2012版 权威·实用

紧扣大纲　直击要点　链接真题　一点就通

全国计算机等级考试6日达标

冲刺模拟+考点速记

全国计算机等级考试命题研究组◎编

四级 软件测试工程师

考点全 真题多

多角度细致点评　讲解精髓　题目精选　针对性强

经过2年的研究与锤炼，从去年的7日达标延伸出来的更适合考生学习的这本6日达标终于付印成册。通过剖析考点、链接真题、达到精讲精练、使考生练一次就能上考场，赢高分。

直击常考点

● 精选常考知识点

计算机等级考试

链接多年真题

- 通过反复练习考题
- 推敲解题方法
- 精选各类考试押题

精选常考知识点、分类讲解、把握核心概念，做到事半功倍，考点中贯穿真题讲解，加深理解，短期内迅速提升。吃透重要考点，考试不再难。

练一次就能上考场

精选各类考试押题，全面覆盖所需考的知识点，通过反复练习考题，推敲解题方法，做到熟练掌握各种命题，巩固知识点，做到点面结合。

考点全 真题多

考点全面，针对性强。上考场前应知的考点通过提取常考知识点，抓住考试重点难点，分类讲解，便于考生专项攻克，迅速掌握命题规律与考试重点。

全国大学生最喜爱的等考品牌
北邮·等考

北京邮电大学出版社
www.buptpress.com

全国计算机等级考试6日达标

（冲刺模拟＋考点速记）

——四级软件测试工程师

全国计算机等级考试命题研究组　编

北京邮电大学出版社
·北京·

内 容 简 介

本书由全国计算机等级考试一线命题研究人员联手多年从事考前培训与阅卷的专家一起为希望快速提高、过关的考生设计的一套高效应试方案。其内容包括：上考场前应知的考点、上考场前应练习的真题、上考场前应练习的题库三大部分。上考场前应知的考点通过提取常考知识点，抓住考试重点难点，分类讲解，贯穿真题、便于考生专项攻克；上考场前应练习的真题是选取最新几套考试真题、把握命题规律、便于考生了解最新考试动态。上考场前应练习的题库是根据最新版考试大纲的要求，由多年研究等级考试考纲、试题及相关政策的老师编写，覆盖所有考点。

此外，本书附赠超值软件，软件中包括 10 套笔试试卷。考试环境和操作界面与真题一致。

本书精心设计应试板块，用最科学的方式引导考生在最短时间内获得最大收获，非常适合考前快速突破过关，也适合高等院校作为相关教学及培训辅导用书。

图书在版编目(CIP)数据

全国计算机等级考试 6 日达标 ：冲刺模拟＋考点速记. 四级软件测试工程师/全国计算机等级考试命题研究组编. --北京：北京邮电大学出版社，2012. 1

ISBN 978-7-5635-2811-0

Ⅰ. ①全… Ⅱ. ①全… Ⅲ. ①电子计算机—水平考试—自学参考资料②软件—测试—水平考试—自学参考资料 Ⅳ. ①TP3

中国版本图书馆 CIP 数据核字(2011)第 223203 号

书　　名：全国计算机等级考试 6 日达标(冲刺模拟＋考点速记)——四级软件测试工程师
作　　者：全国计算机等级考试命题研究组
责任编辑：满志文　姚　顺
出版发行：北京邮电大学出版社
社　　址：北京市海淀区西土城路 10 号(邮编：100876)
发 行 部：电话：010-62282185　传真：010-62283578
E-mail：publish@bupt. edu. cn
经　　销：各地新华书店
印　　刷：北京联兴华印刷厂
开　　本：880 mm×1 100 mm　1/16
印　　张：8
字　　数：317 千字
版　　次：2012 年 1 月第 1 版　2012 年 1 月第 1 次印刷

ISBN 978-7-5635-2811-0　　定价：25. 00 元

前　言

计算机作为一种得到广泛应用的工具，其重要性与日俱增。越来越多的人开始学习计算机知识，很多单位已经把计算机应用能力作为考核、录用工作人员的重要条件之一。各种计算机水平考试也随之应运而生，其中最受欢迎和信赖的就是教育部考试中心所组织的“全国计算机等级考试”。

本书在上一版“7日达标”的基础上，整合考生的心理，深入分析，研究出更适合考生学习和掌握的方法。

本书特点如下：

➢ 精心设计，高效实用。本书由全国计算机等级考试一线命题研究人员联手多年从事考前培训与阅卷的专家一起为希望快速提高和过关的考生设计的一套高效应试方案。其主要内容包括：上考场前应知的考点、上考场前应练习的真题、上考场前应练习的题库三大部分。本书精心设计应试板块，用最科学的方式引导考生在最短的时间内获得最大收获，非常适合考前快速突破过关！

➢ 考点全面，针对性强。上考场前应知的考点通过提取常考知识点，抓住考试重点难点，分类讲解，便于考生专项攻克，迅速掌握命题规律与考试重点。

➢ 穿插真题，专项巩固。本书在每个考点讲解下穿插与考点相关的真题，逐个讲解，透彻分析，便于考生一一理解，专项巩固。

➢ 最新真题，详尽解析。选取最新几套考试真题，让考生亲临考场，将记忆中的考点分散到具体考题当中，更能把握出题思路，吃透真题，把握最新考试动态。

➢ 题库丰富，点面结合。精选各类考试押题，全面覆盖所需考的知识点，通过反复练习考题，推敲解题方法，做到熟练掌握各种命题，巩固知识点，做到点面结合。

➢ 书盘结合，题量超大。盘中提供超大容量试卷库与答案解析，全真模拟环境，便于考生实战演练。

➢ 作者实力强。作者团队是从事等级考试近10年的辅导、培训、命题、阅卷及编写的经验，有较高的权威性，图书质量有保障。

本丛书由全国计算机等级考试命题研究组主编。参与本书编写与资料收集工作的有：秦海泉、陈忠贤、樊圣兰、俞翠兰、陈海霞、袁鸿鹏、陈斌、蔡季平、陈娟娟、刘卉、徐杨阳、白晖、李海磊、顾兴健、王连涛、陈海燕、赵海峰、李晓飞、吴松松、何光明。

