

## 软件测试面试题整理

### 01. 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。

### 02. 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？

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

### 03. 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 04. 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

### 04. 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

### 05. 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。

06. 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策略和测试方法（最好是能先评审）

07. 您认为做好测试计划工作的关键是什么？

1. 明确测试的目标，增强测试计划的实用性

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。

08. 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.

09. 请以您以往的实际工作为例, 10. 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

11. 您以往是否曾经从事过性能测试工作？如果有，12. 请尽可能的详细描述您以往的性能测试工作的完整过程。

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；

13. 您在从事性能测试工作时，14. 是否使用过一些测试工具？如果有，15. 请试述该工具的工作原理，16. 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。

17. 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？

18. 在您以往的工作中，19. 一条软件缺陷（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？

20. 您以往所从事的软件测试工作中，21. 是否使用了一些工具来进行软件缺陷（Bug）的管理？如果有，22. 请结合该工具描述软件缺陷（Bug）跟踪管理的流程。

23. 您认为在测试人员同 24. 开发人员的沟通过程中，25. 如何提高沟通的效率和改善沟通的效果？维持测试人员同 26. 开发团队中其他成员良好的人际关系的关键是什么？

27. 在您以往的测试工作中，28. 最让您感到不 29. 满意或者不 30. 堪回首的事情是什么？

您是如何来对待这些事情的？

31. 在即将完成这次笔试前，32. 您是否愿意谈一些自己在以往的学习和工作中获得的工作经验和心得体会？（可以包括软件测试、过程改进、软件开发或者与此无关的其他方面）

33. 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，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。

34. 你的测试职业发展是什么？

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，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 11, 12 点要求自己，不断的更新自己改正自己，做好测试任务。

35. 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。

36. 你以前工作时的测试流程是什么？

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）。

37. 当开发人员说不 38. 是 BUG 时， 39. 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。

23. 你为什么想离开目前的职务？

因为公司运作情况并不理想，公司需要调整部门体系，公司考虑到缩减部门人员，所以大批量的裁员（有 6, 7 个），这是我的第一份工作，对公司也有较深的感情，因为在这里我找到了职业理想（就是测试），所以公司需要精简人员，我自愿退出。虽然很舍不得，但我将会有新的发挥能力的舞台。

24: 你对我们公司了解有多少？

25: 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。

26: 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。

27: 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。

28.白箱测试和黑箱测试是什么?什么是回归测试?

29. 单元测试、集成测试、系统测试的侧重点是什么？

30. 设计用例的方法、依据有那些？

31. 一个测试工程师应具备那些素质和技能？

32.集成测试通常都有那些策略？

33.你用过的测试工具的主要功能、性能及其他？

34.一个缺陷测试报告的组成

35.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

36.软件测试项目从什么时候开始，？为什么？

37.需求测试注意事项有哪些？

38.简述一下缺陷的生命周期

39.测试分析测试用例注意（事项）？

你在你所在的公司是怎么开展测试工作的？是如何组织的？

你认为理想的测试流程是什么样子？

你是怎样工作的？

软件测试活动的生命周期是什么？

请画出软件测试活动的流程图？

针对缺陷采取怎样管理措施？

什么是测试评估？测试评估的范围是什么？

如果能够执行完美的黑盒测试，还需要进行白盒测试吗？为什么？

测试结束的标准是什么？

软件验收测试除了 alpha,beta 测试以外,还有哪一种？

做测试多久了？

以前做过哪些项目？

你们以前测试的流程是怎样的？

<答：测试计划—测试用例设计—测试执行—测试分析报告>

用过哪些测试工具？

为什么选择测试这行？

<答：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>

为什么值得他们公司雇用？

如果我雇用你，你能给部门带来什么贡献？

如何从工作中看出你是个自动自觉的人

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）

通常你对于别人批评你会有什么样的反应

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

你觉得什么样的人最难相处

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

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

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

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

通常你对于别人批评你会有什么样的反应

有错即改，无措勉之

如果明知这样做不对，你还会依主管的指过去做吗

在公司内部下级是否有申诉渠道？

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

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

你觉得什么样的人最难相处

自以为是的人

什么叫单元测试？

请就软件测试人员应该具备什么样的基本素质说说你的看法。

请就如何在开发中进行软件质量控制说说你的看法

简述软件测试的意义，以及软件测试的分类

1、功能测试，性能测试，界面测试，安全测试（可以简单点，比如只涉及到 COOKIES 里的内容），压力测试（商业性质的网站）等等，B/S 软件也要根据其具体功能采用不同的测试策略。

2、态度、责任心、自信、敏锐的观察力、良好的发散思维

3、先设计后开发模式，加强单元测试，加强代码走查，有一套完整的白盒测试方法。关键是加强开发人员的质量意识，增进程序员向工程师水平发展。

4、意义嘛，就自己想吧。软件测试的分类，这个很多人都按各种方法去分。无明确答案给你。

对测试的理解——基本的测试知识，对测试是否认可？ 75。

3、谈一谈过去自己的工作——了解经历、提供进一步提问的素材，表达能力  
测试技能

测试设计的方法并举例说明——测试技术的使用

测试工具——熟悉程度，能否与当前工作匹配？

如何做计划？如何跟踪计划？——日常工作能力

如果开发人员提供的版本不满足测试的条件，如何做？——与开发人员协作的能力

熟悉 unix 系统、oracle 数据库吗？——是否具备系统知识

做过开发吗？写过哪些代码？——开发技能  
阅读英语文章，给出理解说明？——部分英语能力  
文档的意义——是否善于思考？（最简单的概念，不同层次的理解）  
假如进入我们公司，对我们哪些方面会有帮助？——讲讲自己的特长  
随便找一件物品，让其测试——测试的实际操作能力  
软件测试的方法有？  
软件测试的过程？  
有一个新的软件，假如你是测试工程师，该如何做？

软件测试分哪两种方法？分别适合什么情况？

2. 一套完整的测试应该由哪些阶段组成？分别阐述一下各个阶段。
  3. 软件测试的类型有那些？分别比较这些不同的测试类型的区别与联系。
  4. 测试用例通常包括那些内容？着重阐述编制测试用例的具体做法
  5. 在分别测试 winform 的 C/S 结构与测试 WEB 结构的软件是，应该采取什么样的方法分别测试？他们存在什么样的区别与联系？
  6. 在测试 winform 的 C/S 结构软件时，发现这个软件的运行速度很慢，您会认为是什么原因？您会采取哪些方法去检查这个原因？
  7. 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
- 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，面试官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，而且反问面试官，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。

问题 23 你怎样做出自己的职业选择？

分析 面试官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，想清楚了自已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。

