WEEK 13

Quiz 2

Name
Sherly Lutfi Azkiah Sulistyawati

NIM
2341720241

Class
1I

Major
Information Technology

Study Program
D4 Informatics Engineering

Github link:
https://github.com/sherlyazkiah/AlgoritmadanStrukturData/tree/main/Practice/Week13

1. addLast method is use for add a new node at the end of the list.

```
void addLast(int data){
    Node nu = new Node(data);
    if (isEmpty()){
        head = tail = nu;
    } else{
        //complete here
        tail.n = nu;
        nu.p = tail;
        tail = nu;
    }
    size++;
}
```

This is the example of addLast method. The new node will add after 45 and after 25.

```
public static void main(String[] args){
    DoubleLinkedList dll = new DoubleLinkedList();
    dll.addFirst(data:45);
    dll.addFirst(data:10);
    dll.addFirst(data:10);
    dll.addFirst(data:15);
    dll.addFirst(data:150);
    dll.print();
    dll.deleteFirst();
    dll.print();
    //continue to call addLast, deleteLast, printFromTail,
    dll.addLast(data:25);
    dll.print();
    dll.addLast(data:30);
    dll.print();
```

```
150-15-10-10-45-
15-10-10-45-
15-10-10-45-25-
15-10-10-45-25-30-
```

2. deleteLast method is use for remove the last node.

```
void deleteLast(){
    if (isEmpty()) {
        System.out.println(x:"Linked list is still empty, cannot remove");
    } else if (head == tail) {
        head = tail = null;
        size--;
    } else {
        tail = tail.p;
        if (tail != null) {
            tail.n = null;
        }
        size--;
    }
}
```

This is the example of deleteLast method. The last node (30) will remove from the list.

```
    dll.deleteLast();
    dll.print();
```

```
15-10-10-45-25-30-
15-10-10-45-25-
```

3. printFromTail method is use for traverses and prints all data from tail to head.

```
void printFromTail(){
    Node tmp = tail;
    while (tmp != null) {
        System.out.print("" + tmp.data + "-");
        tmp = tmp.p;
    }
    System.out.println(x:"");
}
```

This is the example of printFromTail method. The data prints from the tail to head.

```
    dll.printFromTail();
```

```
15-10-10-45-25-
25-45-10-10-15-
```

4. getPositionFromTail method is use for find positions relative to the tail.

```
int getPositionFromTail(int data){
    if (isEmpty()) {
        System.out.println(x:"Linked list still empty");
        return -1;
    }
    Node tmp = tail;
    int position = 0;
    while (tmp != null) {
        if (tmp.data == data) {
            return position;
        }
        tmp = tmp.p;
        position++;
    }
    return -1;
}
```

This is the example of getPositionFromTail method. thi s method will return the position of data from last

```
    System.out.println("Position of 45 from tail: " + dll.getPositionFromTail(data:45));
```

```
25-45-10-10-15-
Position of 45 from tail: 1
```

5. getLastPositionFromTail method is use for find the data and the searching start from last.

```
int getLastPositionFromTail(int data){
    if (isEmpty()) {
        System.out.println(x:"Linked list still empty");
        return -1;
    }
    Node tmp = tail;
    int position = 0;
    int lastPosition = -1;
    while (tmp != null) {
        if (tmp.data == data) {
            lastPosition = position;
        }
        tmp = tmp.p;
        position++;
    }
    return lastPosition;
}
```

This is the example of getLastPositionFromTail method. This method will return the last position where the data found and the searching start from last.

```
    System.out.println("Last position of 15 from tail: " + dll.getLastPositionFromTail(data:15));
```

```
25-45-10-10-15-
Last position of 15 from tail: 4
```

6. getAverage method is use for calculates the average of all nodes' data.

```
double getAverage(){
    if (isEmpty()) {
        return 0;
    }
    //complete here
    Node tmp = head;
    int sum = 0;
    while (tmp != null) {
        sum += tmp.data;
        tmp = tmp.n;
    }
    double average = sum/size;
    return average;
}
```

This is the example of getAverage method. It will calculate all the data and divide into the size of the list.

```
    System.out.println("Average of data: " + dll.getAverage());
```

```
25-45-10-10-15-
Average of data: 21.0
```

7. getNodeByIndex retrieves a node by its index.

```
Node getNodeByIndex(int index){
    Node tmp = head;
    //complete by using loop to get node at certain index below
    for(int i=1;i<=index;i++){
        tmp = tmp.n;
    }
    return tmp;
}
void sort(){
    for(int i=0; i<size-1; i++){
        //comparisson data with adjacent
        for(int j=1; j<size-i; j++){
            Node node1 = getNodeByIndex(j);
            Node node2 = getNodeByIndex(j-1);
            if(node2.data > node1.data){
                //SWAP
                int tmp = node2.data;
                node2.data = node1.data;
                node1.data = tmp;
            }
        }
    }
}
```

```
System.out.println("The node in index 2 is: " + dll.getNodeByIndex(index:2).data);
```

```
15-10-10-45-25-
```

```
The node in index 2 is: 10
```

8. getMedian method is use for calculates the median value after sorting the list.

```
double getMedian(){
    if (isEmpty()) {
        return 0;
    }
    //complete to calculate median below
    sort();
    if (size % 2 == 1) {
        return getNodeByIndex(size / 2).data;
    } else {
        return (getNodeByIndex(size / 2 - 1).data + getNodeByIndex(size / 2).data) / 2.0;
    }
}
```

This is the example of getMedian methods.

```
    System.out.println("Median of data: " + dll.getMedian());
```

```
25-45-10-10-15-
```

```
Median of data: 15.0
```

9. Main class

```java
//9. complete main to make a simulation
Run | Debug
public static void main(String[] args){
    DoubleLinkedList dll = new DoubleLinkedList();
    dll.addFirst(data:45);
    dll.addFirst(data:10);
    dll.addFirst(data:10);
    dll.addFirst(data:15);
    dll.addFirst(data:150);
    dll.print();
    dll.deleteFirst();
    dll.print();
    //continue to call addLast, deleteLast, printFromTail,
    dll.addLast(data:25);
    dll.print();
    dll.addLast(data:30);
    dll.print();
    dll.deleteLast();
    dll.print();
    dll.printFromTail();
    //getPositionFromTail, getLastPositionFromTail,
    System.out.println("Position of 45 from tail: " + dll.getPositionFromTail(data:45));
    System.out.println("Last position of 15 from tail: " + dll.getLastPositionFromTail(data:15));
    System.out.println("The node in index 2 is: " + dll.getNodeByIndex(index:2).data);
    //getAverage, getMedian
    System.out.println("Average of data: " + dll.getAverage());
    System.out.println("Median of data: " + dll.getMedian());
}
```