

Fundamentos de la programación y la informática

Test final

21 de junio de 2024

Grados en ingeniería aeroespacial. Turno de tarde
Universidad Rey Juan Carlos

Ejercicio 1 (3 puntos)

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

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

```
e2 := not (x <= 3) or (y >= 0);
```

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 $s2$.

Solución

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

```
s2 := (x > 3) or (y >= 0);
```

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

```
n2 := (x <= 3) and (y < 0);
```

Ejercicio 2 (2.5 puntos)

1. ¿Qué es un *error lógico*?
2. ¿Qué es un *error de compilación*?
3. ¿Qué es error de tipo *defecto en la claridad del código*?
4. ¿Qué es un *error en tiempo de ejecución*?

Respuesta

1. Un *error lógico* es un error en la lógica del programa. Esto es: el programa tiene una sintaxis correcta y se compila sin errores. Pero tiene un comportamiento que no es correcto: hace lo que no tiene que hacer, devuelve datos que están mal, tarda más de lo debido, no acaba nunca, etc.
2. Un *error de compilación* es un error que detecta el compilador porque lo que está escribiendo el programador incumple las normas del lenguaje de programación.
3. Un error de tipo *defecto en la claridad del código* es un error que hace que el código sea confuso, con diseño pobre, difícil de entender por el programador. Un programa con este error se comporta correctamente, hace lo que debe. Pero en el futuro, cuando haya que retocarlo, reutilizar el fuente, adaptarlo, etc, es muy fácil que haya problemas.
4. Un *error en tiempo de ejecución* es un error que aparece cuando el programa se está ejecutando, porque sucede algo que no debería suceder. Por ejemplo un fichero que no se puede usar, o un intento de división entre cero, o que la memoria se ha agotado, o que intentamos acceder a una posición del array que no existe, que la red haya fallado, etc.

Ejercicio 3 (2 puntos)

¿Qué significa *acceso secuencial a un fichero*?

Respuesta

Leer o escribir un fichero en orden, desde el principio al final. Primera posición, segunda posición, tercera, etc. Sin poder saltar de una parte a otra. Como ocurría con las *cassetes* de audio o las cintas VHS.

Se contrapone al *acceso aleatorio* que permite al programador indicar qué parte del fichero se va a leer o escribir, en cualquier orden que sea conveniente. Similar a lo que se puede hacer con un disco de vinilo o con un CD.

Ejercicio 4 (2.5 puntos)

¿Qué es una *palabra reservada* ¿Qué es un *identificador*?

Respuesta

Una *palabra reservada* es una palabra que tiene un significado especial en un lenguaje de programación y que por tanto el programador no puede usar para

otra cosa distinta de la prevista. Por ejemplo, en Pascal, *program*, *var*, *for*, *begin*, etc.

Un *identificador* es una palabra que usa el programador para nombrar de forma única una entidad de su programa: una variable, una constante, una función, un procedimiento, un tipo de datos definido por el programador, etc. Por ejemplo, *notaAlumno*, *cuenta_palabras* o *epsilon*.