Práctica 8: Árboles Binarios de Búsqueda Equilibrados

Estructuras de Datos y de la Información

Implementar el TAD ArbolBinAVL, recordar que es una particularización del TAD ArbolBin, pero que además debe incluir las siguientes operaciones:

```
ESPECIFICACION ArbolBinAVL
USA ArbolBin
PARÁMETROS FORMALES
      TIPOS TipoElemento
FIN PARAMETROS

TIPOS TArbolBinAVL

OPERACIONES
      (*Constructoras no generadoras*)
      PARÁMETROS GENÉRICOS
            Menor: TElem X TElem -> Boolean
      (* o Mayor: TElem X TElem -> Boolean *)
            Iguales: TElem X TElem -> Boolean
      FIN PARÁMETROS GENÉRICOS

      InsertarEnAVL: TArbolBinAVL X TElem -> TArbolBinAVL
      EliminarEnAVL: TArbolBinAVL X TElem -> TArbolBinAVL
      ReequilibrarAVL: TArbolBinAVL -> TArbolBinAVL
      PARCIAL RotacionSimpleDerecha: TArbolBinAVL ->TArbolBinAVL
      PARCIAL RotacionSimpleIzquierda: TArbolBinAVL -> TArbolBinAVL
      PARCIAL RotacionDobleIzquierdaDerecha: TArbolBinAVL -> TArbolBinAVL
      PARCIAL RotacionDobleDerechaIzquierda: TArbolBinAVL -> TArbolBinAVL

      (*Observadoras no selectoras*)
      PARCIAL Minimo: TArbolBinAVL -> TElem
      Pertenece: TElem X TArbolBinAVL -> Boolean
      PARCIAL FactorEquilibrio: TArbolBinAVL -> integer
VARIABLES
      i,i2,i3,d,d2,d3: TArbolBinAVL;
      e, e2, e3: TElem;

ECUACIONES DE DEFINITUD
      DEF(RotacionSimpleDerecha(Construir(i,e,Construir(i2,e2,d2)))
      DEF(RotacionSimpleIzquierda(Construir(Construir(i2,e2,d2),e,d))

DEF(RotacionDobleIzquierdaDerecha(Construir(Construir(i2,e2,Construir(i3,e3,d3))
,e,d)));
DEF(RotacionDobleDerechaIzquierda(Construir(i,e,Construir(Construir(i3,e3,d3),e2
,d2)));
      DEF(Minimo(Construir(i,e,d)));
      DEF(FactorEquilibrio(Construir(i,e,d)));

ECUACIONES

      InsertarEnAVL(CrearArbolBinVacio, e) =
      Construir(CrearArbolBinVacio, e, CrearArbolBinVacio)
      InsertarEnAVL(Construir(i,e2,d), e) =
      SI Menor(e,e2)-> ReequilibrarAVL(Construir(InsertarEnAVL(i,e),e2,d))
      | ReequilibrarAVL(Construir(i,e2,InsertarEnAVL(d,e)))

      EliminarEnAVL(CrearArbolBinVacio,e) = CrearArbolBinVacio
      EliminarEnAVL(Construir(i,e2,d),e) =
      SI Iguales(e2,e) ->
            ReequilibrarAVL(Construir(i,Minimo(d),EliminarEnAVL(d,Minimo(d))))
      | SI Menor(e,e2) -> ReequilibrarAVL(Construir(EliminarEnAVL(i,e),e2,d))
        | ReequilibrarAVL(Construir(i,e2,EliminarEnAVL(d,e)))
```

```
      ReequilibrarAVL(CrearArbolBinVacio) = CrearArbolBinVacio
      ReequilibrarAVL(Construir(i,e,d)) =
            SI FactorEquilibrio(Construir(i,e,d)) = 2 ->
                  SI FactorEquilibrio(d) = 1 ->
                        RotacionSimpleDerecha(Construir(i,e,d))
                  |     RotacionDobleDerechaIzquierda(Construir(i,e,d))
            | SI FactorEquilibrio(Construir(i,e,d)) = -2 ->
                  SI FactorEquilibrio(i) = -1 ->
                        RotacionSimpleIzquierda(Construir(i,e,d))
                  |     RotacionDobleIzquierdaDerecha(Construir(i,e,d))

      RotacionSimpleDerecha(Construir(i,e,Construir(i2,e2,d2))) =
      Construir(Construir(i,e,i2),e2,d2)

      RotacionSimpleIzquierda(Construir(Construir(i2,e2,d2),e,d)) =
      Construir(i2,e2,Construir(d2,e,d))

      RotacionDobleIzquierdaDerecha(Construir(i,e,d)) =
      RotacionSimpleIzquierda(Construir(RotacionSimpleDerecha(i),e,d))

      RotacionDobleDerechaIzquierda(Construir(i,e,d)) =
      RotacionSimpleDerecha(Construir(i,e,RotacionSimpleIzquierda(d)))


      (*Observadoras no selectoras*)
      Minimo(Construir(i,e,d)) = SI EsArbolBinVacio(i) -> e
                                 | Minimo(i)

      Pertenece(e, CrearArbolBinVacio) = FALSE
      Pertenece(e, Construir(i,e2,d)) = SI Iguales(e,e2) -> TRUE
                                       |SI Menor(e,e2) -> Pertenece(e,i)
                                        | Pertenece(e,d)

      FactorEquilibrio(Construir(i,e,d)) = Altura(d) - Altura(i)

FIN ESPECIFICACION
```

Realizar un programa de prueba que cree un árbolAVL, con la siguiente secuencia de inserción: 5, 9, 7, 15, 20, 17, 19 y 23. Para comprobar que las inserciones se realizan correctamente se debe mostrar por pantalla los recorridos preorden, inorden y postorden después de cada inserción.
Después se procederá al borrado de los siguientes nodos: 15, 19, 20 y 23. Al igual que antes, después de cada borrado se debe mostrar por pantalla los recorridos preorden, inorden y postorden después de cada borrado.

Nota 1: Se considera necesario un nivel elevado de encapsulación y abstracción.
Nota 2: Se hará uso de las normas de estilo dictadas en clase (cabecera del fichero, interfaz de la unidad con precondiciones, postcondiciones, excepciones, implementaciones con el análisis de complejidad de cada operación, nombres coherentes de variables y operaciones,...)
Plantilla de cabecera del fichero:

```
{*****************************************************************************
*                                                                            *
*      Módulo:                                                               *
*      Tipo:  Programa()     Interfaz-Implementación TAD ()       Otros()     *
*      Autor/es:                                                             *
*      Fecha de actualización:                                               *
*      Descripción:                                                          *
*                                                                            *
*****************************************************************************}
```