

The main task in this recitation is for you to fix a version of A3, doubly linked lists, so that it implements interfaces Iterator and Iterable. But first a few problems for you to do.

1. Suppose you had to explain to a friend the difference between interfaces Iterator and Iterable. What would you say?

A. Write down in a sentence or two what interface Iterator is for.

B. Write down in a sentence or two what interface Iterable is for.

2. Below is some code that is used to enumerate and print the Strings in some collection. To the right, write down what is wrong with this code:

```
Iterator<String> it= ...;
...
while (it.next()) {
    System.out.println(it.next())
}
```

3. Below is some code that is used to enumerate and print the Strings in some collection. To the right, write down a requirement on the collection for this to actually work and also state when it doesn't work.

```
Iterator<String> it= ...;
...
while (it.hasNext()) {
    System.out.println(it.next());
    System.out.println(it.next());
}
```

4. This problem concerns implementing Iterator and Iterable in your solution to A3, class DLL. Do the following steps to prepare for this—you can use either your solution to A3, if you *know* it is correct, or our solution, which will be on the course website, page lectureNotes, on the row for recitation 07.

- A. Start a new Eclipse project, call it something like a3Iterable.
- B. Fix the project:
 - a. To use YOUR DLL.java: Copy both of your files DLL.java and DLLTest.java from your old a3 project to the new one. You can do this by selecting the old files, doing a copy, selecting the new project's src directory, and doing a paste. Then copy the methods at the end of this page into class DLLTest.
 - b. To use OUR classes: Get our files (see above) and copy them into the project's src directory.
- C. If JUnit4 is not available in the new project, Insert a new JUnit test-case class in the usual way and then delete it.
- D. Look at the code at the bottom of class DLLTest. Look at how it tests interfaces Iterator and Iterable. Of course, there are errors because Iterator and Iterable have not yet been implemented by class DLL.
- E. Write inner class DLLIterator with this specification and header:

```
/** An instance is an iterator over the elements of this list. */  
private class DLLIterator implements Iterator
```
- F. Write method iterator with this specification and header:

```
/** Return an Iterator over the elements of this list. */  
public Iterator iterator()
```
- G. Change the class of declaration of DLL to implement Iterable<E>.
- H. If you did the above correctly, both classes should have no syntax errors. And if you Run class DLLTest and made no mistakes in the above, the test should run without errors.