

CSE 12 – Basic Data Structures and Object-Oriented Design

Lecture 6

Greg Miranda, Fall 2020

Announcements

- Quiz 6 due Friday @ 9am
- Survey 2 due Friday @ 11:59pm
- PA1 due tonight @11:59pm
- PA2 released tomorrow

Code → ~2 weeks
Readme - questions
↑

Topics

- Questions on Lecture 6?
- Generics & Exception Exercises

Questions on Lecture 6?

Generics

- Convert LinkedListString to be a generic

$AList <E>$

↑

```

public interface List<Element> { ← ←
    /* Add an element at the end of the list */
    void add(Element s);

    /* Get the element at the given index */
    Element get(int index);

    /* Get the number of elements in the list */
    int size();
}

```

```

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

```

List < E >
↑

```

public class LinkedList implements StringList {
    Node front;
    int size;

    public LinkedList() {
        this.front = new Node(null, null);
        this.size = 0;
    }

    public String get(int index) {
        Node temp = this.front.next;
        for (int i = 0; i < index; i += 1) {
            temp = temp.next;
        }
        return temp.value;
    }


    public int size() {
        return this.size;
    }

    public void add(String s) {
        Node temp = this.front;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = new Node(s, null);
        this.size += 1;
    }
}

```

Exceptions

- What happens if an invalid index is passed to get()?
- Modify get() to throw an exception if the index is invalid



```
public String get(int index) {  
    Node temp = this.front.next;  
    for (int i = 0; i < index; i += 1) {  
        temp = temp.next;  
    }  
    return temp.value;  
}
```

- Write a test to verify get() throws an exception with an invalid index

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;
```

```
public class TestList {
```

```
→ @Test(expected = IndexOutOfBoundsException.class) ←
    public void testNegativeIndex() { €
        List<String> slist = new ArrayList<String>();
        slist.add("banana");
        slist.get(-1);
    }
}
```

Handwritten red annotations: A checkmark above the `@Test` annotation, an arrow pointing to the `expected` parameter, a red `€` symbol at the end of the method signature, an arrow pointing to `ArrayList`, and the text `ArrayList` written below it.