电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221302006

姓 名 周玉川

（实验） 课程名称 软件工程

理论教师 吴祖峰

实验教师 吴祖峰

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：周玉川 学号：2017221302006 指导教师：吴祖峰**

**实验地点：信软楼西306 实验时间：2018.11.20**

1. **实验名称：质量保证**

**二、实验学时：4**

**三、实验目的：**

1、掌握软件编码规范

2、掌握单元测试的方法，包括白盒测试和黑盒测试

3、能对实际程序进行单元测试，能够撰写简单的测试报告

**四、实验原理：**

1、白盒测试：是通过程序的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)进行测试而不使用用户界面。这种类型的测试需要从代码句法发现内部代码在算法，溢出，路径，条件等等中的缺点或者错误，进而加以修正。

2、黑盒测试：又被称为功能测试、数据驱动测试或基于规格说明的测试，是通过使用整个软件或某种软件功能来严格地测试, 而并没有通过检查程序的源代码或者很清楚地了解该软件的源代码程序具体是怎样设计的。测试人员通过输入他们的数据然后看输出的结果从而了解软件怎样工作。

**五、实验内容：**

1. 编写三角形判断程序，并进行黑盒测试
2. 编写分数统计程序，并进行白盒测试
3. 选择系统中的2个关键模块进行编码实现
4. 使用白盒或黑盒方法测试实现的模块

**六、实验器材（设备、元器件）：**

计算机一台

**七、实验步骤：**

**步骤1：三角形类型的判断**

**步骤2：统计计算**

**步骤3:基于实验2的结果，选择2个关键模块进行编程实现。**

**步骤4：使用白盒或黑盒方法测试实现的模块。（请描述使用的测试方法类别和名称，如黑盒法中的等价类划分法，以及方法基本过程，如等价类表、关键路径等）**

**八、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

## 步骤1：三角形类型的判断

输入三个整数a、b和c分别作为三角形的3条边，判断由这3条边构成的三角形类型是：等边三角形、等腰三角形、一般三角形、或非三角形（即不能构成一个三角形）。

### 1.1编写该程序，注意编程规范

|  |
| --- |
| 代码：  #include <stdio.h>  #include <stdlib.h>  #include <stdbool.h>  bool isPositiveNumber( int num ); //判断正数  int analysis( int borderA, int borderB, int borderC); //判断三角形类型  bool isTriangle ( int a, int b, int c); //判断可以组成三角形  int main(){  int a , b , c;  printf( "输入三角形三个边的长度: \n" ); //提醒输入  scanf( "%d %d %d",&a,&b,&c ); //从屏幕中读取三边长度  printf("\n你输入的数字是: %d, %d, %d\n\n分析结果如下：\n", a, b , c);  switch( analysis(a, b , c) ) //根据判断 三角形类型  {  case -1: printf("Illegal number-->number must bo positive.\n"); break;  case 0: printf("%d,%d,%d can't form a triangle（不能组成三角形）.\n", a, b, c);break;  case 1: printf("它是普通三角形.\n"); break;  case 2: printf("它是等腰三角形.\n"); break;  case 3: printf("它是等边三角形.\n"); break;  default: break;  }    return 0;  }  bool isPositiveNumber( int num ) //判断正数  {  return ( num > 0 );  }  bool isTriangle ( int a, int b, int c) //判断可以组成三角形  {  return ( ( a + b > c ) && ( a - b < c ) &&( b - a < c));    }  int analysis( int borderA, int borderB, int borderC) //判断三角形类型  {  if ( !( isPositiveNumber( borderA ) && isPositiveNumber ( borderB ) && isPositiveNumber( borderC ) ) ){  return -1;  }  else{  if ( isTriangle( borderA, borderB, borderC ) ){ //是三角形  if ( ( borderA == borderB ) && ( borderB == borderC ) )  return 3; //等边三角形返回3  else if ( ( borderA == borderB ) || ( borderB == borderC )|| (borderA == border) )  return 2; //等腰三角形返回2  else{  return 1; //普通三角形  }  }  else{  return 0; //不有三角形  }  }  } |

### 1.2按照等价类划分方法设计测试用例，并进行测试

用等价类划分的方法，建立输入等价类表：

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 三条边的正负 | 三边全为正数（1） | 存在非正数边（2） |
| 三边能否组成三角形 | 可以组成三角形（3） | 不能组成三角形（4） |
|  |  |  |

测试结果

|  |
| --- |
| 用例1：三边全为正数（1） |
| 用例2：存在非正数边（2） |
| 用例3：可以组成三角形（3） |
| 用例4：不能组成三角形（4） |

#### 1.2.1按照边界值方法设计测试用例，并进行测试

三角形问题的边界测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例 | A | B | C | 预期结果 |
| 1 | 0 | 0 | 0 | 输入不为正数 |
| 2 | 100 | 100 | 300 | 不能组成三角形 |
| 3 | 100 | 100 | 100 | 等边三角形 |
| 4 | 100 | 1 | 100 | 等腰三角形 |
| 5 | 3 | 4 | 5 | 普通三角形 |

