Programming on Purpose (CSC 120)
Winter 2024

Lab 9 – Iterables and Iterators

3/7/2024

Objectives

- Practice with interfaces.
- Practice with ADTs.

Preliminary Setup

1. Create a folder for this lab.
2. Download the starter code and save it in the new folder.

You are going to write code in `MyArrayList.java` and in `MyArrayListIterator.java`. The file `ArrayListSandbox.java` has some code to help you test your implementation.

Notice: There are questions for you to answer spread throughout these lab instructions. Please make sure to answer all of these questions as you are working through the lab. If you are not sure about your answer, ask your neighbor or me, and make sure that you know and understand the answer before moving on.

1 Iterables and Iterators

When a collection is *iterable* that means that we can use a loop to *iterate* over the elements of the collection. To provide this functionality, there needs to be a second kind of object, called an *iterator*, that knows how to go through the collection one by one and return the collection's elements one by one.

Java provides two Interface called `Iterable` (`java.lang.Iterable`) and `Iterator` (`java.util.iterator`) that capture the behavior of iterables and iterators. Find the documentation for these two interfaces.

Question: What are the methods that a class implementing each of these interface *has* to implement? (Hint: When a method is labeled as *default* in the Java API for an interface that means that Java will provide a default implementation for this method. That is, a class implementing the interface may but does not have to provide an implementation. For this lab, you can ignore all methods labeled as default.

Iterables and iterators work together as follows: The iterable collection has a method that creates, initializes and returns an iterator object that is *specific* to this collection.

This iterator object has methods for asking for the "next" element in the collection and for checking whether there are any "next" elements left. E.g. given a collection of integers `{5, 2, 7}`, an iterator should first return 5, when the `next` method is called the first time. Then, when the `next` method is called again, it should return 2, etc.

Question: Continuing with the same example, what would `next` return if it was called for a third time?

Question: What would `hasNext` return if we called it after that?

2 Implement an Iterator for MyArrayList and the Iterable Interface

The starter code for `MyArrayListIterator` already has stubs for all methods you have to implement.

Question: What information does a `MyArrayListIterator` need to know about the `MyArrayList` collection it is iterating over?

Hint: Those bits of information need to be passed to the `MyArrayListIterator` constructor. Because iterators work so closely in tandem with the collections they iterate over, this is a case where you are allowed to disobey some information hiding principles and pass references to instance variables of the `MyArrayList` collection to the `MyArrayListIterator` object.

Question: What additional information does the iterator need to keep track of (besides the information it gets from the collection) in order to implement the behavior of the `next` and `hasNext` methods?

Once you have planned your implementation by answering the previous two questions, fill in the code for the constructor, `hasNext`, and `next` methods of the `MyArrayListIterator` class as well as the `iterator` method of the `MyArrayList` class.

Test your code using `ArrayListSandbox.java`.

3 Revise your code

Before turning in any program in this class, remember this mantra:

> Just because it works doesn't mean it's good.

Part of your grade will also come from things like how understandable and readable your code is. You will also be graded on the neatness, presentation, and style of your program code.

Make sure that all modules and functions are documented (even those that you didn't write).

Don't forget to cite who you received help from and include the honor code affirmation in the javadocs of each class that you wrote or modified:

```
I affirm that I have carried out my academic endeavors
with full academic honesty. [Your Name]
```

4 How to submit

Submit the project by uploading `MyArrayList.java` and `MyArrayListIterator.java` to Gradescope.

Make sure to add your partner to the submission so that you both have access to it through Gradescope.