1. 你都用什么测试方法
2. 怎么编写案例
3. 怎么才能够全面的测试到每一个点

1. 你都用什么测试方法

针对不同的产品或者系统或者模块，有不同的测试方法。总体而言有白盒测试和黑盒测试。

2. 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。

3. 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、谈谈软件测试技术，以及如何提高

2、谈谈软件测试职业发展，以及个人的打算

3、谈谈软件测试在企业的地位，也可以结合软件生命周期来谈

有可能清晰的思路比确切的答案更重要

在这里，主要说下笔试和面试的问题，希望大家共同参考。

1，一般公司里实际的软件测试流程是什么样的？你们公司又是怎样的？

2，软件工程师要具有那些素质？

3，你会哪些测试工具？怎么操作？

4，你能不能说下你的3到5年的职业规划（规划）

5，你觉得你来应聘有那些优势？

其余的还好说，但就第4个问题，我感到不好说哦！希望大家给个意见

第一关：首先要自我介绍，自己的性格怎么样，目前的工作经历积累了一些什么经验取得了一些什么值得一说的成果。然后要说说对软件测试怎么看？还有对于软件测试有什么自己的想法。为什么会想到要做这行（因为我的简历上的工作经历没有关于测试方面的）。哦，还有期望薪资。

第二关：认为软件测试人员所要具备的基本素质，如果遇到问题会怎样处理，如果得不到研发人员的配合（就是研发说这个不是问题）你又会怎么处理？然后就是一些基本概念，比如软件测试的流程有哪些？如果我上任了，首先会怎么开始自己的工作计划。

（前两关通过了后面这个就好过多了）

第三关：像我介绍了一下公司的情况，告诉我主要针对什么内容的测试，会不会使用数据库。告诉我大概要做哪些内容，详细的可以上岗以后慢慢熟悉。

大概就这么多了，这对没有经过这一关的不知道有没有帮助，仅供参考吧

我觉得就像李波说的，关键是要给对方留下好印象:)

面试官最后会问你有什么问题要问吗。作为应聘者的你一般不要说没问题问，这会给面试官留下你不太重视这份工作的坏印象。所以如果你想得到这份工作的话应该抓住这最后的表现自己的机会：

你可以问：

1. 贵公司近期和远期的发展目标是什么？
2. 贵公司的主要竞争对手有哪些？

3. 贵公司有多少开发人员有多少测试人员？
4. 贵公司又进一步扩充测试人员的计划吗？
5. 如果我有幸能进入贵公司的话，我有怎么样的发展？
6. 测试人员的沟通能力很重要，贵公司有规范的沟通渠道吗？
7. 请介绍一下贵公司的福利情况。
8. 请问我什么时候能知道结果？

用友面试：

1. 内联接和外连接，自联接有什么区别？  
内联接通常是 2 个表存在主外键关系时使用的，  
内联接查询有 2 种方式实现，  
1 是在 WHERE 子句中指定联接条件  
2 是在 FROM 子句中使用 join...on  
内联接查询通常不仅仅联接 2 表，可以 3 表甚至更多的表  
参与内联接的表的地位是平等的

而外联接中参与联接的表有主从之分。以主表的每行数据去匹配从表的数据列，符合条件的数据将直接返回到结果集中，不符合的用 NULL（空值）填充后再返回到结果集中。

2. SQL 中 \group by\和\order by\有什么不同呢  
一个是对处理的数据进行分组，一个是对处理的数据进行排序

自动测试的好处：

如果你需要反复运行一组测试，那么自动测试将会对你非常有用。

自动测试使你能够应对频繁改变的代码从而跟上周期性回归测试的脚步。

自动测试可以使你能够自动运行主流业务场景从而跟上周期性回归测试的脚步。（原文：It gives you the ability to run automation in main stream scenarios to catch regressions in a timely manner）

自动测试可以帮助你测试大量测试矩阵（在不同操作系统上的不同语言）。自动测试可以使你的测试同时运行在不同的机器上，而手动测试必须不断地继续执行。

自动测试的限制：

花费大。编写测试用例，编写和配置自动化测试框架将会在测试开始时花费比手动测试更多的费用。

无法自动测试一些可视的场景。例如，如果你无法通过代码告诉自动测试工具字体颜色，那么只好使用手动测试。

手动测试的好处：

如果一个测试用例在编码阶段只运行两次，那最好使用手动测试，它将比自动测试花费少得多的费用。

手动测试允许测试员进行更多的随机测试。以我的经验来看，更多的 bug 将会由随机测试发现，而不是自动测试。并且，一个测试员花费越多的时间进行随机测试，发现真正的用户 bug 的几率就越大。

手动测试的限制：

手动进行测试将花费大量的时间。

每次有了新的 build，测试员必须重新运行测试-经过一段时间以后将会非常繁琐和疲惫。

其他的因素：

你将哪些部分进行自动测试也由你使用的工具决定。如果该工具有很多限制，那么这些部分还是手动测试吧。

是否投资的回报值得运行自动测试？是否你自动化测试的产出值得建立和支持测试用例，自动框架和运行测试用例的系统？

自动测试的标准

有两个问题可以用来判断是否应该为你的测试用例进行自动化。

Q1:是否测试场景可以自动化？

A1:是的，并且花费很少。

A2:是的，但是花费很多。

A3:不，不可能进行自动化。

Q2:该测试场景有多么重要？

A1:我必须任何可能的时候都对其进行测试。

A2:我需要有规律地对该场景进行测试。

A3:我只需要测试该场景一次。

如果这两个问题你的答案都是#1，那么你肯定需要自动化该测试。

如果这两个问题你的答案是一个#1 和一个#2，那么你最好自动化该测试。

如果这两个问题你的答案都是#2，那么你应该好好考虑一下是否你值得为自动化测试投资。

如果你无法自动测试，会有什么结果

让我们假设如果你有一个测试必须在任何可能的时间运行，但是却无法自动化它，你的选择是：

再评估 - 是否我真的需要如此频繁地运行它？

如果手动测试它会有多大的花费？

寻找新的测试工具。

考虑使用 test hooks.

