

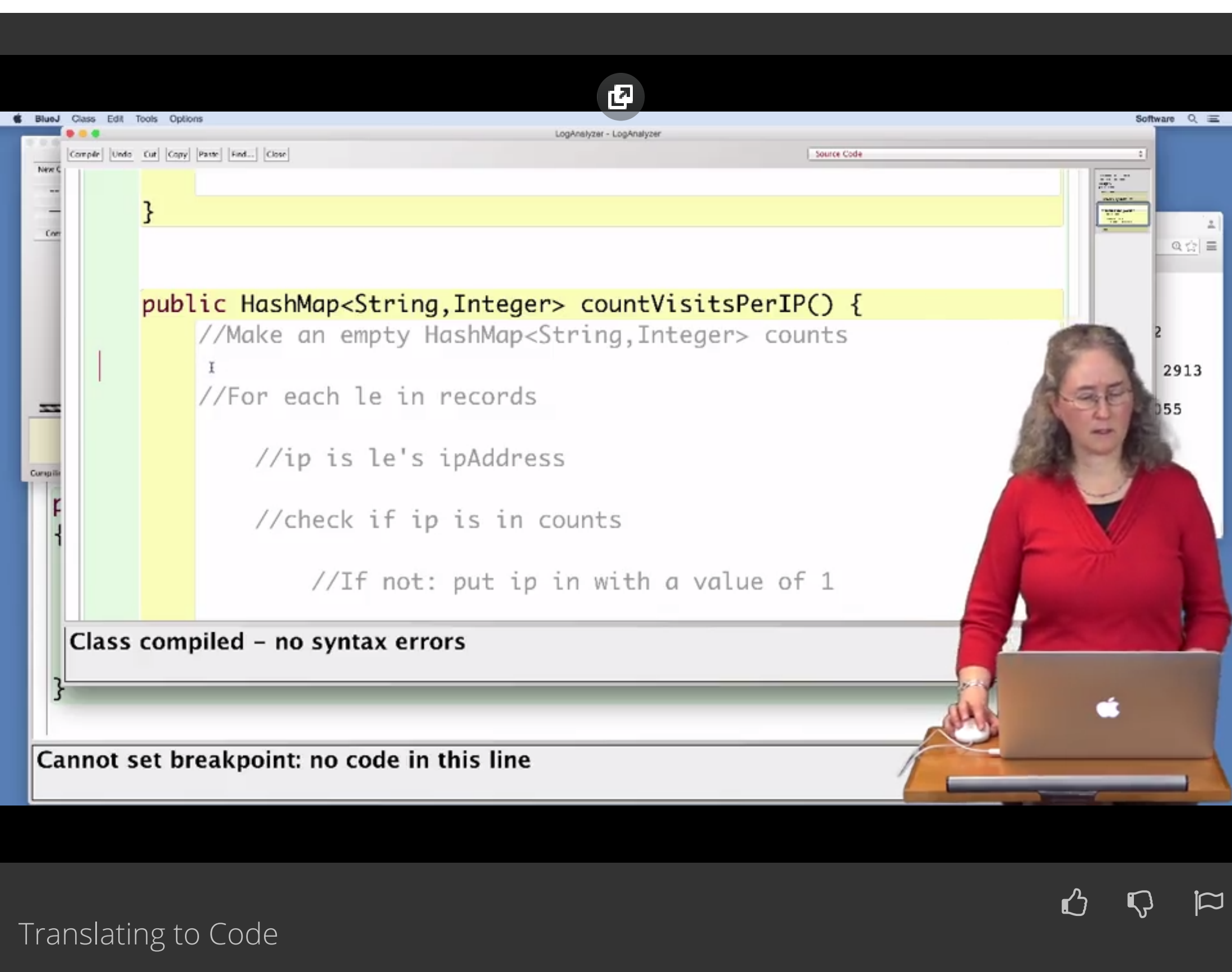
Reading Log Files

Finding Unique IP Addresses

Counting Website Visits

Introduction	1 min
Developing an Algorithm	4 min
Translating to Code	11 min
HashMap for Unique IPs	1 min
Summary	50 sec
Programming Exercise: Counting Website Visits	10 min
Practice Quiz: Counting Website Visits	4 questions

Review



Translating to Code

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

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

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

Interactive Transcript

Search Transcript

English

0:03 We'd like to find out how many times somebody visits a web site. So we're gonna look at a log file that has IP addresses and see, for example, for one IP address, how many times did that IP address appear in the file? That's how many times that person visited the web site. So we have a program here, a class called LogAnalyzer. And we are going to write the method count visits per IP. So how many times does each IP address visit a page? And what we have in here is we're going to put log entries in an ArrayList. It's called records. So we have a constructor that initializes that ArrayList. We then have readFile, which is gonna have the file name of the log entries. And that is going to allow you to select a log entry file. And then it just goes through and reads through all the lines and puts them into records.

1:07 So we are gonna focus on writing countVisitsPerIP and we are gonna use a HashMap to do that.

