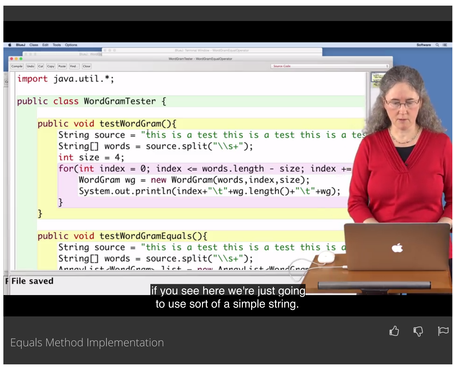
Generating Random Text

Word N-Grams

Review



Equals Method Implementation

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

Interactive Transcript

Search Transcript

English

0:03
We're going to see where it might be useful to write your own equals operator. So we're going to be at word gram tester which I have here, and I have one method called test word gram and and If you see here we're just going to use sort of a simple string. We have this string here that just says, this is a test, this is a test, this is a test of words. So we've got some repetition in there. And what we're going to do with that long string is we're going to create some word grams of size four, and then we're just going to print them out. So I'm going to just run this.

0:46
So I'll create my object and run test word gram and you can see we've got several word grams. They're all four letter words. And you can also notice that if we want to index zero, the first one, this is a test, is the same as the one at index four and also at index eight.

1:14
So, that's just printing out the word gram. So, if we go back to the code You can see that, we go through and we just print out the index position, the links to the word gram and the word gram. How do we do that? lets go look at the word gram class and you'll see in there it has a link method. That's how it knows how to print out links and for all of those it was four, because there's four words, and it also has a two string method, so it knows how to print out each word gram, and essentially what it's doing, is it's going through and just printing out all the words in the word gram by building the string and printing out the string.

2:02
So now, I've also got, we want to focus on equals, so I've also got this other method, here, called test word gram equals, and let's see what that does.

2:15
Again, we're generating word grams from the same string that we had before.

2:21
And this time when we generate them, we're actually creating an array list of type word gram and we're storing them in there instead of printing them, so you can see that in the first for loop,

2:35
then what we're doing is we get the first word gram, and we're going to compare it to all the other word grams to see if any of them are equal. If you remember, the first one at index zero and the one at index four and index eight were all the same. This is a test.

2:54
So let's see what happens if we just run this. So, I'm sorry, compiled,

3:07
So this time we're running test word gram equals.

3:12
Oh and you can see it only matched the first one. And now and if you remember the very first one we're getting here in our code. And then we're comparing it with the first one and then all the others. So it only matches the ones that are exactly the same, they're actually the same object.

3:29
But it's not matching the other ones that have the same words in them. So let's see what we have here.

3:38
So we're going to try and fix this so here again, here's our code, you can see, we're getting the first one, and then we're printing out checking, and then we go through the loop, starting at zero, so we are checking the first one with itself, and then

3:55
we compare first double equals with list.get(k), and we only found one match even though we should have found three matches. So what we're going to do, let's first try to change this to something else. Instead let's try using the .equals operator. So if we say first.equals.

4:27
Let's see if that works. So instead of using the double equals, we're going to use the dot equals. So we'll compile it, let's see if that works better.

4:43
We run it and we still only get one match, the first match, because they're the same object. So what we'd like to do is we'd like to write our own equals operator.

4:54
So we're going to go back to the word gram class over here.

5:00
And we're going to add a new method.

5:07
Equals operator is going to be a return type boolean.

5:16
We're going to have to pass in object o.

5:28
And we're going to create a word gram, calling it other.

5:39
And essentially what we're doing is we will convert the object that we pass in, so that it lets it know it's a word gram. So we will cast word gram oh, now I'm just going to make sure this works, so I'm just going to have it return true.

6:02
And we'll make sure this works before we finish writing the equals method.

6:09
Because all this should do, is it should always return true. So let's see what happens. We should get all the matches. Yes okay so we compile it, and we come over here and.

6:27
So now if we run it, we get everything's a match because we just always return true. So that's not quite what we wanted, but at least we know that we are getting that equals operator.

6:39
Because before when we ran it without that equals operator in there, we only got one match. So we know the code we just put in there is forcing it to at least run the equals method that we just wrote. Alright, so now we need to develop it a little bit further.

7:01
Okay, so the first thing we should probably do with word grams is make sure that if we're comparing two word grams they have the same number of words in them. So let's do that first.

7:13
So we'll check the length of the object, this if that is not equal to

7:30
the other.length. So again, we're passing in some word other. And we want to compare it to the actual object that it's comparing to. If the links are different, hey, we know right away they're not equal. So at that point, we want to return false.

7:54
Now if they are the same, then what we need to do is we need to go through and compare word by word to make sure the first words in each are equal, the second words in each are equal, and so on. So we need a loop for that.

8:08
So we're going to have to create a for loop.

8:18
And then my words dot length is the length of a n gram.

8:38
Okay, so we'll iterate through all the words in my words.

8:43
And we know we have an index position for each of those.

8:47
And then what we want to check is we want to check and see, we compare the first two words, if they don't match, we know right away the n grams, the word grams aren't equal.

8:58
So we'll say if it's not true, that my words, k, in the k's position,

9:12
dot equals. Here we're using the dot equals for two strings.

9:18
Then we want to know the corresponding word in other, which is going to be at other.wordAt(k).

9:34
So if that is not true, that means that those two words do not match.

9:40
Then we know again that's false. And we can just return right away, without checking in any of the other words.

9:46
So we'll just say return false again. And if we manage to get through checking all the words, and we don't find any of them false, that means they all match, then we'll get to the bottom of our method, and return true.

10:00
So let's try this out.

10:03
Okay. So now, we got it compiling and let's see what happens when we run it.

10:21
And it worked, because it found the three matches that are identical at slot zero, four, and eight. So here's a case where sometimes you need to write your own equal operator and we wrote it, and we saw that it worked, and that's pretty exciting. Happy programming!

Downloads

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

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