## 四款主流测试工具的测试流程

### 主流测试工具的测试流程

#### =====winrunner

- 1 启动时选择要加载的插件
- 2 进行一些设置（如录制模式等）
- 3 识别应用程序的 GUI，即创建 map(就是学习被测试软件的界面)
- 4 建立测试脚本（录制及编写）
- 5 对脚本除错及调试（保证能够运行完）
- 6 插入各种检查点（图片，文字，控件等）
- 7 在新版应用程序中执行测试脚本
- 8 分析结果，回报缺陷

#### =====quicktestpro=====

- 1 准备录制

打开你要对其进行测试的应用程序，并检查 QuickTest 中的各项设置是否适合当前的要求。

## 2 进行录制

打开 QuickTest 的录制功能，按测试用例中的描述，操作被测试应用程序。

## 3 编辑测试脚本

通过加入检测点、参数化测试，以及添加分支、循环等控制语句，来增强测试脚本的功能，使将来的回归测试真正能够自动化。

## 4 调试脚本

调试脚本，检查脚本是否存在错误。

## 5 在回归测试中运行测试

在对应用程序的回归测试中，通过 QuickTest 回放对应用程序的操作，检验软件正确性，实现测试的自动化进行。

## 6 分析结果，报告问题

查看 QuickTest 记录的运行结果，记录问题，报告测试结果。

====TestDirect=====

安装好后，先进入站点管理

### 1 创建域及工程

### 2 添加用户

### 3 编辑 licenses 及本服务器

### 4 编辑数据库

--TD

### 1 选择新建的工程进行定制(列表，用户，组，版本等)

### 2 在 require 中增加需求

### 3 把需求转化为 plan

### 4 在 testlab 中由计划新建测试具体用例与执行

### 5 发现 bug，在 defect 中提交 bug

(每一部分都可以相对独立地使用)

=====loadrunner

### 1 制定负载测试计划

(分析应用程序，确定测试目标，计划怎样执行 LoadRunner)

### 2 开发测试脚本

(录制基本的用户脚本，完善测试脚本)

### 3 创建运行场景

(选择场景类型为 Manual Scenario，选择场景类型，理解各种类型，场景的类型转化)

### 4 运行测试

### 5 监视场景

(MEMORY 相关，PROCESSOR 相关，网络容量以及带宽，磁盘相关，WEB 应用程序，IIS5.0，SQL SERVER，NETWORK DELAY 等)

### 6 分析测试结果

(分析实时监视图表，分析事务的响应时间，分解页面，确定 WEBSERVER 的问题，其他有用的功能)

## 软件测试面试题

2007-02-28 17:17

### 软件测试的目的？

测试的目的是想以最少的人力、物力和时间找出软件中潜在的各种错误和缺陷，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。

**Beta 测试：**在客户场地，由客户进行的对产品预发布版本的测试。

软件验收测试合格通过准则：1 软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。2 所有测试项没有残余的一级二级三级的错误。3 立项审批表、需求分析文档、设计文档和编码实现一致。4 验收测试工件齐全（测试计划，测试用例，测试日志，测试通知单，测试分析报告）

软件验收测试包括正式验收测试、alpha 测试、beta 测试三种测试。

系统测试的策略：功能测试，性能测试，外部接口测试，界面测试，强度测试，冗余测试，可靠性测试，恢复测试等

设计系统测试计划需要参考的项目文档有软件测试计划、软件需求工件、和迭代计划。

利用因果图导出测试用例需要经过的一般步骤

- 1.分析程序规格说明的描述中，哪些是原因，哪些是结果。
- 2.分析程序规格说明的描述中语义的内容，并将其表示成连接各个原因与各个结果的因果图
- 3.在因果图上使用若干个特殊的符号标明特定的约束条件
- 4.把因果图转换成判定表
- 5.把判定表中每一列表示的情况写成测试用例

阶段评审与同行评审的区别

同行评审目的:发现小规模工作产品的错误,只要是找错误;

阶段评审目的:评审模块阶段作品的正确性可行性及完整性

同行评审人数:3-7 人人员必须经过同行评审会议的培训,由 SQA 指导

阶段评审人数:5 人左右评审人必须是专家具有系统评审资格

同行评审内容:内容小一般文档 < 40 页, 代码 < 500 行

阶段评审内容: 内容多,主要看重点

同行评审时间:一小部分工作产品完成

阶段评审时间: 通常是设置在关键路径的时间点上!

什么是软件测试？

使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。

软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码的最终复审，是软件质量保证的关键步骤。软件测试是为了发现错误而执行程序的过程。

简述集成测试的过程

根据 IEEE 标准 集成测试划分为 4 个阶段：计划阶段，设计阶段，实现阶段，执行阶段（实施阶段）

计划阶段

- 1) 时间安排 概要设计完成评审后大约一个星期
- 2) 输入 需求规格说明书 概要设计文档 产品开发计划路标
- 3) 入口条件 概要设计文档已经通过评审
- 4) 活动步骤 1.定被测试对象和测试范围 2.评估集成测试被测试对象的数量及难度，即工作量 3.确定角色分工和作任务 4.标识出测试各阶段的时间，任务，约束等条件 5.考虑

一定的风险分析及应急计划 6.考虑和准备集成测试需要的测试工具,测试仪器,环境等资源  
7.考虑外部技术支援的力度和深度,以及相关培训安排 8.定义测试完成标准

5) 输出 集成测试计划

6) 出口条件 集成测试计划通过概要设计阶段基线评审  
设计阶段

1) 时间安排 详细设计阶段开始

2) 输入 需求规格说明书 概要设计 集成测试计划

3) 入口条件 概要设计基线通过评审

4) 活动步骤 1.被测对象结构分析 2.集成测试模块分析 3.集成测试接口分析 4.集成测试策略分析

5.集成测试工具分析 6.集成测试环境分析 7.集成测试工作量估计和安排。

5) 输出 集成测试设计(方案)

6.出口条件 集成测试设计通过详细设计基线评审。

实现阶段

1) 时间安排 在编码阶段开始后进行

2) 输入 需求规格说明书 概要设计 集成测试计划 集成测试设计

3) 入口条件 详细设计阶段

4) 活动步骤 集成测试用例设计 集成测试程序设计 集成测试代码设计(如果需要) 集成测试脚本(如果需要) 集成测试工具(如果需要)

5) 输出 集成测试用例 集成测试规程 集成测试代码 集成测试脚本 集成测试工具

6) 出口条件 测试用例和测试规程通过编码阶段基线评审

执行阶段

1) 时间安排 单元测试已经完成后就可以开始执行集成测试了

2) 输入 需求规格说明书 概要设计 集成测试计划 集成高度设计 集成测试例  
集成测试规程 集成测试代码(如果有) 集成测试脚本 集成测试工具 详细设计 代码 单元测试报告

3) 入口条件 单元测试阶段已经通过基线化评审

4) 活动步骤 执行集成测试用例 回归集成测试用例 撰写集成测试报告

5) 输出 集成测试报告

6) 出口条件 集成测试报告通过集成测试阶段基线评审

文档测试?

文档审核测试目前越来越引起人们的重视,软件质量不是检查出来的,而是融进软件开发中来。文档审核测试主要包括需求文档测试,设计文档测试,为前置软件测试中的一部分。

需求文档测试:主要测试需求中是否存在逻辑矛盾以及需求在技术上是否可以实现;

设计文档测试:测试设计是否符合全部需求以及设计是否合理。

白盒测试有哪几种方法?

