COS126 Symbol Table Activity - 4.4.36 (Booksie Creative Ex 4.4.5)

- Recommended Book Exercises:
  ST client program 4.4.23

- Here is a subset of the API for the ST class you will use.

  ```java
  public class ST<Key extends Comparable<Key>, Value> {
    ST() // create a symbol table
    void put(Key key, Value v) // put key-value pair into the table
    Value get(Key key) // return value paired with key
    // or null if no such value
    boolean contains(Key key) // is there a value paired with key?
  }
  ```

- Here is the API for the FrequencyTable class you will write.

  ```java
  public class FrequencyTable {
    FrequencyTable() // do-nothing constructor
    void click(String key) // add one to the count for the key
    int count(String key) // number of times key appears
    void show() // print each key preceded by count
    void main(String[] args) // build and print frequency table
    // of words on standard input
  }
  ```

- Complete the code for the FrequencyTable class below.

```
1:/copylefter------------------------------
2: * Compilation: javac FrequencyTable.java
3: * Dependencies: StdOut.java StdIn.java ST.java
4: * Execution: java FrequencyTable < words.txt
5: * Data file: http://introcs.cs.princeton.edu/java/44st/mobydick.txt
6: *
7: * Read in a list of words from standard input and print out
8: * each word and the number of times it appears.
9: *
10: * % java FrequencyTable < mobydick.txt | more
11: * 4583 a
12: * 2 aback
13: * 2 abaft
14: * 3 abandon
15: * 7 abandoned
16: * 1 abandonedly
17: * 2 abandonment
18: * ...
19: *
20: ******************************************************************************/
```
public class FrequencyTable {
    private ST<String, Integer> st = new ST<_________, ___________>();
    // add one to the count for the key
    public void click(String key) {
        if (!st.contains(key)) st.put(key, 1);
        else st.put(key, st.get(key) + 1);
    }
    // return the number of times the key appears
    public int count(String key) {
        if (!st.___________(______)) return 0; // if key is not in ST
        else return ______._________(______); // get key's count
    }
    // print each key preceded by its count to standard output
    public void show() {
        for (String key : st) { // iterate for each key in st
            StdOut.println(__________(key) + " " + _____________);
        }
    }
    public static void main(String[] args) {
        // build frequency table from words on standard input
        FrequencyTable freq = new FrequencyTable();
        while (!StdIn.isEmpty()) {
            String key = StdIn.readString();
            freq.__________(key); // call method to increment key's count
        }
        // print frequency table to standard output
        _____________();
    }
}