1:15 So the first thing we are gonna do is we need to make an empty HashMap. And we're going to be mapping a string to an integer. So for each IP address, which is a string, we're gonna map that to the account, which is the number of times that IP address appears in the file. [So first let's make, let's see.](#)

1:43 A HashMap of type String, which is the type of the key, and of type Integer, which is the type of the value for the HashMap.

1:57 We need to give it a name, so we're gonna call it counts and then we have to create a new one. So a new HashMap of type String for the key and Integer for the value. There we go.

2:21 Now that we have it, we want to iterate over all the records that we have, so we're gonna use a for loop for Each log entry.

2:43 And we need a variable name so I'm just gonna call that le.

2:48 We're going to iterate over the records where we've put all the records from the file.

2:57 Okay.

3:13 All right, there's our for loop. And now we're going to look at each LogEntry, one at a time.

3:21 So, first what we'll do from the LogEntry is we'll get the IP address.

3:26 Okay, so we need a variable for that, and that is gonna be of type String. So we'll call that variable, say, IP, which is of type String. And we will use the log entry to get the IP address. So if you remember we have getIpAddress.

3:50 And that should get us the IP address.

3:55 Now that we have the IP address, we have to see if it's already in our HashMap or not, so we'll have to ask a question about that.

4:08 So, we're going to ask if.

4:17 Counts.containsKey, okay. So again, counts is our HashMap and we're using the containsKey function, and we will ask if that IP address is in there or not. And I am actually going to ask first if it's not in there. So we'll put the not there.

4:50 And if it's not in there, then that means we want to put it in there for the first time. When we put it in there for the first time, its count is gonna be 1. So we'll add code for that.

5:01 So we get the name of the HashMap which is counts. And we'll use the put. And we need to put in the IP address, which is the variable IP. And we put in 1 for the count 1.

5:20 Now if it's not in there.

5:25 Then we'll want to. Sorry, if it is in there, then we'll know it's already in there. We need to get the value out. We need to add 1 to it and we need to put it back in. So we'll do that now.

5:44 So essentially we're gonna use counts.put but what we're gonna put in there is we're gonna put the IP address in there again, or basically replace it with, we have to get the old value out and add 1 to it. So we'll have to use counts.get to get the old value of it.

6:06 And then we'll have to add 1 to that.

6:16 And then what else do we have to do? We come down here. So if we get to the end of that and we've put all of our log entries into our HashMap and essentially they're each in there once, with the count of how many times they appear in the file, then we can return the answer, which is just counts.

6:39 So we'll compile this and, let's see, we gotta spell getIpAddress correctly, so we'll fix that.

6:52 And we've compiled it with no syntax errors now.

6:55 All right, so now we wanna test this out. Again, let's just see, countVisitsPerIP is gonna return the HashMap that is all the mappings of the IP addresses to their counts, so you get the whole thing. We need to test it out now. So in order to test it out, we're going to create another class here, which I've started, called CountTester.

7:21 And what we'll do in here is we'll first create a LogAnalyzer, the class that we just had, so, we'll create a LogAnalyzer object. We'll just call it la. And we have to create a new one.

7:43 Okay. So, now we have a LogAnalyzer object. We'll need to pick a file to read from. So I'm going to use the readFile and we'll put in. I happen to have a very small test file to make sure and convince myself that this actually works. So it is called short-test\_log.

8:10 And I'll show you that in a minute, and then I want to now call count visits per IP. And remember that's gonna return a HashMap, so I need to have a HashMap variable to put it in. So I'll do that.

8:24 HashMap of type String Integer.

8:36 I'm gonna call it counts and now that is gonna be assigned the value returned by our LogAnalyzer which is called la and that's gonna call the method we just wrote which is countVisitsPerIP.

8:55 And then once we have that, we can just print out the HashMap that we created. So I'll just have system.out.println on counts.

9:13 Now, let's see if this compiles.

9:17 And we're missing something. Let's see.

9:21 We need to use. We used the wrong kind of quotes. So, we'll put double quotes here.

9:30 And we'll try compiling it again. And we forgot the semicolon at the end. All right good, we're good. So that works, and I want to show you over here my simple file and then we'll run it. So I have this very short test file, and you can see it's got one, two, three, four, seven log entries in there. You can see the second entry which is 152.3.135.44 is in there three times. Now we're gonna run it. So we'll come over to BlueJ. Everything's compiled. We'll run count tester.

10:10 We gotta create the new object first.

10:15 And we'll run, and there we go. And you can see, if I can put both of these up here at the same time. There we go with one.

10:25 So we created a HashMap, and you can see that the one that starts with 152.3 is listed as being there three times. That is, we print out the HashMap and it prints out each IP address equals its count that we came up with. And so the second entry here is 152.3.135.44 with a 3 and you can see the 157.55 one is only in there one time and we got 1 and you can see the 152.3.135 is in there twice. And you can see, we counted it twice. So now I'm more convinced that what we wrote is correct. All right, happy coding.