

一、 单选: 共 20 小题, 每小题 2 分, 满分 40 分。

- 软件测试的目的: (c)
 - 避免软件开发中出现的错误
 - 发现软件开发中出现的错误
 - 尽可能发现并排除软件中潜藏的错误, 提高软件的可靠性
 - 修改软件中出现的错误
- 软件测试是采用 (a) 执行软件的活动。
 - 测试用例
 - 输入数据
 - 测试环境
 - 输入条件
- 导致软件缺陷的最大原因是: (a)
 - 软件需求说明书
 - 设计方案
 - 编码
 - 维护
- 在下列描述中, 关于一个软件缺陷状态完整变化的错误描述是 (d)
 - 打开——修复——关闭
 - 打开——关闭
 - 打开——保留
 - 激活——修复——重新打开
- 在下列描述中, 关于测试与调试的说法错误的是 (d)
 - 测试是显示错误的行为; 而调试是推理的过程;
 - 测试显示开发人员的错误。调试是开发人员为自己辩护;
 - 测试能预期和可控。调试需要想象、经验和思考;
 - 测试必须在详细设计已经完成的情况下才能开始; 没有详细设计的信息调试不可能进行。
- 某次程序调试没有出现预计的结果, 下列 (b) 不可能是导致出错的原因。
 - 变量没有初始化
 - 编写的语句书写格式不规范
 - 循环控制出错
 - 代码输入有误
- 软件缺陷修复的代价最高的阶段为 (a)
 - 发布阶段
 - 需求阶段
 - 设计阶段
 - 编码阶段
- 不属于逻辑覆盖方法的是 (d)。
 - 组合覆盖
 - 判定覆盖
 - 条件覆盖
 - 接口覆盖
- (d) 是选择若干个测试用例, 运行被测程序, 使得程序中的每个可执行语句至少执行一次。
 - 条件覆盖
 - 组合覆盖
 - 判定覆盖
 - 语句覆盖
- (a) 是设计足够多的测试用例, 使得程序中每个判定包含的每个条件的所有情况 (真/假) 至少出现一次, 并且每个判定本身的判定结果 (真/假) 也至少出现一次。
 - 判定 - 条件覆盖
 - 组合覆盖
 - 判定覆盖
 - 条件覆盖
- 软件测试是软件质量保证的重要手段, 下述哪种测试是软件测试的最基础环节? (b)
 - 功能测试
 - 单元测试
 - 结构测试
 - 验收测试
- 单元测试的主要任务不包括 (b)。
 - 出错处理
 - 全局数据结构
 - 独立路径
 - 模块接口
- 如右图所示的 N-S 图, 至少需要 (b) 个测试用例完成逻辑覆盖。
 - 12
 - 48
 - 27
 - 18
- 单元测试中用来模拟实现被测模块需调用的其他功能模块的是 (b)。
 - 驱动模块
 - 桩模块
 - 主控模块
 - 真实的被调用模块
- 集成测试计划应该在 (b) 阶段末提交。
 - 需求分析
 - 概要设计
 - 详细设计
 - 单元测试完成
- 下列关于程序效率的描述错误的是 (c)。
 - 提高程序的执行速度可以提高程序的效率
 - 降低程序占用的存储空间可以提高程序的效率
 - 源程序的效率与详细设计阶段确定的算法的效率无关
 - 好的程序设计可以提高效率
- 下列 (b) 是对程序流程图进行简化后得到的, 它可以更加突出的表示程序控制流的结构, 且不包含复合条件。
 

A. DD-路径图 B. 控制流图 C. MM-路径图 D. 模块调用图

18. 自底向上增量式集成测试中, 下面 (c) 描述是正确的。

A. 测试由桩模块控制 B. 最上面的模块最先测试 C. 父单元用测试过的子单元测试 D. 包含树的深度优先或广度优先遍历过程

19. 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 (d)。

A. 未知 B. 反比 C. 相等 D. 正比

20. 针对是否对无效数据进行测试, 可以将等价类测试分为 (b)

(1) 标准(一般)等价类测试

(2) 健壮等价类测试

(3) 弱等价类测试

(4) 强等价类测试

A. 3) 4) B. 1) 2) C. 1) 3) D. 2) 4)

二、 判断题: 共 20 小题, 每题 1 分, 满分 20 分)

1. 一个程序中所含有的路径数与程序的复杂程度有着直接的关系。 (v)

2. 结构性测试是根据软件的规格说明来设计测试用例。 (x)

3. 错误推测法是根据输出对输入的依赖关系来设计测试用例的。 (x)

4. 软件缺陷属性包括缺陷标识、缺陷类型、缺陷严重程度、缺陷产生可能性、缺陷优先级、缺陷状态、缺陷起源、缺陷来源、缺陷原因。 (v)

5. 对于一个含有 n 个变量的程序, 采用边界值健壮性测试方法来测试程序会产生 $6n+1$ 个测试用例。 (v)

6. 数据流测试是主要用作路径测试的真实性检查。两种形式分别为定义/使用测试、基于程序片的测试。 (v)

