

Fundamentos de la programación y la informática

Examen de teoría. 11 de Junio de 2019

Grado en ingeniería aeroespacial en aeronavegación
Grado en ingeniería aeroespacial en vehículos aeroespaciales
Universidad Rey Juan Carlos

Preparativos

- Ejecuta el script `prepara`
- Esto creará en tu ordenador el fichero `~/fpi/TULOGIN/teoria.txt`, donde TULOGIN es tu nombre de usuario en el laboratorio. Escribe tus respuestas en este fichero.

Ejercicio 1 (2 puntos)

Sean a, b, c, d variables booleanas. Sean x e y variables reales.
Sean las expresiones

```
e1 := not ( ( b or c ) and not ( b and d or a ) );
```

```
e2 := not ( x <= 10 ) or not ( y = 0 );
```

1. A partir de $e1$, escribe una expresión lógica equivalente, más legible, de nombre $s1$.
2. A partir de $e2$, escribe una expresión lógica equivalente, más legible, de nombre $s2$.
3. Escribe una expresión $n1$ que sea la negación de $s1$.
4. Escribe una expresión $n2$ que sea la negación de $s2$.

Solución

```
s1 := (not b and not c) or (b and d or a);
```

```
s2:= (x > 10) or (y <> 0);
```

```
n1 := (b or c) and not(b and d or a);
```

```
n2:= (x <= 10) and (y = 0);
```

Ejercicio 2 (8 puntos)

Como sabes, en un programa mal escrito nos podemos encontrar con errores de compilación, errores de ejecución, errores lógicos y defectos en la claridad del código. El siguiente programa está muy mal escrito. Encuentra y describe brevemente todos los errores y todos los defectos que encuentres.

- Deja claro si se trata de un error o un defecto, aunque no es necesario que especifiques si el error es de compilación, ejecución o lógico.

- Obviamente, el número que aparece al principio de cada línea no forma parte del programa, es el número de línea. Para cada error que encuentres, indica en qué línea está.

Por ejemplo, un error encontrado en el código podría describirse de la siguiente forma:
Línea 23: Error: El identificador 'inexistente' no es válido porque contiene el carácter 'é', que no está permitido.

```

1 {$mode objfpc}{$H-}{$R+}{$T+}{$Q+}{$V+}{$D+}{$X-}{$warnings on}
2 program aeropuertos_01;
3 {Este programa inicia un array de registros conteniendo código y altitud de
4 aeropuerto, luego suma todas las altitudes.}
5
6 const
7     Num_aeropuertos = 2;
8 type
9     tipo_aeropuerto = record
10         código : string;
11         altitud : integer;
12     end;
13     TipoAeropuertos = array[1..Num_aeropuertos] of tipo_aeropuerto;
14
15 function suma_altitudes(aeropuertos:TipoAeropuertos): integer;
16 var
17     i, suma : integer;
18 begin
19     for i = 1 to Num_aeropuertos-1 do begin
20         if aeropuertos[i].altitud > 0 then
21             suma := suma + aeropuertos[i].altitud;
22         else
23             suma := suma + aeropuertos[i].codigo;
24             writeln('WARNING: Encontrado aeropuerto con altitud cero');
25         end;
26     aeropuertos.result := suma;
27 end;
28
29 procedure inicia_aeropuertos(var aeropuertos:TipoAeropuertos);
30 begin
31     aeropuertos[1].codigo := 'MAD';
32     aeropuertos[1].altitud := 667;
33     aeropuertos[2].codigo := 'LHR';
34     aeropuertos[2].altitud := 11;
35 end;
36
37 var
38     tipo_aeropuerto : TipoAeropuertos;
39 begin
40     writeln(inicia_aeropuertos(tipo_aeropuerto));
41     writeln(suma_altitudes((tipo_aeropuerto)));
42 end;

```

solucion

Línea 9: Defecto. Según el criterio que seguimos un tipo debería estar escrito en NotaciónCamello, es decir sin barra baja ni ningún otro elemento que separe las palabras y con la primera letra de cada palabra en mayúscula.

Línea 10: Error. El carácter "ó" no es válido para identificadores, habría que quitarle la tilde.

Línea 18: Error. Falta inicializar la variable suma, concretamente con el valor 0 para que funcione correctamente.

Línea 19: Error. El operador de asignación es :=, no =. Por tanto, la asignación debería ser `i := 1`, no `i = 1`.

Línea 19: Error. El bucle debería ser de 1 a Num_aeropuertos para recorrer el array en su totalidad.

Línea 21: Error. Sobra el ";" al final de la línea ya que interrumpe el `if then else`.

Línea 23: Error. No es posible sumar "suma" que es un integer con `.aeropuertos[i].codigo` que es un string.

Líneas 22-25: Falta el `begin-end` del `else`. Tal y como está escrito el programa, la sentencia `writeln` está fuera del bucle, a pesar de que la tabulación indica lo contrario.

Línea 24: Error. Es una función que tiene efectos laterales.

Línea 26: Error. Debería poner simplemente `result:= suma;`, ya que es como se fija el resultado de una función. Además la sentencia que aparece contiene varios errores: `aeropuertos` no se puede modificar ya que no se ha pasado por referencia (`var`), y aunque se pudiese modificar el record no contiene ningún campo `result`.

Línea 38: Error. El nombre de la variable es el mismo que el del record, identificador duplicado.

Línea 40: Error. El argumento de un subprograma, en este caso `writeln`, no puede ser un procedimiento, porque un procedimiento no devuelve ningún valor.

Línea 41: Error. Sobra un paréntesis de apertura, debería ser `writeln(suma_altitudes(tipo_aeropuerto));`.

Línea 42: Error. El programa debe terminar con un "." no un ";".