Programming Languages and Techniques (CIS120)

Lecture 32
April 8, 2016

Histogram Demo
Poll

Are you here today?

1. yes
Announcements

• HW8: Spellchecker
  – Available on the web site
  – Due: Tuesday
  – Parsing, working with I/O, more practice with collections

• Apply to be a TA!
  – CIS 120 TAs are THE BEST!
  – http://www.cis.upenn.edu/~introtas
Design Example: Histogram.java

A design exercise using java.io and the generic collection libraries.
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**
Which I/O class should we use to open the text file?

1. InputStream
2. FileInputStream
3. FileReader
4. BufferedReader
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?
1. How do we iterate through the text file, identifying all of the words?

   • **Key idea:** Define a class (WordScanner) that implements this interface by reading words from a text file.
Interactive Demo

Histogram.java

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?
Interactive Demo

WordScanner.java