

CyPLP 2024

EMT-3-B

### Verdadero o Falso

Marque con una cruz las respuestas falsas.

- ☒ El nivel de polimorfismo de un lenguaje se refiere pura y exclusivamente a modelar subtipos y herencia.
- ☒ Un lenguaje de programación representa un modelo para resolver problemas computacionales
- ☐ La forma de continuación para el manejo de excepciones es una de las preguntas que debemos respondernos cuando evaluamos este concepto en un lenguaje de programación.

### Pregunta

```
1  procedure Program is
2  e: exception
3  ...
4  Begin
5  ...
6  .. if (condError) then
7      raise e;
8  endif
9  ....
10 Exception
11 when e : begin ...
12             ... (raise)
13             End
14 When others Begin
15             ...
16             End
17 end Program;
```

Para este programa en ADA indique de qué forma simularía el método por reasunción utilizando los constructores del lenguaje cuando se levanta una excepción en la línea 7. Justifique indicando qué instrucciones utilizaría y en qué lugar del código las ubicaría (números de línea). Puede incluir pseudocódigo. Indique la diferencia conceptual entre la respuesta dada y el código original.

### Respuesta

Para simular el método de reasunción se podría agregar una sentencia `Declare` inmediatamente antes de la línea 6, incluyendo un `begin` luego de la sentencia `Declare`.  
Así mismo, luego de la línea 8, se encontrarían los manejadores de excepciones (es decir desde la línea 10 hasta la 16), y luego de los manejadores cruzar `end` para indicar el fin del `Declare` y el resto de sentencias que se encontraban en la línea 9.

SIGUE EN LA HOJA

## Pregunta

Lógico	imperativo
$rel(a,b).$ $rel(b,c).$ $rel(a,c):-rel(a,b),rel(b,c).$	$var\ a,b,c:integer;$ $rel(a,b); //relaciona\ a\ la$ $variable\ a\ con\ b\ y\ retorna\ true$ $rel(b,c); //relaciona\ a\ la$ $variable\ a\ con\ b\ y\ retorna\ true$ $if(rel(a,b)\ and\ rel(b,c))\ rel(a,c)$

Indique cuál de los dos códigos pertenece a un lenguaje lógico y qué componentes (cláusulas) encuentra en él describiéndolos claramente.

Respuesta

El código de la izquierda pertenece a un lenguaje lógico. En el mismo se hace uso de los cláusulas de Horn. En particular, las primeras 2 líneas son hechos, es decir, verdades, que, en este caso, indican que hay una relación entre  $a$  y  $b$  (línea 1) y entre  $b$  y  $c$  (línea 2).

Por su parte, en la línea 3 se utiliza una regla, la cual tiene la forma:

conclusión :- condición

↓

Esto indica que, de cumplirse la condición (en el ejemplo sería  $rel(a,b), rel(b,c)$ ), la conclusión será verdadera.

Cabe recalcar que los comas en la condición son operadores de conjunción (and lógico).

Por último, en el ejemplo, dado que  $rel(a,b)$  es verdadero y  $rel(b,c)$  también (ambos son hechos), entonces  $rel(a,c)$  será verdadero.



## 2) pseudocódigo

procedure Program is

e: exception

Begin

...

Declare

begin

if (cond Error) Then

raise e;

endif

Exception

When e: begin ...

(raise)

End

When others: begin

...

End

end declare

... 3 sentencias que previamente se encontraban

end Program

en la línea 9

⚡

Lo que se logra de esta manera es que, luego de que se levante la excepción e, en la línea 7, la misma se maneje dentro del bloque Declare y luego continúe la ejecución normal del programa, a partir de la línea 9, simulando así el método por resurrección.

Previamente, luego de ~~ser~~ levantarse la excepción

la misma se ~~se~~ manejaba en el programa principal y luego de ser manejada finalizaba la ejecución del mismo.