



Minutas del proyecto

Estructuras de datos 2022 (primer cuatrimestre)

Lógica:

- Cualquier método que modifique los hijos de un nodo, modificara su orden. Por ejemplo, el método “removeNode(...)” al remover un nodo cuyo padre tenía grado n , luego de removerlo el grado del padre será de $n-1$.
- La clase Lógica usa un atributo llamado “mainTree” de tipo `Tree<Pair<Character, Integer>>`. De este modo la clase “GUI” nunca accederá directamente al “TDA Árbol”, sino que lo hará a partir de los métodos que la clase “lógica” le facilite.
- Las excepciones lanzadas por algunos métodos, son de utilidad para hacer un “try-catch” en la clase “GUI” y poder imprimir a partir de las excepciones en la interfaz gráfica lo que ocurrió.
- Los únicos casos de que se pueda eliminar la raíz, es que esta sea el único elemento o que tenga un solo hijo, cumpliendo así los requisitos de la estructura de árbol.
- Todos los métodos incluso los privados, se encuentran comentados para una lectura y comprensión más ágil.

Gui:

- Los rótulos de los nodos pueden ser caracteres, es decir, que podrían ser un numero entero, así como un espacio, un signo o cualquier carácter.
- En la pantalla derecha de la interfaz gráfica se muestra una representación en tiempo real del árbol.
- Cuando se insertan más nodos de los que entran en la pantalla de salida, se habilitara una barra lateral de scroll vertical.
- Para ingresar un nuevo nodo es obligatorio ingresar su padre y el rotulo. Si por alguna razón alguna de estas dos condiciones no se cumple, el nodo no podrá ser agregado.
- Si se inserta un nodo que no existe en la casilla de eliminar nodo, aparecerá la advertencia de que no se encontró el nodo y por ende no podrá ser eliminado.
- La casilla de eliminar nodos K , solo podrá recibir un numero entero. Si no existe ningún nodo con el grado ingresado, no se eliminará nada y aparecerá una advertencia.
- En el código de la “GUI”, hay secciones comentadas para una lectura más ágil y accesible.

Consideraciones generales:

- Se asume que no se ingresaran nodos con rótulos repetidos.
- Al igual que java, nuestro programa es sensible a mayúsculas y minúsculas. Por ejemplo, “m” es distinto de “M”.
- Se confecciono un manejo de errores de manera muy abarcativo. Por ejemplo, si se inserta un grado que no existe se le avisara al usuario, si se ingresa un padre igual que el nodo a insertar también se notificara.
- A algunas clases se les añadió el “@SupressWarnings ()”, opción brindada por el entorno de eclipse para evitar que aparezcan advertencias.
- En una de las computadoras donde fue probado el programa, el archivo “.jar” no arranca al igual que otros programas con la misma extensión desarrollados por otras personas. Esto se da solamente en un dispositivo solo, por un problema de compatibilidad de java. **En los demás dispositivos se ejecutó de manera correcta.**