由于水平有限，加上时间紧迫，书中难免有不足之处，恳请各位同仁和广大读者批评指正。如遇到疑难问题，可以通过以下方式与我们联系：bjbaba@263.net。微博地址：（北邮等考）http://weibo.com/u/2297589741。

全国计算机等级考试命题研究组

2012年1月

目　录

（工欲善其事，必先利其器；知已知彼，百战不殆）

第一部分

上考场前应知的考点

提取常考知识点，抓住考试重点难点，分类讲解，便于及时预习或复习，通过把握重要枝干来延伸细枝末节，在每个考点中插入链接到的相关真题，有助于更深入直观的理解考点，化抽象的概念为具体的题目，使考生在上考场前能够逐个理解掌握。

考点精讲一：软件静态分析的概念及内容

评注：静态错误分析主要用于确定在源程序中是否有某类错误或危险结构。它有以下几种类型：单位分析、引用分析、表达式分析和接口分析。静态错误分析有以下几种：类型和单位分析，引用分析等，其中在静态错误分析中，最广泛使用的技术就是发现引用异常。请考生用心掌握。

历年真题链接

真题 1 如果一个软件产品的功能或特性没有实现，包括主要功能部分丢失，次要功能完全丢失，或错误的声明，这是属于_____。(2009 年 3 月)

A) 致命的错误

B) 严重的错误

C) 一般的错误

D) 微小的错误

答案：B

解析：软件缺陷有四种级别，分别为：

致命的(Fatal)。致命的错误，导致系统或者应用程序崩溃、死机、系统悬挂，或者造成数据丢失、主要功能完全丧失。

严重的(Critical)。功能或特性没有实现，主要功能部分丧失，次要功能完全丧失，或致命的错误声明。

一般的(Major)。这种级别的错误不是很严重，虽然有一些缺陷，但是不影响系统和程序的基本使用。功能没有被很好地实现，没有达到预期要求。

微小的(Minor)。无关紧要的小问题，软件仍然可以使用，不影响功能的实现。

此处为 B 选项，严重的错误。

考点精讲二：白盒测试中的逻辑覆盖问题

评注：白盒测试中的逻辑覆盖往往是四级软件测试工程师考试的重中之重。2010 年 3 月卷一(6)、(7)、(8)连续三道题目均考查此知识点。覆盖率是度量测试完整性的一个工具，通常可以分为逻辑覆盖和功能覆盖。

覆盖率＝被执行到的项数/项总数×100%

逻辑覆盖是以程序内部的逻辑结构为基础的设计测试用例的技术，属于白盒测试。由于覆盖率的不同，又可以分为语句覆盖、判定覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。此处不一一介绍，在各种情况下，对于覆盖率＝被执行到的项数/项总数×100%均可适用。

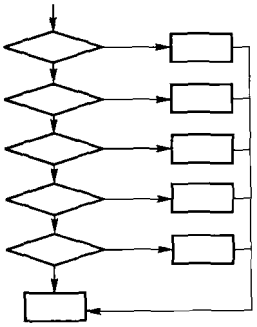
历年真题链接

真题 2 程序的流程图如下图所示，采用路径覆盖法进行测试，则至少需要几个测试用例可以覆盖所有可能的路径_____。(2008 年

4月)

A) 5　　B) 6

C) 7　　D) 8



答案:B

解析:根据程序流程图,我们可以导出程序的控制流图,然后利用环路复杂性的计算方法,最简单的是看该程序控制流图中有五个判定结点数,便可得到该控制流图的环路复杂性为5+1=6,这样便可以确定六个线性无关的基本路径集,然后我们根据这些基本集对应找出六个测试用例,确保基本路径集中每一条路径都可以被执行到,即满足题目中的路径覆盖,由此知若要满足路径覆盖,至少需要设计六个测试用例。

真题3　论述题2:(20分)(2008年9月)

针对以下C语言程序,请按要求回答问题。

已知weekday.c源程序如下:

```
#include<stdio.h>
#include<conio.h>
/*主函数*/
Int main()
{
    Char letter;
    Printf("please input the first letter,"Y"to exit!\n");
    While ((letter = getch())! ="Y")//当输入字母为Y时结束
    {
       Switch(letter)
       {
    Case 'S':
    Printf("%c\n",letter);
    Printf("please input second letter\n");//输入第二个字母
    If((letter = getch()) = ='a')
    Printf("Saturday\n");
    Else if(letter = ='u')
         Printf("Sunday\n");
    Else printf('data error\n);
    Break;
    Case 'F':
    Printf("fridaykn");
    Break;
    Case'M':
    Printf("mondayha");
    Break;
    Case'T':
    Printf("%c\n",letter);
    Printf("please input second letter\a");//输入第二个字母
    If((letter = getch()) = ='u')
    Printf("Tuesday\n");
    Else if(letter = ='h')
    Printf("Thursday\n");
      Break;
    Case'W':
    Printf("Wednesday\n");
    }
       }
       Return 0;
}
```

