

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

Back to Week 2

Lessons

Prev

Next

Telling a Random Story

Using and Improving GladLibs

Introduction

7 min

Brittle Code

3 min

Adding New Labels

4 min

Programming Exercise: Using GladLibs

10 min

HashMap

7 min

HashMap for Unique Words

4 min

HashMap for Flexible Design

7 min

Summary

3 min

Programming Exercise: Improving GladLibs

10 min

Practice Quiz: Using and Improving GladLibs

10 questions

Review

total = 34582

it will tell me that there are 34,582 words in the works of Confucius.

HashMap for Unique Words

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:03
Hi, today we're gonna write some code to find out how many times every word in a file occurs. So we'll find out how many times the word the occurs, how many times the word wonderful occurs, and we're going to start with a working program. Let me show you very quickly what that looks like. This countWords function that we have here, this method countWords, simply creates a file resource, loops over all the words in it, and counts how many total words there are. If I take this program, make an object on my workbench, right click and call countWords, it will ask me for to enter the name of a file. I will use Confucius, and [it will tell me that there are 34,582 words in the works of Confucius](#). Rather than knowing the total number of words, I'd like to find out how many times each individual word occurs. To do that, I'm going to use a map. So I'm going to need to add another local variable here, a HashMap. I'm going to map strings to integers. Each string will be a word that occurs in my file, that's the key in this map. And the value in the map is how many times that word occurs. So I need to create a new one.

1:18
I'm allowed to make a HashMap because I've imported java.util. That's the package in which we find HashMap. Now, as I read the words, rather than incrementing the total, I'm going to ask, have we seen w in the map before? So I'm going to ask whether the key set associated with the map contains w, the word I'm looking for.

1:46
And if it does, I've seen the word before, I'm going to put the value w back in the map. It's already there. I'm going to get the number of times that word occurs and add once. I'll go over again what I'm doing in a minute. If I've never seen the word before, then what I'm going to do is put it in my map

2:09
with the number one, because the word will have occurred one time. Let me compile and make sure I've got that right. So let me go over very quickly what I've just done. I've looked to see, does this word, w, that I just read and converted to lowercase, is it in my keyset? Have I ever seen it before? If I have seen it before, let me get the integer value associated with it. That's the number of times it's already occurred. Add 1 to that, and put that in the map as a key value pair, with the word and an incremented count. If I've never seen it before, this is the first time.

2:48
I've now added all the values to the map and now I'd like to print them out. The way I'm going to print them out is to loop over the key set.

2:59
That's all the words that are keys in my map.

3:02
And I'm going to get the value associated with it, that's the number of occurrences.

3:09
That's the value associated with my word, and if that value is big, and here I'm gonna say if it's bigger than 500, I'm going to print the words that occur a lot.

3:22
Which is, so I'm going to print the number of occurrences, a tab character, and the word itself. So I'm looping over the keyset that allows me to find every key. I'm getting the value associated with that key in the map, and if that's big, I'm going to print it. Big in this case is bigger than 500. So I'm going to right click and create a new object.

3:48
And I am going invoke the method countWords which before, used to count the total number of words, and now it's gonna count all the individual words in Confucius. And here we can see that in the file Confucius there are not that many words that occur more than 500 times. The occurs more that 2,000 times. And occurs 762. If I wanna see a little more in the way of words, I'll say 200, just to make sure that I'm getting something more. And I'll make a new version of this that sits on the object workbench. I'll right click and call countWords. I'll once again count Confucius, and you can see there are more words. Master occurs 484 times. Confucius was the master of many things. You are now going to be the master of programming as you use maps to map keys to values to solve many problems. Have fun coding.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?