

Assignment 2

Due: February 7th, 2013 before 6:30pm

Objectives

- Practice implementing an interface in Java
- Practice with a Node class and references
- Practice reading and understanding specification
- Exposure to more complete testing

Introduction

This assignment introduces the idea of a Container – a data structure that is used to store a collection of items. For simplicity, our containers will only store integers. Later in the course we will encounter two different mechanisms that Java supports to store items of an arbitrary type (inheritance and generics).

Your task is to implement the interface described in the file `IntegerList.java` using a doubly-linked list in the file `IntegerLinkedList.java`.

Quick Start:

1. Download `a2tester.java`, `IntegerLinkedList.java`, `IntegerList.java`, `IntegerNode.java` and `IntegerArrayList.java`
2. Read `IntegerList.java` carefully
3. Compile and run the test program `a2tester.java`
4. If no errors reported by test program, see the Grading section of this document
5. Implement one of the methods in `IntegerLinkedList.java`
6. goto 3

Understanding the test program: a2tester.java

You've been given a program that will test your implementation of the IntegerList interface.

One of the first things you should do after downloading the source code files is to run the test program.

Compile the test program by typing:

```
javac a2tester.java
```

Run the test program by typing:

```
java a2tester
```

You should see the following output:

```
Basic testing of size, addFront, addBack, get  
Failed test: 0 at line 53
```

The tester is reporting that your implementation is failing the very first test. This is hardly surprising, since you haven't written any code yet!

The tester is written to be able to test both an array implementation (provided to you in the file `IntegerArrayList.java`) and the linked list implementation that you are required to implement in `IntegerLinkedList.java`

To see the result of testing your instructor's array based solution, type:

```
java a2tester array
```

You will see there are 23 test cases (numbered 0 to 22), and the array based solution passes them all.

Your goal is to make your linked list based implementation pass all 23 test cases.

Bonus:

Jason will give out a chocolate bar for each new bug¹ that is found in `IntegerArrayList.java`. This bug finding isn't part of the assignment – try it only if you are interested.

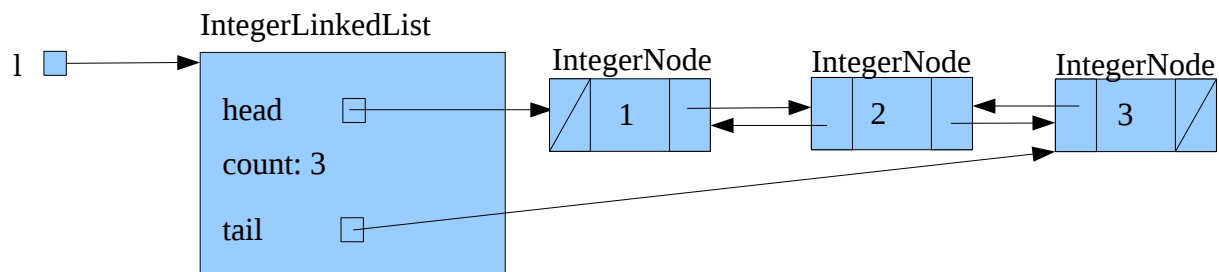
¹ There are currently no known bugs. The first person to post the test case and description of the bug to the assignment 2 connex forum gets the chocolate bar.

Linked Lists

Your implementation must be a doubly-linked list that maintains both a `head` and a `tail` reference. The stub code you've been provided in `IntegerLinkedList.java` already includes the appropriate instance variables.

You've also been provided with a node class (`IntegerNode.java`) that includes a `prev` and `next` references.

If your implementation is correct, the in memory picture of the list `l` containing `{1,2,3}` will be:



Your implementation of `addFront` and `addBack` must be $O(1)$.

What to do

All the coding you have to do in this assignment is in the file `IntegerLinkedList.java`. This code already compiles and runs, it just doesn't do anything useful yet.

You should be sure that you understand the idea of a linked list and the purpose of the node class before you start writing code. There will be examples and code regarding linked lists in the labs and lectures this week to help your understanding.

It is very helpful to draw pictures of what a list looks like before and after an operation (such as `addFront`) to help you understand what the code you are writing is trying to accomplish.

You should do your development incrementally. Implement one method and then re-run the test program.

In order to pass the first set of test cases, you will have to implement the methods: `size`, `addFront`, `addBack` and `get`.

Once you've completed those methods, move on to `toString` and `clear`.

Finally, implement `remove`.

Submission

Submit your `IntegerLinkedList.java` using Connex. **Please be sure you submit your assignment, not just save a draft.**

A reminder that it is OK to talk about your assignment with your classmates, and you are encouraged to design solutions together, but each student must implement their own solution.

We will be using plagiarism detection software on your assignment submissions.

Grading

If you submit something that does not compile, you will receive a grade of 0 for the assignment. It is your responsibility to make sure you submit the correct files.

Requirement	Marks
You submit something that compiles	1
Your code passes the first set of test cases: “Basic testing of size, addFront, addBack, get”	3
Your code passes the second set of test cases: “Testing toString() and clear”	2
Your code passes all the remaining test cases	6
Total	12

To be clear about the above guidelines – you cannot score more than 50% on this assignment unless your code passes ALL the test cases.