16.1.3为什么我们应该录取你？

您可以由我过去的工作表现所呈现的客观数据，明显地看出我全力以赴的工作态度。

16.1.4请谈谈你个人的最大特色。

我的坚持度很高，事情没有做到一个令人满意的结果，绝不罢手。

16.1.5一个测试工程师应具备那些素质和技能？

1、掌握基本的测试基础理论 ；

2、本着找出软件存在的问题的态度进行测试,即客观吧,不要以挑刺形象出现

3、可熟练阅读需求规格说明书等文档 ；

4、以用户的观点看待问题

5、有着强烈的质量意识 ；

6、细心和责任心 ；

7、良好的有效的沟通方式(与开发人员及客户)

8、具有以往的测试经验 ；能够及时准确地判断出高危险区在何处.

16.1.6还有问一下你是怎样保证软件质量的，也就是说你觉得怎样才能最大限度地保证软件质量？

测试并不能够最大限度的保证软件的质量，软件的高质量是开发和设计出来的，而不是测试出来的，它不仅要通过对软件开发流程的监控，使得软件开发的各个阶段都要按照指定的规程进行，通过对各个阶段产物的评审，QA 对流程的监控，对功能及配置的审计来达到开发的最优化。当然测试也是保证软件质量的一个重要方式，是软件质量保证工程的一个重要组成部分。

16.1.7为什么选择测试这行？

它是一个新兴的行业，有发展潜力，而且很锻炼人，需要掌握更多的技能，比做开发要更难

16.1.8为什么值得他们公司雇用？

帮助公司提高软件质量和测试部门的技术水平

16.1.9如果我雇用你，你能给部门带来什么贡献？

分享我的测试经验和测试技能，提高测试部门技术水平

16.1.10 如何从工作中看出你是个自动自觉的人

自动自觉范围太广

1. 工作成果
2. 工作质量

16.1.11 你的工作通常能在时限内完成吗.（我想问一下就是她问这个问题的动机是什么）

在有足够的资源和合理的工作量的情况下，完全可以按时完成，并能比一般人做的更好

16.1.12 通常你对于别人批评你会有什么样的反应

有错即改，无错勉之

16.1.13 如果明知这样做不对，你还会依主管的指过去做吗？

如果你接到一个客户抱怨的电话，你确知无法解决他的问题，你会怎么处理？

弄清楚客户为什么抱怨？是怎么样的问题？

如果是客服问题，提交客服部门解决

如果是质量问题，分析原因，下一版本改进

16.1.14 你在五年内的个人目标和职业目标分别是什么？

分析这个问题是用来了解你的计划能力的，通过这个问题，面试人同时还可以知道你的目标是否符合企业对你的安排。

错误回答我想在将来的某个时候考虑这个问题。如今企业的领导者更换频繁，我认为做太多的个人计划是荒谬可笑的，不是吗？

评论这种回答属于令人反感的一类。首先，当有人想了解你的目标时，"将来的某个时候"这种通俗说法并不奏效。其次，认为企业很脆弱，领导者更换频繁，这种说法毫无疑问会令人反感，而且也是不合理的。最后，认为做计划可笑，看不起这个问题，而且反问面试人，这些都注定了这样的求职者最终会失败。

正确回答从现在起的五年之内，我希望能够在一个很好的职位上待几年，而且最好有一次晋升，然后就期待着下一步。不管是向上提升，还是在企业内横向调动，对我个人来说，我希望找到一家企业——一家愿意做相互投入的企业——待上一段时间。

评论这个问题没有回答得过分具体（那样可能会产生漏洞），而且它表明你有雄心，并且思考过在企业中的成长方式。通过表达横向调动和向上提升的愿望，表明你是一个有灵活性的人。

