1. 选择题

1．软件测试的目的是（ B ）。

A）试验性运行软件 B）发现软件错误 C）证明软件正确 D）找出软件中全部错误

2．软件测试中白盒法是通过分析程序的（ B ）来设计测试用例的。

A）应用范围 B）内部逻辑 C）功能 D）输入数据

3．黑盒法是根据程序的（ C ）来设计测试用例的。

A）应用范围 B）内部逻辑 C）功能 D）输入数据

4．为了提高软件测试的效率，应该（ D ）。

A）随机地选取测试数据 B）取一切可能的输入数据作为测试数据

C）在完成编码以后制定软件的测试计划 D）选择发现错误可能性最大的数据作为测试用例

5．与设计测试用例无关的文档是（ A ）。

A）项目开发计划 B）需求规格说明书 C）设计说明书 D）源程序

6．测试的关键问题是（ B ）。

A）如何组织软件评审 B）如何选择测试用例

C）如何验证程序的正确性 D）如何采用综合策略

7．软件测试用例主要由输入数据和（ C ）两部分组成。

A）测试计划 B）测试规则 C）预期输出结果 D）以往测试记录分析

8．成功的测试是指运行测试用例后（ B ）。

A）未发现程序错误 B）发现了程序错误 C）证明程序正确性 D）改正了程序错误

9．下列几种逻辑覆盖标准中，查错能力最强的是（ D ）。

A）语句覆盖 B）判定覆盖 C）条件覆盖 D）条件组合覆盖

10．在黑盒测试中，着重检查输入条件组合的方法是（ D ）。

A）等价类划分法 B）边界值分析法 C）错误推测法 D）因果图法

11．单元测试主要针对模块的几个基本特征进行测试，该阶段不能完成的测试是（ A ）。

A）系统功能 B）局部数据结构 C）重要的执行路径 D）错误处理

12．软件测试过程中的集成测试主要是为了发现（ B ）阶段的错误。

A）需求分析 B）概要设计 C）详细设计 D）编码

13．不属于白盒测试的技术是（ D ）。

A）路径覆盖 B）判定覆盖 C）循环覆盖 D）边界值分析

14．集成测试时，能较早发现高层模块接口错误的测试方法为（ A ）。

A）自顶向下渐增式测试 B）自底向上渐增式测试 C）非渐增式测试 D）系统测试

15．确认测试以（ A ）文档作为测试的基础。

A）需求规格说明书 B）设计说明书 C）源程序 D）开发计划

16．使用白盒测试方法时，确定测试数据应根据（ A ）和指定的覆盖标准。

A）程序内部逻辑 B）程序的复杂度 C）使用说明书 D）程序的功能

17．程序的三种基本结构是（ B ）。

A）过程子、程序、分程序 B）顺序、选择、循环

C）递归、堆栈、队列 D）调用、返回、转移

18．结构化程序设计的一种基本方法是（ D ）

A）筛选法 B）递归法 C）归纳法 D）逐步求精法

19．软件调试的目的是（ A ）

A）找出错误所在并改正之 B）排除存在错误的可能性

C）对错误性质进行分类 D）统计出错的次数

20．程序三种基本结构的共同特点是（ D ）

A）不能嵌套使用 B）只能用来写简单程序

C）已经用硬件实现 D）只有一个入口和一个出口

21．下面说法正确的是( C )。

A）经过测试没有发现错误说明程序正确 B）测试的目标是为了证明程序没有错误

C）成功的测试是发现了迄今尚未发现的错误的测试 D）成功的测试是没有发现错误的测试

22．覆盖准则最强的是( D )。

A）语句覆盖 B）判定覆盖

C）条件覆盖 D）路径覆盖

23．发现错误能力最弱的是( A )。

A）语句覆盖 B）判定覆盖

C）条件覆盖 D）路径覆盖

24．实际的逻辑覆盖测试中，一般以( C )为主设计测试用例。

A）条件覆盖 B）判定覆盖

C）条件组合覆盖 D）路径覆盖

25．下面( D )方法能够有效地检测输入条件的各种组合可能引起的错误。

A）等价类划分 B）边界值分析

C）错误推测 D）因果图

26．与确认测试阶段有关的文档是( A )。

A）需求规格说明书 B）概要设计说明书

C）详细设计说明书 D）源程序

27．超出软件工程范围的测试是( D )。

A）单元测试 B）集成测试

C）确认测试 D）系统测试

28．软件测试不需要了解软件设计的( D )。

A）功能 B）内部结构

C）处理过程 D）条件

29．( B )方法需要考察模块间的接口和各模块之间的联系。

A）单元测试 B）集成测试

C）确认测试 D）系统测试

