CS 1083

Assignment #10

Author: Yulong Wang

Id: 3713596

1. Source Code:

a. ClassList.java

```
import java.util.NoSuchElementException;

/**
 * Represent a doubly-linked list made up of students
 * @author Yulong Wang
 * @date 2021/11/30
 */
public class ClassList {
    /**
     * The last node in the list
     */
    private StudentNode end;
    /**
     * The first node in the list
     */
    private StudentNode front;
    /**
     * the current size of the list
     */
    private int size;

    /**
     * Contructor
     */
    public ClassList() {
    }

    /**
     * Add a new node representing a specified student at the appropriate position in
the list.
     * @param studentIn Student to be added
     */
    public void add(Student studentIn){
        size +=1;
        if(front == null){
            StudentNode res = new StudentNode(studentIn);
            front = res;
            end = res;
            return;
        }
        StudentNode temp = front;
        while(temp!= null){
            if(temp.data.compareTo(studentIn) > 0){
                if(temp == front){
                    StudentNode res = new StudentNode(studentIn);
                    front.prev = res;
```

```
                res.next = front;
                front = res;
                return;
            }else{
                temp = temp.prev;
            }
            break;
        }
        temp = temp.next;
    };
    if(temp == front && temp.data.compareTo(studentIn) > 0){
        StudentNode res = new StudentNode(studentIn);
        front.prev = res;
        res.next = front;
        front = res;
        return;
    }
    if(temp!=null){
        StudentNode res = new StudentNode(studentIn);
        temp.next.prev = res;
        res.next = temp.next;
        temp.next = res;
        res.prev = temp;
    }else{
        StudentNode res = new StudentNode(studentIn);
        end.next = res;
        res.prev = end;
        end = res;
    }
}

/**
 * return the number of student in the list
 * @return int the number of students
 */
public int getNumStudents(){
    return size;
}

/**
 * creates and return an array containing all of the students in this list,
 * stored in reverse order in the array
 * @return {@link Student[]}
 */
public Student[] getReversedList(){
    Student[] reversedList = new Student[size];
    StudentNode temp = end;
    for(int i=0;i<size;i++){
        reversedList[i] = temp.data;
        temp = temp.prev;
    }
    return reversedList;
```

```
    }

    /**
     * remove from the list the node containing the specified student
     * @param studentOut the student to be removed
     * @throws NoSuchElementException
     */
    public void remove(Student studentOut) throws NoSuchElementException {
        StudentNode temp = front;
        while(temp != null){
            if(temp.data.compareTo(studentOut) != 0){
                temp = temp.next;
            }else {
                if(temp.prev!= null){
                    temp.prev.next = temp.next;
                    if(temp.next != null){
                        temp.next.prev = temp.prev;
                    }else{
                        end = temp.prev;
                    }
                }else{
                    if(temp.next != null){
                        front.next.prev = null;
                        front = front.next;
                    }else{
                        front = null;
                        end = null;
                    }
                }
                size-=1;
                return;
            }
        }
        throw new NoSuchElementException("Student not found");
    }

    /**
     * return a string of the content of the list in order
     * @return {@link String}
     */
    @Override
    public String toString() {
        StudentNode temp = front;
        String res = "";
        while(temp!= null){
            res += temp.data.toString() + "\n";
            temp = temp.next;
        }
        return res;
    }

    /**
```

```
     * An inner class that represents a node in the class
     * @author Yulong Wang
     * @date 2021/11/30
     */
    private class StudentNode{
        /**
         * the student refrenced by this node
         */
        private Student data;
        /**
         * next node
         */
        private StudentNode next;
        /**
         * previous node
         */
        private StudentNode prev;

        /**
         * constructor
         * @param dataIn
         */
        public StudentNode(Student dataIn) {
            this.data = dataIn;
            this.next = null;
            this.prev = null;
        }
    }
}
```

TestDriver.java

```
import java.util.NoSuchElementException;

/**
 * test driver for the ClassList class
 * @author Yulong Wang
 * @date 2021/11/30
 */
public class TestDriver {
    /**
     * main method
     * @param args
     */
    public static void main(String[] args) {
        ClassList classList = new ClassList();

        classList.add(new Student("a","b",1));
        classList.add(new Student("c","d",2));
        System.out.println(classList);

        System.out.println("add to the front:");
```

```
        classList.add(new Student("a","a",3));
        System.out.println(classList);

        System.out.println("add to the end:");
        classList.add(new Student("d","d",4));
        System.out.println(classList);

        System.out.println("add to the middle:");
        classList.add(new Student("c","b",5));
        System.out.println(classList);

        System.out.println("add same student:");
        classList.add(new Student("a","b",1));
        System.out.println(classList);

        System.out.println("reversed linked list");
        Student[] temp = classList.getReversedList();
        for(Student each : temp){
            System.out.println(each);
        }

        System.out.println("\nOriginal Linked List: ");
        System.out.println(classList);

        System.out.print("Length: ");
        System.out.println(classList.getNumStudents()+"\n");

        System.out.println("remove student not exists: d,c (12)");
        try{
            classList.remove(new Student("c","d",12));
            System.out.println(classList);
        }catch (NoSuchElementException e){
            System.out.println(e.getMessage());
        }


        System.out.println("after remove student from the front:");
        classList.remove(new Student("a","a",3));
        System.out.println(classList);

        System.out.println("after remove student from the end:");
        classList.remove(new Student("d","d",4));
        System.out.println(classList);

        System.out.print("Length: ");
        System.out.println(classList.getNumStudents()+"\n");

        System.out.println("after remove student from the middle:");
        classList.remove(new Student("a","b",1));
        System.out.println(classList);

        System.out.println("after remove student from the front:");
```

```
        classList.remove(new Student("a","b",1));
        System.out.println(classList);

        System.out.println("after remove student from the end/front:");
        classList.remove(new Student("c","d",2));
        System.out.println(classList);

        System.out.print("Length: ");
        System.out.println(classList.getNumStudents()+"\n");
    }
}
```

2. Test:

```
b, a (1)
d, c (2)

add to the front:
a, a (3)
b, a (1)
d, c (2)

add to the end:
a, a (3)
b, a (1)
d, c (2)
d, d (4)

add to the middle:
a, a (3)
b, a (1)
b, c (5)
d, c (2)
d, d (4)

add same student:
a, a (3)
b, a (1)
b, a (1)
b, c (5)
d, c (2)
d, d (4)

reversed linked list
d, d (4)
d, c (2)
b, c (5)
b, a (1)
b, a (1)
a, a (3)
```

```
Original Linked List:
a, a (3)
b, a (1)
b, a (1)
b, c (5)
d, c (2)
d, d (4)

Length: 6

remove student not exists: d,c (12)
Student not found
after remove student from the front:
b, a (1)
b, a (1)
b, c (5)
d, c (2)
d, d (4)

after remove student from the end:
b, a (1)
b, a (1)
b, c (5)
d, c (2)

Length: 4

after remove student from the middle:
b, a (1)
b, c (5)
d, c (2)

after remove student from the front:
b, c (5)
d, c (2)

after remove student from the end/front:
b, c (5)

Length: 1
```