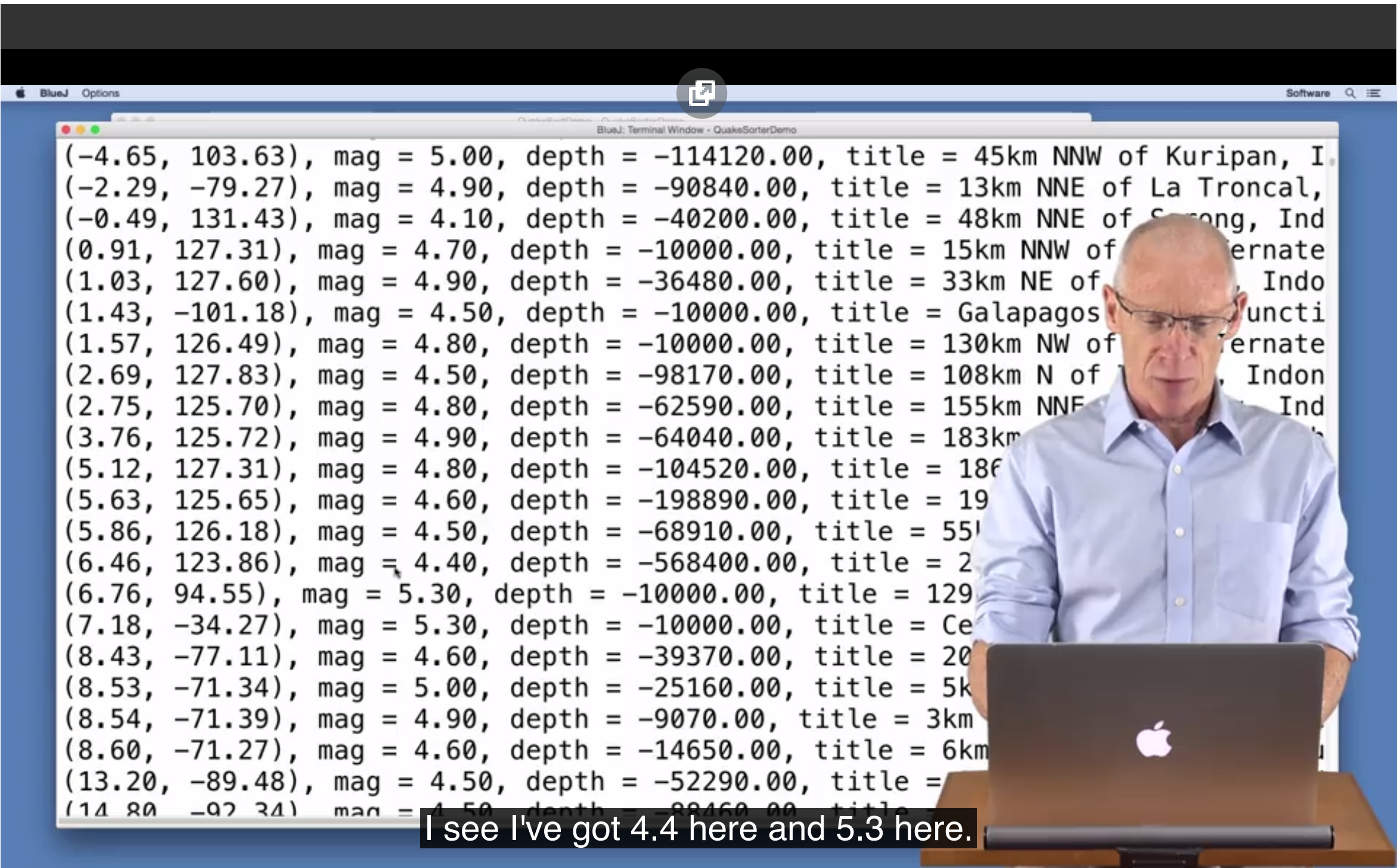


Implementing Selection Sort

Sorting at Scale

Introduction	6 min
Comparable	5 min
Ordering Quakes by Magnitude	8 min
Comparator	5 min
Comparator for Distance from a Location	6 min
Summary	1 min
Programming Exercise: Sorting at Scale	10 min
Practice Quiz: Sorting at Scale	3 questions

Review



I see I've got 4.4 here and 5.3 here.