16.1.15 你怎样做出自己的职业选择？

分析：面试人提出这个问题是为了了解求职者的动机，看看他（她）应聘这份工作是否有什么历史渊源，是否有职业规划，是不是仅仅在漫无目的地申请很多工作。

错误回答：我一直都想在企业界工作。自孩提时代起，我就梦想自己至少也要成为大企业的副总

裁。

评论：除了难以令人相信之外，这种回答还存在一个问题：它表明求职者会对副总裁以下的职位不感兴趣。

正确回答：在上大学四年级前的那个夏天，我决定集中精力在某一领域谋求发展。尽管我是学商业的，但是我不知道自己最终会从事哪一行业的工作。我花了一定的时间考虑自己的目标，想清楚了自己擅长做的事情以及想从工作中得到的东西，最后我得出了一个坚定的结论，那就是这个行业是最适合我的。

评论：这种回答表明，求职者认真地做过一些计划，缩小了自己的关注点，而且也认准了前进的方向。这种回答还表明，求职者理解个人职业规划的重要性，并且有能力做出认真的个人决策。

16.1.16 离职时候工资多少？

说比现在期望薪资少 500 元。

16.2 其他

16.2.1好的测试工程师应具备的素质？

沟通能力、移情能力、技术能力、自信心、外交能力、幽默感、很强的记忆力、 耐心、怀疑精神、自我督促、洞察力。

16.2.2软件测试给你带来什么样的快乐？

测试可以给我带来很多快乐,如果测试出一个项目缺少东西,我会很高兴,因为我对 自己的工作有了新的认识,也为公司做了效益;如果测试出一个项目没有问题,我也很高兴, 因为同事们都在努力,大家都希望为公司做贡献,这就是一个很强大的团队，这是一件多么 另人振奋的事情啊!

16.2.3为什么要在一个团队中开展测试工作？

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。软件测试在整个一个团队中占有非常重要的地位，具体来说就是测试是一个发现软件错误的过程，执行软件测试会以最少的人力和时间，系统的找到软件存在的缺陷和错误，建立起开发人员和使用者对软件的信心。

16.2.4你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？

测试从事过 web 测试，后台测试，客户端软件，进行功能测试，性能测试，编写测试工具，文档的管理等，比较擅长编写测试用例和进行功能测试。

16.2.5请介绍一下你的项目

从几个部分来说，先项目规模，包括项目代码规模，需求规模、用例规模，工作量，进度，质量和成本，然后是整体的测试流程，然后再是角色与职责，接下来是在项目中自己的特色，比如做得最好的是、遇到最大的困难时（如何解决）、最差的是，最后是心得体会。

16.2.6测试过程中，遇到阻塞时，该如何推进？

1.功能基本可以走通但是 bug 太多

因为如果是以此为理由，打回去给开发，理由并不完全站得住。一个是大家对 bug 多的标准不一致，我们说 bug 多，开发不一定认可。这个时候我们需要针对 bug 的情况进行一下分析：

（1）bug 集中，且可以跟其他模块切开

测试发现的 bug 是集中在整个功能的某一个模块中，该模块与整个功能的其他模块可分割，可以单独测试。如果是整个功能都基本正常，只有其中的一个模块有问题，那么可以先对其他模块进行测试，bug 较集中的模块提交给开发重新对代码进行排查，待重新提测后再进行测试，对整体测试时间无影响。

（2）bug 集中，但是跟其他模块关联性强

测试发现的 bug 是集中在整个功能的某一个模块中，该模块与整个功能的其他模块关系较密切，无法单独测试。对其他模块进行测试，只打回这一个模块给开发，待开发重新代码 review 后，确认影响范围重新测试。对整体测试时间有影响，但是影响不大，需要在确认测试范围的时候花费一些时间。

（3）bug 不聚类，多数 bug 都不是严重问题，关联性不强

bug 分散在整个功能的的各个模块，基本是因为整体代码质量不高引起的。但是 bug 都不是什么严重的问题，集中在 UI 显示等模块，这个时候测试需要全功能测试，待开发修复完 bug 后进行修复问题的二轮测试。增加二轮测试，对整体测试时间有影响。

