Ch1 软件测试基础

一、判断是非并改正错误

1. 某用户发现电商平台 A 对其同一个订单进行了多次扣款，这是软件缺陷。（ ）
2. 缺陷是存在于软件中的不希望或不可接受的偏差，一定会导致软件失效。（ ）
3. 调试的主要目的是排错，测试的主要目的是揭错。（ ）
4. 各类测试活动需要等到代码编写完毕才能够展开。（ ）
5. 缺陷是软件运行中出现的不希望或不可接受的偏差。（ ）
6. 达到期望的覆盖度要求是软件测试的主要目标。（ ）
7. 如果测试输入能够执行到包含缺陷的代码就可以发现该缺陷。（ ）
8. 如果可以穷举被测系统的输入空间，那么软件测试可以说明软件的正确性。（ ）
9. 失效是不希望或不可接受的外部行为，说明被测系统中一定存在缺陷。（ ）
10. 软件测试的目的只是发现缺陷。（ ）
11. 缺陷在软件运行期间才能出现。（ ）
12. 软件测试过程按照被测对象规模由小到大的规则推进。（ ）
13. 测试不能说明被测系统是正确的。（ ）
14. 缺陷是指系统表现出的与期望不符的外部行为。（ ）
15. 软件测试是发现缺陷的研发活动，可以说明软件的正确性。（ ）
16. 能够发现缺陷或者至少可能发现缺陷是指测试用例的有效性。（ ）
17. 软件测试是指通过执行被测对象发现其中缺陷的一种研发活动。（ ）
18. 软件测试的主要目的是通过缺陷发现保证软件质量达到既定的目标，同时通过对缺陷的分析和利用实现软件开发中的缺陷预防。（ ）
19. 死机是一种缺陷。（ ）
20. 只有代码才能使用静态测试方法。（ ）
21. 测试用例的有效性是指测试用例随被测对象的变化容易被重复利用的程度。（ ）

二、单项选择（每题有且仅有一个答案）

1. 下面关于测试用例的说法，不正确的是（ ）
   A. 测试用例是测试设计活动的输出结果；
   B. 测试用例是对被测系统的输入数据；
   C. 揭错能力是衡量测试用例质量的维度之一；

D. 测试用例的可视化形式可以是文档化也可以是代码化

2. 下列说法正确的是（ ）

A. 系统产生故障，说明测试输入满足可达性；

B. 系统产生故障，说明测试输入不满足可达性；

C. 系统没有产生故障，说明测试输入满足可达性；

D. 系统没有产生故障，说明测试输入不满足可达性；

3. 下面 4 个评价测试用例质量的属性维度中，（ ）是指测试用例的揭错能力

A. 有效性； B. 仿效性； C. 经济性； D. 可维护性；

4. 下面（ ）不是缺陷

A. 系统分析人员对用户需求的错误理解；

B. 开发人员将“!=”写成“==”；

C. 代码中遗漏语句结束符；

D. 代码没有对调用返回值可能为 null 进行是否为 null 的判断；

5. 下面（ ）必须包含在测试用例里

A、测试用例标识； B、优先级； C、预期结果； D、测试说明；

6. 以破坏测试对象的正常行为目的测试技术被称为（ ）

A. 正向测试； B. 反向测试； C. 代码测试； D. 系统测试

7. 代码如图 1 所示，其中（ ）存在缺陷

```
A、void func_ARG(int **pp, unsigned n)
  {
     int aux;
B、   if (n == 1) {
C、       *pp = &aux;
     } else {
D、       *pp = (int *)malloc(n * sizeof(int));
    }
 }
```

图 1

8. 代码如**图 2** 所示，输入（ ）能产生故障（fault），但不产生失效（failure）；输入（ ）既产生故障，也产生失效

```java
public boolean findInteger(int[] x, int y){
    if (x==null) return false;
    //正确代码为for (int i=0; i<=x.length;i++)
    for (int i=1; i<=x.length;i++){
        if (x[i]==y)
            return true;
    }
    return false;
}
```

图 2

A．x={2,3,4,5}, y=2　　B. x={2,3,4,5}, y=3　　C. x= null, y=4>　　D. 不存在

9. 用于衡量测试用例发现缺陷可能性的属性是（ ）

A、有效性；　B、仿效性；　C、经济性；　D、可维护性；

10. 不考虑系统内部实现结构验证系统是否到达用户需求的测试方法是（ ）

A、黑盒测试；B、白盒测试；C、功能测试；D、非功能测试

三、多项选择（每题有 2 个或 2 个以上答案）

1. 下面描述中（ ）是缺陷

A. 系统长时间没有响应；

B. 代码中遗漏了相关系统特性的实现；

C. 系统的响应时间不满足用户需求；

D. 系统分析人员对用户需求的理解错误；

E. 工程师没有及时修订用户操作手册导致实际操作流程和手册中的指导步骤不符；

F. 移动应用程序闪退；

G. 对于 Java 代码而言，在长生命周期的对象中引用短生命周期对象；

H. 代码中的死循环；

I. 用例规约（use case specification）中缺少对异常处理流程的说明；

J. 系统输出不正确的结果；

2. 代码如图 3 所示，对于输入 ids=""(空字符串)而言，满足（ ）

```
public boolean mc1(String ids){
    if ( ids.length() == 0) // 这里有缺陷，正确的代码为 : if (ids == null)
        return false;
    for (String id : ids.split( regex: ";")){
        if (id.trim().startsWith("5101"))
            return true;
    }
    return false;
}
```

