16.12 Unmodifiable Collections

The `Collections` class provides a set of `static` methods that create **unmodifiable wrappers** for collections. Unmodifiable wrappers throw `UnsupportedOperationException`s if attempts are made to modify the collection. In an unmodifiable collection, the references stored in the collection are not modifiable, but the objects they refer *are modifiable* unless they belong to an immutable class like `String`. Headers for some of these methods are listed in Fig. 16.19. Details about these methods are available at `http://docs.oracle.com/javase/8/docs/api/java/util/C`
All these methods take a generic type and return an unmodifiable view of the generic type. For example, the following code creates an unmodifiable `List` (`list2`) that stores `String` objects:

```
List<String> list1 = new ArrayList<>();
List<String> list2 = Collections.unmodifiableList(lis
```

Software Engineering

Observation 16.5

You can use an unmodifiable wrapper to create a collection that offers read-only access to others, while allowing read–write access to yourself. You do this simply by giving others a reference to the unmodifiable wrapper while retaining for yourself a reference to the original collection.

`public static` **method headers**
`<T> Collection<T> unmodifiableCollection(Collection<T> c)`
`<T> List<T> unmodifiableList(List<T> aList)`
`<T> Set<T> unmodifiableSet(Set<T> s)`
`<T> SortedSet<T> unmodifiableSortedSet(SortedSet<T> s)`
`<K, V> Map<K, V> unmodifiableMap(Map<K, V> m)`
`<K, V> SortedMap<K, V> unmodifiableSortedMap(SortedMap<K, V> m)`

Fig. 16.19

Some unmodifiable wrapper methods.