（4）bug 不聚类，半数 bug 都是较严重的问题

bug 分散在整个功能的的各个模块，基本是因为整体代码质量不高引起的。针对这种情况只能是全部打回去给开发，整体代码 review 后重新提测。提测时间 delay，对整体测试时间有影响。

2.功能实现的与策略不一致

有些时候，开发提测，产品也经过验收通过，但是到测试手里一看，实现跟需求明显是有差异的，这个时候应该怎么办？我们能做的，一定是第一时间找产品确认“开发做成这样你知道吗？”，一般会有两种结果。

1) 产品认可开发的实现。

如果是这种情况，我们那就让产品发需求变更，我们按照变更后的需求对功能进行测试。这种处理方法对整体测试时间基本无影响，只需要增加一个新需求确认的环节。

2) 产品对开发的实现不从，一定要改回来。

这种情况那就没办法了，打回去重新按照需求实现，然后重新提测吧。提测时间 delay，对整体测试时间有影响。

3.出现崩溃等异常完全无法继续测试下去

这种情况没什么好讨论的，直接打回去，等开发修复完全后再重新测试。但是前面已经测试过的部分，需要跟开发确认，如果修改后无影响的，可以不必再次从头开始测试。

16.2.7你们以前测试的流程是怎样的？

测试计划－测试用例设计－测试执行－测试分析报告

16.2.8为什么选择测试这行？

它是一个新兴的行业，有发展潜力，而且很锻炼人，需要掌握更多的技能，比做开发要更难

16.2.9如果时间不够，无法进行充分的测试怎么办？

使用风险分析，确定测试的重点。

由于很少有机会对一个应用软件进行所有可能的测试 (包括所有可能的事件组合、所有的相关性、或者一切可能出错的东西)，对大多数软件开发项目来说，利用风险分析是适当的。这需要判断技能、常识、感觉和经验。如果有正当理由，也可采用正式的方法。需要考虑下列因素：

1) 对于该项目的用途而言，哪种功能最重要?

2) 哪种功能对用户最明显?

3) 哪种功能对安全影响最大?

4) 哪种功能对用户最有用?

5) 对客户来说，该应用软件的哪个部分最重要?

6) 在开发过程中，该应用软件的哪个部分可以最先测试?

7) 哪一部分代码最复杂，容易导致出现错误?

8) 哪一部分的应用程序是在急迫或在惊恐的情况下开发出来的?

9) 哪一部分程序与过去项目中引起问题的部分相类似/有关?

10) 哪一部分程序与过去项目中需要大量维护的部分相类似/有关?

11) 需求和设计的那些部分不清楚或不容易读?

12) 开发人员认为在应用软件中哪些部分是高风险的?

13) 哪些问题能造成最差的发行?

14) 哪些问题最能引起用户抱怨?

15) 哪些测试可以容易地覆盖多种功能?

16) 哪些测试在覆盖高风险部分的测试时使用时间最少?

16.2.10 你是否了解以往所工作的企业的软件测试过程？如果了解，请试述在这个过程中都有哪些工作要做？分别由哪些不同的角色来完成这些

工作？

软件测试部门配合系统分析人员软件需求分析讨论，并根据需求说明书制定《项目测试计划》，编写测试用例，建立测试环境。 软件测试人员负责软件开发部门的新产品测试及原有产品的升级测试，负责软件问题解决过程跟踪，负责软件开发文档开发工作的规范化及管理开发部门的产品文档，制作用户手册及操作手册，负责产品的上线测试，监督软件开发过程的执行，提高产品质量。 需求人员连同系统分析人员&测试人员开会讨论需求。系统分析人员写出需求分析说明， 并连同系统分析人员&测试人员&需求人员开会讨论可行性。系统分析人员写出详细设计说明书，程式人员编码，给出系统流程图。交与测试人员，测试人员给出 Bug 统计表。

