17 - REVIEW Java Iterators

Java Iterators

Iterators are supported in the Java Collection Framework: interface `Iterator<T>`.
They exploit Java Generics (as collections do).

Iterators are usually defined as nested classes (non-static private member classes): each iterator instance is associated with an instance of the collection class.

Usually we have a nested private class in the collection that implements `Iterator<T>` (implements the method defined in the interface: `hasNext()`, `next()`) and the collection implements `Iterable<T>`, meaning it exposes the method `iterator()` that retrieve an iterator (an instance of the nested private class)

example:

```language-java
public class BinTree<T> implements Iterable<T>{

        //private nested class
        private class TreeIterator implements Iterator<T>{
                private Stack<BinTree<T>> stack = new Stack<>();

                private TreeIterator(BinTree<T> n){
                        if(n.val != null)
                                stack.push(n);
                }
                public boolean hasNext(){
                        return !stack.empty();
                }

                // preorder traversal
                public T next(){
                        if(this.hasNext())
                                throw new NoSuchElementException();

                        BinTree<T> n = stack.pop();

                        if(n.right != null)
                                stack.push(n.right);
                        if(n.left != null)
                                stack.push(n.left);

                        return n.val
                }

                public void remove(){
                        throw new UnsupportedOperationException();
                }

        }

        BinTree<T> left;
        BinTree<T> right;
        T val;
        // ...
```

```
        // tree methods: insert, lookup, ...

        public Iterator<T> iterator(){
                return new TreeIterator(this);
        }
}
```

Java enanched `for` "for each" which exploits iterators

```language-java
for(Integer i : myBinTree)
        System.out.println(i);
```