测试结果

|  |
| --- |
| 用例1： |
| 用例2： |
| 用例3： |
| 用例4： |
| 用例5： |

## 步骤2：统计计算

步骤2：输入不超过50个数（以-1作为输入结束标志），计算其中有效的学生分树的个数，及有效分数的总分和平均值。

### 2.1.编写程序，注意编程规范

#include <stdio.h>

int main() {

int marks[50];

int mark, i, availble = 0, average;

int sum = 0;

int number = 0;

printf\_s("请输入分数(-1 结束输入)\n");

scanf\_s("%d", &mark);

while (mark != -1)

{

if (number >=50)

{

printf\_s("存储数据量已经超过50个\n"); break;

}

marks[number] = mark;

number++;

printf\_s("请输入分数(-1 结束输入)\n");

scanf\_s("%d", &mark);

}

for (i = 0; i <= number; i++)

{

if (marks[i] >= 0 && marks[i] <= 100)

{

availble++;

sum = sum + marks[i];

}

else;

}

average = sum / availble;

printf\_s("有效分数个数为： %d \n 有效分数平均分为：%d\n", availble, average);

return 1;

}

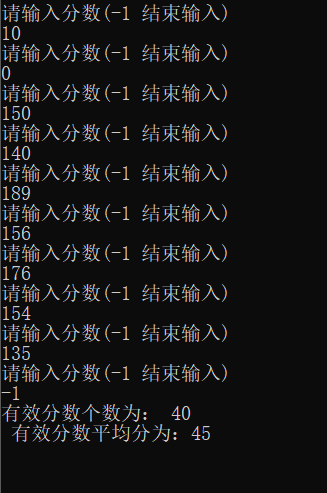
### 2.2.导出程序的控制流图，确定独立路径集合（基本路径集合）



### 2.3.为每条路径设计测试用例，并进行测试

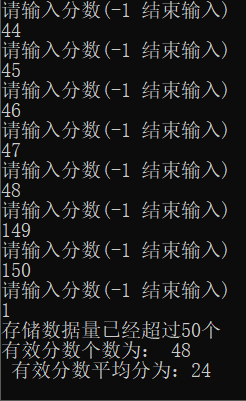
I.输入总数不超过50个并且存在无效分数，手动中止输入

测试结果如图：



II．输入总数超过50个，且存在无效分数，自动终止输入

输出结果截图如下：



## 步骤3、4：测试

**本小组选用了选题中的登陆模块和教室查询模块来进行具体测试**

### 3.1登录模块

|  |
| --- |
| 代码：  #include <bits/stdc++.h>  #define N 1001  using namespace std;  struct student{  int num ;  char pass[15];  }stu[N];  char txt[20] = "test2.txt";  void logIn();  void Register();  bool hasExist();  void readTxt(){ //读文件  FILE\*fp = fopen( txt,"r+");  if (fp == NULL){  perror("Error:");exit(0);  }  int num;  while ( fscanf(fp,"%d",&num)!=EOF){  stu[num].num= num;  fscanf(fp,"%s",stu[num].pass);  }  fclose(fp);  }  void writeTxt(struct student t){ //写入文件内  FILE\*fp = fopen( txt,"ab+");  if (fp == NULL){  perror("Error:");exit(0);  }  if ( fseek(fp,0L,SEEK\_END) == -1) {  perror("Error");  exit(0);  }  fprintf(fp,"\n%d %s\n",t.num,t.pass);  fclose(fp);  }  void print(){ //打印学生信息  int i=1;  for ( ; i < N ; i++){  if (stu[i].num)  printf("idNum: %d password: %s\n",stu[i].num,stu[i].pass);  }  }  int main(void){ //主函数  int type = 1;  memset(stu,sizeof(stu),0);  fprintf(stdout,"开始读取存档学生信息\n");  readTxt();printf("<----Read Successful---->\n");  printf("\n读取信息为：\n");  print();  printf("登录(输入1)或者注册(输入2): ");  scanf("%d", &type);  switch( type ){  case 1:logIn(); break;  case 2:Register();break;  default: printf("输入正确形式");break;  }    return 0;    }  bool hasExist( int num){ //判断学生已经存在  return ( num <1001 && num > 0 && stu[num].num == num );  }  void logIn(){ //用户登录  int num;  printf("输入学号: ");  scanf("%d",&num);  if ( num > N ) //非法输入处理  {  printf("Illegal input\n");  return ;  }  if (!hasExist( num )){  printf("没有找到用户\n");  int choose = 0 ;  printf("是否想要注册(输入1选择是,否则为否): ");  scanf("%d",&choose);  if (choose == 1)  Register();  else{  printf("You choose No,Exit\n");  }  }  else{  char ch[15];  int count = 0;  while( count < 3){  printf("输入密码: ");  scanf("%15s",ch);  if ( strcmp(ch,stu[num].pass) == 0){  printf("登录成功<----Login Successful-->\n");  break;  }  else{  count++;  if (count >= 3)  printf("密码错误超过三次，请重新登录.\n");  else  printf("密码错误，请重新输入,还有%d次机会\n",3-count);  }    }    }  }  void Register(){  int num;  printf("输入学号（小于1000）：");  scanf("%4d",&num);  if ( num > N ) //非法输入处理  {  printf("Illegal input\n");  return ;  }  if ( hasExist(num) )  {  printf("Has already exist\n");  return ;  }  stu[num].num = num;  printf("输入密码（长度少于15）: ");  scanf("%15s",stu[num].pass);  writeTxt(stu[num]);  printf("申请成功<---Register Successful-->");    } |

