

Implementa lo que se pide respetando estrictamente los nombres de los métodos y de las clases. Cada archivo deberá tener un encabezado con tus datos. (Se sancionará con 5 pts menos cada omisión. NO deberás agregar atributos, pero puedes utilizar todas las variables locales que necesites, tampoco es recomendable que agregues métodos de más. Respeta las convenciones vistas en clase e indenta correctamente tu código (Se pierden 5 pts si no se hace). Siempre que puedas reutiliza código. **Cualquier abuso de mal desempeño en la implementación se sancionará hasta llegar a no tomar en cuenta la respuesta.** En caso de utilizar impresiones en consola para probar tu programa es muy importante que los borres o comentarías antes de entregar tu programa. Sólo valida aquello que explícitamente se pida, entonces sino se pide no es necesario validar. Al finalizar debes entregar tu programa en tu carpeta de drive que se te compartió.

Durante el examen está estrictamente prohibido comunicarse con cualquier persona por cualquier medio, sólo está permitido comunicarte conmigo. No debes consultar ningún material más que los archivos que te he proporcionado en el drive y las ligas que se mencionan al final. En todo momento debes grabar localmente la sesión de zoom y debes ir al breakroom asignado. Una vez que entres debes comenzar a compartir la pantalla. Durante todo el examen debes permanecer grabando y compartiendo la pantalla. En la misma computadora en la que realizas el examen debes tener la sesión zoom. Ningún otro dispositivo electrónico debes tener a la mano (Computadora, Tablet, etc). Al terminar tu programa sube los archivos java (extensión .java) y deja que zoom genere el archivo .mp4, ya que se generó el o los archivos .mp4 cárgalos a la misma carpeta donde subiste tu examen.

Escribe la clase MyABBFinal, la cual esta implementada con genéricos y hereda de la clase MyABBFinalPadre, la cual también está parametrizada. La clase MyABBFinal sólo agrega el implementar la interfaz Iterable<E>. El Iterator que regresará el método iterator(), sólo implementará los métodos hasNext() y el método next(), es decir no implementará el método remove(). El iterador debe iterar el árbol de manera que el recorrido que se haga sea un recorrido por nivel y la complejidad debe ser lineal. Como recordatorio el algoritmo de recorrido por nivel es el siguiente.

Se utiliza una cola para este proceso

- Insertar el nodo raíz en una fila.
- Mientras la fila no se vacíe:
 - Sacar un elemento de la fila y procesar el nodo.
 - Insertar en la fila los hijos del nodo procesado (si existen)

Te sugiero utilizar la LinkedList que trae java implementada como cola. Esto se hace de la siguiente manera.

```
Queue<TIPODEDATO> cola=new LinkedList<>();
```

Además si vas a implementar el iterador como una clase anónima (lo cual te lo sugiero) y necesitas hacer algun paso

inicial que regularmente va en el constructor, lo puedes hacer simplemente poniendo esas acciones entre una llave que abre y una que cierra o sea { instrucciones }.

La interfaz Queue, tiene los métodos peek(), que te regresa el siguiente elemento en salir pero sin borrarlo y regresa null si la cola estaba vacía. También tiene el método poll(), que saca un elemento de la cola y lo regresa, si la cola estaba vacía regresa null. Y también tiene el método add(E), que guarda el elemento en la cola.