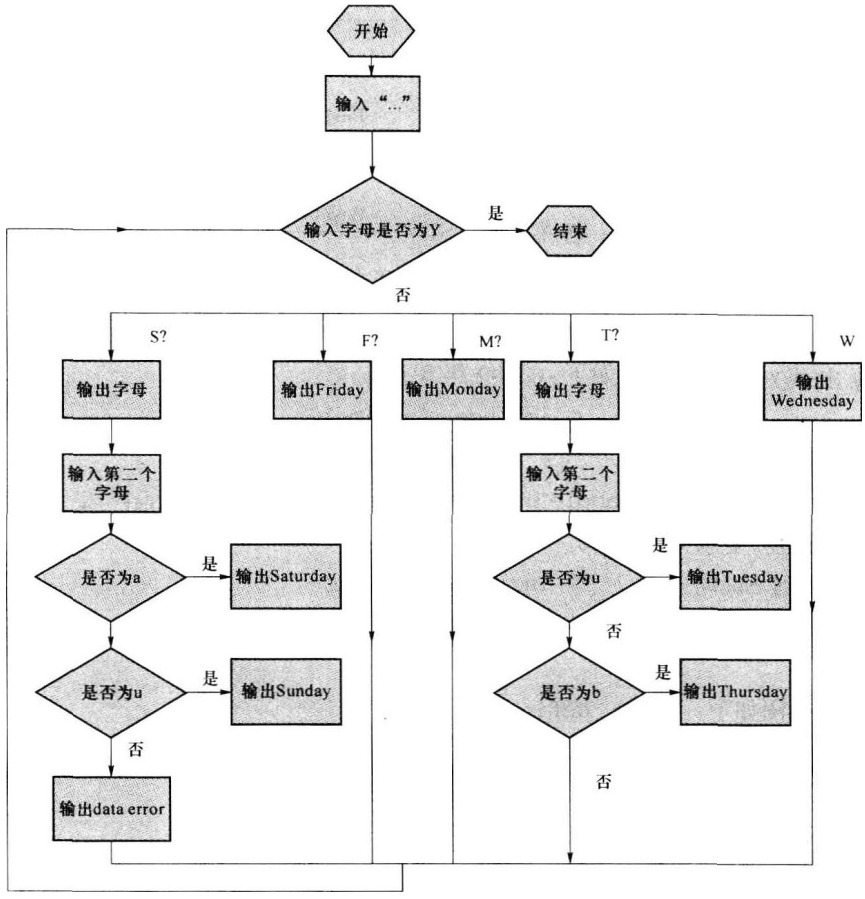
(1) 画出主函数main的控制流程图;(8分)

(2) 设计一组测试用例,使main函数的语句覆盖率尽量达到100%;(11分)

(3) Main函数的语句覆盖率能否达到100%?如果认为无法达到,需说明原因。(1分)

论述题2解析:

(1)



(2) [path1]1,2

[path2]2,3

[path3]2,3,4,5,22

[path4]2,3,6,7,22

[path5]2,3,6,8,22

[path6]9,10,22

[path7]11,12,22

[path8]13,14

[path9]13,14,15,16

[path10]13,14,17,18,22

[path11]13,14,8,22

[path12]19,20,22

[path13]19,8,22

生成测试用例,确保基本路径集中每条路径的执行:

path1. 输入数据 y,输出的结果程序结束返回值 0

path2. 输入数据 s,输出结果"s 输入第二个字母"

path3. 输入数据 s 后,输入数据 a,输出结果"Saturday"

path4. 输入数据 s 后,输入数据 u,输出结果"Sunday"

path5. 输入数据 s 后,输入数据非 a 非 u,输出结果"data error"

path6. 输入数据 f,输出结果"friday"

path7. 输入数据 m,输出结果"monday"

path8. 输入数据 t,输出结果"输入第二个字母"

path9. 输入数据 t 后,输入数据 u,输出结果"tuesday"

path10. 输入数据 t 后,输入数据 h,输出结果"thursday"

path11. 输入数据 t 后,输入数据非 u 非 h,输出结果"data crror"

path12. 输入数据 w,输出结果"Wednesday"

path13. 输入数据 defalt,输出结果"data error"

(3) 不能达到 100%,一些独立的路径如此程序中的【输入】,往往不是完全孤立的,有时候它是程序正常的控制流的一部分,这是这些路径的测试可以是另一条路径测试的部分。

真题 4　论述题 1:一个栈(Stack)对象有三种状态:S1——栈空;S2——栈非空也非满;S3——栈满。则各个状态的条件如下:(2009 年 3 月)

S1:(t0)创建栈对象时初始化,这是系统做的

(t1)在 S2 状态下执行置空运算 setEmpty()

(t2)在 S3 状态下执行置空运算 setEmpty()

(t3)在 S2 状态下执行出栈运算 Pop()

S2:(t4)在S1状态下执行进栈运算Push()

(t5)在S3状态下执行出栈运算Pop()

S3:(t6)在S2状态下执行进栈运算Push()

为简化问题,假设栈Stack的容量为2,栈元素的数据类型为整数。要求:

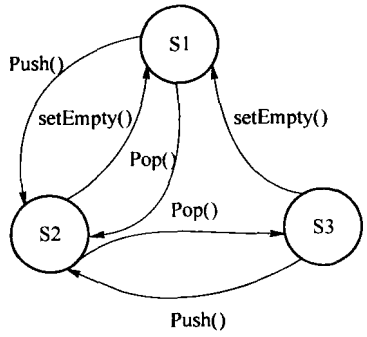
(1)根据题意,画出栈对象的状态迁移图;

(2)计算该状态迁移图的MeCabe环路复杂性;

(3)确定基本的测试路径,要求测试路径从S1出发最后回到S1,同时在状态转换时注明转换条件。

✱ 解析:

(1)根据题意,状态转换图如下:



(2)从图上可以看出,该图分为五个区域,所以$V(G)=5$。

此外,图中边数E为6,结点数N为3,则$V(G)=E-N+2=6-3+2=5$。

(3)根据上图,可得基本测试路径如下:

