Departamento de Computación
Asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS (3303)
Año 2024

PRÁCTICO Nº 8

TEMA: Prueba

Objetivos

Con este práctico se espera que el estudiante pueda:

- Aplicar los criterios de prueba de caja blanca definiendo el grafo de flujo de control y calculado la complejidad ciclomática en aquellos casos en que sea necesario.
- Corregir los errores detectados en el código.
- Aplicar los criterios de prueba de caja negra.

En este práctico realizaremos pruebas de software aplicando criterios de caja blanca y de caja negra, esas pruebas las realizaremos a mano, para ayudarlos a seguir de manera organizada el procedimiento y poder realizar el informe correspondiente a la resolución de cada ejercicio de manera clara y completa tal vez pueda ayudarte consultar el documento: Cómo se hace el proceso de prueba de software de escritorio.

Primera parte: Prueba Estructural – Prueba de Caja Blanca

1) Aplicar las técnicas de prueba estructural a los siguientes ejercicios, y analizar los resultados, por ejemplo pensar ¿Qué errores se detectaron con una técnica y cuáles no?

Para cada inciso:

- definir los casos de prueba de acuerdo a cada criterio de caja blanca (cobertura de sentencias, cobertura de arcos, cobertura de condición, y cobertura de caminos).
- detectar los errores aplicando los criterios de caja blanca y proponer las modificaciones necesarias para salvar el/los error/es.

a) **Algoritmo de Euclides** (permite calcular el máximo común divisor)

```
Program Euclides;
var
  x,y: Integer;
begin
  read(x); read (y);
  while( x <> y )
     begin
       if ( x > y) then
             x := x – y;
       else
             y := y – x;
     end
  return x;
end.
```

b) **Búsqueda de un elemento en un arreglo**

```
Function pertenece (desiredElement:Integer, tabla: array of Integer): boolean
Var
  found: boolean;
  counter, numberOfItem: Integer;
begin
  found:= false;
  counter:= 1;
  while ((not found) and (counter < numberOfItem))
    begin
      if (tabla[counter] = desiredElement)
                found:= true;
      counter:= counter+1;
  endWhile
   if (found) then
       return:= true
   else
       return:= false
end.
```

c) **Mesetas**

Dado un arreglo de números enteros ordenados, imprimir la suma de cada una de sus mesetas, donde una meseta es una secuencia de números iguales.

```
Procedure suma_mesetas(arreglo: Arreglo of Integer);
var
  suma, valorOld , x : Integer;
begin
  suma:= 0;
  valorOld:= arreglo[1];
  x:= 1;
 while (x <= arreglo.lenght(arreglo))
   begin
      if (arreglo[x]<>valorOld) then
        begin
          print(suma);
          suma:= 0;
        endIf
      suma:= suma + arreglo[x];
      valorOld:= arreglo[x];
      x:= x + 1;
endwhile;
```

d) Capicúa

La función esCapicua(String cadena) retorna verdadero cuando la cadena es capicúa y falso si no lo es .

```
public boolean esCapicua(String cadena){
  int i = cadena.length();
  int j;
  boolean res = false;

  while ( j < i ){
    if ( cadena.charAt(j) == cadena.charAt(i) ) {
        j= j+1;
        i= i-1;
    } else {
        i= -100;
    }
  }

  if ( j == i ) {
     res=false;
  }
  return res;
}
```

e) Fibonacci

Dado el siguiente código que calcula y muestra por pantalla el número enésimo de la serie Fibonacci. Recuerda que la serie de Fibonacci se calcula

$$F_n = \begin{cases} 0, & \text{si } n = 0 \\ 1, & \text{si } n = 1 \\ F_{n-1} + F_{n-2}, & \text{si } n > 1 \end{cases}$$

```
public  void  fibonacci (int n) {
 int actual, ant1, ant2;
 ant1 = 0;
 ant2 = 0;
 if ((n == 1) || (n == 2)) {
    actual = 1;
 } else {
    i= 1;
    while  ( i<n ) {
      actual = ant1 + ant2;
      ant2 = ant1;
      ant1 = actual;
      i= i + 1;
      System.out.println("Fibonacci("+ n +")=" + actual);
    }
 }
}
```

Segunda parte. Prueba de Caja Negra

2) Cálculo matemático 1

a) Un programador ha implementado un programa que realiza el siguiente cálculo matemático √(a+b) /√(c-a), donde a, b y c son números enteros. Diseñe los casos de prueba utilizando el criterio de valor límite.

b) División de números enteros, determine el criterio de caja negra más apropiado.

c) Verificar si una fecha es válida, determinar el criterio de caja negra más apropiado.

d) El problema del triángulo (dado los lados de un triángulo, decir si el triángulo es equilátero, Isósceles o escaleno), determine el criterio de caja negra más apropiado.

e) Un subprograma tiene como entrada 3 parámetros enteros x, y, z. Los valores de x e y representan un intervalo [x, y]. El subprograma tiene como misión estudiar la pertenencia del valor z al intervalo. Las salidas posibles del subprograma son:

- Extremo: si z coincide con uno de los extremos del intervalo.
- Medio: si z es el punto medio del intervalo, pero no está en la situación anterior.
- Interior: si z está en el intervalo, pero no en las situaciones anteriores.

- Exterior: si z no está en el intervalo.
- Error: si la entrada es errónea.

Determine el criterio de caja negra más apropiado.

3) Cálculo matemático 2

Dada la función cálculoMatemático(a, b, c, d: Integer): float la cual realiza el siguiente cálculo:

$$\frac{\sqrt{a*(b+\sqrt{c-10})}}{d}$$

Diseñe los casos de prueba utilizando el criterio de caja negra: tabla de decisión.

4) Calculadora

Implemente una calculadora numérica en Java con las siguientes operaciones:

d) operaciones básicas (suma, resta multiplicación y división),
e) factorial de un número,
f) promedio de una lista de números,
g) raíz y potencia de un número.

a) Diseñe las pruebas para cada una de las operaciones anteriores utilizando el criterio de prueba de caja negra valor límite.

5) Clasificador de triángulos

Se desea hacer la prueba de un programa Java que determina a que clasificación pertenece un triángulo según sus ángulos (triángulo acutángulo, rectángulo, obtusángulo).
Utilice tablas de decisión y clases de equivalencia para diseñar los casos de prueba del programa anterior.

6) Empresa de transporte

La empresa de transporte “El Rápido” utiliza un programa para calcular la tarifa del boleto que debe abonar cada pasajero según: el destino (Córdoba cuesta $540 , y a Buenos Aires cuesta $1200); la antelación en la que se obtiene el boleto (sin descuento si se saca el mismo día, 15% de descuento si saca dentro de la semana anterior al viaje, y 25% si se saca con más de de una semana de anticipación); y la edad del pasajero (los mayores de 65 años tienen un 40% de descuento).

Aplica la técnica de prueba funcional de caja negra con el criterio de clases de equivalencia débil para realizar la prueba de este sistema teniendo en cuenta que el mismo realiza el cálculo a partir de la edad del pasajero, el destino, la fecha de viaje y la fecha de compra del pasaje.

a) Determina las clases de equivalencia.

b) Determina el conjunto T de casos de pruebas indicando por cada caso cuál es el resultado esperado.