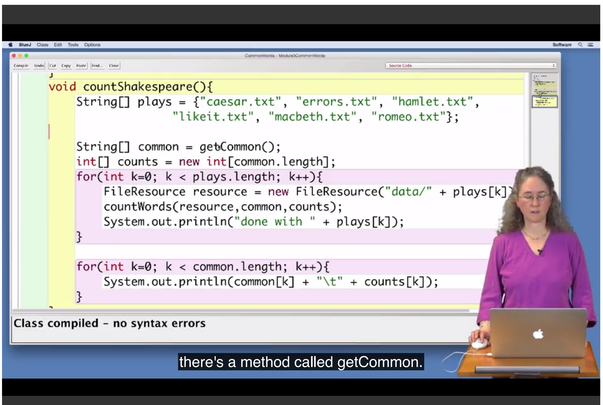

Counting with Arrays

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

Interactive Transcript

Search Transcript

English

0:03
What are the most common words in the English language? There are many ways to find common words. Using search tools to find shared data that has been curated from thousands and thousands of available text helps here. Others have already done such to figure out what the most common words are. For example, they have determined that the word the is used than any other word. Did Shakespeare use common words? We can use our edu.duke classes, public domain versions of Shakespeare's plays, and some simple array code to help answer this question. In this example, we will examine several of Shakespeare's plays and count how many times he uses the most common words. All right, so I've started the code for counting the most common words in Shakespeare's plays. So here's the code, let's look at it. I've got this method called Count Shakespeare. And you can see here, actually, this is a new thing. This shows you how you can initialize an array. Right here we've got the array plays, which is an array of strings. And we can just put curly brackets and list the items in the array. And so what I've done here is I've listed six data files that we've gotten that have text from six of Shakespeare's plays. We've also got another string array called common and there's a method called getCommon. Let's go look at that and see what that does. Here's getCommon. Here you can see there's another data file. This one is called common.txt and that's where I've already put 20 common words. The most common words that somebody else has determined are the most common English words. We're just going to read that and then since I happen to know there are 20 words in there, that's important. I can create the common string array of size 20 because arrays have to know the actual size. So, reading that in, and this loop here all it does is it goes though and reads them in one at a time and puts them in there. So we read in the 20 most common words. That's all getCommon does. Let's go back down here. So we've got the plays, we got the common word, and now we have a loop which is gonna go over and it's gonna read in each play. I've got the plays stored in a data folder. So you can see I'm adding data/ right before the name of the play. And then I'm gonna call count words which is going to count for each word in the play. It's going to check and see if it's one of the common words and if it is, it's going to count how many times each common word appears.

2:47
And then we just print a message that says hey we're done with the plays. Once we've counted all the common words, then we go through, and we're gonna print for each common word. So for example the, we're gonna print the and then how many times it occurs in all six of Shakespeare's plays. So let's go ahead and run this and see what happens. We'll compile it. It's compiled.

3:13
We will create an object. Then let's go ahead and run it. countShakespeare.

3:21
And you can see it doesn't work quite right. It's only counting of, a lot. So let's go back and look at the code. It turns out I haven't given you all the code. We still have to write one of the methods.

3:39
All right, so let's go see. We are going to have to write index of. So what index of does is it takes a list of words and a word, and we would like it to look for the word and a list of words or the array list of words and see if it's in there and count how many times it appears. Or actually we want it to give you the location the index of word appears In there. So what we're gonna have to do is we're gonna have to loop over all the words in the array list and check to see if word is equal to one of them. And if it is, we'll return the location of where it is. So let's just go ahead and start that. All right, so we will start with a four loop.

4:33
So in order to loop over the array we'll need a for loop. So let's say we'll need a variable int k. We'll start at zero.

4:46
And then as long as we don't go off the end of the array. So we'll say k less than list.length. And then we need to update our counter k by one.

5:06
Okay, so we'll go through and then what we need to do Is for each case slot, we need to check and see if the word matches the word that's in that slot. So we'll ask if, what is the word in list k? And we need to compare it to word, so we'll use dot equals.

5:41
And if it equals it, we found it, so we can return the location of it.

5:51
Return k. So this loop will go through and check to see does word equal this word, this word, this word, and if we find it inside k, we return k. If we don't find it, we need to indicate we don't find it so we need to fix this return that's after the for loop, after we've looked at everything. We'll change this to minus one, which is not a position in the array. And that will indicate that we didn't find it. Now, let's see how index of is being used. And we come over here.

6:27
Index is being used in countWords, which is right here. And you can see here we crawl index of for each word. We're getting each word out of the file. We pass the common words, and we pass the word. And then we check to see does it match? If it matches, then we're gonna use the counts array to update that count. So for the word the, we'll keep track of how many times we find it. Every time we find it we will update the counter counts, which is in the same slot as the word the which is in the common array. All lright, let's try and run this now. We'll compile it.

7:09
No syntax errors, that's good. We'll come over here and we'll run it.

7:18
So we're going to run countShakespeare. There we go. So, what this does is, first of all, you can see I've got six data files. Caesar.txt, errors.txt, Hamlet, likeit, Macbeth, and Romeo. And we've read all of those in and for each word in that file we've checked to see if it matches one of these common words. Here you can see the common words are the, of, and, and so on. And you can see that the appears in the combination of those six texts 4,237 times. And then you can see another one "for" appears 1,071. So you can see how many times each of these common words appear. So it looks like Shakespeare did use a lot of common words. Now how do we know that this is actually correct? They're just big numbers, how do we know? So, what I'm gonna do is I'm now gonna modify my program so I can run it on a really small file just to make sure and convince myself that I'm counting these numbers correctly. So, I will come back over here. And you can see here, I have this file called small.txt where I just put a few words in there. And what I wanna do now is modify my code.

8:40
So that I look at this file, small.txt. So back here in countShakespeare, What I'm gonna do is I'm gonna comment out this line here.

8:57
These two lines and I'm going to create my string plays just to be that one file. So I will say string,

9:10
Plays is going to equal just the one file, which is small.txt.

9:22
That's all I have to do, and now I'm gonna to run it. And this time, it will just look at that one file. So let's compile it. We got it compiled. We need to run it.

9:43
And there we go. So let's see. So if you look in small.txt, you can see the word the appears one, two, three times. If you look over here, you can see me counted the three times. That looks good. Of appears twice. We got two and and appears twice. We got a appears once. Okay, so it looks like it worked. Now I'm really more convinced that I counted the Shakespeare words correctly. All right, that's it. Thanks.

Downloads

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

Would you like to help us translate the transcript and subtitles into additional languages?