Using linked memory

1/15/16

Green slides taken (w/ minor modifications) from Cynthia Lee’s CS 2 slides on http://www.peerinstruction4cs.org, licensed under Creative Commons Attribution-NonCommercial 4.0 International License.
Announcements

• Reading: For Monday: rest of Chapter 3
• HW out soon
• Extra credit opportunity: MLK day convocation
• Quiz today!
Recall: Exceptions

- Code that throws an exception:
  ```java
  public void method(...) throws ExceptionType {
      ...
      throw new ExceptionType(...);
      ...
  }
  ```

- Code that catches it:
  ```java
  try {
      //do stuff that may generate exceptions
  } catch(ExceptionType ex) {
      //handle the exception
  }
  ```
public void testRemoveEmpty() {
    try {
        empty.remove();
        fail();
    } catch (NoSuchElementException ex) {
    }
}
public class LinkedBag<T> implements Bag<T> {
    private Node head;
    // optionally: could have size

    public class Node {
        public T value;
        public Node next;

        public Node(T value, Node next) {
            this.value = value;
            this.next = next;
        }
    }

    ...

    An “inner class”. Objects can only be created in context of the main class.
How our class looks in memory

-head points to first node

Null next pointer means “end of the list”
What line of code will correctly complete this method?

```java
public void add(T newItem) {
    Node newNode = new Node(newItem, head);
    __________________________;
}
```

A) No line is needed. The code is correct as written.
B) head = head.next;
C) head = newNode;
D) newNode.next = head;

Question to think about/discuss: what is the purpose or effect of passing head as an argument here?
What line of code will correctly complete this method?

public void add(T newItem) {
    Node newNode = new Node(newItem, head);
    __________________________;
}

A) No line is needed. The code is correct as written.
B) head = head.next;
C) head = newNode;
D) newNode.next = head;

Question to think about/discuss: what is the purpose or effect of passing head as an argument here?
We implement size() by traversing the list. Which of the following tests if curr is at the end of the list?

```java
public int size() {
    Node curr = head;
    int count = 0;
    while(XXX) {
        count++;
        ...
    }
    return count;
}
```

A. curr == head
B. curr != null
C. curr.next != head
D. curr.next != null
E. Not exactly one of the above
We implement size() by traversing the list. Which of the following tests if curr is at the end of the list?

```java
public int size() {
    Node curr = head;
    int count = 0;
    while(XXX) {
        count++;
        ...
    }
    return count;
}
```

A. curr == head
B. curr != null
C. curr.next != head
D. curr.next != null
E. Not exactly one of the above
Which of the following advances curr to point to the next Node in the list?

public int size() {
    Node curr = head;
    int count = 0;
    while(curr != null) {
        count++;
        XXX
    }
    return count;
}
Which of the following advances curr to point to the next Node in the list?

```java
public int size() {
    Node curr = head;
    int count = 0;
    while(curr != null) {
        count++;
        XXX
    }
    return count;
}
```

A. curr++;
B. curr.next;
C. curr = curr.next;
D. Node temp = curr.next;
    curr = temp;
E. Not exactly one of the above (C & D)