路径1:S1 —Push()→ S2 —Pop()→ S1

路径2:S1 —Push()→ S2 —Push()→ S3 —Pop()→ S2 —Pop()/SetEmpty()→ S1

路径3:S1 —Push()→ S2 —Push()/SetEmpty()→ S3 —Pop()→ S2 —Push()→ S1

路径4:S1 —Push()→ S2 —Push()→ S3 —SetEmpty()→ S1

路径5:S1 —Push()→ S2 —Push()→ S3 —Pop()→ S2 —Pop()→ S1

真题5 论述题3:已知C源程序如下:(2009年3月)

```
# include <stdio.h>
/ * A simple mailing list example using an array of structures. * /
# include <stdi.h>
# include <stdlib.h>

define MAX 4

struct addr {
  char name[30];
  char street[40];
  char city[20];
unsigned long int zip;
} addr_list[MAX];

void init_list(void),enter(void);
void deleteAddr(void),list(void);
int menu_select(void),find_free(void);
int main(void)
{
  char choice;

  init_list();/ * initialize the structure array * /
  for(;;){
    choice = menu_select();
    switch(choice){
      case 1:enter();
        break;
      case 2:deleteAddr();
        break;
      case 3:list();
        break;
      case 4:exit(0);
    }
  }
  return 0;
}

/ * Initialize the list. * /
void init_list(void)
{
  register int t;

  for(t = 0;t<MAX; ++ t) addr_list[t].name[0] = \0;
}

/ * Get a menu selection. * /
Int menu_select(void)
{
  char s[80];
  int c;
  printf("1. Enter a name\n");
  printf("2. Delete a name\n");
  printf("3. List the file\n");
  printf("4. Quit\n");
  do {
    printf("\nEnter your choice:");
    gets(s);
    c = atoi(s);
  }while(c<1 || c>4);
  return c;
```

```
}

/ * Input addresses
into the list. * /
void enter(void)
{
  int slot;
  char s[80];

  slot = find_free();
  if(slot = = -1){
      printf("\nList
Full");
    return;
  }

      printf (" Enter
name:");
    gets (addr _ list
[slot].name);
      printf (" Enter
street:");
   gets(addr_list[slot].
   street);
   printf("Enter city:");
      gets (addr _ list
[slot].city);
   printf("Enter zip:");
   gets(s);
    addr _ list[slot].
zip = strtoul(s, \0,10);
}

/ * Find an unused struc-
ture. * /
 int find_free(void)
 {
   register int t;
    for(t = 0; addr _
    list[t].name[0]
    &&t<MAX; ++t);
  if(t = = MAX) return
-1; / * no slots free * /
    return t;
}
  / * Delete an ad-
dress. * /
 void deleteAddr(void)
 {

   register int slot;
   char s[80];

    printf("enter record
# :");
   gets(s);
   slot = atoi(s);

    if(slot > = 0 &&
    slot<MAX)
     addr_list[slot].
     name[0] = \0;
 }

 / * Display the list
 on the screen. * /
 void list(void)
 {
   register int t;
    for(t = 0;t<MAX;
++t){

     if(addr_list[t].
name[0]){

      printf("%s\n",
      addr _ list[t].
      name);
      printf("%s\n",
      addr _ list[t].
      street);
      printf("%s\n",
      addr _ list[t].
      city);
       printf("%lu\
       n", addr _ list
       [t].zip);
     }
   }
   printf("\n\n");
   }
```

(1) 画出 main 函数的控制流程图。

(2) 设计一组测试用例，使该程序所有函数的语句覆盖率尽量达到 100%。如果认为该程序的语句覆盖率无法达到 100%，需说明原因。

论述题 3 解答：

(1) main 函数的控制流程图如下(限于篇幅只显示语句块，不显示具体语句)：

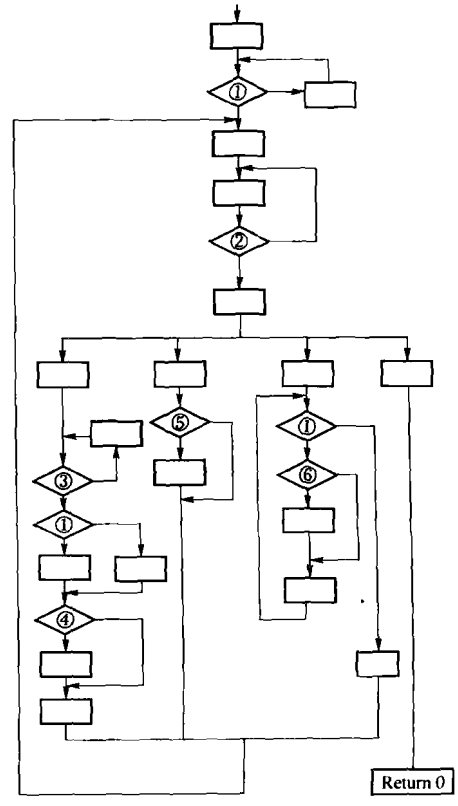
图中各判断语句如下：

① t<MAX

② c<1 ‖ c>4

③ : addr _ list[t]. name[0] && t<MAX

④ : slot==-1



⑤ ：slot＞＝0&&slot＜＝MAX

⑥ ：addr_list[t].name[0]

