

Fundamentos de la programación y la informática

Test final. 16 de junio de 2025

Grados en ingeniería aeroespacial. Turno de tarde

Universidad Rey Juan Carlos

- Contesta brevemente las siguientes preguntas.

Ejercicio 1 (2.5 puntos)

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

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

```
e2 := not (x > 6) and (y >= 10);
```

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

Respuesta

En este ejemplo, ninguna expresión usa simultáneamente los operadores *and* y *or*, por tanto no son necesarios los paréntesis. Aunque no se ha penalizado su uso porque pueden aportar claridad.

```
s1 := a or not b or not c or d;  
s2 := x <= 6 and y >= 10;  
n1 := not a and b and c and not d;  
n2 := x > 6 or y < 10;
```

Ejercicio 2 (2.5 puntos)

Dentro del mismo programa en Pascal, ¿se puede usar el mismo identificador para dos cosas completamente distintas? ¿Sí? ¿No? ¿Depende? Si depende, ¿de qué depende?

Respuesta

Depende. En Pascal, como en la mayoría de lenguajes de programación, se puede usar el mismo nombre en dos variables locales o parámetros, siempre que estén en subprogramas distintos ¹

En cambio, dentro de un mismo subprograma, no se puede repetir un nombre para dos cosas distintas. Por ejemplo, no puede haber dos variables con el mismo nombre, ni una variable y un procedimiento que se llamen igual ².

Ejercicio 3 (2.5 puntos)

¿Qué es una *precondición*? ¿Qué puede suceder en un programa que no maneje correctamente las precondiciones?

Respuesta

Una precondición es algo que debe cumplirse antes de ejecutar un subprograma. Si no se cumple, el programa compilará, pero probablemente fallará o dará resultados incorrectos. Un programa bien escrito comprueba todas las precondiciones de todos sus subprogramas, y actúa de forma adecuada si alguna falla³.

Ejercicio 4 (2.5 puntos)

¿Qué hace esta sentencia, para qué sirve?

```
if x mod 2 = 0 then
  writeln('tal tal')
else
  writeln('cual cual');
```

Respuesta

Si el resto de la división entera de x entre 2 es 0, escribe *tal tal*. Esto es, si el número es par. En otro caso, número impar, escribe *cual cual*.

¹Los nombres siempre deberían ser razonables y descriptivos, pero aún así, el mismo nombre puede referirse a cosas distintas. Por ejemplo 'i' puede ser el índice en dos estructuras muy diferentes, 'fichero', 'n', 'auxiliar', 'entrada', 'nombre', 'clave' son identificadores habituales que en distintos ámbitos se referirán a distintas entidades, 'peso' puede ser 'masa' o puede ser 'relevancia', 'tiempo' puede ser cronológico o meteorológico, etc etc.

²Ahora bien, si una variable local o un parámetro tiene el mismo nombre que un procedimiento, el nombre pasa a referirse a la variable o al parámetro, y el procedimiento queda oculto. Esto no da error de compilación, pero ya no se puede usar ese procedimiento en ese lugar

³Si las precondiciones son sencillas se pueden comprobar en el propio subprograma, si no, en una nueva función con este propósito.