白盒测试也称结构测试或逻辑驱动测试,它是知道产品内部工作过程,可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行,按照程序内部的结构测试程序,检验程序中的每条通路是否都有能按预定要求正确工作,而不顾它的功能,白盒测试的主要方法有逻辑驱动、基路测试等,主要用于软件验证。“白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。“白盒”法是穷举路径测试。

软件测试面试题(软通动力,博彦科技,奇虎,瑞星,中软)

2007-07-27 14:34

## 1. 软通动力面试笔答

### 1. 白箱测试和黑箱测试是什么?什么是回归测试?

白箱测试是在看懂程序代码和设计方案的前提下,进行软件的测试。这种测试注重于源代码的覆盖率,同时需要测试者具备较高的技术水平。白箱测试的优点是可以对代码有详细的审查,能找出隐藏在代码中的错误,从而确保高质量的代码;缺点是很多时候不能看完所有的代码,不能找出欠缺的代码,同时白箱测试和用户如何使用软件无关。

黑箱测试的优点是测试者无需熟悉软件内部结构,并且根据蓝图在早期就可以制定测试方案,并不依赖于开发者的工作进展,而且黑箱测试简单易行,对测试者的技术要求不高;但是,黑箱测试主要是功能上的测试,只能覆盖只有一小部分的输入,不能保证程序的所有部分都被测试到。

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。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。

回归测试包括两部分:函数本身的测试、其他代码的测试。

在对被修改的函数重新测试。如果函数的设计功能没有变化,直接运行函数测试就可以了。如果修改了设计功能,则要根据增减的功能点,增加或删除测试用例。另外,还要完成白盒覆盖。

函数代码的修改可能导致调用该函数的代码产生错误,所以需要测试其他代码。如果函数是私有函数并且未涉及到全局变量,应运行类测试,否则应运行工程测试。在函数列表中选择类测试或工程测试,编译运行测试工程,即可执行对其他代码的回归测试。

### 2. 单元测试、集成测试、系统测试的侧重点是什么?

单元测试:以代码检查、逻辑覆盖

集成测试:增加静态结构分析、静态质量度量

系统测试:根据黑盒测试结果,采用白盒测试

单元测试是在软件开发过程中要进行的最低级别的测试活动,在单元测试活动中,软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

集成测试,也叫组装测试或联合测试。在单元测试的基础上,将所有模块按照设计要求,组装成为子系统或系统,进行集成测试。实践表明,一些模块虽然能够单独地工作,但并不能保证连接起来也能正常的工作。程序在某些局部反映不出来的问题,在全局上很可能暴露出来,影响功能的实现。

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。

### 3. 设计用例的方法、依据有那些?

白盒测试用例设计有如下方法:基本路径测试\等价类划分\边界值分析\覆盖测试\循环测试\数据流测试\程序插桩测试\变异测试.这时候依据就是详细设计说明书及其代码结构吧,恩,这个真不确定

黑盒测试用例设计方法:基于用户需求的测试\功能图分析方法\等价类划分方法\边界值分析方法\错误推测方法\因果图方法\判定表驱动分析方法\正交实验设计方法.依据是用户需求规格说明书,详细设计说明书

#### 4.一个测试工程师应具备那些素质和技能?

掌握基本的测试基础理论

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

可熟练阅读需求规格说明书等文档

以用户的观点看待问题

有着强烈的质量意识

细心和责任心

良好的有效的沟通方式(与开发人员及客户)

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

①、  
、沟通能力

一名理想的测试者必须能够同测试涉及到的所有人进行沟通,具有与技术(开发者)和非技术人员(客户,管理人员)的交流能力。既要可以和用户谈得来,又能同开发人员说得上话,不幸的是这两类人没有共同语言。和用户谈话的重点必须放在系统可以正确地处理什么和不可以处理什么上。而和开发者谈相同的信息时,就必须将这些活重新组织以另一种方式表达出来,测试小组的成员必须能够同等地同用户和开发者沟通。

②、移情能力

和系统开发有关的所有人员都处在一种既关心又担心的状态之中。用户担心将来使用一个不符合自己要求的系统,开发者则担心由于系统要求不正确而使他不得不重新开发整个系统,管理部门则担心这个系统突然崩溃而使它的声誉受损。测试者必须和每一类人打交道,因此需要测试小组的成员对他们每个人都具有足够的理解和同情,具备了这种能力可以将测试人员与相关人员之间的冲突和对抗减少到最低程度。

③、技术能力

就总体言,开发人员对那些不懂技术的人持一种轻视的态度。一旦测试小组的某个成员作出了一个错误的断定,那么他们的可信度就会立刻被传扬了出去。一个测试者必须既明白被测软件系统的概念又要会使用工程中的那些工具。要做到这一点需要有几年以上的编程经验,前期的开发经验可以帮助对软件开发过程有较深入的理解,从开发人员的角度正确的评价测试者,简化自动测试工具编程的学习曲线。

④、自信心

开发者指责测试者出了错是常有的事,测试者必须对自己的观点有足够的自信心。如果容许别人对自己指东指西,就不能完成什么更多的事情了。

## ⑤、外交能力

当你告诉某人他出了错时，就必须使用一些外交方法。机智老练和外交手法有助于维护与开发人员的协作关系，测试者在告诉开发者他的软件有错误时，也同样需要一定的外交手腕。如果采取的方法过于强硬，对测试者来说，在以后和开发部门的合作方面就相当于“赢了战争却输了战役”。

## ⑥、幽默感

在遇到狡辩的情况下，一个幽默的批评将是很有帮助的。

## ⑦、很强的记忆力

一个理想的测试者应该有能力将以前曾经遇到过的类似的错误从记忆深处挖掘出来，这一能力在测试过程中的价值是无法衡量的。因为许多新出现的问题和我们已经发现的问题相差无几。

## ⑧、耐心

一些质量保证工作需要难以置信的耐心。有时你需要花费惊人的时间去分离、识别和分派一个错误。这个工作是那些坐不住的人无法完成的。

## ⑨、怀疑精神

可以预料，开发者会尽他们最大的努力将所有的错误解释过去。测试者必须听每个人的说明，但他必须保持怀疑直到他自己看过以后。

## ⑩、自我督促

干测试工作很容易使你变得懒散。只有那些具有自我督促能力的人才能够使自己每天正常地工作。

## 11、洞察力

一个好的测试工程师具有“测试是为了破坏”的观点，捕获用户观点的能力，强烈的质量追求，对细节的关注能力。应用的高风险区的判断能力以便将有限的测试针对重点环节。

### 5.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。

### 6.你用过的测试工具的主要功能、性能及其他？

WinRunner (WR) 是一个基于 Windows 的企业级功能测试工具，它在业务应用正式部署之前，通过自动捕获、检测和重放用户对应用系统的交互操作，来发现系统缺陷，确保那些跨越多个应用程序和数据库的业务流程在初次发布就能避免故障的出现，保证系统对所有关键业务处理功能、处理流程的正确，保障应用的质量和准备工作的最优化

主要功能:

1) 轻松创建测试: 用 WinRunner 创建一个测试，只需在应用软件中操作记录下一个标准的业务流程，例如下一张订单或建立一个新的商家账户，WinRunner 将直观地记录该流程。即使技术知识有限的用户，也能通过在 GUI 上单击鼠标而生成完整的测试。用户还可以直接编辑测试指令来满足各种复杂测试的需求

2) 插入检查点: 在建立一个测试的过程中可以插入检查点，以在查找潜在错误的同时，将预想的结果和实际测试结果进行比较。在插入检查点后，WinRunner 会收集相应的性能指标，在测试运行时对其一一验证。WinRunner 允许使用几种不同类型的检查点，包括文本、GUI、位图和数据库等。例如用一个位图检查点，可以确认一个位图图像是否出现在指定的位置上。

WinRunner 的数据库检验功能能够自动标示出被修改的数据

3) 检验数据: 除了创建并运行测试，WinRunner 还能验证数据库的数值，从而确保交易的准确性。例如，在测试创建时，可以设定哪些数据库表格和记录资料需要检测。在重放时，测试程序就会核对数据库内的实际数值与预想的数值。WinRunner 能自动显示检测结果，在有更新/修改、删除或插入的记录上会用突出标识引起注意

4) 增强测试: 为了彻底全面地测试一个应用程序，用户需要了解对于不同类型的数据它是如何运行的。WinRunner 的 DataDriver Wizard 使用户只需单击几下鼠标，就能简单地将一个记录下的业务流程转化为一个数据驱动的测试，来反映多个用户各自独特且真实的操作行为

5) 运行测试: 在建立测试，并插入检查点和做一些必要的功能添加后，就可以开始运行测试。当 WinRunner 执行测试时，它会自动操作应用程序，正如一个真实用户根据记录流程执行着每一步的操作，而且它的意外处理功能为测试排除干扰，包括消息和警报

6) 分析结果: 一旦测试运行后，就需要分析测试结果。WinRunner 的互动式的报告工具通过提供详尽的、易读的报告，其中会列出在测试中发现的差错和出错的位置，来帮助用户解释所得到的结果。这些报告对在测试运行中发生的重要事件进行描述，如出错内容和检查点等。单击按钮，还能进一步获取任何未被包括在此测试范围内的错误的详尽资料。这些结果都可以通过 MI 的测试管理工具 TestDirector 来查阅

7) 维护测试: 随着时间推移，开发人员会对应用程序做进一步的修改，这时，需要增加额外的测试。WinRunner 会帮助用户创建可重复使用的测试，以大大节省时间和资源，充分利用测试投资

## 7. 一个缺陷测试报告的组成

缺陷的标题，缺陷的基本信息，复现缺陷的操作步骤，缺陷的实际结果描述，期望的正确结果描述，注释文字和截取的缺陷图象。

缺陷的标题;

缺陷的基本信息;

测试的软件和硬件环境;

测试的软件版本;

缺陷的类型;

缺陷的严重程度;

缺陷的处理优先级。

复现缺陷的操作步骤;

缺陷的实际结果描述；  
期望的正确结果描述；  
注释文字和截取的缺陷图像。

8.基于 WEB 信息管理系统测试时应考虑的因素有哪些？

- 一、功能测试
  - 1、链接测试
  - 2、表单测试
  - 3、Cookies 测试
  - 4、设计语言测试
  - 5、数据库测试
- 二、性能测试
  - 1、连接速度测试
  - 2、负载测试
  - 3、压力测试
- 三、可用性测试
  - 1、导航测试
  - 2、图形测试
  - 3、内容测试
  - 4、整体界面测试
- 四、客户端兼容性测试
  - 1、平台测试
  - 2、浏览器测试
- 五、安全性测试

9.软件本地化测试比功能测试都有哪些方面需要注意？

软件本地化测试的目的：

软件本地化测试的测试策略：1.本地化软件要在各种本地化操作系统上安装并测试。2.源语言软件安装在另一台相同源语言操作系统上，作为对比测试。3.重点测试因本地化引起的软件的功能和软件界面的错误。4.测试本地化软件的翻译质量。5.手工测试和自动测试相结合。

10.软件测试项目从什么时候开始，？为什么？

软件测试应该在需求分析阶段就介入,因为测试的对象不仅仅是程序编码,应该对软件开发过程中产生的所有产品都测试,并且软件缺陷存在放大趋势.缺陷发现的越晚,修复它所花费的成本就越大.

11.需求测试注意事项有哪些？

一个好的需求应当具有一下特点：

完整性：每一项需求都必须将所要实现的功能描述清楚，以使开发人员获得设计和实现这些功能所需的所有必要信息。

正确性：每一项需求都必须准确地陈述其要开发的功能。

一致性：一致性是指与其它软件需求或高层（系统，业务）需求不相矛盾。

可行性：每一项需求都必须是在已知系统和环境的权能和限制范围内可以实施的。

无二义性：对所有需求说明的读者都只能有一个明确统一的解释，由于自然语言极易导致二

义性，所以尽量把每项需求用简洁明了的用户性的语言表达出来。

健壮性：需求的说明中是否对可能出现的异常进行了分析，并且对这些异常进行了容错处理。

必要性：“必要性”可以理解为每项需求都是用来授权你编写文档的“根源”。要使每项需求都能回溯至某项客户的输入，如 Use Case 或别的来源。

可测试性：每项需求都能通过设计测试用例或其它的验证方法来进行测试。

可修改性：每项需求只应在 SRS 中出现一次。这样更改时易于保持一致性。另外，使用目录表、索引和相互参照列表方法将使软件需求规格说明书更容易修改。

可跟踪性：应能在每项软件需求与它的根源和设计元素、源代码、测试用例之间建立起链接链，这种可跟踪性要求每项需求以一种结构化的，粒度好（fine-grained）的方式编写并单独标明，而不是大段大段的叙述。

## 12.简述一下缺陷的生命周期

• 软件缺陷的生命周期指的是一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程。

简单的软件缺陷生命周期：

- 1、发现——打开：测试人员找到软件缺陷并将软件缺陷提交给开发人员；
- 2、打开——修复：开发人员再现、修复缺陷，然后提交测试人员去验证；
- 3、修复——关闭：测试人员验证修复过的软件，关闭已不存在的缺陷。

但是这是一种理想的状态，在实际的工作中是很难有这样的顺利的，需要考虑的各种情况都还是非常多的。

复杂的软件缺陷生命周期：

- 1、新建一个软件缺陷，这个软件缺陷是（open）状态，进行 bug 审查，不是代码问题，就是设计需要修改；
- 2、新建一个软件缺陷，这个软件缺陷是（open）状态，进行 bug 审查，以后修改的，就可以延期；
- 3、新建一个软件缺陷，这个软件缺陷是（open）状态，进行 bug 审查，实际没有这个 bug，可以将其关闭；
- 4、新建一个软件缺陷，这个软件缺陷是（open）状态，看是否清楚可重现，如果不能重现，就是缺少信息，需要返回到（open）状态；如果能够重现，就进行修正，修正后关闭，进行回归测试。