16.2.11 你所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）

有功能测试，性能测试，可靠性测试，安全性测试，负载测试，压力测试，安装/卸载测试，启动/停止测试，兼容性测试，互连测试，文档测试，恢复测试，回归测试，可使用性测试，容量测试。功能测试只对软件的功能是否满足用户需求来做测试。性能测试需要和压力和负载测试联合起来。

16.2.12 你自认为测试的优势在哪里？

优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我有信心在工作中得以发挥。

16.2.13 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。你应该怎么做？

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，并由上级做出决定。

16.2.14 你是如何制定时间进度表的？

首先确定三个大的时间段 项目开始时间 项目结束时间 开发转系统测试时间，在根据测试各个阶段的工作量和项目资源制定计划、设计、执行、评估、验收阶段的时间。设计和执行的时间一般较多。

16.2.15 介绍一下整体项目流程

公司将测试分为了五个阶段 计划 设计 执行 验收和评估.。在测试计划阶段 我们主要工作是编写测试计划 对项目做一个整体的规划 其中进度的安排、人力物力的分配、总体测试策略的制定和风险的评估比较重要。设计阶段我们主要是编写测试策略和测试用例。 在执行阶段，当我们拿到开发转过来的版本以后，首先是安装测试环境，然后执行用例。发现缺陷我们会提交缺陷报告，然后交给开发人员进行修改，之后进行循环测试！至于说具体系统测试会循环多少轮，是根据项目实际情况和版本的质量来决定的。在评估阶段我们主要是编写测试总结报告，其中对测试过程的总结和版本质量的评价要体现在测试总结报告里面最后就是软件的验收，验收阶段我们会和客户一同进行产品的最后检查！

16.2.16 你是如何制定测试过程中的时间进度表的？

确定项目开始 截止和转测试的时间 然后根据我们的经验值 去合理划分这五个阶段的时间 在设计和执行阶段 比重比较大

16.2.17 测试工作进行到一半时，发现时间不够，你是如何处理的？

1) 先看加班 加人能不能解决 如果不行就找重点 提出优先级高的用例执行
2) 向上级申请加派测试员
3) 和客户协商，推迟产品的发布时间

16.2.18 怎样保证你所负责的模块通过了测试？

首先是了解用户的需求，设计好的测试用例，严格的进行用例的评审，认真的执行测试用例，对自己提交的 Bug 进行详细的描述。 反复测试增强测试的准确性，通过冒烟回归随机测试挖掘缺陷提高测试工作质量，把各个模块整体运行发现未曾出现的错误，完善测试用例。

16.2.19 软件测试人员和测试组长的职责分工

普通测试人员：

创作相关的测试计划和测试案例

识别可自动测试的区域

参与组内的测试计划和测试案例以及测试脚本分析工作

手动或自动测试

按照需求规格说明查证并验证各项功能

发现并报告 bug，跟踪其状态

初步评估 bug 对产品其他部分的影响

测试组长：

确定测试的策略 评估 bug 对用户的影响

跟踪关键 bug 状态

管理测试工作和对象的资源

参与面试新人

交流状态和存在问题，并驱动问题的解决

促进组内的交流

16.2.20 如果你是测试组长你是如何对项目及组员进行管理的？

首先要充分了解要测试的项目，参考开发文档同时与开发人员及时沟通，要对产品十分的熟悉。员工方面多与员工沟通，了解员工的擅长的工作，根据员工擅长的工作进行分配，能力强的多分配，这样可以说测试工作快速稳定的进行！最终的测试工作就会一帆风顺！

