

Conceptos y Paradigmas de Lenguajes de Programación - 2024 - Segundo Parcial 1ra Fecha, T2  
14/06/2024

Realice el parcial con lapicera, de otra forma se desaprobará el ejercicio.  
Se considera presentismo cuando se realiza completamente un ejercicio.

Nota: 3

Comando				
1	a	b	a	b
2	a	b	a	b
3	a	b	a	b
4	a	b	a	b
Resultado Final				

Ejercicio 1 (30p)

Realice la pila de ejecución para el siguiente código:

- a) por cadena estática b) por cadena dinámica Nota: La forma de evaluación de este lenguaje es de izquierda a derecha.  
Abs es una función que retorna el valor absoluto de la operación recibida.

```

Program Main;
Var h, a, b, c: integer;
m, n: array[1..3] of integer;

Procedure Uno(val res a: integer; nombre b:
integer);
var h: integer;
begin
  h:=5;
  a:=f+2;
  b:=b+3;
  n:=a+10;
end;

Function F: integer;
Var b: integer;
Begin
  b:=1+h;
  if (a < 3 and h > 0) then
  begin
    n(abs(a)):=n(abs(a))+1;
    a:=a+1;
  end;
end;
    
```

```

if (b < 3) then
begin
  m(b):=m(b)+n(abs(a))-2;
  m(b):=m(b)*2;
  b:=b+1;
end
else
begin
  b:=3;
end
return m(b);
end if de la función F

begin if del main
  a:=1; b:=1; h:=1;
  for c:=1 to 3 do begin
    m(c):=abs(4-c)*(-1);
    n(c):=c*2;
  end;
  Uno(m(c), n(abs(4-a)));
  for c:=1 to 3 do write (m(c), n(c));
end;
    
```

Ejercicio 2

- a) (10pts) Clasifique las siguientes estructuras de datos de acuerdo a lo visto en la práctica. Justifique en cada caso:

<p>i) Java</p> <pre> class Afiliado {   private String nyap; private int edad;   private String domicilio;   private float[] credenciales;   public void setEdad(int edad){     this.edad=edad;   }   public int getEdad(){     return this.edad;   }   //sigue el mismo en la otra columna     </pre>	<p>...en este espacio se encontrarían el resto de setters y getters</p> <pre> public float getCredencial(int i){   return this.credenciales[i]; }     </pre> <p>ii) Python</p> <pre> piezas = ("tuerca", "tornillo", "clavo") herramientas = [(1, "tuerca"), (2, "tornillo"), (3, "clavo"), (4, "martillo"), (5, "destornillador")]     </pre>
--	--

- b) (10 pts) Responda si las siguientes afirmaciones son V o F. Justifique en cada caso

- i) Los chequeos dinámicos siempre aseguran que el lenguaje sea débilmente tipado F  
ii) Las tuplas de python y las listas de pascal son ejemplos de correspondencia finita F  
iii) Los lenguajes funcionales, al igual que los lógicos, son débilmente tipados F

Ejercicio 3

- a) (15 pts) Dado el siguiente código en Java, establezca cuáles de las opciones indicadas más abajo son válidas como camino de ejecución. Justifique su selección con una breve descripción del flujo de ejecución, caso contrario no se considerará válida la respuesta)

```

1 public class JavaConExcepciones {
2
3   public static void main(String[] args) {
    
```

```

4      try {
5          for (int i = 1; i <= 5; i++) {
6              if (i==1) {
7                  System.out.println(Integer.toString(i));
8                  rethrow("Uno");
9              }
10             else {
11                 if (i==2 || i==3) {
12                     switch(i) {
13                         case 2:
14                             System.out.println(Integer.toString(i));
15                             rethrow("Dos");
16                             break;
17                         case 3:
18                             System.out.println(Integer.toString(i));
19                             rethrow("Tres");
20                             break;
21                     }
22                 }
23             }
24             else {
25                 if (i==4) {
26                     System.out.println(Integer.toString(i));
27                     rethrow("Cuatro");
28                 }
29             }
30         }
31     } catch (ThirdException | FourthException e) {
32         System.out.println(e.getMessage());
33     }
34 }
35
36 static void rethrow(String s) throws ThirdException, FourthException {
37     try {
38         try {
39             switch(s) {
40                 case "Uno":
41                     throw new FirstException("Excepción 1");
42                     break;
43                 case "Dos":
44                     throw new SecondException("Excepción 2");
45                     break;
46                 case "Tres":
47                     throw new ThirdException("Excepción 3");
48                     break;
49                 case "Cuatro":
50                     throw new FourthException("Excepción 4");
51                     break;
52             }
53         } catch (SecondException e) {
54             ThirdException e1=new ThirdException("Excepción 3");
55             throw e1;
56         }
57     } catch (FirstException e) {
58         FourthException e1=new FourthException("Excepción 4");
59         throw e1;
60     }
61 }
62 }

```

- Se imprime en pantalla "1", "Excepción 4", "2", "Excepción 3", "3", "Excepción 3", "4", "Excepción 4" y luego termina  
 i) Se imprime en pantalla "1", "Excepción 1", "2", "Excepción 2", "3", "Excepción 3", "4", "Excepción 4" y luego termina  
 ii) Se imprime en pantalla "1" y luego "Excepción 3" y luego termina  
 iii) Ninguna de las anteriores

b) (15 pts) Indique si eliminando el break de la línea 42 el resultado es el mismo. Indique qué se imprime en ese caso

#### Ejercicio 4

(20pts). Marcar si son verdaderas o falsas las siguientes afirmaciones. Acompañar la respuesta con una justificación, caso contrario, NO se tomarán como válidas

- |    |   |   |                                    |
|----|---|---|------------------------------------|
| a. | Tanto Java como C exigen que los switch tengan cláusula default   | V | <input checked="" type="radio"/> F |
| b. | Los hechos en un lenguaje lógico permiten definir relaciones entre objetos del lenguaje                                     | V | <input type="radio"/> F            |
| c. | El tipo recursión es fácilmente implementable en Java   | V | <input checked="" type="radio"/> F |
| d. | En PL/1 si se genera una excepción, se ejecuta el manejador correspondiente y el programa termina automáticamente           | V | <input checked="" type="radio"/> F |
| e. | La sentencia else de python en el manejo de excepciones se ejecuta siempre sin importar si hubo ejecución de manejador o no | V | <input type="radio"/> F            |