

Dec 08, 10 7:16

ReviewSolution.java

Page 1/3

```
// #15

public class Node {
    private int datum;
    private Node next;
    public Node(int d, Node n) {
        datum = d;
        next = n;
    }
    public Node next() { return next; }
    public int datum() { return datum; }
    public void setNext(Node next) { this.next = next; }
}

public class List {
    private Node head;
    public List() { head = null; }
    public void addToFront(int item) {
        head = new Node(item, head);
    }
    public void removeHead() {
        if (head != null)
            head = head.next();
    }
    public int average() {
        int sum = 0,
            num = 0;
        for (Node current = head; current != null; current = current.next()) {
            sum += current.datum();
            num++;
        }
        return sum / num;
    }
}

// # 16

public class Node{

    private int datum;
    private Node next;

    public boolean contains(int item) {
        try {
            return datum == item || next.contains(item);
        } catch (NullPointerException npe) {
            return false;
        }
    }
}

// # 17

public class Stopwatch() {

    private boolean isRunning;
    private int time;
}
```

```

private long timeMark;

public Stopwatch() { // unnecessary
    isRunning = false;
    time = 0;
    timeMark = 0;
}

public void start() {
    if (!isRunning) {
        timeMark = System.currentTimeMillis();
        isRunning = true;
    }
}

public void stop() {
    if (isRunning) {
        time += System.currentTimeMillis() - timeMark;
        isRunning = false;
    }
}

public int getTime() {
    return time;
}
}

// # 18 a
static int arrayToInt(int[] array) {
    int toReturn = 0;
    for (int i = 0; i < array.length; i++)
        toReturn = toReturn * 10 + array[i];
    return toReturn;
}

// b
static int stringToInt(String str) {
    int toReturn = 0;
    for (int i = 0; i < str.length(); i++)
        toReturn = toReturn * 10 + ((int) (str.charAt(i) - '0'));
    return toReturn;
}

// # 20
public class LR implements LibraryRecord {

    private HashMap<String, String> patronBooks, bookPatrons;

    public LR() {
        patronBooks = new HashMap<String, String>();
        bookPatrons = new HashMap<String, String>();
    }

    public void checkedOut(String patron, String book) {

```

Dec 08, 10 7:16

**ReviewSolution.java**

Page 3/3

```
        if (! patronBooks.containsKey(patron))
            patronBooks.put(patron, "");
        patronBooks.put(patron, patronBooks.get(patron) + "," + book);
        if (! bookPatrons.containsKey(book))
            bookPatrons.put(book, "");
        bookPatrons.put(book, bookPatrons.get(book) + "," + patron);
    }

    public String booksCheckedOut(String patron) {
        return patronBooks.get(patron);
    }

    public String patronsCheckedOut(String book) {
        return bookPatrons.get(book);
    }
}
```