

EjercicioGenealógico

Estructuras de Datos

Tema 4: árboles

1º Grado en Ingeniería de la Computación

© Profesor Dr. Carlos Grima Izquierdo (www.carlosgrima.com)

URJC (www.urjc.es)

Programa en C++ un árbol genealógico.

Tendremos una clase (o struct) "Nodo" con, al menos, los siguientes atributos:

- El elemento que se quiere guardar en el nodo. En nuestro caso será un objeto de tipo "string" que representará el nombre de una persona dentro del árbol genealógico.
- Un puntero al nodo padre. NULL si no hay padre (estamos en la raíz del árbol).
- Un puntero al primer hijo. NULL si no hay.
- Un puntero a su hermano derecho. NULL si no hay.

Tendremos una clase "ArbolGenealogico" que será un conjunto de nodos enlazados unos a otros. Tendrá los siguientes dos atributos:

- Puntero al nodo raíz. Al igual que en una lista enlazada sólo necesitamos el puntero al primer elemento, en un árbol sólo necesitamos el puntero al nodo raíz. Será NULL si el árbol está vacío.
- El tamaño actual del árbol ("n").

Primero programaremos varios métodos privados en el árbol:

- Un método para obtener el número de hijos directos que tiene un nodo. El único parámetro es el punto al nodo cuyo número de hijos queremos obtener. Devuelve el susodicho número de hijos. La complejidad temporal es $O(\text{hijos})$, siendo hijos el número de hijos directos del nodo pasado como argumento. Esta complejidad también la tienen otros métodos de este ejercicio.
- Un método para obtener el puntero a uno de los nodos hijos directos de otro nodo. Tiene dos parámetros: el puntero al nodo padre, y el número de hijo que queremos obtener (el primer hijo es el número 0). Devuelve el puntero al nodo que contiene ese hijo.
- Un método para buscar el nodo que contiene un contenido dado. Este método es recursivo y le pasamos dos parámetros: el nodo raíz del subárbol en donde buscar, y lo que queremos buscar. Mira a ver si el nodo raíz contiene lo buscado (caso trivial). Si no, busca recursivamente en cada uno de los subárboles que cuelgan del nodo raíz. La complejidad temporal de este método vemos intuitivamente que es $O(n)$, ya que en el peor caso habrá que mirar todos los nodos del árbol (n nodos). Sin embargo el cálculo riguroso (usando $T(n)$) no la sabemos calcular en este curso, pues necesitaríamos estimar el número medio de hijos por nodo, la altura y profundidad del árbol, etc. Lo mismo ocurre con casi todo el resto de métodos: el cálculo de su complejidad se sale del ámbito de esta asignatura, aunque intuitivamente vemos que será $O(n)$.

- Un método recursivo para imprimir un subárbol por pantalla. Imprime el árbol en forma de esquema sangrado mediante tabulaciones (mira las pruebas de caja negra más abajo). Tendrá dos parámetros: el puntero al nodo raíz del subárbol, y el número de tabulaciones que tenemos que imprimir justo antes de imprimir dicho nodo raíz. Nota que, metiendo este método (que es de interfaz) en la clase ArbolGenealogico (que es de modelo), estamos violando el principio de separación entre interfaz y modelo (lo hacemos por simplicidad).
- Un método recursivo para eliminar todos los nodos de un subárbol (es decir liberar la memoria de todos esos nodos). El único parámetro es un puntero al nodo raíz del subárbol que queremos eliminar. Piensa que, antes de eliminar un nodo, hay que eliminar previamente todos sus nodos descendientes, de tal modo que el último nodo eliminado es la raíz del subárbol. Es decir vamos eliminando nodos desde abajo hacia arriba.

