Problema 9

UNIVERSIDAD DE CANTABRIA

Escribir un programa que permita obtener la derivada de una función de la variable X, expresada mediante un árbol binario, que contenga:

- **números reales constantes**
- **la variable X**
- **y los operadores binarios +,-,*,/ y ^ (elevar a).**

Los métodos de la clase a desarrollar son:

- **El constructor, que recibe como argumento de entrada el árbol con la expresión a derivar.**
- **Un método que realiza la derivada de la expresión.**
- **Un método que realiza la simplificación de una expresión.**

Problema 9 (cont.)

Para llevar a cabo la derivación y simplificación se utilizarán las siguientes reglas:

Der(cte) = 0 Der(X) = 1 Der(U±V) = Der(U) ± Der(V) Der(U*V) = U*Der(V) + V*Der(U) Der(U/V) = ((V*Der(U)) - (U*Der(V)))/V**2 Der(U^V) = V*(U^(V-1))*Der(U)	S±0=S 0+S=S 0-S=-S S*0=0 0*S=0 S*1=S 1*S=S 0/S=0 S^0=1 S^1=S

Problema 9 (cont.)

UNIVERSIDAD DE CANTABRIA

Pseudocódigo:

```
metodo deriva (Iterador<ElementoDeExpresion> iter)
retorna ArbolBinCE <ElementoDeExpresion>

  si iter es una hoja entonces
    si
      iter.contenido es una Constante =>
        retorna nuevo ArbolBinCE (cero)
      iter.contenido es una Variable  =>
        retorna nuevo ArbolBinCE (uno)
    fsi
  si no // es un operador
```

Problema 9 (cont.)

```
    si
      iter.contenido = ‘+’ | ‘-’ =>
         retorna nuevo ArbolBinCE (‘+’ o ‘-’,
           deriva(ramaIzquierda),
           deriva(ramaDerecha)
      iter.contenido = ‘*’ =>
         retorna nuevo ArbolBinCE (‘+’,
           ramaIzquierda*deriva(ramaDerecha),
           deriva(ramaIzquierda)*ramaDerecha)
      ...
    fsi
  fsi
fmétodo
```

Problema 9 (cont.)

```
método simplifica
    (ArbolBin<ElementoDeExpresion> arbol)
retorna ArbolBin<ElementoDeExpresion>
     boolean hayCambios;
     ArbolBin<ElementoDeExpresion> nuevo=arbol
     hacer
       hayCambios:=false;
       nuevo:=simplificaUnaVez(nuevo)
     mientras hayCambios
     retorna nuevo
fmétodo
```

El metodo `simplifica` usa un metodo auxiliar `simplificaUnaVez`, que hace una simplificación y anota en `hayCambios` si ha habido cambios o no

Problema 9 (cont.)

```
metodo simplifica_una_vez
    (ArbolBin<ElementoDeExpresion> arbol)
retorna ArbolBin<ElementoDeExpresion>
  IterARbolBin<ElementoDeExpresion> iter =
   arbol.iterador();
  si iter es una hoja entonces
        retorna arbol;
  fsi;
  si
    iter.contenido = '+' | '-' =>
      si ramaIzquierda de iter es = 0 entonces
        hayCambios=true;
        retorna simplificaUnaVez
           (ramaDerecha de iter);
      fsi
```

Problema 9 (cont.)

```
            // Igual para la rama izquierda
                        ..
  fsi

  // si llegamos aquí, no ha habido simplificación
  retorna nuevo ArbolBinCE<ElementoDeExpresion>
    (iter.contenido(),
      simplificaUnaVez (RamaIzquierda de iter),
      simplificaUnaVez (RamaDerecha de iter));
fmétodo
```