黑盒测试

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 注册 | 输入不存在的学号(1) | 其（2） |
| 登录 | 存在学号，正确密码(3) | 其他数（4） |
|  |  |  |

等价类划分：

边界测试：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例 | 学号 | 密码 | 是否注册 | 注册学号 | 预期结果 |
| 1 | 100 | xin | / | / | 成功登录 |
| 2 | / | / | / | 512 | 注册成功 |
| 3 | 522 | / | 是 | 522 | 不存在学号，然后注册 |

|  |
| --- |
| 用例1： |
| 用例2： |
| 用例3： |
| 注册前后txt文档对比：  注册前    注册后 |

### 3.2教室查询模块

代码：

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define EMPTY 0  #define FULL 1  /\*教室信息由文本提前准备好，装在state.txt中，1代表教室已满，0代表教室为空\*/  void Init(int a[][10]);  int main() {  FILE \*p = NULL;  int num0=0,num1 = 0, num2 = 0,num=0,num3=0;  Init(Room\_state);  fopen("state.txt","r+");  while(1){  L:printf("请输入你要查询的教室号：");  scanf("%d", &num0);  if(num0%100==0){  printf("教室信息输入错误,请重试！\n");  goto L;  }  num1=num0/100;  num2=num0/10%10;  num3=num0%100%10;  if(!(1<=num1&&num1<=4&&num2>=0&&num2<=1&&num3>=0&&num3<=5)) {  printf("教室信息输入错误,请重试！\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  goto L;  }  if (Room\_state[num1][num2\*10+num3] == EMPTY){  printf("教室可用！\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("输入1进入租借界面,输入0退出！\n");  L1: scanf("%d",&num);  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  {  if(num==0)  goto L;  else if(num==1){  printf("请上传学生证信息！\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  }  else {  printf("操作有误，请重新输入！\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  }  goto L1;  }  }  else if (Room\_state[num1][num2\*10+num3] == FULL){  printf("教室不可用！\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  }  }  return 0;  }  void Initial(int a[][10]) {  FILE \*p = NULL;  int i=0,j=0;  p = fopen("D:\\state.txt", "r+");  for (i = 0; i < 5; i++) {  for (j = 1; j < 10; j++) {  if (fscanf(p, "%1d", &a[i][j])== EOF){  printf("读取失败！\n");  break;  }    }  }  } |

黑盒测试：

等价类划分：

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 教室号位数 | 3位（1） | 其他数（2） |
| 第一位数 | 1-4 （3） | 其他数（4） |
| 后两位数 | 01-15（5） | 其他数（6） |

边界测试：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例 | 第一次 | 第二次 | 第三次 | 第四次 | 预期结果 |
| 1 | 10 | 102 | 1023 | / | 第二次成功 |
| 2 | 012 | 112 | 412 | 512 | 第二、三次成功 |
| 3 | 100 | 201 | 315 | 416 | 第二、三次成功 |

|  |
| --- |
| 用例1：输入教室号2位、3位、4位： |
| 用例2：第一位数输入0、1、4、5： |
| 用例3：后两位数输入00、01、15、16： |

**九、总结及心得体会：**

1. 大致了解如何对软件质量保证。
2. 学习黑白盒测试，等价分类，以及边界值测试。
3. 认真学习书本内容，方便实践实验。
4. 团队合作很重要，讨论分工必不可少。
5. 系统设计对工程的实现帮助很大。

**十、对本实验过程及方法、手段的改进建议：**

1. 希望老师多指点出学生的不足，加以改进。
2. 实验好处很多，多多益善。

**十一、成员**

|  |  |  |
| --- | --- | --- |
| 队长 | 周玉川 | 2017221302006 成绩：100 |
| 成员 | 郝若帆 | 2017221302013 成绩：100 |
| 孙玉龙 | 2017221302011 成绩：100 |
| 陈子琦 | 2017221302007 成绩：100 |
| 夏靖森 | 2017221302018 成绩：100 |

**报告评分：**

**指导教师签字：**