Iterators

Iterators provide:

1. an alterative to for-loops to go through an entire Collection class linear data structure.
2. a way to safely add and remove elements while iterating through a collection.

Iterator Methods

The iterator includes three methods that help us traverse a collection:

- **hasNext()**: returns true if the iterator has at least one more element to iterate through.
- **next()**: returns the next element in the collection as long as *hasNext()* is true and throws a ***NoSuchElementException*** if *hasNext()* is false.
- **remove()**: removes the current element in the collection and throws an ***IllegalStateException*** if the method is called again after it's already removed the current element.

ListIterator

The *ListIterator<>* is an Iterator which provides additional functionality when working with *List<>* classes. It allows you to traverse the list forward or backward and the ability to add and update elements in the list, in addition to the other iterator methods.

The *ListIterator<>* includes the following methods to help us traverse a list:

- **hasNext()**: returns true if the iterator has at least one more element to iterate through.
- **next()**: returns the next element in the collection as long as *hasNext()* is true and throws a ***NoSuchElementException*** if *hasNext()* is false.
- **remove()**: returns the next element in the collection as long as *hasNext()* is true and throws a ***NoSuchElementException*** if *hasNext()* is false.
- **add(element)**: inserts an object immediately before the current cursor position. The element must be the type that the iterator has said it holds.
- **set(element)**: replaces (updates) the last element returned by next() and throws an IllegalStateException if the method is called before an element has been returned by next().
- **hasPrevious()**: returns true if the iterator has at least one more previous element to iterate through.
- **previous()**: returns the next element in the collection as long as *hasPrevious()* is true and throws a ***NoSuchElementException*** if *hasPrecious()* is false.