30．调试应该由( B )完成。

A）与源程序无关的程序员 B）编制该源程序的程序员

C）不了解软件设计的机构 D）设计该软件的机构

31．基线是软件生存期中各开发阶段的一个特定点，它可作为一个检查点，当采用的基线发生错误时，我们可以返回到最近和最恰当的（ C ）上。

A）配置项 B）程序 C）基线 D）过程

32．版本管理是对系统不同版本进行的（ A ）过程

A）标识与跟踪 B）标识变更 C）发布变更 D）控制变更

33．在变更控制中，管理各个用户存取和修改一个特定软件配置对象的权限是( D )。

A）异步控制　　　 B）同步控制

C）存取控制　 　　 D）基线控制

34. 在配置管理中，“登入”和“检出”处理实现了两个重要的变更控制要素，即同步控制和( B )。

A）异步控制　　　 B）同步控制

C）基线控制　　　 D）存取控制

35. 按照软件配置管理的原始指导思想，受控制的对象应是( C )。

A）软件元素　　 B）软件配置项

C）软件项目　　　 D）软件过程

36. 软件配置项是软件配置管理的对象，指的是软件工程过程中所产生的( B )。

A）接口　　 B）软件环境　　　C）信息项 D）版本

37．在变更控制中，可用来确保由不同用户所执行的并发控制是( B )。

A）异步控制　 B）同步控制　　C）存取控制　 D）基线控制

38．以下说法错误的是( A )。

A）软件配置管理简称SCI

B）软件配置项是配置管理的基本单位

C）软件配置实际上是一动态的概念

D）软件工程过程中某一阶段的变更均要引起软件配置的变更

39．在设计人机界面时，应主要考虑的因素有（D）。

1. 系统响应时间　B）错误处理　C）用户求助机制　D）以上都是

40．人机界面的风格大致经过了（ D ）代的演变

A）一 B）二 C）三 D） 四

1. 填空题

1．软件测试的目的是尽可能多地发现软件中存在的 错误 ，将测试 测试结果 作为纠错的依据。

2．测试阶段的基本任务是根据软件开发各阶段的 文档资料 和程序的 内部结构 ，精心设计一组 测试用例 ，利用这些实例执行 程序 ，找出软件中潜在的各种 错误 和 缺陷 。

3．测试用例由 输入数据 和预期的 输出数据 两部分组成。

4．软件测试方法一般分为两大类： 动态测试 方法和 静态测试 方法。

5．动态测试通过 运行程序 发现错误。根据 测试用例 的设计方法不同，动态测试又分为 黑盒测试 与 白盒测试 两类。

6．静态测试采用 人工检测 和 计算机辅助静态分析 的手段对程序进行检测。

7．人工审查程序偏重于 编码质量 的检验，而软件审查除了审查 编码 还要对各阶段 软件产品 进行检验。

8．计算机辅助静态分析利用 静态分析 工具对测试程序进行 特性 分析。

9．黑盒法只在软件的 接口 处进行测试，依据 需求规格 说明书，检查程序是否满足 功能 要求。

10．白盒法必须考虑程序的 内部结构 和 处理过程 ，以检查 处理过程 的细节为基础，对程序中尽可能多的逻辑路径进行 测试 。

11．白盒测试是 结构 测试，被测对象是 源程序 ，以程序的 内部逻辑 为基础设计测试用例。

12．逻辑覆盖是对程序内部有 判定 存在的逻辑结构设计测试用例，根据程序内部的逻辑覆盖程度又可分为 语句覆盖 判定覆盖 条件覆盖 判定/条件覆盖 条件组合覆盖 路径覆盖6种覆盖技术。

13．实际的逻辑覆盖测试中，一般以 条件组合 覆盖为主设计测试用例，然后再补充部分用例，以达到 路径 覆盖测试标准。

14．循环覆盖是对程序内部有 循环 存在的逻辑结构设计测试用例，它通过限制 循环次数 来测试。

15．基本路径测试是在程序 控制流程图 基础上，通过分析控制构造的 环路 复杂性，导出 基本路径 集合，从而设计测试用例。

16．黑盒测试是 功能 测试，用黑盒技术设计测试用例有4种方法：等价类划分 边界值分析 错误推测 因果图。

17．等价类划分从程序的 功能 说明，找出一个输入条件（通常是 一句话 或 一个短语 ），然后将每个输入条件划分成两个或多个 等价类 。

18．边界值分析是将测试 边界 情况作为重点目标，选取正好等于、刚刚大于或刚刚小于边界值 的测试数据。如果输入或输出域是一个有序集合，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。