## 13.测试分析测试用例注意（事项）？

### 1.为什么要写用例：

我们编写测试用例，有如下的好处：

便于团队交流：假如说一个测试团队有 10 个成员，大家测试的时候都各自为政，没有统一的标准，测试的效率无疑会大打折扣；如果大家都遵循统一的用例规范去写，就会解决这一问题。

便于重复测试：大家知道，软件在实际开发过程中是会有不同版本的，比如会从 1.0 升级到 10.0，那么如果不写测试用例的话，在测试 10.0 版本的时候，你能完全记得 1.0 版本时你做过哪些测试吗？测试用例就像一个备忘录一样，便于重复测试。

便于跟踪统计：这一点是针对测试经理或是项目经理来说的，项目负责人通过看测试用例的执行情况，就能了解到项目目前的概况，比如已经执行了哪些测试，还有哪些测试没有执行，测试没有通过的地方主要集中在哪些模块等。

便于用户自测：尤其是项目软件，有的时候用户希望自己测试一下软件产品，但是用户大都

是非专业人士，他需要根据你写好的用例来更好的检验产品的质量  
说了这么多编写测试用例的优点，那它有没有缺点呢？有一个明显的缺点就是需要花费大量的时间，通常编写测试用例的时间比实际执行测试的时间还要长，这一点大家会在实际工作中有深刻的体会

## 2.什么时候写用例：

什么时候写用例？这个问题没有统一的标准答案，但有一点可以肯定，就是测试用例要尽早编写。大家认为在哪个阶段开始写用例比较好呢？

通常，我们都会在测试设计阶段来写用例，即《需求规格说明书》和《测试计划》都已完成之后

## 3.由谁来写测试用例

有的读者会说，当然是测试人员来写用例了！

可是测试人员又会有不同的角色，一般分为测试经理，测试设计人员，测试执行人员和测试工具开发人员等，一般测试用例是由测试设计人员来编写，由测试执行人员来执行，这就要求测试设计人员有一定的用例设计经验，并对被测试的系统有深入的了解。

但是在很多小公司里面，区分的不是这么明显，一个测试人员往往会身兼数职，既是测试组长，又是测试设计人员，又是测试执行人员。项目组里就你一个测试工程师，你不写用例谁写啊！

## 4.根据什么写测试用例

我们编写测试用例的唯一标准就是用户需求，具体的参考资料就是《系统需求规格说明书》和软件原型，其中软件原型指的是没有嵌入全部源代码的软件界面，比如我做一个电子商务网站，为了尽快能给用户演示，我只是用 html 语言作一些静态页面，并没有编写动态的程序，这就是一个软件原型，它也看作是需求的一部分。

## 二. 瑞星笔试题(15 道)

1.一台计算机的 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？

你的子网掩码不对。

不可能出现 255.255.255.64 的子网掩码。

另外，这个题也不能说成“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。

针对此题：

.....

子网掩码为 255.255.128.0 时，是同一子网。

子网掩码为 255.255.255.0 时，是同一子网。

子网掩码为 255.255.255.128 时，不是同一子网。

子网掩码为 255.255.255.192 时，不是同一子网。

.....

2.internet 中 e-mail 协议,IE 的协议,NAT 是什么,有什么好处,能带来什么问题?DNS 是什么,它是如何工作的?

NAT 全称 Network Address Translation, 中文解释为“网络地址转换”。NAT 是一种 IETF (Internet Engineering Task Force) 的标准, 简单描述其功能就是让处于内网的计算机能够通过 NAT 的作用透明的访问外网的互联网资源。NAT 的功能一般集成在路由器、防火墙或者单独的 NAT 设备中。

DNS 全名是 Domain Name System, 透过 DNS 系统, 我们可以由一部机器的 domain name 查其 IP, 也可以由机器的 IP 反查它的 domain name, 除此之外 DNS 还与 Mail System 结合, 提供 Mail routing 的功能.

DNS 分为 Client 和 Server, Client 扮演发问的角色, 也就是问 Server 一个 Domain Name, 而 Server 必须要回答此 Domain Name 的真正 IP 地址。而当地的 DNS 先会查自己的资料库。如果自己的资料库没有, 则会往该 DNS 上所设的 DNS 询问, 依此得到答案之后, 将收到的答案存起来, 并回答客户。

3.PROXY 是如何工作的?

Proxy 是什么呢, 是代理。普通的因特网访问是一个典型的客户机与服务器结构: 用户利用计算机上的客户端程序, 如浏览器发出请求, 远端 WWW 服务器程序响应请求并提供相应的数据。而 Proxy 处于客户机与服务器之间, 对于服务器来说, Proxy 是客户机, Proxy 提出请求, 服务器响应; 对于客户机来说, Proxy 是服务器, 它接受客户机的请求, 并将服务器上传来的数据转给客户机。它的作用很象现实生活中的代理商。因此 Proxy Server 的中文名称就是代理服务器。

Proxy Server 的工作原理是: 当客户在浏览器中设置好 Proxy Server 后, 你使用浏览器访问所有 WWW 站点的请求都不会直接发给目的主机, 而是先发给代理服务器, 代理服务器接受了客户的请求以后, 由代理服务器向目的主机发出请求, 并接受目的主机的数据, 存于代理服务器的硬盘中, 然后再由代理服务器将客户要求的数据发给客户。

4.win2k 系统内 AT 命令完成什么功能,Messenger 服务是做什么,怎么使用?

AT 命令可在指定时间和日期、在指定计算机上运行命令和程序。

Messenger 服务:发送和接收系统管理员或者“警报器”服务传递的消息。

5 进程,线程的定义及区别

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是 CPU 调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

一个线程可以创建和撤销另一个线程;同一个进程中的多个线程之间可以并发执行

6,32 位操作系统内,1 进程地址空间多大,进程空间与物理内存有什么关系?

7 网络攻击常用的手段,防火墙如何保证安全.

8 如何配静态 IP,如何测网络内 2 台计算机通不通,PING 一次返几个数据包?

9 WIN9X 与 WINNT 以上操作系统有"服务"吗,服务是什么,如何停止服务?

10 AD 在 WIN2KSERVER 上建需什么文件格式,AD 是什么?XP 多用户下"注销"与"切换"的区别.

11 UDP 可以跨网段发送吗?

12 最简单的确认远程计算机(win2K 以上)某个监听端口是正常建立的?

13 软件测试的定义,测试工作是枯燥反复的,你是如何理解的?黑盒,白盒,回归,压力测试的定义.

14 winrunner,loadrunner 是什么,区别

15 磁盘分区如何分类,请举例说明安装操作系统的注意事项.