Ordering Quakes by Magnitude

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

Interactive Transcript

Search Transcript

English

0:03  
Hi. We're going to sort some earthquake data so we can have a better understanding of what the quake data means. Are there powerful earthquakes? Less powerful earthquakes? Are there earthquakes close to where we are, and more. So I've got QuakeSorter Demo which is a very simple program. I'm creating a parser, I'm opening up the data file that we've already saved, I'm gonna read that data file and fill the list with Quake entries, I'm gonna call it collections.sort which is a very fast built-in sorter In the java.util package, and then I'm gonna print out the entries. So, I've got my QuakeSortDemo. I'm going to create a new object.

0:46  
And then I'm going to run the test sort.

0:51  
Now that's a lot of earthquake data. And as I look through, I was hoping that we would have the earthquake data, so that I could see the powerful quakes that had a high magnitude and then the less powerful quakes earlier, but, as I scroll through here, [I see I've got 4.4 here and 5.3 here](#). This quake data is kind of all over the map and now I realize that looking over here it's the latitudes of the quakes that are increasing. So I've sorted from the South Pole, which is a very low latitude, all the way to the quakes that might be near the North Pole. And we can see that, for example, that at this quake whose latitude is 66, they're all up near Alaska. So this is not the way I want to sort. I want to sort based on magnitude. So I'm going to have to go into the Quake entry class, because as you know, the objects being sorted implement the comparable interface. So I can see here that Quake entry implements comparable.

1:53  
That's right here.

1:56  
I'm gonna look at that compare to method, and I can see that this quake entry is sorting based on the latitude and the longitude. And that's not what I want. What I want to be able to do is sort by magnitude. Now I'm gonna have to modify the QuakeEntry class, which means I'm going to lose the functionality it had here. So I should make a copy of this code to make sure that I don't lose this sort by latitude functionality. And in the next lesson we'll see that we can sort by many different ways at once, but not using the techniques I have here, which is simply changing compared to, so I'm going to comment out all of this code which as a reminder returns a -1 or a 1 or a 0. So I'm commenting that code out completely and I'm going to look at my magnitude and the loc's magnitude and if.

2:52  
And I can come up here to remind myself my magnitude is an instance field. So if my magnitude is less than the loc.getMagnitude,

3:12  
I'm gonna return a negative one. Because the sort.compareTo method, that interface has a requirement that when one object is less than another it returns a value less than zero. And now I'm going to have to say if my magnitude is greater than the magnitude of the other object, I'm gonna return positive one because I'm greater and that's what the contract calls for.

3:46  
And we can see using laws of mathematics that if it's not less than and it's not greater than, it must be equal to. So I'll simply return zero in that case, meaning they must be equal. I don't have to check because I know if it was less than I would have returned negative one, if it's positive I'm gonna return positive one if it's greater than, so cannot find symbol getMagnitude. So I can look in here and see it says getMagnitude right there, and the problem is that this is a loc, and it says getMagnitude without the parentheses. So I better make sure I call that method, not getMagnitude by itself, but getMagnitude with parentheses, that's an easy thing to forget.

4:29  
Class compiled, no syntax errors.

4:33  
I'm going to take my Quake Sort demo, compile it, make an object and invoke that sorter.

4:47  
As these have gone scrolling by I can see this gigantic earthquake of 7.0 that occurred in the Solomon Islands is the last one I see. And the early quakes all have teeny tiny magnitudes. Negative zero zero zero. These are all very small quakes. So the quake sorter did work the way I wanted it to, by putting the smaller quakes first, and the larger quakes at the end, as a reminder what I had to do to make that happen was change the compare to method. So that, if my magnitude is less than the others, I return -1. If my magnitude is greater than the others, I return positive 1. Otherwise, I returned zero. As it turns out, because these are all doubles, I can actually shorten this code by saying let's return double.compareTo my magnitude and the other magnitude.

5:52  
So I'm going to remove this code by deleting it.

6:05  
See if that compiles.

6:09  
And the java.lang.Double cannot be .compareTo so apparently I've named this one incorrectly which I often do.

6:20  
So I will use my handy replacement function in Bluejay to find the name of the method that I want.

6:31  
And because I'm failing to do that. There we go. Double dot compare, not compare to. So I'm comparing these two.

6:41  
I just did not know the name of the right method to call.

6:45  
I didn't read the documentation as I should of. So now I've compiled quake sorter demo.

6:53  
And I need to make sure I get that one right.

7:05  
Make a new object.

7:10  
Run it. And I still get the magnitudes as they're supposed to be. So just as a reminder, I rewrote that code relying on code that already existed, which is a good thing to do in general, rather than reinventing the wheel. Sometimes you forget what's already done for you. That's it, sorted by magnitude smallest to biggest. Let's write some more code later.

Downloads

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

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