19．在测试程序时，根据经验或直觉推测程序中可能存在的各种错误，称为 错误推测法 。

20．因果图的基本原理是通过画 因果 图，把用自然语言描述的 功能说明 转换为 判定表 ，最后为判定表每一列设计一个测试用例。

21．测试的综合策略是在测试中，联合使用各种 测试 方法。通常先用 黑盒 法设计基本的测试用例，再用 白盒 法补充一些必要的测试用例。

22．软件测试过程中需要3类信息：软件配置 、 测试配置 和 测试工具 。

23．软件测试一般经过4个测试：单元测试 集成测试 确认测试 系统测试 。

24．单元测试 指对源程序中每一个程序单元进行测试，检查各个模块是否正确实现规定的功能，从而发现模块在编码中或算法中的错误，它涉及 编码 和 详细设计 的文档。

25．单元测试主要测试模块的5个基本特征：模块接口 局部数据结构 重要的执行路径 错误处理 边界条件。

26．在单元测试中，需要为被测模块设计 驱动 模块和 桩 模块。 驱动模块 用来模拟被测模块的上级调用模块， 桩模块 用来代替被测模块所调用的模块。

27．集成测试指在 单元 测试基础上，将所有模块按照设计要求组装成一个完整的系统进行的测试。也称 组装 测试或 联合 测试。

28．集成测试的方法有两种： 非渐增式测试 渐增式测试 。

29．渐增式测试有两种不同的组装模块的方法： 自顶向下结合 自底向上结合 。

30．自顶向下渐增式测试不需要编写 驱动 模块，只需要编写 桩 模块，其步骤是从 模块开始，沿着被测程序的 顶层 的控制路径逐步向下测试，它有两种组合策略： 软件结构图 和深度优先策略 宽度优先策略 。

31．自底向上渐增式测试不需要编写 桩 模块，只需要编写 驱动 模块。

32．确认测试指检查软件的功能 与 性能 是否与 需求规格 说明书中确定的指标相符合，又称 有效性 测试。

33．确认测试在模拟环境下运用 黑盒 测试方法，由专门测试人员 和 用户 参加的测试。

34．确认测试开始前需要制定 测试 计划，结束后要写出 测试分析 报告。其测试用例要选用实际运用 的数据。

35．软件配置审查的任务是检查软件的所有文档资料的 完整性 和 正确性 。

36．调试也称 纠错 ，是在成功的测试之后才开始进行，其目的是确定错误的 原因 和 位置 ，并改正错误。

37．调试技术包括 简单调试 归纳法调试 演绎法调试 回溯法调试

38．回溯法调试是从程序产生错误的地方 出发，而归纳法调试是从 测试结果发现的线索 入手。

39．被测试程序不在机器上运行，而是采用人工检测和计算机辅助分析检测的手段称为

 静态 测试。

40．用等价类划分法设计一个测试用例时，使其覆盖 尽可能多的 尚未被覆盖的合理等价类。

41．用等价类划分法设计一个测试用例时，使其覆盖 一个 不合理等价类。

42．在单元测试时，需要为被测模块设计 驱动模块与桩模块 。

43．在集成测试时有两种测试方法，它们是 渐增式和非渐增式 。

44．软件测试是为了 发现错误 而执行程序的过程。

45．运行被测程序的方法称为 动态 测试。

46．动态测试中，主要测试软件功能的方法称为 黑盒 法。

47．选择测试用例，使得被测程序中每个判定的每个分支至少执行一次，这种逻辑覆盖标准称为 判定覆盖 。

48．要覆盖含循环结构的所有路径是不可能的，一般通过限制 循环次数 来测试。

49．用等价类划分法设计测试用例时，如果被测程序的某个输入条件规定了取值范围，则可确定一个合理的等在和 两个不合理的等价类 。

50．凭经验或直觉推测程序中可能存在的错误而设计测试用例的方法是 错误推测法 。

51．集成测试中的具体方法是 渐增式和非渐增式测试方法 。

52．确认测试阶段的两项工作是 进行确认测试和软件配置审查 。

53．在单元测试中，测试一个模块时，需要设计 驱动模块和桩模块 。

54．软件配置管理，简称SCM，它用于整个软件工程过程。其主要目标是： 标识变更　控制变更　确保变更正确地实现　报告有关变更

55．SCM是一组管理整个软件生存期各阶段中 变更 的活动。

56．基线的作用是把各阶段的开发工作划分得更加明确，便于检查与确认阶段成果。因此，基线可以作为项目的一个 检查点 。

1. 名词解释

1．软件测试：软件测试指为了发现软件中的错误而执行软件的过程。它的目标是尽可能多地发现软件中存在的错误，将测试结果作为纠错的依据。

2．静态测试：指被测试的程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。

3．动态测试：指通过运行程序发现错误

4．黑盒测试：指把测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求，又称为功能测试或数据驱动测试。

5．白盒测试：把测试对象看成一个打开的盒子，测试人员需了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。

