Programming Fundamentals
Programming Assignment 4 - Set ADT

For this assignment, you are to write a program which implements a *Set* data structure using a singly linked list implementation and a driver program that will test this implementation.

The Set ADT is a linear collection of data that will support the following operations:

1. add(x) – adds the integer *x* to the collection. The resulting collection should not contain any duplicate values.

2. delete(x) – deletes the integer *x* from the set.

3. exists(x) – returns true if the integer *x* exists in the set and false otherwise.

4. toString() – returns a string representing the set as a space separated list.

The time complexity of all operations should be $O(N)$, where N is the size of the collection.

Program Design Requirements

1. Your program should consist of 3 classes: *LinkedNode*, *Set*, and *Test*.

2. The *Test* class is the **driver** class for your program (should contain only the main method).
   Inside the main method, you should create a new *Set* object, then enter an infinite loop in which the user is prompted to enter one of 3 commands:
   ```
   add x
   del x
   exists x
   ```

   `x` stands for an actual number (some integer).

   The `add` and `delete` operations should perform the necessary operation (inserting or deleting the given integer) and then display the contents of the set using the *toString()* method. The `exists` command should output `true` or `false` depending on whether *x* exists in the set.

3. The *Set* class should implement the Set ADT by adding the appropriate variables and methods. There should be 1 method for each of the required operations.

4. The *LinkedNode* class should contain an integer value and a pointer to the next LinkedNode.

5. Make sure to validate all inputs. This means re-prompting the user if he or she entered something invalid.

Additional Requirements

1. Your code should follow good coding practices, including good use of whitespace (indents and line breaks) and use of both inline and block comments.
2. You need to use meaningful identifier names that conform to standard Java naming conventions.
3. At the top of **each file**, you need to put a block comment with the following information: your name, course name, semester, and assignment name.
4. **The output of your program should match exactly the sample program output given at the end.**

What to Turn In

1. You will turn in a **single .zip** file using BlackBoard. The name of this file should be in this format:
   P4-<Name>.zip
   where <Name> is your actual name.
2. Inside of the .zip file, there should be 3 .java files, each corresponding to one of the 3 classes that you will define. Make sure each .java file is named the same as the class (Set.java, LinkedNode.java, Test.java)

What You Need to Know for This Assignment

- How to parse input and split it into an array
- How to use linked lists to implement data structures

HINTS

- To parse the command, you will need to first use the `nextLine()` method of `Scanner`, then you can split it into 2 by using the `split()` method of the `String`. This method will return a `String` array that will contain both parts of the command (index 0 will contain the command itself, and 1 will contain the integer x).
- You can use the `Integer.parseInt()` method to extract the actual `int` value from the `String`.
- The `LinkedNode` class was one of the examples shown during the lecture so you can use it.
- Make sure you follow standard programming conventions for use of comments, identifier names, whitespace, etc. These are easy points that everyone should get.

Sample Program Output

```
Programming Fundamentals
NAME: [put your name here]
PROGRAMMING ASSIGNMENT 4 - SET

Enter command: add 5
5
Enter command: add 8
8 5
Enter command: add 6
6 8 5
Enter command: add 24
24 6 8 5
Enter command: add 6
24 6 8 5
Enter command: exists 6
true
Enter command: exists 13
false
Enter command: exists 24
true
Enter command: del 6
24 8 5
Enter command: del 12
24 8 5
Enter command: add 8
24 8 5
Enter command: del 8
24 5
…
```