图 3

A．可达性； B. 感染性； C.传播性；

3. 下列说法正确的是（ ）

A．测试用例必须包含测试输入和预期结果两个部分；

B．若测试用例不具备感染性，则一定不能发现缺陷；

C．需求定义不完整是缺陷；

D．内存泄漏是缺陷；

4. 代码如图 4 所示，for 循环中 i>0 存在缺陷,应为 i>=0,则输入（ ）不能发现该缺陷

```
public int findLast (int[] x, int y)
{
//Effects: If x==null throw NullPointerException
//   else return the index of the last element
//   in x that equals y.
//   If no such element exists, return -1
   for (int i=x.length-1; i > 0; i--)
   {
      if (x[i] == y)
      {
         return i;
      }
   }
   return -1;
}
```

图 4

A. x={2,3,5}, y=2； B. x=Null, y=4； C. x = {1,2,4}, y=4； D. x={1}, y=4；

四、问答题

1、代码如图 5 所示，请根据代码回答如下问题

1) Identify the defect （识别代码中的缺陷）

2) If possible, identify an input that does not execute the defect. （如果存在，给出一个不能执行到该缺陷的输入）

3) If possible, identify an input that executes the defect, but does not result in a fault. （如果存在，给出一个可以执行到该缺陷但不能产生故障的输入）

4) If possible， identify an input that results in a fault, but not a failure . （如果存在，给出一个可以产生故障但不产生失效的输入）

```
4       /**
5        *  if arr is null throw NullPointException
6        *  else return the number of occurrences of 0 in arr
7        */
8   @   public int numZero(int[] arr) {
9           int count = 0;
10          for (int i = 1; i < arr.length; i++)
11              if (arr[i] == 0)
12                  count++;
13          return count;
14      }
```

图 5

2. 代码如图 6 所示，请根据代码回答如下问题

1) Identify the defect

2) If possible, identify an input that does not execute the defect.

3) If possible, identify an input that executes the defect, but does not result in a fault.

4) If possible， identify an input that results in a fault, but not a failure .

5) For the given test **(arr = {0，1，0}，expected = 2)**， identify the first faulty state （对于给定的测试用例**(arr = {0，1，0}，expected = 2)**，给出代码运行时产生的第一个故障状态）

```java
/**
 *  if arr is null throw NullPointException
 *  else return the index of the last 0 in arr
 *  return -1 if 0 does not occur in arr
 */
public int lastZero(int[] arr){
    for( int i = 0; i < arr.length; i++)
        if(arr[i] == 0)
            return i;
        return -1;
}
```

for (int i = arr.length-1; i>=0; i--)

图 6

1) Identify the defect.
2) If possible, identify an input that does not execute the defect.
3) If possible, identify an input that executes the defect, but does not result in a fault.
4) If possible identify an input that results in a fault, but not a failure.
5) For the given test (arr = {0, 1, 0}, expected = 2), identify the first faulty state

3. Answer questions (a) - (d) for the mutant (line 8) in the *findVal()* method.

(Mutant：变异体，即故意置入代码中的缺陷)

```java
1  //Effects: If numbers null throw NullPointerException
2  // else return LAST occurrence of val in numbers[]
3  // If val not in numbers[] return -1
4   public static int findVal (int numbers[], int val)
5  {
6    int findVal = -1;
7    for (int i=0; i<numbers.length; i++)
8   // for (int i=(0+1); i<numbers.length; i++) (mutant)
9    if (numbers [i] == val)
10     findVal = i;
11   return (findVal);
12 }
```

a) If possible, find a test input that satisfies reachability but not infection for the Mutant.

b) If possible, find a test input that satisfies infection, but not propagation for the Mutant .

c) If possible, find a test input that strongly kills mutant m .

d) If possible, find a test input that weakly kills mutant m .

4. Answer questions a)-e) for the following code segment

```
30      /**
31       *  if arr is null throw NullPointException
32       *  else return number of positive elements in arr
33       */
34  @   public int countPositive(int[] arr){
35          int count = 0;
36          for( int i = 0; i < arr.length; i++)
37              if(arr[i] >= 0)
38                  count++;
39              return count;
40      }
41
```

a) Identify the defect.

b) If possible, identify a test case that does not execute the defect.

c) If possible, identify a test case that executes the defect, but does not result in a fault.

d) If possible, identify a test case that results in a fault, but not a failure.

e) For the given test case (arr = {-4, 2, 0, 2}, expected = 2), identify the first error state.

Be sure to describe the complete state.

5. 指出下面 C/C++代码中的缺陷（提示：不止一个缺陷）

```
int main(int argc, char* argv[])
{
    int i, length;
    char *string1 = "Hello Everyone!";
    char *string2 = (char *)malloc(10);

    length = strlen(string2);
    printf("The initial length of string2 is:%d\n,The initial content is :%s\n",length,string2);


    for (i = 0; string1[i] != '\0'; i++) {
        string2[i] = string1[i];
    }

    length = strlen(string2);

    printf("The end length of string2 is:%d\n,The initial content is :%s\n",length,string2);

        return 0;
}
```