6．语句覆盖：设计足够的测试用例，使被测程序中每个语句至少执行一次

7．判定覆盖：指设计足够的测试用例，使被测程序中每个判定表达式至少获得一次“真”值或“假”值，从而使程序的每个分支至少都通过一次，因此判定覆盖又称分支覆盖

8．条件覆盖：指设计足够测试用例，使判定表达式中每个条件的各种可能的值至少出现一次。

9．判定/条件覆盖：设计足够的测试用例，使得判定表达式中每个条件的所有可能取值至少出现一次，并使每个判定表达式所有可能的结果也至少出现一次。

10．条件组合覆盖：指设计足够的测试用例，使得每个判定表达式中条件的各种可能的值的组合都至少出现一次。

11．路径覆盖：设计足够的测试用例，覆盖被测程序中所有可能的路径

12．测试用例：指为寻找程序中的错误而精心设计的一组测试数据

13．驱动模块：指用来模拟被测模块的上级调用模块，其功能比真正的上级模块简单得多，它只完成接受测试数据，以上级模块调用被测模块的格式驱动被测模块，接收被测模块的测试结果并输出。

14．桩模块

15．单元测试

16．集成测试

17．确认测试

18．渐增式测试

19．非渐增式测试

20．调试

21．人的因素的含义

22．基线

23．软件配置管理

24．软件配置项

25.软件质量

1. 简答题

1．为什么说软件测试是软件开发中不可缺少的重要一环，但不是软件质量保证的安全网？

2．软件测试的目的是什么？为什么把软件测试的目的定义为只是发现错误？

3．软件测试应当遵循什么原则？为什么要遵循这些原则？

4．软件测试的步骤是什么？这些测试与软件开发各阶段之间的关系？

5．软件测试的过程是什么？

6．单元测试、集成测试和确认测试各自主要目标是什么？它们之间有什么不同？相互有什么关系？

7．什么是黑盒测试与白盒测试？它们都适应哪些测试？

8．简述软件测试与软件调试的区别。

9．软件配置管理的任务。

10．试述第三代界面的优点。

11．试述人机界面的设计过程。

12、SQA策略主要分哪三个阶段

13、测试计划应包括内容

14、软件测试阶段是如何划分的

15、简述软件测试过程

1. 综合题

1．变量的命名规则一般规定如下：变量名的长度不多于40个字符，第一个字符必须为英文字母，其他字母可以英文字母、数字以及下划线的任意组合。请用等价分类法设计测试用例。

2．某商场在“五一”期间，顾客购物时收费有4种情况：普通顾客一次购物累计少于100元，按A类标准收费（不打折），一次购物累计多于或等于100元，按B类标准收费（打9折）；会员顾客一次购物累计少于1000元，按C类标准收费（打8折），一次购物累计等于或多于1000元，按D类标准收费（打7折）。测试对象是按以上要求计算顾客收费模块，按照路径覆盖法设计测试用例。

3．下面是某程序的流程图：

（1）计算它的环路复杂性。

（2）为了完成基本路径测试，求它的一组独立的路径。

4.分析比较面向对象的软件测试与传统的软件测试的异同

1、传统的单元测试的对象是软件设计的最小单位——模块。当考虑面向对象软件时，单元的概念发生了变化，此时最小的可测试单位是封装的类或对象，而不再是个体的模块。传统单元测试主要关注模块的算法实现和模块接口间数据的传递，而面向对象的单元测试主要考察封装在一个类中的方法和类的状态行为。

面向对象软件没有层次的控制结构，因此传统的自顶向下和自底向上集成策略就不再适合，它主要有以下两种集成策略： 基于类间协作关系的横向测试；基于类间继承关系的纵向测试。

系统测试一般不考虑内部结构和中间结果，因此面向对象软件系统测试与传统的系统测试差别不大。面向对象软件测试的整体目标和传统软件测试的目标是一致的，即以最小的工作量发现尽可能多的错误，但是面向对象测试的策略和战术有很大不同。测试的视角扩大到包括复审分析和设计模型，此外，测试的焦点从过程构件(模块)移向了类。

