

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

Back to Week 3

Lessons

Prev

Next

Which Countries Export...?

Weather CSV Problem

Hottest Day in a Year: Comma Separated Values

2 min

Converting Strings to Numbers

4 min

Maximum Temperature: Developing an Algorithm

5 min

Java for Nothing—null: When You Don't Have an Object

4 min

Maximum Temperature: Translating into Code

4 min

Maximum Temperature: Testing Code

3 min

Maximum Temperature from Multiple Datasets

5 min

Maximum Temperature Refactored

4 min

CSVMax: Summary

41 sec

Programming Exercise: Parsing Weather Data

10 min

Practice Quiz: Weather Data

9 questions

But instead of this one being the largest, this one is only gonna be the current

Maximum Temperature from Multiple Datasets

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

Interactive Transcript

Search Transcript

English

0:03
Now that we can find the hottest temperature on a single day, let's look over our range of days and find the hottest temperature over that range.

0:11
For this I've created a new method called hottestInManyDays that we'll use to do this calculation. And we'll be using a directory resource for this which will allow us to select any number of files over ones to compare.

0:31
As we've done for many examples, we'll create a File, and that will be what the DirectoryResource returns when we say selectedFiles, and we'll iterate over that. And now that we have that file, we'll use that to create a new FileResource.

1:03
And now that we have that file resource, we can actually call the hottest hour in file method that we created earlier with a CSV parser, just like we did in our test actually. And so this code right here that we've already got in our test is exactly what we're gonna do, but now we're gonna do it inside of a loop so that we can do it as many times as needed.

1:28
But instead of this one being the largest, this one is only gonna be the current temperature and we're gonna have to compare that against the largest so far. So just like in the previous example, we're gonna have to keep track of what we think is the largest one so far. So to do that we will create a CSVRecord

1:54
largestSoFar, I'll even use the same name, and initially it will be nothing, so I'll set that to null.

2:03
And then once I'm looking through the loop here, I'm gonna check to see if the largestSoFar is empty would

2:15
mean we haven't assigned it yet, then I will go ahead and re-assign it to the current one that we just got.

2:28
Otherwise I'm going to have to compare the two of them again. And, again, since this code is gonna be very, very similar, I'm actually going to go ahead and literally copy and paste it from my previous implementation.

2:50
And I'm gonna get the currentTemp, I call that one currentRow. I like currentRow. I'm gonna call this one currentRow as well, again so that we have a similarity of names. And so I'm gonna get the current temperature and save that as a double currentTemp. I'm gonna get the largest temperature out of largest so far. I'm gonna compare them. If it's greater I'm gonna go ahead and replace it, and then I'm gonna close my loop and so now I've iterated over many days. The main difference is I've called hottest hour in file instead of calling it for a particular row in that day, and once again largestSoFar is gonna be my answer.

3:42
So I'm gonna return that at the end.

3:46
I'm gonna compile my code just to make sure that I didn't make any silly mistakes, and it compiles, so now let's go on to testing.

3:54
For testing, once again I've created a test method, test hottestInManyDays. In this case I call the hottestInManyDays method, it takes no arguments, so there's nothing to pass, but it returns a CSVRecord largest for me to use. I've printed out that the hottest temperature was, and I've also printed out what day it was. I've changed the field that I get to date UTC from time EST, since it could be any day, and now I want some extra information. I want to know what day it actually occurred on. So I'm going to go ahead and compile this, and come over to the BlueJ environment. I'm going to create a new CSVMax object.

4:37
I'm gonna call the test hottest in many days, and I'm gonna go ahead and choose the first two days from 2015 just to test, because I've tried those two times before, so I know that the hottest day for January 1st was 51.1 and the hottest for January 2nd was 54. So I would expect that the answer is 54 on January 2nd, and sure enough that's what my answer is.

5:07
Now that I have some confidence that it's working, I'm gonna try it on a bigger data set. And so now I'm gonna test for many days and I'm gonna see what it was like for the whole of the year 2014, which is the last year for which we have complete data. And now I'm gonna go ahead and say open and apparently that was 98.1 occurring on July 8th at 10:51 PM. So now I feel somewhat confident that my code is working. I tried it on a small example, just two days, and then I scaled it up to try it on an entire year. Something that would be very difficult for me to test by myself, but that smaller data set gave me some confidence that my larger data set was working correctly.

Downloads

Lecture Video

mp4

Subtitles (English)

WebVTT

Transcript (English)

txt

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