## 12.6 Exercises

## **Factual**

- 1. What methods are required for a class that is Iterable?
- 2. Which of the following is true about the java.util.Se t and the java.util.List interfaces?
  - If we add String[][] objects to a Set and a List, the size of the set will always be less than or equal to the size of the list.
  - The java.util.ArrayList class is an implementation of the java.util.List interface.
  - ☑ The Set and List interfaces extend the Iterator interface.
  - ✓ The Set and List interfaces extend the Iterable interface.
- 3. Suppose we have a class that implements <a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Iterator</a><a>Itera
  - Problem 1

An Iterable is required to have the iterator() method, which returns an Iterator.

- Problem 2
  - If we add String[][] objects to a Set and a List, the size of the set will always be less than or equal to the size of the list. Sets only have unique items, while lists can have duplicates, so if we add the same elements to both the list will always have at least as many elements as the set.
  - The java.util.ArrayList class is an implementation of the java.util.List interface. One implementation of the List interface in

The Set and List interfaces extend the Iterable interface. Sets and Lists in Java can be used in enhanced for loops, which means that they are Iterable.

Problem 3

An Iterator must override hasNext(), which returns a boolean indicating whether there are more elements in the Iterator, and next(), which returns the next item.

## Conceptual

- 1. Why do we want to override the .equals method?
  - Problem 1

The .equals() method inherited from Object only checks if two items have the same memory address. This is undesireable behavior for many user-written classes in Java.

## Metacognitive

1. In lecture, you built the ArraySetIterator class. Modify the lecture class to take in a Comparator<T> and an item of generic type T called ref in the constructor. This new iterator should only return items greater than T. For reference, the code for ArraySetIterator is included below.

```
private class ArraySetIterator implements Iterator<T> {
    private int pos;

public ArraySetIterator() {
        pos = 0;
    }

public boolean hasNext() {
        return pos < size;
    }

public T next() {
        T returnItem = items[pos];
        pos += 1;
        return returnItem;
    }
}</pre>
```

2. Problem 7 from the Spring 2018 Midterm 2

```
Problem 1
```

```
public class ArraySetGreaterIterator implements Iterator<T> {
    private int pos;
    private T ref;
    private Comparator<T> comp;
    public ArraySetGreaterIterator(T ref, Comparator<T> comp) {
        this.ref = ref;
        this.comp = comp;
    7
    @Override
    public boolean hasNext() {
        return pos < size;</pre>
    3
    @Override
    public T next() {
        T returnItem = items[pos];
        while (comp.compare(returnItem, ref) <= 0) {</pre>
            pos += 1;
            returnItem = items[pos];
        return returnItem;
    3
}
```

Problem 2

Solutions and walkthrough are linked here and on the course website.

Previous 12.5 Chapter Summary

Next 13. Asymptotics I

Last updated 1 year ago

