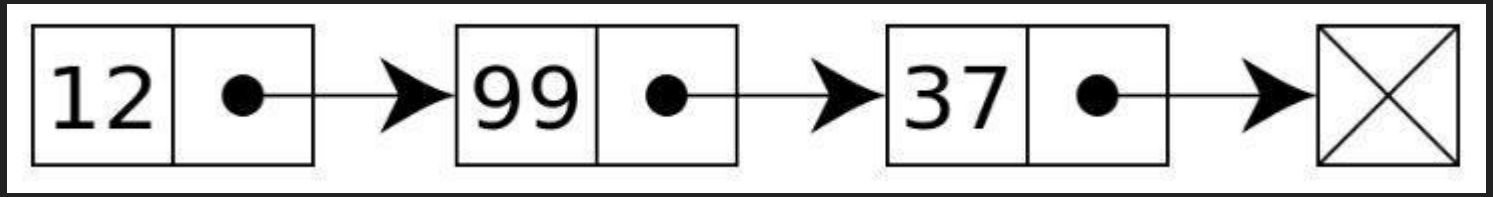
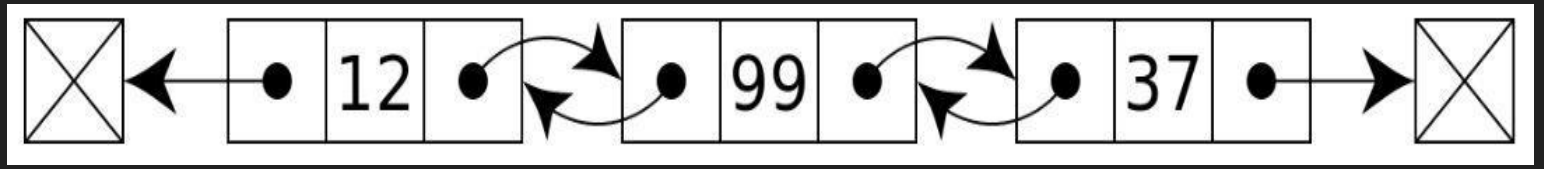
Computer Programming Lab 9

2018.05.04

Linked list(Concept)





Linked list(ListNode class)

```
public class ListNode {

int data;

ListNode next;

public ListNode(int data) {

this.data = data;

this.next = null;

}

public ListNode(int data, ListNode next) {

this.data = data;

this.next = next;

}

}
```

Linked list(LinkedIntList class)

```
public class LinkedIntList {

private ListNode front;

public LinkedIntList() {

front = null;

}

methods go here

}
```

Task

Fill methods on LinkIntList class

1. public void add(int value) : Adds the given value to the end of the list
2. public int get(int index) : Returns value in list at given index.
3. public int remove() : Removes and returns the list's first value.

Optional task

Fill methods on OptionalLinkedList class

public class ListNode {

String word;

ListNode next;

.. below same as slide3

}

Optional task

```
Public class OptionalLinkedList{

private ListNode front;

public OptionalLinkedList() {

front = null;

}

methods go here

}
```

Optional task

1. public String get(int index) : Returns value in list at given index.
2. public void append(ListNode r) : Adds the given linked list to the end of the current list.
3. public void rev() : reverses current linked list.
4. public bool substring(String s) : check whether current linked list has substring s or not.

Optional task example

OptionalLinkedList Instance test

an => apple => a => day => keeps => the => doctor => away

System.out.println(test.get(1))

-> apple

System.out.println(test.get(3))

-> day

Optional task example

OptionalLinkedList Instance test

an => apple => a => day => keeps => the => doctor => away

ListNode Instance foo

she => sells => seashells => on => the => seashore

test.append(foo)

OptionalLinkedList Instance test(after append)

an => apple => a => day => keeps => the => doctor => away => she => sells => seashells => on => the => seashore

Optional task example

OptionalLinkedList Instance test

an => apple => a => day => keeps => the => doctor => away

test.rev()

OptionalLinkedList Instance test(after rev)

away => doctor => the => keeps => day => a => apple => an

Optional task example

OptionalLinkedList Instance test

an => apple => a => day => keeps => the => doctor => away

test.substring(“aykeepsth”) : true

test.substring(“leeday”) : false

test.substring(“daykeepsthe”) : true