1选择题

1．B 2．B 3．C 4．D 5．A 6．B 7．C 8．B 9．D 10．D

11．A 12．B 13．D 14．A 15．A 16．A 17．B 18．D 19．A 20．D

21．C 22．D 23．A 24．C 25．D 26．A 27．D 28．D 29．B 30．B

31．C 32．A 33．D 34．B 35．C 36．B 37．B 38．A 39．D 40．D

1. 填空题

1. 错误 测试结果

2. 文档资料 内部结构 测试用例 程序 错误 缺陷

3. 输入数据 输出数据

4. 动态测试 静态测试

5. 运行程序 测试用例 黑盒测试 白盒测试

6. 人工检测 计算机辅助静态分析

7. 编码质量 编码 软件产品

8. 静态分析 特性

9. 接口 需求规格 功能

10. 内部结构 处理过程 处理过程 测试

11. 结构 源程序 内部逻辑

12. 判定 语句覆盖 判定覆盖 条件覆盖 判定/条件覆盖 条件组合覆盖 路径覆盖

13. 条件组合 路径

14. 循环 循环次数

15. 控制流程图 环路 基本路径

16. 功能 等价类划分 边界值分析 错误推测 因果图

17. 功能 一句话 一个短语 等价类

18. 边界 边界值 第一个 最后一个

19. 错误推测法

20. 因果 功能说明 判定表 判定表

21. 测试 黑盒 白盒

22. 软件配置 测试配置 测试工具

23. 单元测试 集成测试 确认测试 系统测试

24. 单元测试 编码 详细设计

25. 模块 模块接口 局部数据结构 重要的执行路径 错误处理 边界条件

26. 驱动 桩 驱动模块 桩模块

27. 单元 组装 联合

28. 非渐增式测试 渐增式测试

29. 自顶向下结合 自底向上结合

30. 驱动 桩 顶层 软件结构图 深度优先策略 宽度优先策略

31. 桩 驱动

32. 功能 性能 需求规格 有效性

33. 黑盒 专门测试人员 用户

34. 测试 测试分析 实际运用

35. 完整性 正确性

36. 纠错 原因 位置

37. 简单调试 归纳法调试 演绎法调试 回溯法调试

38. 程序产生错误的地方 测试结果发现的线索

39．静态

40．尽可能多的

41．一个

42．驱动模块与桩模块

43．渐增式和非渐增式

44．发现错误

45．动态测试

46．黑盒法

47．判定覆盖

48．循环次数

49．两个不合理的等价类

50．错误推测法

51．渐增式和非渐增式测试方法

52．进行确认测试和软件配置审查

53．驱动模块和桩模块

54．标识变更　控制变更　确保变更正确地实现　报告有关变更

55．变更

56．检查点

1. 名词解释

1. 软件测试指为了发现软件中的错误而执行软件的过程。它的目标是尽可能多地发现软件中存在的错误，将测试结果作为纠错的依据。

2. 静态测试指被测试的程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。

3. 动态测试指通过运行程序发现错误。

4. 黑盒测试指把测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求，又称为功能测试或数据驱动测试。

5. 白盒测试指把测试对象看成一个打开的盒子，测试人员需了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。

6. 语句覆盖指设计足够的测试用例，使被测程序中每个语句至少执行一次。

7. 判定覆盖指设计足够的测试用例，使被测程序中每个判定表达式至少获得一次“真”值或“假”值，从而使程序的每个分支至少都通过一次，因此判定覆盖又称分支覆盖。

8. 条件覆盖指设计足够测试用例，使判定表达式中每个条件的各种可能的值至少出现一次。

9. 判定/条件覆盖指设计足够的测试用例，使得判定表达式中每个条件的所有可能取值至少出现一次，并使每个判定表达式所有可能的结果也至少出现一次。

10. 条件组合覆盖指设计足够的测试用例，使得每个判定表达式中条件的各种可能的值的组合都至少出现一次。

11. 路径覆盖指设计足够的测试用例，覆盖被测程序中所有可能的路径。

12. 测试用例指为寻找程序中的错误而精心设计的一组测试数据。

13. 驱动模块指用来模拟被测模块的上级调用模块，其功能比真正的上级模块简单得多，它只完成接受测试数据，以上级模块调用被测模块的格式驱动被测模块，接收被测模块的测试结果并输出。

14. 桩模块指用来代替被测试模块所调用的模块，其作用是返回被测试模块所需的信息。

15. 单元测试指对源程序中每一个程序单元进行测试，检查各个模块是否正确实现规定的功能，从而发现模块在编码中或算法中的错误。

16. 集成测试指在单元测试基础上，将所有模块按照设计要求组装成一个完整的系统进行的测试。也称组装测试或联合测试。

17. 确认测试指检查软件的功能与性能是否与需求规格说明书中确定的指标相符合，又称有效性测试。

18. 渐增式测试指逐个把未经过测试的模块组装到已经过测试的模块上去，进行集成测试。每加入一个新模块进行一次集成测试，重复此过程直到程序组装完毕。

19. 非渐增式测试指首先对每个模块分别进行单元测试，然后把所有的模块按设计要求组装在一起进行测试。

20. 调试指确定错误的原因和位置，并改正错误的过程，也称纠错。

21. 1）人对感知过程的认识，包括视觉、阅读时的认知心理、记忆、归纳与演绎推理等；

 2）用户已有的技能和行为方式；

 3）用户所要求的完成的整个任务以及用户对人机交互部分的特殊要求。

22. 已经通过正式复审和批准的某规约或产品，它因此可以作为进一步开发的基础，并且只能遵循正式的变化控制过程得到改变。

23. 软件配置管理，简称SCM，它用于整个软件工程过程。其主要目标是：标识变更、控制变更、确保变更正确地实现、报告有关变更。SCM是一组管理整个软件生存期各阶段中变更的活动。

