CS 240: Creating APIs with Lambdas Transcript

Start slide description.

Jerod Wilkerson presents a recorded lecture with the aid of a slide deck. All visual content is described in the audio.

End slide description.

[00:00:01] **JEROD WILKERSON:** Lambda expressions allow us to create powerful APIs that would be difficult to create without them, and so I want to show you an example of that. So, here I have a class called "String Selector example," which has one static method in it, has a select method, and that method can take a collection of strings, so some number of strings and a predicate function.

[00:00:25] So, remember from my previous video, a predicate function is a function that if you pass it a string, it'll return true or false, so that allows me to create a method that could return a subset of strings.

[00:00:39] So, in this method, the select method takes a collection of strings, and whatever function I give it and it will apply that function to determine which strings it should return.

[00:00:51] So, it starts by creating an array list to hold the strings that it's going to return, and then it iterates through this list of strings and it calls the the predicate function, it calls this selector.test. Remember, predicates have a test method, so it calls the test method passing each string one at a time.

[00:01:09] And for each case where that predicate returns true, that's a string I want to have returned, so we'll add that string to the array list. Then at the end it just returns those strings.

[00:01:20] So that provides an API for me. Now, any time I need to select some set of strings that meet some criteria, I can just call this method parser criteria in the

lambda function and I'll get the strings that match that criteria, so it makes a very reusable function. So, here's an example of how I could use that.

[00:01:39] So, if I call that with a bunch of lines in this example, I'm calling it with lines from Robert Frost poem "The Road Not Taken," which you likely read that in an English class somewhere.

[00:01:52] So, if I call. If I have a string variable or I'm sorry, as a strings collection that has all those lines from the poem and I call the select method with this predicate, X is the variable and I'm just returning true, that means my predicate returns true no matter what the string is.

[00:02:14] That would say select all the strings. So, that'll give me all of the lines of the poem. And then this code on the right is the code I showed you in the previous slide. This is just the select method using the selector.

[00:02:25] So, if I call it this way with this lambda expression, I get all the lines, but I can call it with different lambda expressions and get different subsets of the poem.

[00:02:34] So, if I call it with this lambda expression where I say each string starts with "I" space, then I'll get all the lines that start with the word "I," so this will be the result. If I call it with "and," this will be the result.

[00:02:49] And if I call it with "X contains road," I will have, looks like, I have a little spacing problem here, but I'll have two lines, the two that start that have the word "road" in them. So, let's actually look at that in my code editor.

[00:03:05] So, I have basically the same example here. Here's the poem. So, I've created this list that has all of the lines of the poem in it. I have "my select" and "my select method," which is the exact same code that I've been showing you.

[00:03:21] And I have a “main method.” So, we’ll open up the main method. So, the first thing I’m going to do, I’m going to print a heading to show which lines of the poem should print out.

[00:03:36] So, the first thing I do is I printed heading for all the lines, and then I call the selector with this lambda expression that’s going to just select every line, so that gives me my list of strings, and then I just need to print them somehow, so I wrote a little method print strings that we can see there right here.

[00:03:56] It just iterates through the list of strings and prints them out and let’s see what’s the suffix doing? If I have something, I want to print at the end, like an extra new line or something, I could pass that in.

[00:04:08] Okay. So, I get all the lines back from this lambda because its lambda expression says just return true no matter what the string is. And so, then I call print strings, and they have a line separator that just allows me to separate the different sections.

[00:04:25] Maybe it makes more sense if I run this first and then we’ll go through the code or after that. Okay, so here I just ran the code, and I ran the main method, and you can see, the first I get all in the prints, the whole thing out.

[00:04:38] Then I get lines, starting with “I” in the prints, those lines starting with, “and” then it prints those, lines containing “road” and it prints those. And you can see what the code looks like here. It’s basically what was on the slides.

[00:04:53] I called the select method multiple times with different lambdas getting different results. So, that becomes a really useful API that I can use, and I can create lots of methods like this for use in whatever programs that I’m writing.