

Introduction

Searching Earthquake Data

Module Learning Outcomes / Resources	10 min
Introduction	7 min
Relationships Between Classes	4 min
Licensing and APIs	3 min
Location class documentation and license resources	10 min
Coding a Magnitude Filter	10 min
Coding the Closest Quakes	12 min
Summary	2 min
Programming Exercise: Searching Earthquake Data	10 min
Practice Quiz: Searching Earthquake Data	6 questions

Filtering Data

Review

Filtering and Searching Quake Data

We're going to use Bluejay to gain some experience with filtering and

Coding a Magnitude Filter

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

Interactive Transcript

Search Transcript

English

0:02
Hi! [We're going to use Bluejay to gain some experience with filtering and](#) searching earthquake data. We'll start with the existing earthquake client class which reads and prints quake data. We can use this class to print CSV data from either the live USGS data feed, or from data we've recorded, so you can load it from a file on your computer, rather than over the internet.

0:28
Since the data is in CSV format, you could also copy and paste this output Into a spreadsheet and look at it there. This can help you check and debug the code you'll write. This live coding lesson will help you build experience reading and searching quake data, which will use the same techniques and ideas to search any kinda data. You'll be able to create new ideas for filtering data so that you can search the quake data to find patterns of simply finding where quakes occur.

0:59
You'll see code to filer quake data by the magnitude of a quake or by how far quakes are from a specific location. You'll also see that using live data can make it difficult to debug your code since the data may change. We've captured data you can use to help debug and you can do that as well. Let's start the coding now. First thing we're gonna do is just process the data using the create CSV method we've written. And we can either read data from a file, which is what we're gonna do first. Or we could read it from a link. So let's just go ahead and run createCSV.

1:41
So I'm gonna create an object, and run the createCSV method, and you'll see it's gonna run, and it's gonna find from our file, it's got 1,518 quakes and it's got information about the quakes.

2:03
So you could actually take that data, and cut and paste it, and put it into a spreadsheet because it is in CSV format. And then that would actually allow you to look at the data in another way, and kind of decide if your program is working correctly. So we're not gonna do that right now, but that's something you could do. So what we want to do now is go ahead and figure out, what are the big quakes from that file?

2:27
So we're gonna write a method called big quakes, which essentially is starting out the same way that we just started out the other one. We're going to read in all of those earthquakes from our data file, and then we got to figure out what the big quakes are.

2:48
So, let's do that.

2:53
So, as you can see what happens here in our code, is we have this line here, ArrayList, quake entry list. So what we are doing is we are reading with our parser and putting all of the earthquakes that we are reading into an array list of type quake entries. So we can iterate over those which is what I'm gonna do now.

3:22
So for quake entry we'll just call it QE.

3:37
So for each quake entry, we're gonna have to ask a question and find out how big that quake was.

3:45
So, we'll have an if statement. [SOUND] And we'll have to get the magnitude of the quake, so QE is our quake that's in our array list. And so we'll use the method, get magnitude.

4:07
And then we'll compare it. What I'm gonna do is just ask if the quake is bigger than 5.0, I want to know all those quakes that are bigger than 5.0.

4:22
And if it is, I'm gonna print it out.

4:29
So I'll just print the quake out.

4:35
And let's compile that and see if it works.

4:39
So it compiled no syntax errors. We'll just go ahead and run it. So again, we're gonna create a new object. And this time, we're gonna run the method bigQuakes and let's see what the output is.

4:56
Okay, so you can see all of these quakes here have magnitude greater than five.

5:12
Now another way we could do this is we could write a filter for this. Because what if we wanna get big quakes and we wanna do other things. So I'm gonna instead, I'm gonna do this a different way. So I'm gonna go up here and I've already started this method up here. It's called filter by magnitude and when I'm passing in our two parameters. I'm passing in an array list of our quake data, of quake entries, and I'm also passing in a number which is the size. I want all quakes bigger than that number. And then what I've done is, I'd like to create an array list of all such earthquakes.

5:51
So I've created a new array list here called answer, and what I want to do is go through the quake data that we're passing in that array list. And I want to figure out what are all those earthquakes bigger than magmen, which is our number that we're also passing it as a parameter. So we'll just add code here.

6:16
So again we're gonna create a for loop for each QuakeEntry From our parameter quakeData

6:38
We're gonna ask the same question.

6:42
If qe and we're gonna use the getMatnitude function.

6:48
And here we're gonna see if it's greater than our magMin, which is the parameter we're passing in, the number.

7:02
And if it is, we want an ArrayList of all such quakes.

7:13
So if we find a match, we're gonna add it to the ArrayList answer. So, answer .add(qe. And we'll just check and see if this compiles, and it does. Now in order to use this new method we wrote filtered by magnitude, we'll come back down here to big quakes. And I'm just gonna comment out this part that we wrote here cuz we're going to do it in a different way this time we're gonna actually use our methods. I'm just gonna put a big comment around this and instead we're gonna write, we're gonna call. The method we just wrote, and that filter will then return an array list of all of the quakes that have the large magnitude.

8:07
So we're gonna have to create an array list to receive the answer. So an array list, of type quake entry

8:23
So I'm gonna call this ArrayList listBig.

8:28
These are just the big quakes.

8:33
And that's gonna equal where I'm gonna call the method. So that method is called, filter by magnitude. Then we're gonna have to pass it, the array list of all the quakes, which is called list.

8:52
And we're also gonna have to pass it A number, so I'm just gonna pass it 5.0

9:08
>> So that will give us an array list and then what we'll have to do is once we get that array list then we can just print out the big earthquakes so I'll do that now. We'll create a for loop just to loop over that particular array list.

9:27
So, for each QuakeEntry qe that's in the listBig ArrayList,

9:41
What we'll do is print these out.

9:50
Okay so let's see if this compiles, and it does. So we'll go ahead and run this.

10:03
So we'll create our object, and we'll run bigQuakes.

10:08
And there it goes. So again we get all the quakes that are bigger than five and you can see that. Now notice we're taking advantage of two string which knows how to print out an earthquake. It prints it out in this nice format that's got the location, the magnitude, the depth, and the title. And so we're also taking advantage of using that two string. Just wanted to point that out.

Downloads

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

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