24. 软件配置项是软件工程中产生的信息项，它是配置管理的基本单位，对已成为基线的SCI，虽然可以修改，但必须按照一个特殊的正确的过程进行评估，确认每一处的修改。

25.软件产品具有满足规定的或隐含要求能力要求有关的特征与特征总和(ISO 8492)。

简答题

1. ① 软件测试是软件开发中不可缺少的重要一环，原因是：

·       测试的工作量约占整个项目开发工作量的40%左右，几乎一半。如果是关系到人的生命安全的软件，测试的工作量还要成倍增加。

·       软件测试代表了需求分析、设计、编码的最终复审。

② 软件测试不是软件质量保证的安全网，因为软件测试只能发现错误，不能保证没有错误。

2. 软件测试的目的有：

① 软件测试是为了发现错误而执行程序的过程。

② 一个好的测试用例能够发现至今尚未发现的错误。

③ 一个成功的测试是发现了至今尚未发现的错误。

    软件测试的目标定义为只是发现错误，原因是软件测试可以有两个目标，一个是预防错误，另一个是发现错误。由于软件开发是人的创造性劳动，人的活动不可能完美无缺，错误可能发生在任何一个阶段，因此预防错误这一目标几乎是不可实现的，所以软件测试的目标定义为只是发现错误。

3. 软件测试应当遵循原则如下：

① 用例由输入数据和预期的输出数据两部分组成，因为这样便于对照检查，做到有的放矢。

② 用例不仅选用合理的输入数据，还要选择不合理的输入数据。因为当以特殊方式使用程序时，会突然发现程序中有许多错误，故使用预期的不合理的输入数据进行程序测试，比用合理的输入数据收获要大，从而能更多地发现错误，提高程序可靠性。

③ 除了检查程序是否做了它应该做的事，还应该检查程序是否做了它不应该做的事，因为如果程序做了它不应该做的事，即使程序能做它应该做的事，程序也是错误的。

④ 应制定测试计划并严格执行，因为这样可以排除随意性。

⑤ 长期保留测试用例，因为测试用例的设计耗费很大的工作量，而修改后的程序可能有新的错误，需要进行回归测试，故必须将测试用例作为文档保存，使测试具有可重复性，同时测试用例是将来系统维护测试与确认的依据，保存测试用例也为以后的维护提供方便。

⑥ 对发现错误较多的程序段，应进行更深入的测试，因为发现错误较多的程序段，其质量较差，同时在修改错误过程中又容易引入新的错误。

⑦ 程序员避免测试自己设计的程序，因为测试目的是找错。从心理学角度讲，程序员大多对自己的程序存有偏见，总认为没有错误或错误不大，另外程序员对需求规格说明的理解而引入的错误则更难发现，应该由别人或另外的机构来测试会更客观、更有效。

4. ① 软件测试的步骤如图7-3所示。



图7-3 软件测试的步骤

② 这些测试与软件开发各阶段之间的关系如图7-2所示。因为系统测试已经超出了软件工程的范围，所以图7-2中的系统测试不包括在内。

5. 软件测试是一个规则的过程，包括测试设计、测试执行以及测试结果比较等。

① 测试设计：根据软件开发各阶段的文档资料和程序的内部结构，利用各种设计测试用例技术精心设计测试用例。

② 测试执行：利用这些测试用例执行程序，得到测试结果。

③ 测试结果比较：将预期的结果与实际测试结果进行比较，如果二者不符合，对于出现的错误进行纠错，并修改相应文档。修改后的程序还要进行再次测试，直到满意为止。如果测试发现不了错误，可能由于测试配置考虑不周到，应考虑重新制定测试方案，设计测试用例。

6. 单元测试、集成测试和确认测试各自主要目标是：

① 单元测试的主要目标是检查各个模块是否正确实现规定的功能，从而发现模块在编码中或算法中的错误。

② 集成测试的主要目标是检查与设计相关的软件体系结构的有关问题。

③ 确认测试的主要目标是检查已实现的软件是否满足需求规格说明书中确定的各种需求。

单元测试、集成测试和确认测试之间的不同如下：

① 测试内容不同：单元测试集中于单个模块的功能和结构检验，其测试内容主要包括模块接口、局部数据结构、重要的执行路径、错误处理和边界测试；集成测试集中于模块组合的功能和软件结构检验，其测试内容主要包括模块组装中可能出现的问题，即数据穿过接口可能丢失、一个模块可能破坏另一个模块的内容、子功能组装可能不等于主功能、全程数据结构问题、误差累积问题；确认测试集中于论证软件需求的可追溯性，主要包括测试软件功能和性能是否与软件需求一致、测试软件配置的所有程序与文档是否正确完整而且一致。

