软件工程期末试题及答案（六）

选择题

1．软件测试的目的是（ ）。

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（ ）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（ ）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（ ）。

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）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（ ）。

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（ ）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（ ）。

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（ ）。

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（ ）。

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）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（ ）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（ ）。

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）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（ ）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（ ）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（ ）。

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（ ）

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）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（ ）

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( )。

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( )。

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）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( )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( )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）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( )。

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( )。

A）功能 B）内部结构

C）处理过程 D）条件

29．( )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( )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（ ）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（ ）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( )。

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( )。

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( )。

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( )。

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( )。

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）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（　）。

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（ ）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 ，将 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 和程序的 ，精心设计一组 ，利用这些实例执行 ，找出软件中潜在的各种 和 。

3．测试用例由 和预期的 两部分组成。

4．软件测试方法一般分为两大类： 方法和 方法。

5．动态测试通过 发现错误。根据 的设计方法不同，动态测试又分为 与 两类。

6．静态测试采用 和 的手段对程序进行检测。

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

8．计算机辅助静态分析利用 工具对测试程序进行 分析。

9．黑盒法只在软件的 处进行测试，依据 说明书，检查程序是否满足 要求。

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

11．白盒测试是 测试，被测对象是 ，以程序的 为基础设计测试用例。

12．逻辑覆盖是对程序内部有 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 、 、 、 、 和 6种覆盖技术。

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

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

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

16．黑盒测试是 测试，用黑盒技术设计测试用例有4种方法： 、 、

和 。

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

18．边界值分析是将测试 情况作为重点目标，选取正好等于、刚刚大于或刚刚小于 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 元素和 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 。

20．因果图的基本原理是通过画 图，把用自然语言描述的 转换为 ，最后为 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 方法。通常先用 法设计基本的测试用例，再用 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： 、 和 。

23．软件测试一般经过4个测试： 、 、 和 。

24． 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 和 的文档。

25．单元测试主要测试 的5个基本特征： 、 、 、 和 。

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

27．集成测试指在 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 测试或 测试。

28．集成测试的方法有两种： 和 。

29．渐增式测试有两种不同的组装模块的方法： 和 。

30．自顶向下渐增式测试不需要编写 模块，只需要编写 模块，其步骤是从 模块开始，沿着被测程序的 的控制路径逐步向下测试，它有两种组合策略： 和 。

31．自底向上渐增式测试不需要编写 模块，只需要编写 模块。

32．确认测试指检查软件的 与 是否与 说明书中确定的指标相符合，又称 测试。

33．确认测试在模拟环境下运用 测试方法，由 和 参加的测试。

34．确认测试开始前需要制定 计划，结束后要写出 报告。其测试用例要选用 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 和 。

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

37．调试技术包括 、 、 和 。

38．回溯法调试是从 出发，而归纳法调试是从 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

测试。

40．用等价类划分法设计一个测试用例时，使其覆盖 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 不合理等价类。

42．在单元测试时，需要为被测模块设计 。

43．在集成测试时有两种测试方法，它们是 。

44．软件测试是为了 而执行程序的过程。

45．运行被测程序的方法称为 测试。

46．动态测试中，主要测试软件功能的方法称为 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 。

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 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 。