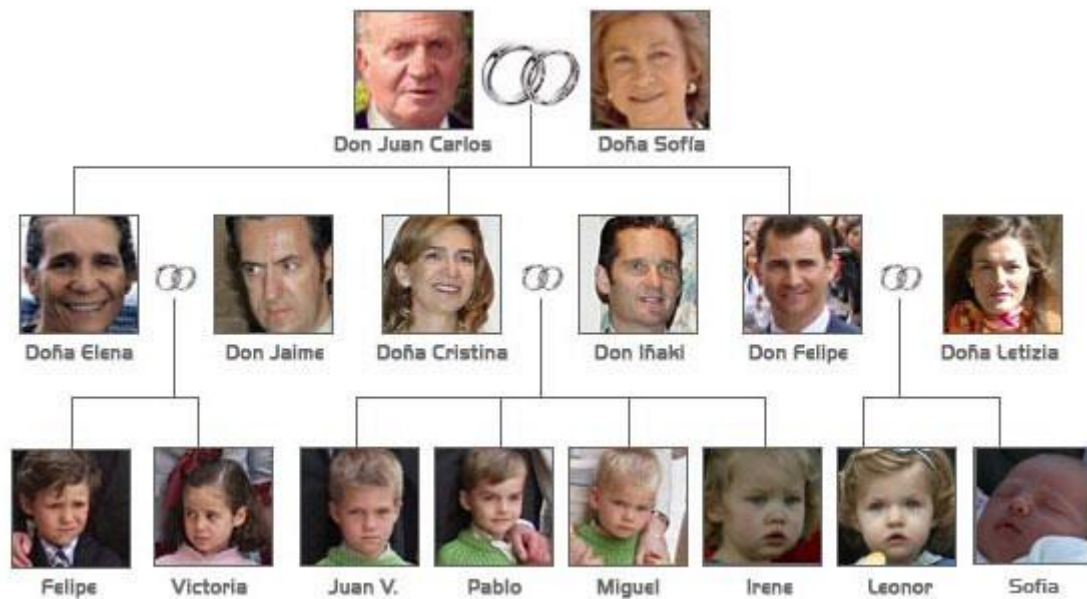
Los métodos públicos del árbol serán los siguientes:

- Constructor. Construye un árbol con un solo nodo (su raíz). Le pasamos el contenido de dicha raíz.
- Método para insertar un nuevo nombre en el árbol. Pasamos el nombre que queremos insertar y el nombre de su padre. El árbol buscará el nodo que contenga el nombre del padre (usando el método recursivo “buscar()” descrito arriba) e insertará el nuevo nombre como su último hijo directo (es decir: el hijo más a la derecha). El nombre del padre debe existir en el árbol (precondición).
- Método para imprimir todo el árbol por pantalla. Llamará al método privado recursivo homónimo, pasándole el puntero a la raíz del árbol completo (es decir, en la primera llamada al método recursivo el subárbol será todo el árbol). Nota que, metiendo este método (que es de interfaz) en la clase ArbolGenealogico (que es de modelo), estamos violando el principio de separación entre interfaz y modelo (lo hacemos por simplicidad).
- Destructor. Elimina todos los nodos del árbol. Llamará al método privado recursivo homónimo, pasándole el puntero a la raíz del árbol completo (es decir, en la primera llamada al método recursivo el subárbol será todo el árbol).

Nota que puede haber muchas más precondiciones que las enunciadas aquí. Pon todas las que creas necesarias. Recuerda que las precondiciones están para ayudarte a desarrollar más rápido el programa al poder encontrar errores más rápido que si no las pusieras.

El main nos pedirá el nombre de nuestro antepasado más lejano, y luego mostrará un menú que permita insertar un nuevo nombre o imprimir todo el árbol.

Un ejemplo podría ser con la antigua familia real española, obviando los/as consortes:



```

C:\WINDOWS\system32\cmd.exe
Introduce tu antepasado mas lejano: JuanCarlos
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Elena
Escriba el nombre de su padre: JuanCarlos
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 2
JuanCarlos
  Elena
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Cristina
Escriba el nombre de su padre: JuanCarlos
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Felipe
Escriba el nombre de su padre: JuanCarlos
  
```

```

0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 2
JuanCarlos
    Elena
    Cristina
    Felipe
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: FelipeFroilan
Escriba el nombre de su padre: Elena
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Victoria
Escriba el nombre de su padre: Elena
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 2
JuanCarlos
    Elena
        FelipeFroilan
        Victoria
    Cristina
    Felipe
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: JuanU
Escriba el nombre de su padre: Cristina
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Pablo
Escriba el nombre de su padre: Cristina
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Miguel
Escriba el nombre de su padre: Cristina
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Irene
Escriba el nombre de su padre: Cristina
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 2
JuanCarlos
    Elena
        FelipeFroilan
        Victoria
    Cristina
        JuanU
        Pablo
        Miguel
        Irene
    Felipe
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Leonor
Escriba el nombre de su padre: Felipe
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 1
Escriba el nuevo nombre: Sofia
Escriba el nombre de su padre: Felipe

```

```
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 2
JuanCarlos
    Elena
        FelipeFroilan
        Victoria
    Cristina
        JuanU
        Pablo
        Miguel
        Irene
    Felipe
        Leonor
        Sofia
0. Salir
1. Insertar nuevo nombre
2. Mostrar arbol genealogico
Introduce opcion: 0
Presione una tecla para continuar . . .
```