

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

Back to Week 2

Lessons

Prev

Next

Telling a Random Story

Module Learning Outcomes / Resources

10 min

Introduction

2 min

High-level Design Concepts

5 min

ArrayList

6 min

ArrayList for Unique Words

7 min

ArrayList Advantages and Issues

7 min

Summary

3 min

Programming Exercise: Telling a Random Story

10 min

Practice Quiz: Telling a Random Story

6 questions

Using and Improving GladLibs

Review

BlackJ Options

BlackJ Terminal Window - WordCounting

Software

unique words: 5895

So I'll find out how many times the occurs Montague,

ArrayList for Unique Words

Like

Dislike

Flag

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

>

Interactive Transcript

Search Transcript

English

0:02

Hi, we're gonna show an example of how the array list structure works. We'll show code that keeps track of how many times each word in a file occurs. The file or it could be a URL, could be a large file, like text from Shakespeare's Romeo and Juliet, or like the sayings of Confucius. We don't know how many different words there are before we start reading the file. So we can't easily use an array. Instead we'll use two ArrayList structures. One that holds strings and one that holds integer values. We'll need integer instead of int. We'll explain why. The kth value of the array list, whose name is myFreqs, will store an integer. The number of occurrences of the kth value in the instance variable array list, myWords. The names are names of instance variables that we'll see in the code. As shown here we see two occurrences of the with a string and an integer stored in the location with index zero in the two different array lists. Similarly there were three occurrences of the word dog, and one occurrence of the word green. Let's write code. We've got a class, WordFrequencies. I'm gonna open it up, and take a look to see how it works. And one way we can easily see how it works, before we look at the code, is simply to run it. I've created an instance of it on my object work bench. I'm going to right click and run the tester method.

1:31

I'm gonna open the file romeo.txt, the entire text of Shakespeare's Romeo and Juliet.

1:37

And I can see that there were 5,895 unique words. What I like to do with ArrayList is count how many times each word occurs. [So I'll find out how many times the occurs Montague](#), Capulet, all the words from Romeo and Juliet.

1:53

Just going through this code very quickly to highlight the key features before I add the counting. I have an instance variable, myWords, which is initialized to be able to hold strings in the constructor.

2:08

In tester, I call the function findUnique. Which creates a file resource and because it had no parameters that allowed me to open up any file I wanted. It looks over every word, converts each word to lowercase.

2:23

If the word has never been seen before, which I find out by using the array list method .indexOf which is a method whose same name we've seen in the string class. If it has never been seen, I add it to my words. Now what I want to do is have a parallel array. I will call it myFreqs, for my frequencies. So I need another instance variable.

2:46

This has to store integer values, and unfortunately with array lists in Java, I must use integer. You've seen this before with integer.parseInt. Similar to the class Double.parseDouble, I'll explain this as I go, I need to store integers. I'm going to create a place for it. I'm going to initialize that in the constructor to a new ArrayList of Integer values.

3:13

And then If I've never seen the word before, that means this is the first occurrence. So, I'm gonna add to the end of the array list the value 1. This is just like the idea of, the first time you see a word, it's occurred once. But, if I've seen it before, this is in my else statement.

3:34

If I've seen it before I know where it is because index tells me. So what I'm going to do is find the value that's in myFreqs, this is the number of times it's already occurred.

3:48

I can get this value using the get method and then I can set the value.

3:55

In myFreqs, to value +1. So, let me make clear what I've just done there. I've accessed the value at the location specified by index, which was returned to me by the dot index of method. And the idea here is that if the word the, occurs in location 500 of my words, the frequency, or number of times that occurs is in location 500 of myFreqs. They match up exactly. This will, I hope, compile.

4:28

Cannot find myFreqs.

4:31

So, I've looked up here and I can see myFreqs, myFreqs myFreqs, that's because I forgot to say set, the name of the method to set the value is set.

4:43

Now I'm ready to run it.

4:45

But I haven't printed any values. After I've printed them, so after I've tested them all and found them I will literally print every single word and how many times it occurs. So I'm gonna use the standard four loop that loops over array values or array list values.

5:03

Size is the method that tells me how many there are.

5:06

And I'm going to print The number of occurrences which is in myFreqs, a tap character and then the word itself.

5:26

That compiled I'll create a new one by right clicking and creating a new object.

5:34

And then I'm going to run the tester method on this object by right clicking.

5:42

There's the right click, run tester. Let's count romeo. And you can see how many times different words have occurred in Romeo. For example, the has occurred 677 times, Romeo 48 and Juliet 23. Another word that occurs often is Juliet with a period. I haven't taken into account punctuation at all. And from occurs 86 times. Let's make sure that part of the code is clear.

6:11

I've accessed the value in myFreqs and stored it in the Int value. And then I've set the value at index to value +1 because array lists use integer rather than int, this is a two step process. Find the value and then store the value. That's what we need to do with integer which would be a little bit different than we had with int. We'll see that in a later video. Let's look down here. MyFreqs.get(k) returns the kth value of the frequency array list. And myWords.get(k) returns the corresponding word. The 23rd frequency is the number of times the 23rd word has occurred. Now that you know a little more about array lists, let's write some code.

Downloads

Lecture Video

mp4

Subtitles (English)

WebVTT

Transcript (English)

txt

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