Programming Languages and Techniques (CIS120)

Lecture 32
November 19, 2018

Histogram Demo
Chapter 28
Announcements

• HW8: SpellChecker
  – Due tomorrow, November 20th

• Thanksgiving Break:
  – No lab sections
  – Wednesday, November 21st: Bonus Lecture offered ONLY at the 11:00-noon timeslot (everyone welcome to attend)
  – (Dr. Sheth away; Dr. Fouh will cover)
  – Topic: Code is Data (fun, but not relevant to HW or exam materials)

• NOTE: Office Hours schedules are different due to Thanksgiving
  – Check the calendar

• Interested in becoming a TA?
  – Applications are available for CIS 110, 120, 160, 121 – due November 28th
  – We encourage folks of underrepresented groups to apply.
Midterm Exam Scores

AVERAGE: 68
MEDIAN: 70
STD. DEV: 12.07
MAX: 95
HW09: Create your own Game

• Instructions available soon.

• Checkpoint: Describe your game, what features you will use.
  – DUE: Next Week! (After break)
  – Not a huge time commitment
  – Intended to get you thinking about your project. Be creative!
  – Submission via Gradescope

• Game Project
  – Due: MONDAY, December 10th at midnight
  – No late days!
A design exercise using java.io and the generic collection libraries

(SEE COURSE NOTES FOR THE FULL STORY)
Problem Statement

Write a program that, given a filename for a text file as input, calculates the frequencies (i.e. number of occurrences) of each distinct word of the file. The program should then print the frequency distribution to the console as a sequence of “word: freq” pairs (one per line).

Histogram result:
The : 1
Write : 1
a : 4
as : 2
calculates : 1
command : 1
console : 1
distinct : 1
distribution : 1
e : 1
each : 1
file : 2
filename : 1
for : 1
freq : 1
frequencies : 1
frequency : 1
given : 1
i : 1
input : 1
line : 2
number : 1
occurrences : 1
of : 4
one : 1
pairs : 1
per : 1
print : 1
program : 2
sequence : 1
should : 1
text : 1
that : 1
the : 4
then : 1
to : 1
word : 2
Text File

Find Words

Count

Print

Printed Histogram
Decompose the problem

- Sub-problems:
  1. How do we iterate through the text file, identifying all of the words?
  2. Once we can produce a stream of words, how do we calculate their frequency?
  3. Once we have calculated the frequencies, how do we print out the result?

- What is the interface between these components?
- Can we test them individually?
How to produce a stream of words?

1. How do we iterate through the text file, identifying all of the words?

```java
public interface Iterator<T> {
    // returns true if the iteration has more elements
    public boolean hasNext();
    // returns the next element in the iteration
    public T next();
    // Optional: removes last element returned
    public void remove();
}
```

- **Key idea**: Define a class (WordScanner) that implements this interface by reading words from a text file.
Coding: Histogram.java

WordScanner.java

Histogram.java