(1 小时答题)

### 三. 中软的面试题

#### 一. 简答题.

1. 避免死锁的方法有哪些?

产生死锁的四个必要条件:

- (1) 互斥条件: 一个资源每次只能被一个进程使用。
- (2) 请求与保持条件: 一个进程因请求资源而阻塞时, 对已获得的资源保持不放。
- (3) 不剥夺条件: 进程已获得的资源, 在未使用完之前, 不能强行剥夺。
- (4) 循环等待条件: 若干进程之间形成一种头尾相接的循环等待资源关系。

这四个条件是死锁的必要条件, 只要系统发生死锁, 这些条件必然成立, 而只要上述条件之一不满足, 就不会发生死锁。

2. 在 Sybase 数据库中注册用户与数据库用户有什么区别?

3. 在 MS SQL\_Server 数据库中通过什么约束保证数据库的实体完整性

可以通过建立唯一的索引、PRIMARY KEY 约束、UNIQUE 约束或 IDENTITY 约束来实现实体完整性

4. 内存有哪几种存储组织结构.请分别加以说明

5. JAVA 中的 Wait() 和 notify()方法使用时应注意些什么?

6. 用户输入一个整数.系统判断,并输出是负数还是非负数,请设计测试用例.

7. 操作系统中的同步和互斥解决了什么问题

8. UNIX 中 init

#### 二. 编写类 String 的构造函数,析构函数和赋值函数

已知类 String 的原型为

```
class string
{
public:
```

```
string(const char *str=null);//普通构造函数
string(const string &other);//拷贝构造函数
---string(void);
string &operate=(const string &other);//赋值函数
private:
char * m-data;//用于保存字符串
```

```
};
```

请编写 string 的上述 4 个函数

三. 有关内存的思考题

1. void getmemory(char \*p)

```
{ p=(char*)malloc(100);
```

```
}
```

void test(void)

```
{
```

```
char * str =null;
```

```
getmemory(str);
```

```
strcpy(str,"hello,world");
```

```
printf(str);
```

```
}
```

请问运行 Test 函数会有什么样的结果

2. char\*getmemory(void)

```
{ char p[]="hello world";
```

```
return p;
```

```
}
```

void test(void)

```
{
```

```
char *str=null;
```

```
str=Getmemory();
```

```
printf(str);
```

```
} 请问运行 Test 函数会有什么样的结果.
```

三. 奇虎面试题

前三道程序题

(下面的题不排序, 有笔试题, 也有面试题)

4、怎么划分缺陷的等级?

5、怎么评价软件工程师?

6、软件工程师的素质是什么?

7、怎么看待软件测试?

8、软件测试是一个什么样的行业?

9、图书(图书号, 图书名, 作者编号, 出版社, 出版日期)

作者(作者姓名, 作者编号, 年龄, 性别)

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。

10、你的职业生涯规划

11、测一个三角形是普通三角形、等腰三角形、等边三角形的流程图，测试用例。

12、写出你常用的测试工具。

13、Iordrunner 分哪三部分？

14、希望以后的软件测试是怎么样的一个行业？

15、.软件测试项目从什么时候开始？

我答：从软件项目的需要分析开始。

问：为什么从需求分析开始？有什么作用？

#### 四. 北京博彦科技笔试+面试

#### 笔试题

1. 文件格式系统有哪几种类型？分别说说 win95、win98、winMe、w2k、winNT、winXP 分别支持那些文件系统。

FAT(File Allocation Table)是“文件分配表”的意思。

对我们来说，它的意义在于对硬盘分区的管理。FAT16、FAT32、NTFS 是目前最常见的三种文件系统。

Win95: FAT16 和 FAT32

Win98: FAT16,FAT32

winMe:FAT16,FAT32

w2k: FAT(FAT16),FAT32,NTFS

winNT: FAT16/FAT32/NTFS

winXP:FAT16,FAT32,NTFS

2. 分别填入一个语句，完成下面的函数，通过递归计算数组 a[100]的前 n 个数之和。

```
Int sum ( int a[],int n )
```

```
{  
    if (n>0) return_____ sum(a[], n--) + a[n] _____;  
    else return_____ a[n]_____;// 其实就是 a[0]  
}
```

//一直递归到 0，然后逐级返回，实现累加

3. 写出你所知道的 3 种常用的排序方法，并用其中一种方法设计出程序为数组 a[100]排序。

常用的排序算法有很多：

冒泡，快速排序，直接插入，希尔排序，选择排序，堆排序，归并排序！

就举冒泡排序(c++)：

```
void bubblesort()
```

```
{  
    for (i = 1; i < max; i ++)  
    {
```

```

for (j = max - 1; j >= i; j --)
if (a[j + 1] < a[j]) //小则交换
{
    a[0] = a[j + 1];
    a[j + 1] = a[j];
    a[j] = a[0];
}
}
}

```

4. 什么是兼容性测试？兼容性测试侧重哪些方面，请按照优先级用矩阵图表列出。

（这题的第二问我不会答，所以原题目记得不是很清楚，大家能看明白问什么就好）

兼容性是指协调性，

1. 硬件上就是说你的电脑的各个部件，CPU，显卡等等组装到一起以后的情况，会不会相互有影响，不能很好的运作。
2. 软件上就是说你的电脑的软件之间能否很好的运做，会不会有影响啊？还有软件和硬件之间能否发挥很好的效率工作，会不会影响导致系统的崩溃。

#### 1、平台测试

市场上有很多不同的操作系统类型，最常见的有 Windows、Unix、Macintosh、Linux 等。Web 应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。

因此，在 Web 系统发布之前，需要在各种操作系统下对 Web 系统进行兼容性测试。

#### 2、浏览器测试

浏览器是 Web 客户端最核心的构件，来自不同厂商的浏览器对 Java、JavaScript、ActiveX、plug-ins 或不同的 HTML 规格有不同的支持。例如，ActiveX 是 Microsoft 的产品，是为 Internet Explorer 而设计的，JavaScript 是 Netscape 的产品，Java 是 Sun 的产品等等。另外，框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示。不同的浏览器对安全性和 Java 的设置也不一样。

测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设置的适应性。

5. 我现在有个程序，发现在 WIN98 上运行得很慢，怎么判别是程序存在问题还是硬件系统存在问题？

可能是病毒或者恶意程序啊,如果是程序运行慢那肯定是软件问题,

硬件问题主要表现在点不亮机子和报警.

解决方法,安装杀毒软件和优化软件.