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 。

51．集成测试中的具体方法是 。

52．确认测试阶段的两项工作是 。

53．在单元测试中，测试一个模块时，需要设计 。

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： 、 、

、 。

55．SCM是一组管理整个软件生存期各阶段中 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 。

1. 名词解释

1．软件测试

2．静态测试

3．动态测试

4．黑盒测试

5．白盒测试

6．语句覆盖

7．判定覆盖

8．条件覆盖

9．判定/条件覆盖

10．条件组合覆盖

11．路径覆盖

12．测试用例

13．驱动模块

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

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。

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。

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。

1. 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。

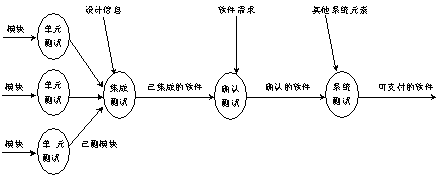


图7-3 软件测试的步骤

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。

1. 综合题

1．① 划分等价类下表。

等价类划分表

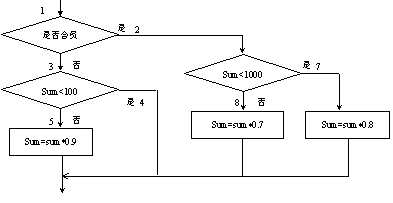
|  |  |  |
| --- | --- | --- |
| 输入条件 | 合理等价类 | 不合理等价类 |
| 长度 | (1)小于40个字符  (2)等于40个字符 | (5)大于40个字符 |
| 第一个字符 | (3)英文字母 | (6)非英文字母 |
| 其他字母 | (4)英文字母、数字或下划线的任意组合 | (7)空格 (8)标点符号 (9)运算符号  (10)其他可显示字符 |

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 |

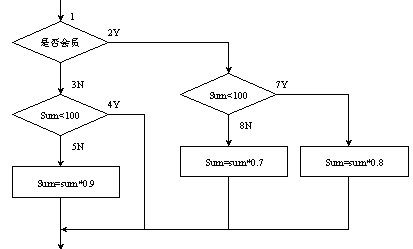
② 为合理等价类设计测试用例，表中有三个合理等价类，两个例子。

|  |  |  |
| --- | --- | --- |
| 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：

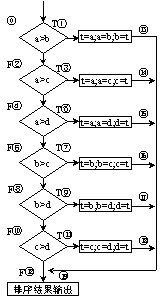
是会员，累计消费900元，覆盖路径127

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。

一、填空题

1．程序设计语言的特性主要有心理特性、工程特性和 技术特性 三个方面。

2．程序语言的工程特性主要表现为可移植性、 开发工具的可利用性 、软件的可重用性和可维护性。

3．为了提高程序的易读性，同时为减少错误，提高软件开发效率，编码时应注意养成良好的 程序设计风格 。

4．程序加注释对理解程序提供了明确指导，根据作用不同注释分 序言性注释 和 功能性注释 。

5．软件需求分析之后，软件的设计、编码、测试与语言的特性有很大的关系，这个特性主要是语言的 技术 特性。

6．语句构造的原则是 简单直接 ，不能为了追求效率而使代码复杂化。

7． FORTRAN 语言上世界上第一个被正式推广应用的计算机语言，它不仅面向科学计算，数据处理能力也极强。

8． 汇编语言 是面向机器的，可以完成高级语言无法满足要求的特殊功能，如与外部设备之间的一些接口操作。

9．为了使数据定义更容易理解和维护，一个语句说明多个变量时，各变量名按

字典 排列。

10． 编码 是将详细设计得到的处理过程的描述转换为基于某种计算机语言的程序。

二、选择题

1．结构化程序设计主要强调的是（ B）

A.程序的执行效率 B程序的易读性 C程序的规模 D程序可移植性

2．下列选项中，不属于程序设计语言心理特性的表现形式的是（C）。

A简洁性 B歧义性 C可移植性 D局部性和顺序性

3．下列选项中与选择程序设计语言无关的因素是（ A）。

A程序设计风格 B软件执行的环境 C软件开发的方法 D项目的应用领域

4．项目应用领域是选择程序设计语言的关键因素，下列不属于项目应用领域类型的选项是（D）。

A系统软件 B数据处理和数据库应用 C人工智能 D.UNIX操作系统

5．不适合做科学工程计算的语言是（ D）。

A.FORTRAN语言 B.C语言 C.Pascal语言 D.Lisp语言

6．源程序中的序言性注释应置于每个模块的其始部分，该注释内容不应有（B）。

A模块的用途 B语句的功能 C数据描述 D模块的功能

7．下列适用于开发人工智能领域的语言是（ C ）

A Ada语言 B C 语言 C Prolog语言 D Cobol语言

8．程序设计语言的工程特性其中之一表现在（ B ）

A开发工具的可利用性 B简洁性

C数据结构的描述性 D数据库的可操作性

9．下列不合适作为数据处理与数据应用的语言是（ A ）

A Cobol语言 B SQL语言 C 4GL D Ada语言

10．提高程序效率的根本途径并非在于（ D ）

A选择良好的设计方法 B选择良好的数据结构

C选择良好的算法 D 对程序语句做调整

三、简答题

1、程序语言有哪些共同特征?

