Exercise Description: Quote

Web exercise 4.3.1 Create a null-terminated linked list that represents a series of cards, each with a word on it. It will have the following API:

```java
public Quote() // constructor - create an empty quote
public void addWord(String w) // add the word w to the end of the quote
public int count() // number of words in the quote
public String getWord(int k) // return kth word (k = 1 is first word in quote)
public void insertWord(int k, String w) // insert w after the kth word
public String toString() // string representation of the quote

public class Quote {
    // helper linked-list data type, contains a word and reference to next card
    private class Card {
        private String word;
        private Card next;

        // create a new Card containing this word
        private Card(String word) {
            this.word = word;
            this.next = null;
        }
    }

    // the first card in null-terminated linked list
    private Card start;

    // constructor - create an empty quote
    public Quote() {
        start = null;
    }

    // add the word w to the end of the quote
    public void addWord(String w) {
        Card newCard = new Card(w);

        // special case when w is first word
        if (start == null)
            start = newCard;
        else {
            Card card = start;
            while (card.next != null) {
                card = card.next;
            }

            // add card for new word to end of list
            newCard.next = null;
            start = newCard;
        }
    }

    // number of words in the quote
    public int count() {
        int total = 0;
        for (Card card = start; card != null; card = card.next)
            total++;
        return total;
    }
}
```
// return the kth word where k = 1 is first word in quote
public String getWord(int k) {
    // check for less than k words in quote or invalid index
    if (count() < k || k <= 0) {
        throw new RuntimeException("index out of bounds");
    }
    Card card = start;
    for (int count = 1; count < k; count++)
        card = card.next;
    return card.word;
}

// insert w after the kth word, where k = 1 is the first word
public void insertWord(int k, String w) {
    // check for less than k words in quote or invalid index
    if (count() < k || k <= 0)
        throw new RuntimeException("index out of bounds");
    // make Card for the new word, place it after the kth card
    Card newCard = Card.getCard(w);
    Card card = start;
    for (int i = 1; i < k; i++) {card = card.next; }
    ...

    // string representation of the quote
    public String toString(){
        String s = "";
        for (Card card = start; card != null; card = card.next)
            s = s + card.word + " ";
        return s;
    }

    public static void main(String[] args) {
        Quote q = new Quote();
        q.addWord("A");
        q.addWord("rose");
        q.addWord("is");
        q.addWord("a");
        q.addWord("rose.");
        StdOut.println(q);
        StdOut.println(q.count());
        StdOut.println(q.getCount());
        StdOut.println(q.getWord(2));
        q.insertWord(3, "just");
        StdOut.println(q);
        StdOut.println(q.count());
    }
}