7. 软件只要经过严格严谨的内部测试之后, 可以做到没有缺陷。 (x)

8. 测试用例应由测试输入数据和对应的实际输出结果这两部分组成。 (x)

9. 测试是可以穷尽的。 (x)

10. 测试自动化是万能的。 (x)

11. 软件缺陷可能会被修复, 可能会被保留或者标识出来。 (v)

12. 每一个软件项目都有一个最优的测试量。 (v)

13. 黑盒测试往往会造成测试用例之间可能存在严重的冗余和未测试的功能漏洞。 (v)

14. 代码审查工作属于静态测试。 (v)

15. 软件测试是一个过程, 包含若干活动, 运行软件进行测试只是活动之一。 (v)

16. 回归测试是在软件修改后再次运行以前为查找错误而执行程序曾用过的测试用例。 (v)

17. 集成测试是为确定软件系统是否满足验收标准以及使客户决定是否接受而进行的正式测试。 (x)

18. 测试按照测试层次可以划分成为单元测试、集成测试和系统测试。 (v)

19. 只要能够达到 100% 的逻辑覆盖率, 就可以保证程序的正确性。 (x)

20. 永远有缺陷类型会在测试的一个层次上被发现, 并且能够在另一个层次上逃避检测。 (v)

三、 简答题: 共 4 小题, 每题 5 分, 满分 20 分。

1. 优秀的软件测试工程师应具备哪些素质?

答: 具有探索精神、具有良好的计算机编程基础、故障排除能手、坚持不懈的精神、具有创新精神和超前意识、追求完美、判断准确、具有整体观念, 对细节敏感、团队合作精神, 沟通能力

2. 有二元函数 $f(x,y)$, 其中 $x \in [1,12]$, $y \in [1,31]$; 请写出该函数采用基本边界值分析法设计的测试用例。

答: { <1,15>, <2,15>, <11,15>, <12,15>, <6,15>, <6,1>, <6,2>, <6,30>, <6,31> }

3. 黑盒测试与白盒测试各有哪些优缺点, 应该如何结合才能解决漏洞和冗余问题?

答: 功能性测试具有两大优点: 功能性测试与软件如何实现无关; 测试用例开发可以与实现并行进行, 因此可以压缩总的项目开发时间。缺点: 测试用例之间可能存在严重的冗余, 还会有位测试的软件漏洞。结构性测试局限于已经完成的代码行为当中, 离代码太近。因此可以结构性测试指标去解决冗余和漏洞问题。如果发现同一条程序路径被多个功能性测试用例遍历, 就可以怀疑这种冗余不会发生新的缺陷, 如果没有达到一定的 DD—路径覆盖, 则可知在功能性测试用力中存在漏洞。因此路径测试可以提供作为功能性测试交叉检查的一组指标。

4. 有一段程序如下，请设计测试用例以满足语句覆盖要求。

```
void DoWork (int x,int y,int z){
    int k=0,j=0;
    if ( (x>3)&&(z<10) )
    {
        k=x*y-1;
        j=sqrt(k);
    }
    //语句块 1
    if ( (x==4)|| (y>5) )
    {
        j=x*y+10;
    } //语句块 2
    j=j%3;
    //语句块 3
}
```

答：要实现 DoWork 函数的语句覆盖，只需设计一个测试用例就可以覆盖程序中的所有可执行语句。

四、 综合题：共 2 小题， 每题 10 分， 满分 20 分。

1. 使用基本路径测试方法，为以下程序段设计测试用例。

- (1) 画出程序的控制流程图，编号已经给出
- (2) 计算程序的循环复杂度，导出程序基本路径集中的独立路径条数
- (3) 导出基本路径集，确定程序的独立路径
- (4) 根据 (3) 中的独立路径，设计测试用例（确保基本路径集中的每一条路径的执行）的输入数据和预期输出。

```
void Do (int X,int A,int B){
1      if ( (A>1)&&(B=0) )
2          X = X/A;
3      if ( (A=2)|| (X>1) )
4          X = X+1;
5  }
```

由于控制流程图假设的是单条件，因此对于复合条件，可将其分解为多个单个条件，并映射成控制流程图。

1: A>1; 2: B=0 ; 3: X = X/A ; 4: A=2 ; 5: X>1 ; 6: X = X+1; 7: }

2. 场景要求：“.....对功率大于 50 马力的机器、维修记录不全或已运行 10 年以上的机器，应给予优先的维修处理.....”。这里假定，“维修记录不全”和“优先维修处理”均已在别处有更严格的定义。请建立决策表。

- (1) 确定规则的个数
- (2) 列出所有的条件桩和动作桩
- (3) 填入条件项
- (4) 填入动作项，得到初始决策表
- (5) 简化决策表，合并相似规则

软件测试技术 (B 卷) 答案及评分标准

一、单选题 (20×2 = 40 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	C	A	A	D	D	B	A	D	D	A
题号	11	12	13	14	15	16	17	18	19	20
答案	B	B	B	B	B	C	B	C	D	B