(2) 设计测试用例时，关键需要注意将 t 的值达到 Max，即起码要输入四个 name 才行，这样当 t＝＝Max 才能为真，才可以执行相应的分支语句。

具体输入如下：

输入四个 name:1

小明

凤凰街

南京

210000

1

小红

南京路

上海

120000

1

王明

达成路

上海

120000

1

李明

南京路

北京

100000

打算输入第五个:1

删除一个 name:2

小明

列出所有：　3

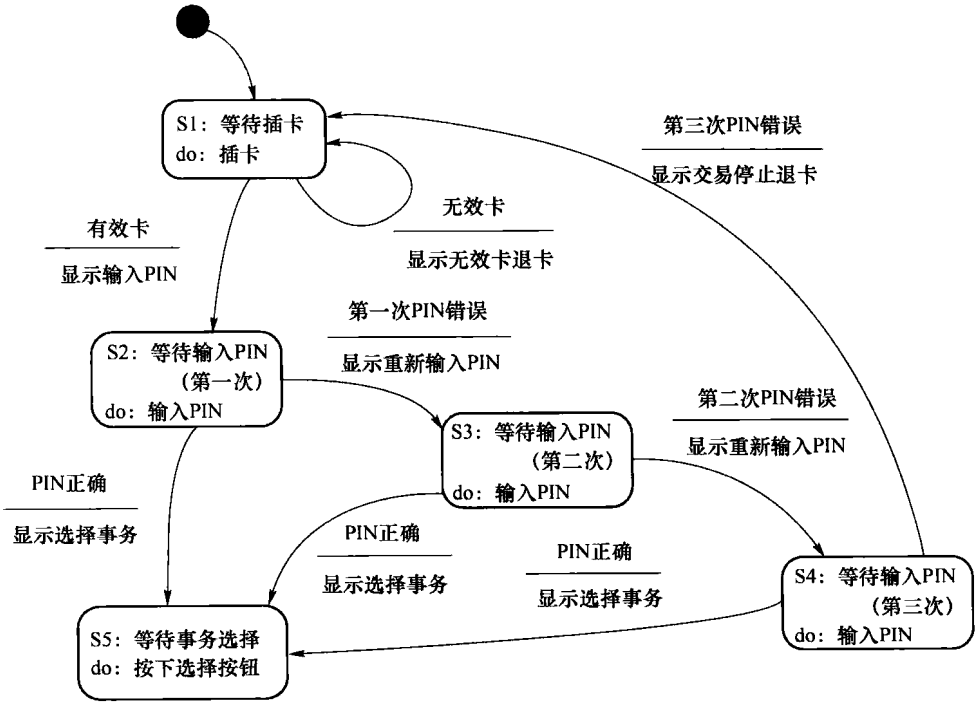
退出：　4

这样，程序中每一条语句都能够被执行。

上述测试用例能够使每条语句均执行，故语句覆盖率能够达到100%。

真题 6　论述题 1:(20 分)(2009 年 9 月)

下图是一个简单的 ATM 机中描述验证信用卡 PIN 活动的有限状态机。其中包含五个用“□”表示的状态和八个用“→”表示的转移。转移上的标签所遵循的是：横线上方是引起转移的事件，横线下方是与该转移相关联的行动。该有限状态机允许储户有三次输入 PIN 的机会，如果三次都输入错误，则停止交易退卡。



请完成下列工作：

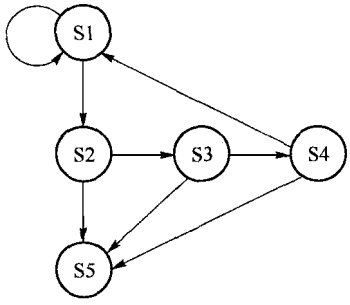
(1) 请给出与此有限状态机等价的控制流图；(4 分)

(2) 确定基本测试路径集；(9 分)

(3) 设计测试用例以覆盖基本测试路径。(7 分)

论述题 1 解析：

(1) 把有限状态自动机图的状态用结点代替，迁移用弧线代替，则可画出相应的控制流图。



(2) 上图的环路复杂性度量 V(G)为 5(区域数或者 V(G)=E−N+2)。

所以基本路径集合共有 5 条独立的路径：

S1—S1…

S1—S2—S5

S1—S2—S3—S4—S1…

S1—S2—S3—S5

S1—S2—S3—S4—S5

(3) 为每一条独立路径个设计一组测试用例，以便强迫程序沿着该路径至少执行一次

① 插入无效卡

② 插入有效卡——正确输入 PIN 码——按下事务选择按钮

③ 插入有效卡——第一次错误输入 PIN 码——第二次错误输入 PIN 码——第三次错误输入 PIN 码

④ 插入有效卡——第一次错误输入 PIN 码——第二次正确输入 PIN 码——按下事务选择按钮

⑤ 插入有效卡——第一次错误输入 PIN 码——第二次错误输入 PIN 码——第三次正确输入 PIN 码——按下事务选择按钮

真题 7　论述题 3：(10 分)(2009 年 9 月)

QESuite Web Version 1.0 中，已知 Lead 1.0 邮件系统测试项目的部分信息如下：

- 测试版本：总计两个测试版本，按日期排序分为 Lead1.0_090703，Lead1.0_090801
- 功能分类视图中部分待测区域及人员分配定义如下表所示：

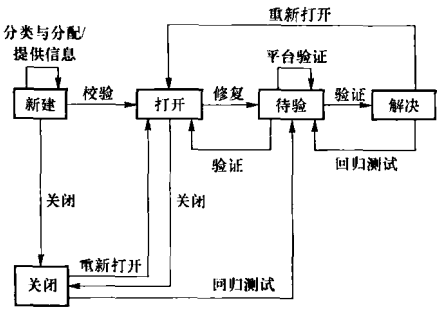
功能区域	功能区域	测试人员	开发人员
安全		Test1	Developer1
邮件系统	邮件管理	Test2	Developer2
	发邮件	Test2	Developer2
	……	……	……
	收邮件	Test2	Developer2
性能	并发处理能力	Test3	Developer3
……	……	……	……

已知在功能区域“性能/并发处理能力”下有一个软件问题处于“打开/修复失败”状态，下表中是该软件问题当前的部分操作历史信息，请依据 QESuite Web Version 1.0 的软件问题生命周期定义和题中提供的相关人员分配和测试版本信息，填写表中空缺的操作历史信息。(每空 1 分)

操作序号	操作者	执行操作	操作后的问题状态	测试版本
1		新建		Lead1.0_090703
2			打开/再现	
3		修复/修复		
4			打开/修复失败	Lead1.0_090801

论述题 3 解析：

解答：下图是 QESuite Web Version 1.0 的问题报告工作状态转换流程。



操作1中,在新建状态下,应该由测试人员进行校验工作,然后更改问题状态为打开。

操作2中,结果状态为打开/再现,说明是测试人员由打开状态进行校验,最后变为打开/再现。此时版本号不变,为Lead1.0_090703。

操作3中,状态为修复/修复,说明是开发人员进行了修复操作,最后状态变为待验,此时版本号变更为Lead1.0_090801。

操作4中,操作后的问题状态为打开/修复失败,说明是测试人员进行了验证操作,执行操作为验证。

所以,最后的表格为:

操作序号	操作者	执行操作	操作后的问题状态	测试版本
1	Test5	新建	打开	Lead1.0_090703
2	Test5	校验	打开/再现	Lead1.0_090703
3	Developer3	修复/修复	待验	Lead1.0_090801
4	Test5	验证/不同意	打开/修复失败	Lead1.0_090801

