Singly Linked List

Computer Science Department
Eastern Washington University
Yun Tian (Tony) Ph.D.

Review

- Storage Type
  - Array-based(sequential)
  - Linked
  - Hashing
- Linear and Non-linear Data Structure
  - Linear: each item in collection has one predecessor and one successor at most.
  - Non-linear: each item in collection may have more than one predecessor or successor.

Outlines Today

- Two Java Concepts
- Singly Linked List

Java Object Reference

- Java uses ***new*** operator to allocate memory space for an Object.
  - then use its constructor to initialize the space.
- We use a **reference variable** to store the address or the location of the new object.
- We use the object reference to access its data and its methods.
  - E.g. image.getPixel(100,50),

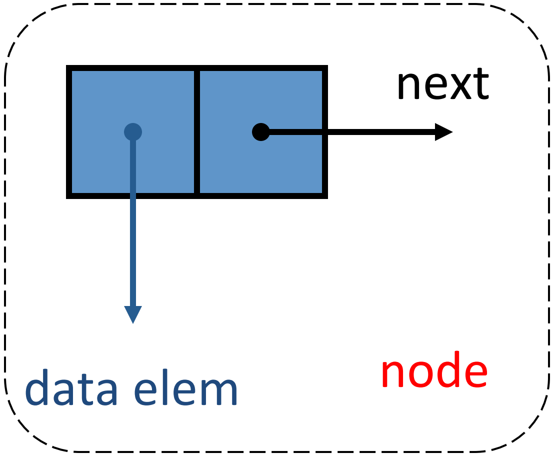
Java Recursive Types

- Recursive types are classes containing members of the same type.

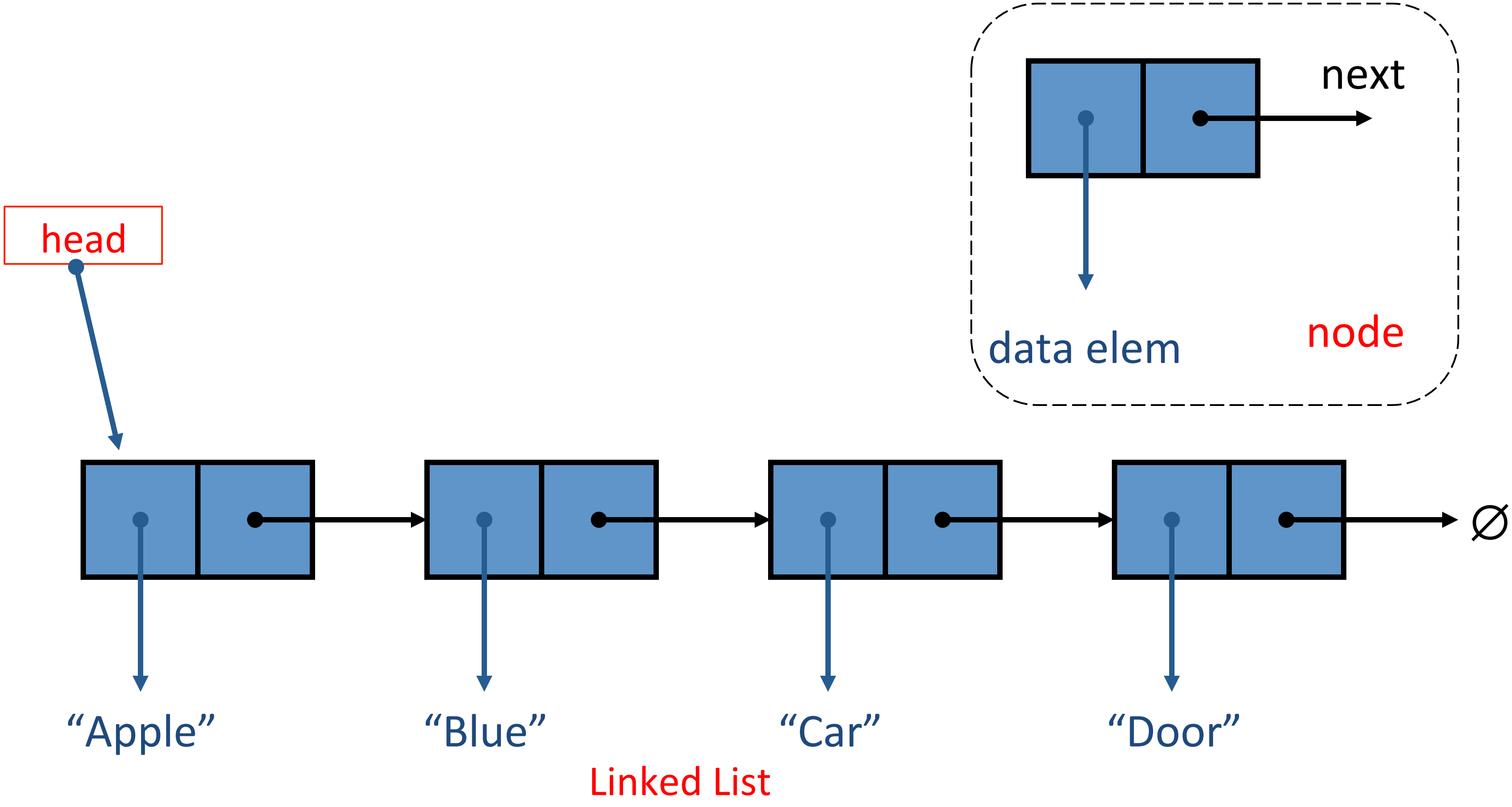
```
public class Employee {
    int age;
    int gender;
    String address;
    Employee boss;
……..
}
```