6. 翻译，中——英，有关 P2P 点对点文件传输的原理。

7. 翻译，英——中，有关互联网的发展对商务、学习、交流的影响。

笔试完了是初步的面试

先问了个问题：FAT16/FAT32/NTFS 哪个的安全性最好，为什么？（不会答）

NTFS 文件系统是一个基于安全性的文件系统，是 Windows NT 所采用的独特的文件系统结构，它是建立在保护文件和目录数据基础上，同时照顾节省存储资源、减少磁盘占用量的一种先进的文件系统。使用非常广泛的 Windows NT 4.0 采用的就是 NTFS 4.0 文件系统，相信它所带来的强大的系统安全性一定给广大用户留下了深刻的印象。Win 2000 采用了更新版本的 NTFS 文件系统——NTFS 5.0，它的推出使得用户不但可以像 Win 9X 那样方便快捷地操作和管理计算机，同时也可享受到 NTFS 所带来的系统安全性。

又做了两道题，

一题是关于 C++ 类的继承，看程序写出输出结果，A 是虚类，B 继承 A，跟一般 C++ 的书上的习题差不多。

一题是写出在 32 位机器下，计算几个变量的 size，

最后用英文介绍一下自己

黑盒测试的测试用例设计方法

目前黑盒测试的测试用例设计方法有 5 种：

1. 等价类划分
2. 边界值分析
3. 错误推测法
4. 因果图
5. 功能图

一、等价类划分

等价类划分设计方法是把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少量具有代表性的数据作为测试用例。

等价类是指某个输入域的子集合。在该子集中，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他值的测试。

等价类划分有两种不同的情况：有效等价类和无效等价类。设计时要同时考虑这两种等价类。

下面给出 6 条确定等价类的原则：

1. 在输入条件规定了取值范围或值的个数的情况下，则可以确立一个有效等价类和两个无效等价类。
2. 在输入条件规定了输入值的集合或者规定了“必须如何”的条件（如“必须为偶数”）的情况下，则可以确立一个有效等价类和一个无效等价类。
3. 在输入条件是一个布尔量的情况下，可以确立一个有效等价类和一个无效等价类。
4. 在输入数据的一组值（假定 n 个），并且程序要对每一个输入值分别处理的情况下，可以确立 n 个有效等价类和一个无效等价类。
5. 在输入数据必须遵守的规则的情况下，可以确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。
6. 在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价

类进一步的划分为更小的等价类。

在确立了等价类后，可建立等价类表，列出所有划分出的等价类。然后从划分出的等价类中按以下的 3 个原则设计测试用例：

- ? 为每一个等价类规定一个唯一的编号
- ? 设计一个新的测试用例，使其尽可能多的覆盖尚未被覆盖的有效等价类，重复这一步，直到所有的有效等价类都被覆盖为止。
- ? 设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止。

例：程序规定：输入三个整数作为三边的边长构成三角形。当此三角形为一般三角形、等腰三角形、等边三角形时，分别作计算。用等价类划分方法为该程序进行测试用例设计。

解：设 a、b、c 代表三角形的三条边。

1) 分析题目中给出的和隐含的对输入条件的要求：

- a) 整数
- b) 3 个数
- c) 非零数
- d) 正数
- e) 两边之和大于第三边
- f) 等腰
- g) 等边

2) 列出等价类表并编号

3) 列出覆盖上述等价类的测试用例，如下表：

## 二、边界值分析法

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。其次，应但选取正好等于、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。

基于边界值分析方法选择测试用例的原则：

1. 如果输入条件规定了值的范围，应取刚达到这个范围的边界值，以及刚刚超过这个范围边界的值作为测试输入的数据。
2. 如果输入条件规定了值的个数，应用最大个数、最小个数、比最小个数少一、比最大个数多一的数作为测试输入的数据。
3. 根据规格说明的每个输出条件，使用前面的原则 1。
4. 根据规格说明的每个输出条件，使用前面的原则 2。
5. 如果程序的规格说明给出的输入域或输出域是有序集合，则应选取集合的第一个元素和最后一个元素作为测试用例数据。
6. 如果程序中使用了一个内部数据结构，应当选择这个内部数据结构边界上的值作为测试用例。
7. 分析规格说明，找出其他可能的边界条件。

## 三、错误推测法

错误推测法就是根据经验和直觉推测程序中所有可能存在的各种错误，从而有针对性地设计测试用例的方法。

基本思路：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：输入数据和输出数据为 0 的情况。

例：现有一个学生标准化考试批阅试卷，产生成绩报告的程序。其规格说明如下：程序的输入文件由一些有 80 个字符的记录组成，所有记录分为 3 组，如图：

1、标题：改组只有一个记录，其内容是成绩报告的名字。

2、各题的标准答案：每个记录均在第 80 个字符处标以数字 2。该组的记录：

第一个记录：第 1~3 个字符为试题数（1~999）。第 10~59 个字符是 1~50 题的标准答案（每个合法字符表示一个答案）。

第二个记录：是第 51~100 题的标准答案。

…….

3、学生的答案：每个记录均在第 80 个字符处标以数字 3。每个学生的答卷在若干个记录中给出。

学号：1~9 个字符

1~50 题的答案：10~59。当大于 50 题时，在第二、三、……个记录中给出。

学生人数不超过 200，试题数不超过 999。

程序的输出有 4 个报告：

a)按学号排列的成绩单，列出每个学生的成绩、名次。

b)按学生成绩排序的成绩单。

c)平均分数及标准偏差的报告

d)试题分析报告。按试题号排序，列出各题学生答对的百分比。

解答一：采用边界值分析方法，分析和设计测试用例。分别考虑输入条件和输出条件，以及边界条件。下表列出了输入条件及相应的测试用例。

下表为输出条件及相应的测试用例表。

解答二：采用错误推测法还可补充设计一些测试用例：

1. 程序是否把空格作为回答
2. 在回答记录中混有标准答案记录
3. 除了标题记录外，还有一些的记录最后一个字符即不是 2 也不是 3
4. 有两个学生的学号相同
5. 试题数是负数。

#### 四、因果图法

因果图法是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的测试用例设计方法。

利用因果图生成测试用例的基本步骤：

1. 分析软件规格说明描述中那些是原因，那些是结果，并给每个原因和结果赋予一个标识符。
2. 分析软件规格说明描述的语义。找出原因和结果之间、原因和原因之间的关系，根据这些关系，画出因果图。
3. 在因果图上用一些记号表明约束或限制条件。
4. 把因果图转换为判定表。
5. 把判定表的每一列拿出来作为依据，设计测试用例。

例：第一列字符必须是 A 或 B，第二列字符必须是一个数字，在此情况下进行文件的修改，但如果第一列字符不正确，则给出信息 L；如果第二列字符不是数字，则给出信息 M。

解 1、画出因果关系表和因果图。

2、根据因果图建立判定表。

按条件的各种组合情况产生对应的动作。原因 1 和原因 2 不能同时成立，故可排除这种情况。

从判定表可设计出测试用例：6 个测试用例是所需的数据。

本 文 来 自 CSDN 博 客 ， 转 载 请 标 明 出 处 ：  
<http://blog.csdn.net/Alina77/archive/2008/04/10/2277646.aspx>