Laboratory Exercise 4

Implementation of ADTs - linked lists

[1] *See p52 7.2.3. Vol.2 subject guide in your first year Programming class*

1. Review the concept of a *constructor*[1] in Java.

   For example, what is a constructor?
   How can we use a constructor to define a new type?
   What do we have to take care of when writing a constructor method, in terms of its name and of polymorphism?

2. Analyse the following class. What do the constructors do?

```
public class Node {
  private String item;
  private Node next;

  public Node(String newItem) {
    item = newItem;
    next = null;
  } // Constructor

  public Node(String newItem, Node nextNode) {
    item = newItem;
    next = nextNode;
  } // Constructor
}
```

3. Using constructor approach, construct a new class of data structure Node (for a linked list) which contains two fields of different type:

   - item: of Object type (this means it could be of anything)
   - next: of Node type

4. Analyse the statements below. What do they do in each case?

   (a) `Node n = new Node("pen");`

   (b) `Node n = new Node("pencil", null);`

   (c)
```
Node n = new Node("pen", null);
n = new Node("pencil", null);
```

   (d)
```
Node n = new Node("an apple?");
n = new Node("ate", n);
n = new Node("cat", n);
```

(e)
```
public class Node {
  private String item;
  private Node next;

  public Node(String newItem) {
    item = newItem;
    next = null;
  } // Constructor

  public Node(String newItem, Node nextNode) {
    item = newItem;
    next = nextNode;
  } // Constructor
} // end Node

public class list {
  public static void main(String [] args) {
    Node n = new Node("an apple?");
    n = new Node("ate", n);
    n = new Node("cat", n);
  } // end main
} // end list
```

5. Following the above, write and add four methods which allow the access to the value of each field into the class Node. For example, you may write following four methods and include them into the class:

   - setItem(Object newItem): to set the value of item field, i.e. to write the newItem in the item field of a node.
   - getItem(): to read the value of item field.
   - setNext(Node nextNode): to set the value of next field, i.e. to write the nextNode in the next field of a node.
   - getNext(): to read the value of next field.

6. Write a method that displays all the items in a list.
   Hint: the method takes the head of a list as the input and display the item field of each node in the list.

7. Write a method that constructs an empty list (of Objects).
   Hint: the method returns a null.

8. Write a method that checks if a list is empty.
   Hint: the method takes the head of a list and returns a *true* if the head has a null value.

9. Write a *recursive* method to count the number of nodes in a list.
   Hint: the method takes the head of a list and returns the number of the nodes in the list.

10. (Optinal - for advanced students) Using the approaches in this lab exercise, implement the ADT *appointmentBook*.
    Hint: use the interfaces for ADT *appointmentBook* introduced in the lecture.