

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

Back to Week 4

Lessons

Prev

Next

Using Java Beyond BlueJ

Mainstream Java

Introduction

1 min

Understanding Exceptions

4 min

Handling Exceptions

2 min

Declaring Exceptions

1 min

Throwing Exceptions

1 min

Reading Files with Java.nio

6 min

Hello Around the World with Eclipse

4 min

Summary

45 sec

Where To Go From Here

10 min

Review

Exceptions

Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 82  
at java.lang.String.charAt(String.java:646)  
at Test.doStuff(Test.java:5)  
at Test.main(Test.java:8)

• Exceptions: you’ve probably seen them...

• But what do they mean?

• How do you use them?

Duke UNIVERSITY

But what exactly are exceptions?

Understanding Exceptions

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

>

Interactive Transcript

Search Transcript

English

0:03

Hi. You've probably seen exceptions and know that they mean your program crashes because something went wrong. [But what exactly are exceptions?](#) And why does your program crash in this particular way? How could you use them in your own programs?

0:20

An exception means that the program could not do something that it tried to do. Sometimes this indicates a mistake in the code such as when you access an invalid index in a string or an a ray. However, sometimes an exception indicates a circumstance beyond the programmer's control. The user asked your program to open a file that does not exist, or their internet connection went down while your program was trying to download information. Real programs have to deal with such problematic situations and exceptions are actually one of the nicer ways to handle these types of problems. This exception message which is likely similar to something you've seen before, is called a stack trace. When your program has an unhandled exception, one that indicates something went wrong and your program did not say how to deal with it, your program crashes and prints a stack trace If you look at this in detail it starts by telling you there was an exception. Something went wrong.

1:24

Then it says in thread main. We're not gonna talk about threads which are a fairly advanced concept. And unless you start using them you can ignore these few words of the message.

1:36

Next, a message tells you what kind of exception happened. That is, what sort of problem the program had? In this example, I made a mistake in my program and accessed a string at an invalid index. So, I got a string index out of bounds exception. Many exception types are nicely descriptive of the problem. But you can always check the documentation for more information.

2:01

Next the message provides some more information about the problem. Here, it tells me what the invalid index was. I tried to access element 82 which was not legal for that string. The information here will depend on the type of exception you have,. Lastly, the message has the stat trace. That is the list of methods which had been called but not yet returned when the problem occurred. Here the problem happened inside of strings.charAt method. Specifically on line 646 ofsString.java in the java library.

2:39

The next line shows what method which was do stuff and test that Java on line five. This is the first method that we see that is in my code, rather than a library, which is generally where you would want to start looking for the problem. Last, it shows where main code do stuff on line 8 of tester Java we just said that your program crashes and prints a stack trace if it does not handle the exception. Your program can, however, handle an exception, specifying what to do to deal with the problem.

3:13

Handling exceptions involves three new concepts, try, catch and finally, which you will learn more about shortly. After you learn about try, catch and finally, you will learn about exception propagation, where a method that does not know how to deal with an exception will propagate the exception to its caller. The caller can then either handle the exception or propagate it to its caller and so on. You do not have to do anything explicit to propagate the exception which is one of the key benefits of exception based error handling.

3:47

But you do sometimes need to declare that the method might throw an exception. Finally, you will learn about throwing your own exceptions, when your code discovers that a problematic situation has happened. Thank you.

Downloads

Lecture Video

mp4

Subtitles (English)

WebVTT

Transcript (English)

txt

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