

**Write a Java Program to Find the Middle Node of a Linked list in a Single-pass**

```
public class LinkedListTest {  
    public static void main(String args[]) {  
        //creating LinkedList with 5 elements including head  
        LinkedList linkedList = new LinkedList();  
        LinkedList.Node head = linkedList.head();  
        linkedList.add( new LinkedList.Node("1"));  
        linkedList.add( new LinkedList.Node("2"));  
        linkedList.add( new LinkedList.Node("3"));  
        linkedList.add( new LinkedList.Node("4"));  
  
        //finding middle element of LinkedList in single pass  
        LinkedList.Node current = head;  
        int length = 0;  
        LinkedList.Node middle = head;  
  
        while(current.next() != null){  
            length++;  
            if(length%2 ==0){  
                middle = middle.next();  
            }  
        }  
    }  
}
```

```
        current = current.next();  
    }
```

```
    if(length%2 == 1){  
        middle = middle.next();  
    }
```

```
    System.out.println("length of LinkedList: " + length);
```

```
    System.out.println("middle element of LinkedList : " +  
middle);
```

```
    }
```

```
}
```

```
class LinkedList{
```

```
    private Node head;
```

```
    private Node tail;
```

```
    public LinkedList(){
```

```
        this.head = new Node("head");
```

```
        tail = head;
```

```
    }
```

```
public Node head(){  
    return head;  
}
```

```
public void add(Node node){  
    tail.next = node;  
    tail = node;  
}
```

```
public static class Node{  
    private Node next;  
    private String data;  
  
    public Node(String data){  
        this.data = data;  
    }
```

```
    public String data() {  
        return data;  
    }
```

```
    public void setData(String data) {
```

```
        this.data = data;
    }

    public Node next() {
        return next;
    }

    public void setNext(Node next) {
        this.next = next;
    }

    public String toString(){
        return this.data;
    }
}
```

alvas-education-foundation/3rdOnline Java Compiler - Online J...jdoodle.com/online-java-compiler/

Sponsored: Bitcoin Virtual Hackathons- Earn crypto building OSS. New prizes/webinars every week [Sign up + learn more](#)

```
1 public class LinkedListTest {
2
3
4
5
6 //creating LinkedList with 5 elements including head
7 LinkedList linkedList = new LinkedList();
8 LinkedList.Node head = linkedList.head();
9 linkedList.add( new LinkedList.Node("1"));
10 linkedList.add( new LinkedList.Node("2"));
11 linkedList.add( new LinkedList.Node("3"));
12 linkedList.add( new LinkedList.Node("4"));
13
14 //finding middle element of LinkedList in single pass
15 LinkedList.Node current = head;
16 int length = 0;
17 LinkedList.Node middle = head;
18
19 while(current.next() != null){
20     length++;
21     if(length%2 == 0){
22         middle = middle.next();
23     }
24     current = current.next();
25 }
26
27 if(length%2 == 1){
28     middle = middle.next();
29 }
30
31 System.out.println("length of LinkedList: " + length);
32 System.out.println("middle element of LinkedList : "
33                     + middle);
34 }
35
36 }
37
38 class LinkedList{
39     private Node head;
40     private Node tell;
41 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4Stdin Inputs

☐ Interactive

CommandLine Arguments

Execute

Result

CPU Time: 0.22 sec(s), Memory: 35836 kilobyte(s)compiled and executed in 0.959 sec(s)

length of LinkedList: 4  
middle element of LinkedList : 2

Type here to search

ENG 4:29 PM 6/15/2020