LinkedList

Alexandre Bergel
http://bergel.eu
11/09/2017

LinkedList

As you already know, a linked list is a very convenient abstract data type

Consider the following interface

```
interface ILinkedList {
  public void addFirst(Object o);
  public boolean includes(Object o);
  public int size();
}
```

Propose an implementation of this interface

Some tests first

```java
public class LinkListTest {
    private ILinkList l;

    @Before
    public void setUp() throws Exception {
        l = new LinkList();
     }

    @Test
    public void testDefault() {
        assertTrue(l.isEmpty());
        assertEquals(0, l.size());
    }

    @Test
    public void adding() {
        l.addFirst(42);
        l.addFirst("Happy World");
        assertEquals(2, l.size());
        assertTrue(l.includes("Happy World"));
        assertTrue(l.includes(42));
        assertFalse(l.includes(43));
    }
}
```

The class LinkList

```java
public class LinkList implements ILinkList {
	private Cell first;

	public LinkList() {
		first = new NullCell();
	}

	@Override
	public void addFirst(Object o) {
		first = new ConcreteCell(o, first);
	}

	@Override
	public boolean includes(Object o) {
		return first.includes(o);
	}

	@Override
	public int size() {
		return first.size();
	}

	@Override
	public boolean isEmpty() {
		return this.size() == 0;
	}
}
```

The Cell interface and NullCell class

```
interface Cell {
    boolean includes(Object o);
    int size();
}

public class NullCell implements Cell {
    @Override
    public boolean includes(Object o) {
        return false;
    }

    @Override
    public int size() {
        return 0;
    }
}
```

The ConcreteCell class

```java
class ConcreteCell implements Cell {
    private Object value;
    private Cell next;

    public ConcreteCell(Object o, Cell cell) {
        value = o;
        next = cell;
    }

    @Override
    public boolean includes(Object o) {
        return value.equals(o) || next.includes(o);
    }

    @Override
    public int size() {
        return 1 + next.size();
    }
}
```

Other operations?

Can you implement these two operations?

- indexOf(Object) that returns the position of the element in the chain
- addLast(Object) that add an element at the end of the chain

Send your answer to u-cursos