Linked List Homework

In the java examples we’ve shown so far, we’ve stored primitive variables of type int in our data structures. Storing such variables simplifies the program examples, but it’s not representative of how you use data storage structures in the real world. Usually, the data items (records) you want to store are combinations of many fields. For a personnel record, you would store last name, first name, age, Social Security number, and so forth. For a stamp collection, you would store the name of the country that issued the stamp, its catalog number, condition, current value, and so on. In your homework, you need to show how objects, rather than variables of primitive types, can be stored. This time you need to use a linked list.

The provided classes are: a reference data type , and the Link class for singly linked list and doubly linked list. Notice that the Link class does not have a int or double as its data element. Instead, now it contains a reference variable field.

**Part A. Singly Linked List.** Start by creating a singly linked list to use as your data structure for this task. In addition to the first(head)field that we practiced with in class, you will also maintain a last(tail)field. Include the below standard linked list features/methods in your class

* Constructor
* isEmpty()
* insertFirst()
* insertLast()
* deleteFirst()
* displayList()

For each method, be sure to test your work by adding test code to a main method in a SingleListedListTestApp class.

**Part B. Doubly Linked List.** Create a second program that achieves the same goal as your first one, but uses a doubly-linked list rather than a singly-linked list. You will need to first create a Link class, that will have an extra field (called previous) for the backward reference. Link is the class you will use to build your doubly linked list.

For this program, after creating the fields and the constructor method, you need to add the following methods:

* constructor
* isEmpty()
* insertFirst()
* insertLast()
* deleteFirst()
* deleteLast()
* insertAfter()
* deleteKey()
* displayForward()
* displayBackward()

***Warning***: you might be tempted to try copying and pasting from the methods you wrote above, then modifying the code. However, these methods in a doubly-linked list class are *significantly* different, and more streamlined. You should write them from scratch rather than confuse things by trying to reuse old code.

For each method, be sure to test your work by adding test code to a main method in a doublyListedListTestApp class.

**Submission- you** need to submit the following java files

1. Reference data type
	1. Data member
	2. Constructor(s)
	3. getMethod(s)
	4. setMethod(s)
2. SinglyLinked.java
	1. Link class for singly linked list
	2. Singly linked list class
	3. Main method to test singly linked list class
3. DoublyLined.java
	1. Link class for doubly linked list
	2. Doubly linked list class
	3. Main method to test doubly linked list class