16.2.21 什么时候开始搭建测试环境？由谁搭建？如何进行产品的集成？

测试开始之前搭建测试环境

由测试工程师搭建

产品的集成由开发人员完成

16.2.22 你所做的项目中采用了哪些测试方法？进行回归测试吗？

功能测试 界面测试 安装测试 性能测试

进行了回归测试 一般在缺陷修复之后进行验证的过程中进行回归测试

16.2.23 上级如何检查你的工作？

查看项目周报 开周例会

16.2.24 QA 是如何检查你的工作的？

检查项目过程及文档 参与周例会

16.2.25 在你所做的项目中有需要测试的项目过程吗？有，请介绍。

有啊，我们在奇瑞，华师，网校项目中都进行了测试。在测试过程中，我们先根据用户需求进行测试计划，编写测试用例，在开发部完成项目并进行产品集成，交由配置管理员纳入配置库之后，我们从配置库签出安装包，搭建测试环境，开始进行测试，测试过程中我们提交缺陷，开发人员修复缺陷，直到所有缺陷修改完毕，做测试总结。最终完成测试过程。

16.2.26 怎样保障你所负责的模块通过了测试？

仔细分析需求，制定测试计划与策略，详细编写测试用例因实际情况合理修改并执行，进过严格的评审，总结出真实测试报告。

首先在设计阶段要保证所设计的用例的覆盖率，发现缺陷的能力；在提交缺陷报告时要对缺陷进行详细确切的描述，方便开发人员进行修复；修复后认真进行回归测试，直到产品符合用户需求。

1.通过反复测试来增强测试的准确性

2.还可以通过冒烟、回归、随机测试来挖掘冒烟发现的缺陷，提高测试工作的质量

3.把各个模块整体运行时来发现未曾出现的错误

4.完善测试用例

16.2.27 你是如何了解到你说项目中的成员？

在项目立项公告中

16.2.28 是否成立了独立的测试组？测试人员在项目中测试的职责？

是。验证需求是否正确实现 发现软件中存在的缺陷 确认软件缺陷被修复。

16.2.29 测试结果分析如何？如何产生和被记录？

在项目测试之后，我们对缺陷进行了统计分析，并生成了测试报告文档。在此次项目中所有的缺陷都已修复并关闭。所有的缺陷都记录在缺陷管理工具中，并导出了缺陷报告

16.2.30 认为软件测试过程中较常见的困难是什么？如何有效克服这些困难？（根据自己实际测试中遇到的情况来写的）

①Bug 的重现问题：有些 Bug 只是偶尔出现的，根本就不知道具体需要什么条件才能重现 Bug.

解决方法：将不能重现的 Bug，利用截图的方式记录下来。并说明一系列的操作步骤

②Bug 的更新：旧的 Bug 修改好之后，很多时候会引发更多 Bug 的出现。解决方法：对更新的功能模块重点的测试之后，再重新测试和更新的功能密切的模块，会不会产生新的 Bug.

③与开发人员的沟通和对业务流程理解的分歧，经常缺少需求文档解决方法：根据需求说明书和 Bug 情况，多多和开发人员进行交流

16.2.31 在实际项目中你是如何做测试计划？

在做计划之前，我们要先了解这个项目的大致情况：

比如测试的是什么产品？

是新程序还是维护升级的？

是独立程序还是由多个小程序组成的？

产品的质量目标是什么？

产品的功能需求和性能指标必须得到所有人的一致认可。

为了深入了解项目，测试人员应该及早介入项目

对工作量的大小、时间点的安排作出了解。

16.2.32 你什么时候开始制定测试计划?是否发生过变更？如何进行变更？

在需求完成并得到确认之后开始制定测试计划

没有发生变更

变更流程：

项目经理向部门经理提交《计划变更申请单》。其中应当说明：变更原因、变更内容及变更可能对项目造成的影响和影响范围 。

由项目管理部组织对项目经理提交的《计划变更申请单》进行评审，评审人中应包括高层经理、CCB 成员，如果评审结论是不同意变更项目计划，则退回《计划变更申请单》，项目要按照原计划执行。如果评审结论是同意变更项目计划，则执行变更

修改项目计划

项目计划及其下属计划的评审和审批

工作产品纳入配置库

16.2.33 你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 2．边界值分析法 3．错误推测法 4．因果图方法