Concept of Singly Linked List

- A singly linked list is a concrete data structure consisting of a sequence of nodes.
- Each node only stores
  - a data element
    - usually a reference to data
  - a link to the next node
    - usually a reference to next node.



Concept of Singly Linked List



LinkedList Class Implementation

```
public class LinkedList implements Iterable<Object> {
        private class Node {
                private Object data;
                private Node next;

                private Node( Object data, Node next ) {
                        this.data=data;
                        this.next = next;
                }
                private Node( Object data ) {
                        this(data, null);
                }
                private Node() {}  // Can we leave out this empty constructor?
        }//end of node

        private Node head;
        private int size;

        //………….to be continued on next page
```

How does this differs from using LinkedList<E> implements Iterable<E>?

LinkedList Class Implementation

```
public LinkedList() {
    this.head = null;
    this.size = 0;
}
public boolean  isEmpty() {
    return (this.head == null); // or we use   return (this.size == 0)
}

public void addFirst( Object data )
{
    this.head = new Node(data,   __this.head);
    this.size ++;
}
//…………to be continued on next page
```

LinkedList Class Implementation

```
   public void addLast( Object data ) {
      if(isEmpty() )
           addFirst(data);_
      else {
         Node cur = this.head;
         while( __cur.next != null_) {
               cur = cur.next;
          }
         __cur.next_= new Node(data, null );
         this.size ++;
      }
}
 //………….to be continued on next page
```

LinkedList Class Implementation

```
public Object remove( int index ) throws IllegalArgumentException {
    if( index < 0 | | index >= this.size )
        throw new IllegalArgumentException("index out of range on call to remove " + index);
    Node cur, prev;
    cur = this.head;
    prev = null;
    Object data;
    if(index ==0) {  //edge case when remove the first node
        data = this.head.data;
        this.head = this.head.next;
        this.size --;
        return data;
    }
    for(int i = 0; i < index; i ++) {
        prev = cur;
        cur = cur.next;
    }
    prev.next = cur.next;
    this.size --;
    return cur.data;
}
```

Summary

- Concept of Singly Linked List
- How to implement it?
  - LinkdList Class
  - Inner Node Class
  - addFirst()
  - addLast()
  - remove(index)

Next class

- More operation on LinkedList
  - – Remove(Object obj)
  - – List Iterator