

ITEC 324

Programming Assignment #4

- Specification for class **AddressList**
 - Implement class **AddressList** as specified below. In your implementation, all methods except **addToFront** and **isEmpty** must be recursive (or must use a recursive helper method).

AddressList
- head: ListNode - currentNode: ListNode
+ isEmpty(): Boolean + addToFront(String name, String tel, String email, String address, String dob): void + addToBack(String name, String tel, String email, String address, String dob): void + toString(): String + reverseToString(): String + reverse(): AddressList + sizeOf(): int + phoneNumberByName(String name): String + emailByName(String name): String + nameByPhoneNumber(String tel): String + dobByName(String name): String

- In the methods **phoneNumberByName**, **emailByName**, **nameByPhoneNumber**, and **dobByName**:
 - if there is no matching data, it should return "**No matching data**"
 - duplicate data in the list will not be considered. If there is more than one matching item, the first one is to be returned.
 - Please be certain that you follow this specification exactly so that my test program will work. I suggest that you write your own test driver to test your methods.
 - If necessary, you can add private methods and/or fields to the class.
 - You may **NOT** add public fields or methods to the class.
 - For **reverseToString** do not simply call **reverse** and **toString**.
 - Do not use the LinkedList class provided in java.util.* package in JDK
 - You should write a class **ListNode** which represents an element of the single linked list. You may or may not use all or part of the following code when you write this class.

```
public class AddressList
{
    /**
     * This nested, private class represents a node of a singly linked list.
     */
    private class ListNode
    {
        private String name;
        private String tel;           // Telephone number
        private String email;
        private String addr;         // Address
        private String dob;          // Date of birth
        private ListNode next;       // Pointer to the next node

        private ListNode(String name, String tel, String email, String addr, String dob)
        {
            this.name = name;
        }
    }
}
```

```

        this.tel = tel;
        this.email = email;
        this.addr = addr;
        this.dob = dob;
    } // end of the constructor

    public String getName() { return name; }
    public String getTel() { return tel; }
    public String getEmail() { return email; }
    public String getAddr() { return addr; }
    public String getDob() {return dob; }

    public void setName(String name) { this.name = name; }
    public void setTel(String tel) { this.tel = tel; }
    public void setEmail(String email) { this.email = email; }
    public void setAddr(String addr) { this.addr = addr; }
    public void setDob(String dob) { this.dob = dob; }

    public ListNode getNext() { return next; }
    public void setNext(ListNode link) { next = link; }
} // end of class ListNode

// =====
// Include your code here
// =====
} // end of class AddressList

```

- You are to write a graphical user interface called **ListGUI** that will allow a user to execute all of the methods in the class diagram above. The interface should have textfields for entering all required data and the output of each command should be displayed in a textfield or textarea with scrollbars.

You should submit your programming assignment 4 to D2L.

Since the UML will be very simple, you don't need to submit UML for this assignment.