有黑盒和白盒两种测试种类，黑盒有等价类划分法，边界分析法，因果图法和错误猜测法。白盒有逻辑覆盖法，循环测试路径选择，基本路径测试。

例子：在一次输入多个条件的完整性查询中。利用等价类划分法则和边界分析法则，首先利用等价类划分法，可以一个或多个结果是 OK 的测试用例，然后确认多个 NG 的测试用例， 然后利用边界值分析法，可以对结果分别是 OK 和 NG 的测试用例进行扩展和补充。 请详述设计测试用例的方法？（只是列出一个测试用例思考的方向，具体设计靠经验）

①黑盒测试用例根据业务需求说明书来设计，分为：

等价划分法边界值分析法错误推测法因果图法逻辑覆盖法

②白盒测试用例通过研究代码与程序结构可以分为以下两种方式：

静态测试：通过静态的检查程序代码、界面、文档中可能存在的错误的过程。

|-测试代码编写的规范性 |-测试界面 |-测试相关需求说明和用户手册是否符合实际要求

动态测试：通过路径和分支测试。测试用例主要根据以下六种覆盖测试方法设计

|-语句覆盖 |-判定覆盖 |-条件覆盖 |-判定/条件覆盖 |-组合覆盖 |-路径覆盖

16.2.34 你认为做好测试用例设计工作的关键是什么？

1) 明确测试的目标，增强测试计划的实用性

2) 坚持“5W”规则，明确内容与过程，'what''why''when''where''how'

3) 采用评审和更新机制，保证测试计划满足实际需求

4) 分别创建测试计划与测试详细规格、测试用例

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

16.2.35 在你以往的工作中，一条软件缺陷（或者叫 Bug）记录都包括哪些内容？如何提交高质量的软件缺陷（Bug）记录？

检测时间，系统环境，硬体环境，严重程度，程式版本，确认人，功能模块，问题描述，详细操作步骤，是否会重现。

问题描述和详细操作步骤要尽可能的详细。Bug 应该尽量用书面语，对与严重程度比较高的缺陷要在相同环境下在测试一遍。

在 C/S 模式下，如果条件满足可以使用替换法来确认是 client 端的问题还是 server 端的问题。

16.2.36 你在五年内的个人目标和职业目标分别是什么？

从现在起的五年之内，我希望能够在一个很好的职位上待几年，而且最好有一次晋升，然后就期待着下一步。不管是向上提升，还是在企业内横向调动，对我个人来说，我希望找到一家企业——一家愿意做相互投入的企业——待上一段时间。

16.2.37 怎样做出自己的职业选择？

在上大学四年级前的那个夏天，我决定集中精力在某一领域谋求发展。尽管我是学商业的，但是我不知道自己最终会从事哪一行业的工作。我花了一定的时间考虑自己的目标，想清楚了自己擅长做的事情以及想从工作中得到的东西，最后我得出了一个坚定的结论，那就是这个行业是最适合我的。

16.2.38 离职原因

每次面试必问的问题大概就是离职原因，建议不要提到上家公司不好或者是领导不好这些比较消极的原因，可以给面试官一些无关痛痒的原因比如想找一个离家近的公司或者因为搬家了上一家公司离家太远不太方便等。不管你离职的真正原因是什么都要回答积极的方面。

16.2.39 面试官一般会问，您还有什么想问的吗？

一般分几种情况:

第一种是双方满意，先表示感谢，然后积极主动的提问，比如，确认是否有复试及时间，刚才面试中自己有哪些不足（表现你的上进心）

第二种情况是双方感觉一般般，自己感觉很 low,基本的套路是，先表示感谢，坦白地说对自己今天表现不是非常满意，还可以表现得更好，还想得到这个机会，询问面试官能否给一些建议（表现出你的学习意愿）

第三种情况是面试情况非常糟糕，这种情况下，也要先表示感谢。基本的思路是，分两种情况

1.面试官人不错技术很好，表达你的学习意愿和想进公司的意愿

2.面试官技术不太好，但是人很差，不尊重人，什么也不说，直接走人

当然，也可以参考以下问题，相信会给你更多灵感。