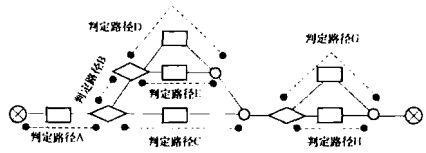
真题8 下图是一个判定路径覆盖图,其中用"◇"表示判定语句,用"□"表示处理语句,用"○"表示判定汇合点,用"●—●"表示判定路径,若有一个测试用例覆盖了判定路径A、B、E、H,则判定路径覆盖率为_____。(2009年3月)

A) 57.1%

B) 78.6%

C) 90%

D) 100%



答案:A

解析:覆盖率公式:DDP覆盖率=被评价到的判定路径数量/程序中判定路径总数×100%=4/7×100%=57.1%。由图可以看出,图中一共有七条判定路径,而该测试用例覆盖了判定路径A、B、E、H四条。所以该测试用例的判定路径覆盖率为4/7=57.1%。

考点精讲三:软件的性能测试

评注:软件性能测试在近三年连续5套试卷中均在第一大题的11、12题左右出现,一般为必考内容。软件的性能是软件的一种非功能特性。它关注的不是软件是否能够完成特定的功能而是在完成该功能时展现出来的及时性。

软件性能的指标:定义一系列软件性能指标,目的是为了能够客观地度量软件的性能。合适的软件性能指标不仅需要能够以客观的方式进行度量,而且要与人类对软件性能的主观感受相一致。

主要的软件性能指标如下:

(1) 响应时间。

(2) 系统响应时间和应用延迟时间。

(3) 吞吐量。

(4) 并发用户数。

(5) 资源利用率。

历年真题链接

真题9 在软件性能测试中,下列指标中哪个不是软件性能的指标_____。(2008年4月)

A) 响应时间

B) 吞吐量

C) 资源利用率

D) 并发进程数

答案:D

解析:软件的性能是软件的一种非功能特性。它关注的不是软件是否能够完成特定的功能而是在完成该功能时展现出来的及时性。为了能够客观地度量软件的性能,人们定义了一系列性能指标,以便于能够在不同情况下度量软件的性能。通常合理的软件性能指标有响应时间、系统响应时间和应用延迟时间、吞吐量、并发用户数、资源利用率五种,而题目选项D中"并发进程数"不能作为性能指标,故答案为D。

真题10 下列关于软件性能测试的说法中,正确的是_____。(2008年4月)

A) 性能测试的目的不是为了发现软件缺陷

B) 压力测试与负载测试的目的都是为了探测软件在满足预定性能需求的情况下所能负担的最大压力

C) 性能测试通常要对测试结果进行分析才能获得测试结论

D) 在性能下降曲线上,最大建议用户数通常处于性能轻微下降区与性能急剧下降区的交界处

答案:C

✱ 解析:性能测试通用模型(PTGM 模型)的最后一步就是测试结果分析,说明性能测试通常要对测试结果进行分析才能获得测试结论,故 C 正确。

真题 11　以下关于软件性能的说法中,正确的是_____。(2008 年 9 月)

A) 软件性能与该软件的实现算法无关

B) 软件的吞吐量越大,其平均响应时间总是越短。

C) 给软件的可用资源越少,其平均响应时间越短

D) 对于一个网络,其支持的同时发送请求的用户数越大,该网站的性能越好

答案:D

✱ 解析:软件性能指标主要有响应时间、系统响应时间和应用延迟时间、吞吐量、并发用户数、资源利用率五种。软件实现的算法与系统响应时间和应用延迟时间是直接相关的,所以软件的性能也必定与实现算法是有关系的。吞吐量是指系统在单位时间内处理请求的数量,对于无并发的应用系统而言,吞吐量是与响应时间严格的反比关系,但对于其他系统则不一定。资源数量与平均响应时间没有直接关系,因为在具体一个时间点,所占用的资源往往并不多,即若处理单个请求,在每个时间点都可能会有许多资源被闲置,当处理多个请求时,平均响应时间也与资源调配是否合理有关,与资源数量没有直接关系。另外,并发用户数是性能的一个重要指标,所以选项 D"对于一个网络,其支持的同时发送请求的用户数越大,该网站的性能越好",说的就是这点,完全正确。

真题 12　以下哪种软件测试属于软件性能测试的范畴?_____。(2008 年 9 月)

A) 接口测试

B) 压力测试

C) 单元测试

D) 易用性测试

答案:B

✱ 解析:依据测试目标不同,可以把软件性能测试及与性能有关的其他测试分为以下几类:①性能测试(Performance Testing);②并发测试(Concurrency Testing);③压力测试(Stress Testing);④可靠性测试(Reliability Testing);⑤负载测试(Load Testing);⑥配置测试(Configuration Testing);⑦失效恢复测试(Recovery Testing)。

真题 13　以下分析技术中,哪一种技术不属于基于性能计数器的分析技术?_____。(2008 年 9 月)

A) 内存分析

B) 处理器分析

C) 通信中断分析

D) 进程分析

答案:C

✱ 解析:性能计数器是指与性能有关的资源利用率指标,基于性能计数器的分析技术有内存分析、处理器分析、磁盘 I/O 分析、进程分析。

真题 14　以下哪种软件测试不属于软件性能测试的范畴_____。(2009 年 3 月)

A) 配置测试

B) 健壮性测试

C) 失败恢复测试

D) 负载测试

答案:B

✱ 解析:软件性能测试分为以下几类:

性能测试:测试软件的性能与软件需求规格说明是否相符。

并发测试:模拟多个用户并发使用软件,以测试软件是否存在与并发有关的缺陷。

压力测试:在较大的业务压力下,即系统运行环境超常的情况(如提供超常数量、频率或总量资源)下,测试软件是否存在功能和性能上的缺陷。

可靠性测试:在比较大的业务压力情况下进行的软件可靠性测试。

负载测试:不断增加软件的业务压力,探测软件在保证预定性能指标(如响应时间)的情况下所能负担的最大压力。

配置测试:通过调整软件的运行环境,测试不同的环境配置对软件性能的影响程度。

失效恢复测试:很多系统应当具有一些容错的能力,在出现某些故障时,仍然能够让用户继续使用下去。失效恢复测试是指验证系统从故障中恢复能力的测试。

B 项的健壮性测试属于系统测试的方法。

真题 15　以下目标中,哪个是软件性能测试的目标_____。(2009 年 3 月)