答：程序语言的共同特征(似为特性更符合题意)是指程序设计语言的语言特性，包括心理特性、工程特性和技术特性三个方面。语言的心理特性对人机通信的质量有主要的影响；语言的工程特性对软件开发成功一否有重要的影响，此外，语言的技术特性也会影响软件设计的质量。

2、在项目开发时选择程序设计语言通常考虑哪些因素?

答：通常要考虑的因素有：

项目的应用领域、软件开发的方法、软件执行的环境、算法和数据结构的复杂性以及软件开发人员的知识等。

3、第4代语言(4GL)有哪些主要特征?

答：4GL的主要特征是：

(1)友好的用户界面

(2)兼有过程性和非过程性双重特性

(3)高效的程序代码

(4)完备的数据库

(5)应用程序生成器

4、举例说明各种程序设计语言的特点及适用范围?

答：以常见的一些编程语言为例：

FORTRAN语言，它不仅面向科学计算，数据处理能力也极强。适用于科学工程计算

Pascal语言，具有很强的数据和过程结构化能力，它语言简明，数据类型丰富，程序结构严谨。它是第一个体现结构化编程思想的语言。用于描述结构化算法和科学计算,操作系统的编写

C语言：功能强大，支持复杂的数据结构，可大量运用指针，具有丰富灵活的操作运算符及数据处理操作符，它还具有汇编语言的某些特征，使程序运行效率高。可用于许多领域，特别是在操作系统等方面有较大优势。

Coble语言，它具有极强的数据定义能力，程序说明与硬件环境说明分开、数据描述与算法描述分开。结构严谨层次分明，广泛用于商业数据处理。

SQL：结构化查询语言，用于对数据库进行存取管理。

汇编语言和Ada语言，主要用于实时处理和编写系统软件。

Prolog、Lisp语言。Lisp是一种函数型语言,特别适用于组合问题中符号运算和表处理，用于定理证明、树的搜索和其他问题的求解。Prolog是一种逻辑型语言，它提供了支持知识表示的特性，每一个程序由一组表示事实、规则和推理的句子组成，比较接近与自然语言。这两种语言主要用于人工智能系统开发。

C++、Java语言,是面向对象的设计语言，用于面向对象的方开发软件。

5、什么是程序设计风格?为了具有良好的设计风格，应注意哪些方面的问题?

答：程序设计风格是指一个人编制程序时所表现出来的特点、习惯、逻辑思路等。

要形成良好的程序设计风格，应从源程序文档化、数据说明、语句构造、输入输出和追求效率几个方面加以注意。

一、填空题

1．软件原来没有帮助信息，使用不方便，现在要增加帮助信息。这种维护性活动称之为 完善性维护 。

2．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。

3．调试技术包括 简单调试 、 归纳法调试 、 演绎法调试 和 回溯法调试 。

4．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。

二、选择题

1.某个应用软件原来是在DOS环境下运行的，现在要把它移植到Windows环境下来运行，这种修改软件的过程称为（ B）

A校正性维护 B适应性维护 C完善性维护 D预防性维护

2.产生软件维护的副作用是指（ D）

A开发软件时的错误 B运行时的错误

C隐含的错误 D因修改软件而造成的错误

3．软件维护是指（ C）

A维护软件的正常运行 B软件的配置更新

C对软件的改进，适应和完善 D软件开发期的一个阶段

三、简答题

1、软件维护有哪些内容？

a、校正性维护b、适应性维护c、完善性维护d、预防性维护

2.软件维护的特点是什么？

a、非结构化维护和结构化维护

b、维护的困难性

c、软件强维护的费用

3.软件维护的流程是什么？

 a、制定申请维护报告b、审查申请报告并批准c、进行维护并作详细记录d、覆审

4.软件维护的副作用有哪些？

 编码、数据、文档副作用

5.什么是软件的可维护性？可维护性的度量的特性是什么？

软件的可维护性：软件能够被理解、校正、适应及增强功能的容易程度。

可维护性的度量的特性是：可理解性、可测试性、可修改性、可靠性、可移植性、可使用性和效率。

6.提高可维护性的方法有哪些？

A、建立明确的软件质量目标。B、利用先进的软件开发技术和工具。

C、建立明确的质量保证工作。D、选择可维护的程序设计语言。

E、改进程序文档。