二、判断题 (20×1 = 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	√	×	×	√	√	√	×	×	×	×
题号	11	12	13	14	15	16	17	18	19	20
答案	√	√	√	√	√	√	×	√	×	√

三、简答题 (4×5 = 20 分)

1. 具有探索精神、具有良好的计算机编程基础、故具有整体观念，对细节敏感、团队合作精神，沟通

2. 具有良好的计算机编程基础，有一定的软件开发

3. 善于同软件开发人员沟通；善于同领导沟通

4. 掌握一些自动化测试工具；善于学习的能力

5. 提高自己的表达能力；了解业务知识

6. 具有探索精神；故障排除能手

7. 坚持不懈的精神；具有创新精神和超前意识

8. 追求完美；判断准确；具有整体观念，对细节敏感

9. { <1,15>, <2,15>, <11,15>, <12,15>, <6,15>, <6,17>, <6,18>, <6,20>, <6,30>, <6,31> }

10. 功能性测试具有两大优点：功能性测试与软件如何实现无关；测试用例开发可以与实现并行进行，因此可以压缩总的项目开发时间。缺点：测试用例之间可能存在严重的冗余，还会有位测试的软件漏洞。结构性测试局限于已经完成的代码行为当中，离代码太近。因此可以结构性测试指标去解决冗余和漏洞问题。如果发现同一条程序路径被多个功能性测试用例遍历，就可以怀疑这种冗余不会发生新的缺陷，如果没有达到一定的 DD—路径覆盖，则可知在功能性测试用力中存在漏洞。因此路径测试可以提供作为功能性测试交叉检查的一组指标。

11. 要实现 DoWork 函数的语句覆盖，只需设计一个测试用例就可以覆盖程序中的所有可执行语句。
➤ 测试用例输入为：{ x=4、y=5、z=5 }
- ```
graph TD; 1((1)) --> 2((2)); 2 --> 3((3)); 3 --> 4((4)); 4 --> 5((5)); 5 --> 6((6)); 6 --> 7((7)); 7 --> 4;
```

四、综合题 (每题 10 分，共计 20 分)

1.
- (1) 画出控制流图,如右图所示:
- (2) 计算环形复杂度:  
10 (条边) - 7 (个节点) + 2 = 5
- (3) 导出独立路径 (用语句编号表示)  
路径 1: 1→2→3→4→5→6→7  
路径 2: 1→4→5→6→7  
路径 3: 1→2→4→6→7  
路径 4: 1→2→4→5→7  
路径 5: 1→2→3→4→5→7

(4) 测试用例

| 用例号 | 路径            | 输入数据 |   |   | 预期输出 |
|-----|---------------|------|---|---|------|
|     |               | A    | B | X |      |
| TC1 | 1→2→3→4→5→6→7 | 3    | 0 | 6 | 3    |
| TC2 | 1→4→5→6→7     | 0    | 1 | 3 | 4    |

|     |             |   |   |   |   |
|-----|-------------|---|---|---|---|
| TC3 | 1→2→4→6→7   | 2 | 1 | 1 | 2 |
| TC4 | 1→2→4→5→7   | 3 | 1 | 0 | 0 |
| TC5 | 1→2→3→4→5→7 | 3 | 0 | 3 | 1 |

## 2. 解答:

(1) 确定规则的个数: 这里有 3 个条件, 每个条件有两个取值, 故应有  $2 \times 2 \times 2 = 8$  种规则。

(2) 列出所有的条件桩 and 动作桩:

|        |              |
|--------|--------------|
| 条<br>件 | 功率大于 50 马力吗? |
|        | 维修记录不全吗?     |
|        | 运行超过 10 年吗?  |
| 动<br>作 | 进行优先处理       |
|        | 作其他处理        |

(3) 填入条件项。可从最后 1 行条件项开始, 逐行向上填满。

如第三行是: Y N Y N Y N Y N

第二行是: Y Y N N Y Y N N

等等。

(4) 填入动作项。这样便得到形如图的初始判定表。

|        |              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------------|---|---|---|---|---|---|---|---|
| 条<br>件 | 功率大于 50 马力吗? | Y | Y | Y | Y | N | N | N | N |
|        | 维修记录不全吗?     | Y | Y | N | N | Y | Y | N | N |
|        | 运行超过 10 年吗?  | Y | N | Y | N | Y | N | Y | N |
| 动<br>作 | 进行优先处理       | x | x | X |   | X |   | X |   |
|        | 作其他处理        |   |   |   | X |   | x |   | x |

初始判定表

(5) 化简。合并相似规则后得到结果图。

|        |              | 1 | 2 | 3 | 4 | 5 |
|--------|--------------|---|---|---|---|---|
| 条<br>件 | 功率大于 50 马力吗? | Y | Y | Y | N | N |
|        | 维修记录不全吗?     | Y | N | N | - | - |
|        | 运行超过 10 年吗?  | - | Y | N | Y | N |
| 动<br>作 | 进行优先处理       | x | x |   | X |   |
|        | 作其他处理        |   |   | x |   | x |

化减后的判定表