

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

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

Interactive Transcript

Search Transcript

English

0:03

Hello, we're going to look at code and concepts to investigate real time earthquake data. Real time means that your program will be able to access data recorded seconds or [minutes before your program is run](#). Your program will also be able to read a file of recorded data, which will make it easier to determine if the program is correct as you develop and test the code. These programs will illustrate how to design classes that use other classes. And eventually, how interfaces, a new Java concept are used. These programming ideas will be used to study searching and sorting. Fundamental concepts in computer science and programming that help in understanding data and quickly accessing the data you want.

0:47

You'll also see how to use APIs or application programming interfaces to parse and transform data, formatted using industry standards.

0:58

These programs will be the basis of an exciting set of projects, but will also be the basis of possible further explorations in software engineering and computer science.

1:09

Our Capstone Project will use the same concepts. The second specialization in Java programming designed by our friends from UCSD and available now on the Coursera platform uses the same earthquake data as the focus of their first course. We hope you'll be intrigued and curious enough to explore these concepts and these datas on your own. We recorded real-time earthquake data on November 11, 2015 with the program you'll study in this lesson. We use the earthquake data to create a Google map, whose URL you see here. This snapshot shows earthquakes of low magnitude in California in the United States. The key you see shows that a yellow star is an earthquake, whose magnitude is less than 1.5. Quakes who's magnitude is less than 2.0 are typically considered microearthquakes, you can see that there are a lot of these in California. We can also use the data from this program to show quakes in all parts of the world. Here is a map showing quakes of greater magnitude in areas near Southeast Asia, Japan and Indonesia. The map makes it easier to see clusters of quakes. We created these maps by transforming the real time data into a CSV file, where CVS stands for comma separated values. Data in this format can be loaded directly into a Google map. You'll see how to transform data and how to use it as you develop the Java programs in this lesson. The data is obtained from real time data feeds from the United States Geological Survey website at [earthquake.usgs.gov](#). The data downloads in an XML format, a standard, widely used format for structured data. XML stands for Extensible Markup Language. JSON is another standard format that is increasingly used for transmitting and storing data, as well.

3:13

JSON stands for JavaScript Object Notation, though the format is widely used in all programming languages. These are widespread standards, but parsing data in these formats can be somewhat tricky or tedious. As you can see here, the latitude and longitude data are formatted between tags that specify a particular kind of data. All modern programming languages provide an API for parsing formatted data. But as part of this course, we will provide an API that hides some of the details to help make parsing the data more in line with our use of collections and iterables.

3:52

The programs you use will be able to parse and transform the data into other formats. For example, we need a CSV format for creating a Google Map.

4:02

Other mapping services use CSV data, as well.

4:06

As you can see, the latitude and longitude are stored as simple values separated by commas from other data about an individual earthquake.

4:15

Transforming data from XML format to CSV format is relatively straightforward, because the parser creates a Java object that helps with the programming.

4:26

Parsing and Transforming Data is a common application that helps with solving problems and interpreting data. The code to transform XML to CSV is shown here. Parsing is not always simple, so we have created an API to make it easier for you to write programs to process earthquake data. The API creates a collection of earthquake entry objects where QuakeEntry here is a Java class.

4:55

Here you can see an EarthQuakeParser object being created to parse XML data. You'll be able to look at the parsing code if you want, which uses standard API to parse the XML.

5:09

The API we've designed can use the data that comes from a file or a URL. This will facilitate testing the program on the same data stored in a file or to use real-time data that might be different every time the program runs.

5:24

Here you see the source of the data being specific as a URL. This is the real-time data feed from the USGS quake data.

5:33

The parser uses the datasource, reads the XML and returns an array list of QuakeEntry objects. Let's look briefly at what a QuakeEntry is. There are two classes you'll use, QuakeEntry and location. The QuakeEntry class holds the data that's relevant to an earthquake and that comes from the USGS XML feed. As you can see, there are four instance fields we'll use in understanding earthquake data. The field myLocation is a location object, which will help as you write code to find where quakes are close to each other or to you.

6:13

The magnitude of the earthquake is a measure of how strong or forceful it is. The depth of an earthquake determines how far underground or under the ocean a quake is located and all quakes from the USGS feed have a description related to the location of the quake.

6:30

Location is a separate class. We could have used double values to store the latitude and longitude directly, but you'll see why a separate class is useful later.

6:40

One class often uses another or contains another, as the QuakeEntry class contains a location object. The location object was adapted from the Android standard libraries. Android code is based on Java and is the software that runs more smartphones than any other software in the world. Phones typically can help you find directions, because they know where you are based on GPS sensors. The Android location class is a robust and well-tested class you'll be able to use in your programs.

Earthquake Data

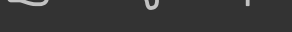
- Using earthquake data as basis for study



Duke UNIVERSITY

minutes before your program is run.

Introduction



Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

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