② 测试的方法不同：单元测试总是使用白盒测试法，为被测模块设计驱动模块和桩模块；集成测试使用渐增式测试和非渐增式测试，渐增式测试又有分为自顶向下结合法和自底向上结合法；确认测试总是使用黑盒测试法。

③ 发现的错误不同：单元测试发现的错误主要是在编码阶段产生的错误，集成测试发现的错误主要是在设计阶段产生的错误，确认测试发现的错误主要是在需求分析阶段产生的错误。

④ 涉及的文档不同：单元测试涉及编码和详细设计文档，集成测试涉及详细设计文档和概要设计文档，确认测试涉及软件需求规格说明书和用户手册。

三者相互关系是：单元测试、集成测试和确认测试是顺序实现的。首先单元测试对各个模块进行测试，然后集成测试以单元测试为基础，将所有已测模块按照设计要求组装成一个完整的系统，对模块组合的功能和软件结构检验进行测试，最后确认测试是以集成测试为基础，测试集成的软件是否满足需求规格说明书中确定的各种需求。

7. ① 黑盒测试指把测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求，又称为功能测试或数据驱动测试。

② 白盒测试指把测试对象看成一个打开的盒子，测试人员需了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。

③ 白盒测试适应的测试有单元测试、逻辑覆盖（按逻辑覆盖程度不同，有语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。）、循环覆盖（限制循环次数，有单循环和嵌套循环）和基本路径测试。

④ 黑盒测试适应的测试有：确认测试、等价类划分、边界值分析、错误推测和因果图。

8. 软件测试与软件调试在目的、技术和方法等方面存在很大的区别，主要表现在：

① 测试从一个侧面证明程序员的失败，而调试是为了证明程序员的正确。

② 测试从已知条件开始，使用预先定义的程序，且有预知的结果，不可预见的只是程序是否通过测试。调试一般以不可知的内部条件开始，除统计性调试外，结果是不可预见的。

③ 测试是有计划的，并要进行测试设计，而调试是不受时间约束的。

④ 测试是一个发现错误、改正错误、重新测试的过程，而调试是一个推理过程。

⑤ 测试的执行是有规程的，而调试的执行往往要求程序员进行必要的推理及知觉的飞跃。

⑥ 测试经常由独立的测试组在不了解软件设计的前提下完成，而调试必须由了解详细设计的程序员完成。

⑦大多数测试的执行和设计可由工具支持，而调试时，程序员能利用的工具主要是调试器。

12、以检测为重：产品制成之后进行检测，只能判断产品质量，不能提高产品质量。以过程管理为重：把质量的保证工作重点放在过程管理上，对制造过程中的每一道工序都要进行质量控制。以新产品开发为重：在新产品的开发设计阶段，采取强有力的措施来消灭由于设计原因而产生的质量隐患。

13、一个测试计划应包括：产品基本情况、测试需求说明、测试策略和记录、测试资源配置计划表、问题跟踪报告、测试计划的评审、结果等。

14、软件测试的阶段划分为：规格说明书审查；系统和程序设计审查；单元测试；集成测试；确认测试；系统测试；验收测试。

15、软件测试过程主要包括如下6个活动：测试计划；测试需求分析；测试设计；测试规程实现；测试执行；总结生成报告。

1. 综合题

1．① 划分等价类下表。

等价类划分表

|  |  |  |
| --- | --- | --- |
| 输入条件 | 合理等价类  | 不合理等价类  |
| 长度  | (1)小于40个字符 (2)等于40个字符  | (5)大于40个字符  |
| 第一个字符  | (3)英文字母  | (6)非英文字母  |
| 其他字母  | (4)英文字母、数字或下划线的任意组合  | (7)空格 (8)标点符号 (9)运算符号 (10)其他可显示字符  |

② 设计测试用例下表。

测试用例表

|  |  |  |
| --- | --- | --- |
| 测试数据  | 测试范围  | 期望结果  |
| (1) s\_name12  | 等价类(1),(3),(4)  | 有效  |
| (2) a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s\_tr  | 等价类(2),(3),(4)  | 有效  |
| (3) a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s\_trff  | 等价类(5)  | 无效  |
| (4) 234name  | 等价类(6)  | 无效  |
| (5) ab gh  | 等价类(7)  | 无效  |
| (6) ab!2f  | 等价类(8)  | 无效  |
| (7) fg+ghh  | 等价类(9)  | 无效  |
| (8) H@gh  | 等价类(10)  | 无效  |