A) 检查软件的容错能力

B) 发现压力下软件功能的缺陷

C) 发现软件的安全漏洞

D) 检查用户界面是否易于使用

答案:C

解析:软件测试的目标不仅仅是发现(和改正)性能缺陷(Performance Bug),还包括探索和规划软件的实际性能。具体软件性能测试包括以下几方面的目标:

①发现缺陷;②性能调优;③能力检验与规划。

真题16 以下关于软件性能测试的说法中,不正确的是_____。(2009年9月)

A) 发现软件缺陷是性能测试的目的之一

B) 压力测试与负载测试的目的都是为了探测软件在满足预定的性能需求的情况下所能负担的最大压力

C) 性能测试通常需要对测试结果进行分析才能获得测试结论

D) 检验软件的最大负载是性能测试的目的之一

答案:B

解析:软件性能测试包括三个目标:①发现缺陷;②性能调优;③能力检验与规划。A说法正确。

压力测试是指在较大的业务压力下,即系统运行环境超常的情况下,测试软件是否存在功能和性能上的缺陷。负载测试是指不断增加软件的业务压力,探测软件在保证预定性能指标的情况下所能负担的最大压力。压力测试和负载测试是有区别的,两者都需要对软件施加业务压力,但根本目的完全不同,负载测试是探测软件处理能力的极限,而压力测试是利用压力揭示潜在缺陷。B说法错误。

性能测试通常需要对测试结果进行分析才能获得测试结论,C说法正确。

性能测试的目的之一,就是检验软件的最大负载,D说法也正确。

真题17 以下哪种软件测试不属于广义软件性能测试的范畴_____。(2009年9月)

A)并发测试

B)压力测试

C)兼容性测试

D)负载测试

答案:C

解析:软件的性能是软件的一种非功能特性,它关注的不是软件是否能够完成特定的功能,而是在完成该功能时展示出来的及时性。根据测试目的的不同,可以把软件性能测试以及与性能有关的其他一些测试分为:①性能测试;②并发测试;③压力测试;④可靠性测试;⑤负载测试;⑥配置测试;⑦失效恢复测试。

兼容性测试属于系统测试的范畴,不是软件性能测试。故本题选C。

考点精讲四:软件可靠性测试

评注:软件的可靠性一般为必考题目,往往以一个选择题的形式出现,一般出现在第13题左右。软件可靠性有两方面的含义:

(1) 在规定条件下,规定时间内,软件不引起系统失效的概率。

(2) 在规定时间周期内,在所述条件下执行所要求的功能的能力。

软件可靠性的基本指标是"在规定条件下,规定时间内,软件不引起系统失效的概率",通常用平均无失效时间(Mean Time To Failure,MTTF)来直观表示可靠性,所谓MTTF,是指软件运行后,到下一次发生失效的平均时间。

要求考生对软件可靠性的概念和含义深刻理解和掌握。

历年真题链接

真题18 下列关于软件可靠性测试的说法中,错误的是_____。(2008年4月)

A) 发现软件缺陷是软件可靠性测试的主要目的

B) 软件可靠性测试通常用于有可靠性要求的软件

C) 在一次软件可靠性测试中,执行的测试用例必须完全符合所定义的软件运行剖面

D) 可靠性测试通常要对测试结果进行分析才能获得测试结论

答案:A

解析:软件可靠性测试的目的是收集软件测试时揭示软件故障的情况,并对其进行整理从而为分析和预测软件的可靠性提供帮助,与其他软件测试不同的是,软件可靠性测试的目的不在于通过测试揭示软件中的缺陷并通过修改这些缺陷来提高软件的可靠性,而是通过受控的软件测试过程来预测软件在实际运行中的可靠性。从软件的可靠性测试过程可知,软件可靠性测试通常用于有可靠性要求的软件,在一次软件可靠性测试中,执行的测试用例必须完全符合所定义的软件运行剖面,可靠性测试通常要对

测试结果进行分析才能获得测试结论。

真题 19　以下关于软件可靠性测试的说法中，正确的是_____。（2008 年 9 月）

A）定义软件运行剖面是软件可靠性测试的重要步骤

B）软件可靠性测试使用的测试用例应该满足分支覆盖

C）软件可靠性测试可以在单元测试中实施

D）软件可靠性预测模型的作用是指导软件可靠性测试中的测试用例生成

答案：A

解析：软件可靠性测试过程包括五个步骤：①确定可靠性目标；②定义软件运行剖面；③设计测试用例；④实施可靠性测试；⑤分析测试结果。所以说定义软件运行剖面是软件可靠性测试的重要步骤，故选项 A 正确。

真题 20　以下关于软件可靠性测试的说法中，正确的是_____。（2009 年 3 月）

A）软件运行剖面的定义需要符合软件的实际运行情况

B）测试用例的生成必须采用白盒测试方法

C）软件可靠性测试通常能够比黑盒测试发现更多的错误

D）软件可靠性测试必须在集成测试中实施

答案：A

解析：软件可靠性有两方面含义：

① 在规定的条件下和规定的时间内，软件不引起系统失效的概率。

② 在规定的时间周期内，在所述条件下程序执行所要求的功能的能力。

软件可靠性测试的目的不在于通过测试揭示软件中的缺陷并通过修改这些缺陷来提高软件的可靠性，而是通过受控的软件测试过程来预测软件在实际运行中的可靠性。

因为软件可靠性的预测依赖于出错数据的统计，软件在可靠性测试中的运行情况必须完全符合软件的实际运行情况，否则预测的软件可靠性只能适用于所使用的测试用例，而不能反映实际运行时的可靠性。由此可得 A 正确。

