Prácticas sobre expresiones

Fundamentos de la programación y la informática
Grado en ingeniería aeroespacial en vehículos aeroespaciales 2019-2020
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

Práctica 2.1. Casting

En este ejercicio probarás el *casting* de tipos de datos

1. Crea el directorio

   `~/fpi/practica02`

2. Configura tu editor para que muestre los caracteres invisibles. Observa cómo representa los tabuladores y cómo representa los espacios. Presta mucha atención al sangrado de tu programa, en este ejercicio y en todos los que escribas durante el resto del curso.

3. Escribe un programa llamado `~/fpi/practica02/ahormados.pas`

   Será parecido al programa *casting* del tema 2, pero no idéntico.

4. El programa debe hacer al menos 4 conversiones de tipos de dato correctas.

5. Escribe también un par de ejemplos incorrectos. Una vez que compruebes que dan error de compilación, *comenta* esas líneas.

   (Comentar una línea es una expresión habitual en programación. No significa que hagas un comentario explicando la línea, sino que deshabilites la línea convirtiéndola en un comentario. No tendrá efecto, pero seguirá presente en el código)

6. Usa solamente constantes y expresiones. Ni funciones ni variables ni sentencias if-then-else.

Práctica 2.2. IVA

Escribe un programa llamado `~/fpi/practica02/iva.pas` que, a partir de una constante con un precio sin IVA, muestre el importe del IVA y el precio final con IVA.

Y que a partir de una constante con un precio con IVA, muestre el importe del IVA y el precio sin IVA.

- En todos los casos entenderemos siempre que se trata del IVA general (el 21%).
- En contabilidad hay ciertas normas sobre el redondeo de los céntimos del IVA. Pero aquí lo ignoraremos.
- Usa solamente constantes y expresiones. Ni funciones ni variables ni sentencias if-then-else.

Ejemplo:

Al ejecutar tu programa, saldrá algo similar a esto:

```
Precio sin IVA: 200.00€  IVA: 42.00€   Precio con IVA: 242.00€

Precio con IVA: 1000€   IVA: 173.55€  Precio sin IVA: 826.44€
```

Práctica 2.3. Construcción si-entonces en lenguaje natural

Las siguientes expresiones tienen la imprecisión propia del lenguaje natural. Supongamos que queremos escribirlas de forma algorítmica. Prepara un fichero llamado `~/fpi/practica02/no_ambiguo.txt` y para cada una de ellas:

- Indica si parece una implicación o parece una equivalencia (muchas son discutibles, no tienen una única solución correcta)
- Vuelve a redactarla sin ambigüedades. Observa que no se te pide un programa en Pascal, sino una redacción en lenguaje natural, pero sin ambigüedades.

Expresiones:

1. Si vas a cambiar de carril, pon el intermitente.
2. Busque, compare y si encuentra algo mejor, cómprelo.
3. Si lo pruebas, repites.
4. Si me quieres, vende la moto.

5. Si la bolsa se rompe, tráigala y le daremos otra.

Práctica 2.4. Expresiones lógicas

Este ejercicio enumerará una serie de expresiones lógicas en lenguaje natural, tendrás que escribirlas en pascal, afirmadas y negadas, escribir un programa que las muestre en pantalla y elegir la versión más clara.

A continuación se detallan una serie de equivalencias lógicas, en lenguaje natural.

- Si el cliente tiene 65 o más años, o si tiene 16 o menos, o tiene una discapacidad, tiene precio reducido. Y si no, tiene precio normal.
- Si tengo capacidad, constancia y se dan las circunstancias adecuadas, tendré un gran éxito profesional. Y si no, seré del montón.
- Si voy deprisa o si el autobús sale tarde, llegaré a tiempo. En otro caso, llegaré tarde.
- Si se enciende la alarma y no es un simulacro, entonces me muevo. Y si no, me quedo quieto.

(Fíjate en que son equivalencias porque llevan una expresión del tipo *y si no, no.* En otro caso, serían simples implicaciones)

Escribe estas equivalencias en Pascal, en un programa llamado `~/fpi/practica02/equivalencias.pas`. Hazlo similar al ejemplo *bisiesto* de la transparencia 28 del tema 2, según la siguiente especificación:

1. A partir de cada equivalencia escribirás dos expresiones booleanas, por tanto escribirás un total de 8 expresiones booleanas.
2. En una de las expresiones el segundo término de la equivalencia estará en positivo, y en otra, en negativo.

   Ejemplo:

   Tiene precio reducido

   ```
   (edad >= 65) or (edad <= 16) or discapacidad
   ```

   Tiene precio normal

   ```
   (edad < 65) and (edad > 16) and not discapacidad
   ```

   Usa en ambos casos las constantes *edad* y *discapacidad* (no te inventes una constante distinta como *capacidad*)

   De la misma forma, escribe

- Una expresión para tener éxito y una equivalente para ser del montón. A partir de las constantes capacidad, constancia y circunstancias.
- Una expresión para llegar a tiempo y otra equivalente para llegar tarde. A partir de las constantes ir_deprisa y autobus_retrasado
- Una expresión para moverse y otra equivalente para quedarse quieto. A partir de las constantes alarma y simulacro.

3. De cada par de expresiones, indica, en un comentario, la que te parece más adecuada para un programa. Esto es, la más legible, la que tenga en total menos negaciones. Si en algún caso la claridad de ambas formas te parece similar, indícalo también.
4. Usa solamente constantes y expresiones. Ni funciones ni variables ni sentencias if-then-else.

Práctica 2.5. Sustentación

Escribe un programa llamado `~/fpi/practica02/sustentacion.pas` que calcule la fuerza de sustentación de un avión en Newtons a partir de la densidad del aire, la velocidad del avión en m/s, la superficie alar y el coeficiente de sustentación. Solo puedes usar constantes y expresiones (no funciones ni sentencias if-then-else). Invéntante los valores de entrada.

- Emplea la notación de esta página de wikipedia.
- Usa solamente constantes y expresiones. Ni funciones ni variables ni sentencias if-then-else.