2．① 被测模块的程序流程图如图所示。



被测模块的程序流程图

② 按照路径覆盖法设计测试用例如下：

是会员，累计消费900元， 覆盖路径127

是会员，累计消费2000元， 覆盖路径128

不是会员，累计消费80元， 覆盖路径134

不是会员，累计消费300元， 覆盖路径135

3．(a) 环路复杂性=判断数＋1＝6＋1＝7（个）

(b) 路径1： 0 —①— 13 — 19

 路径2： 0 —②—③— 14 — 19

 路径3： 0 —②—④—⑤— 15 — 19

 路径4： 0 —②—④—⑥—⑦— 16 — 19

 路径5： 0 —②—④—⑥—⑧—⑨— 17 — 19

 路径6： 0 —②—④—⑥—⑧—⑩— 18 — 19

 路径7： 0 —②—④—⑥—⑧—⑩— 12 — 19

4．(1) 划分等价类见表1。

表1 等价类划分表

|  |  |  |
| --- | --- | --- |
| 输入条件 | 合理等价类  | 不合理等价类  |
| 长度  | (1) 小于40个字符 (2) 等于40个字符  | (5) 大于40个字符  |
| 第一个字符  | (3) 英文字母  | (6) 非英文字母  |
| 其他字母  | (4) 英文字母、数字或下划线的任意组合  | (7) 空格 (8) 标点符号 (9) 运算符号 (10) 其他可显示字符  |

(2) 设计测试用例见表2。

表2 测试用例表

|  |  |  |
| --- | --- | --- |
| 测试数据  | 测试范围  | 期望结果  |
| (1)s\_name12  | 等价类(1),(3),(4)  | 有效  |
| (2)a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s\_tr  | 等价类(2),(3),(4)  | 有效  |
| (3)a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s\_trff  | 等价类(5)  | 无效  |
| (4)234name  | 等价类(6)  | 无效  |
| (5)ab gh  | 等价类(7)  | 无效  |
| (6)ab!2f  | 等价类(8)  | 无效  |
| (7)fg+ghh  | 等价类(9)  | 无效  |
| (8)H@gh  | 等价类(10)  | 无效  |

5．等价划分法

① 划分等价类并编号，如下表所示。

十六进制整型常量输入条件的等价类表

|  |  |  |
| --- | --- | --- |
|   输入数据  | 合理等价类  | 不合理等价类  |
| 十六进制整数  | 1. 0x或0X开头1～2位数字串 2. 以－0x打头的1～2位数字串  | 3. 非0x或非－打头的串 4. 含有非数字且（a,b,c,d,e,f）以外字符 5. 多于5个字符 6. －后跟非0的多位串 7. －0后跟数字串 8. －后多于3个数字  |
| 十六进制数范围  | 9. 在－7f～7f之间  | 10. 小于－7f 11. 大于7f  |

② 为合理等价类设计测试用例，表中有三个合理等价类，两个例子。

|  |  |  |
| --- | --- | --- |
| 测试数据  | 期望结果  | 覆盖范围  |
| 0×23  | 显示有效输入  | 1，9  |
| －0×15  | 显示有效输入  | 2，9  |

③ 为每个不合理等价类至少设计一个测试用例。

|  |  |  |
| --- | --- | --- |
| 测试数据  | 期望结果  | 覆盖范围  |
| 2  | 显示无效输入  | 3  |
| G12  | 显示无效输入  | 4  |
| 123311  | 显示无效输入  | 5  |
| －1012  | 显示无效输入  | 6  |
| －011  | 显示无效输入  | 7  |
| －0134  | 显示无效输入  | 8  |
| －0x777  | 显示无效输入  | 10  |
| 0x87  | 显示无效输入  | 11  |

6．(1) 被测模块的程序流程图如下图所示。



(2) 按照路径覆盖法设计测试用例如下：

是会员，累计消费900元，覆盖路径127

是会员，累计消费2000元，覆盖路径128

不是会员，累计消费80元，覆盖路径134

不是会员，累计消费300元，覆盖路径135

2．



4．变量的命名规则一般规定如下：变量名的长度不多于40个字符，第一个字符必须为英文字母，其他字母可以是英文字母、数字以及下划线的任意组合。请用等价分类法设计测试用例。

5．根据下列条件使用等价划分法设计测试用例。

某8位微机，其十六进制常数定义为：以0x或0X开头的数是十六进制整数，其值的范围是－7f～7f（表示十六进制的大小写字母不加区别），如0X13，0X6A，－0X3c。

6．某商场在“五一”期间，顾客购物时收费有4种情况：普通顾客一次购物累计少于100元，按A类标准收费（不打折），一次购物累计多于或等于100元，按B类标准收费（打9折）；会员顾客一次购物累计少于1000元，按C类标准收费（打8折），一次购物累计等于或多于1000元，按D类标准收费（打7折）。测试对象是按以上要求计算顾客收费的模块，按照路径覆盖法设计测试用例。