真题 21　以下关于软件可靠性与硬件的可靠性主要区别的说法中，正确的是_____。（2009 年 9 月）

A）软件的每个拷贝都是完全一样的，而按照设计生产出来的同规格硬件总有微小差别

B）软件经常面临恶意的使用者，而硬件没有恶意的使用者

C）软件的使用者通常遍及整个世界，而硬件的使用者通常只局限于某个地区

D）软件的失效都是逻辑错误引起的，而硬件的失效都不是逻辑错误引起的

答案：A

解析：软件可靠性和软件可靠性测试的研究在很大程度上借鉴了硬件可靠性的研究，但由于软件和硬件的不同特点导致软件可靠性和硬件可靠性也有很大的不同，这也是软件可靠性研究仍然很不成熟的重要原因。具体而言，软件可靠性和硬件可靠性的区别包括：①唯一性。软件设计出来后，所有副本都是一模一样的；硬件设计出来后，每个按照设计生产的硬件都不可能完全相同；②物理退化。一个正确的硬件器件会因为物理退化在某时刻失效，但正确的软件则不会因为物理退化而发生失效；③逻辑复杂性。软件具有复杂的内部逻辑，而硬件的内部逻辑则相对简单；④版本更新。硬件的版本更新通常很慢，而软件的版本更新比较频繁。

B 说法错误，硬件也有恶意使用者。随着国际化的加深，硬件的生产标准越来越统一，使用者也遍布很多地区，C 说法错误。硬件失效通常不是逻辑错误导致的，但也有例外，D 说法太过绝对。综合以上，本题应选 A。

考点精讲五：面向对象软件测试

评注：此考点为必考考点，而且比较复杂，尤其涉及了封装，集成的类的单元测试，以及面向对象集成测试自身的特点。所以请考生多下功夫，分析、掌握其本质原理。

封装、继承和多态是面向对象软件区别于传统的结构化软件的三个主要特点，然而这些特点都可能给测试带来困难。在面向对象中，封装一方面是指一组相关的变量和方法被封装到一个类中，则类的成员方法对成员变量有依赖性，故成员方法通常不能实现独立的功能，需要在不同的实例状态下才能展示出来，甚至有的成员方法要在特定

的实例状态下才能执行,因此在测试面向对象软件时,不能简单地对每个类的成员方法进行测试,在调用任何成员方法之前还必须保证相应的实例处于该方法的预期工作状态,即设计类的测试用例时,不仅要考虑各成员方法的输入参数,还要考虑如何设计调用的序列。多态就是指对类的引用可以与多个类的实现绑定,绑定有静态和动态之分,所以为达到较高的测试充分性,应对所有可能的绑定都进行测试。再来看继承和继承与多态的复合对测试的影响,假设类B是类A的子类,如果类A已进行了充分的测试,若按传统的测试充分性准则,在测试类B时可以把关注点放在类B自身定义的成员变量和成员方法上,但在实际测试类B时,这样的测试往往会不够充分,还是要对类B继承类A的成员方法进行测试,而且对于一棵继承树上的多个类,仅对处于叶结点的类进行测试也是不充分的。

历年真题链接

真题22 以下关于面向对象软件测试的说法中,错误的是_____。(2009年9月)

A) 对于面向对象程序集成测试而言,大突击集成可能导致测试不充分

B) 面向对象软件只能采用白盒测试,不能采用黑盒测试

C) 在存在多态的情况下,为了提高测试的充分性需要对所有可能的绑定都进行测试

D) 单个成员方法的测试属于面向对象程序单元测试考虑的范畴

答案:B

解析:由于大突击集成面对的是整个软件的所有代码,几乎没有什么方法能够为此生成合适的测试用例集,所以大突击集成可能导致测试不充分,A说法正确。

在面向对象中,一般有少数的方法需要单独进行测试,D说法正确。

真题23 以下关于面向对象软件测试的说法中,正确的是_____。(2009年9月)

A) 对于一个类的测试,一个测试用例只能包含对该类的一个方法的一次调用

B) 基于判定表的测试不能用于面向对象程序的单元测试

C) 不变式边界测试可用于类层次的测试,其目的是测试功能组合

D) 对于抽象类,需要进行单元测试

答案:D

解析:在面向对象中,很难对单个成员方法进行充分的测试,这是因为多个成员方法会通过成员变量产生相互依赖关系。合理的测试是将这些相互依赖的成员方法放在一起进行测试,故A说法错误。

基于判定表的测试,又称为组合功能测试,既可以用于传统软件测试,也可以用于面向对象软件测试,B说法错误。

不变式边界测试是一种基本的和高效的类层次的测试技术。类层次测试的一个主要困难是成员变量的某些状态可能不会出现,这就是所谓的类不变式。不变式边界测试首先准确定义类的不变式,其次寻找成员方法的调用序列以违反类不变式,这些调用序列即可作为测试用例。不变式边界测试的目的不是测试功能组合,C的说法错误。

对于抽象类,需要进行单元测试。但是构造抽象类的驱动程序显然比构造其他类的驱动程序复杂,因为在测试抽象类时,需要为抽象类构造一个子类,并实现所有抽象类没有实现的成员方法。D正确。

考点精讲六:Web应用软件的测试方法

评注:对于Web应用软件的特点,考生需要掌握,这是四级软件测试工程师的必考内容,一般出题为选择题的16、17题左右,题目不难,所以牢固的掌握Web应用软件测试的相关内容,一般不会失分。

普通软件由开发人员设定结构,而Web应用软件在固定的结构中填充相应的内容。Web应用软件的结构也与普通软件大不相同。Web应用软件的系统测试包括功能测试、性能测试、易用性测试、内容测试、安全性测试、接口测试等。

1. 功能测试

(1) 链接测试。

(2) 表单测试。

(3) Cookie测试。

2. 性能测试

(1) 并发测试。

(2) 负载测试和压力测试。

(3) 配置测试和性能调优。

3. 易用性测试

4. 内容测试