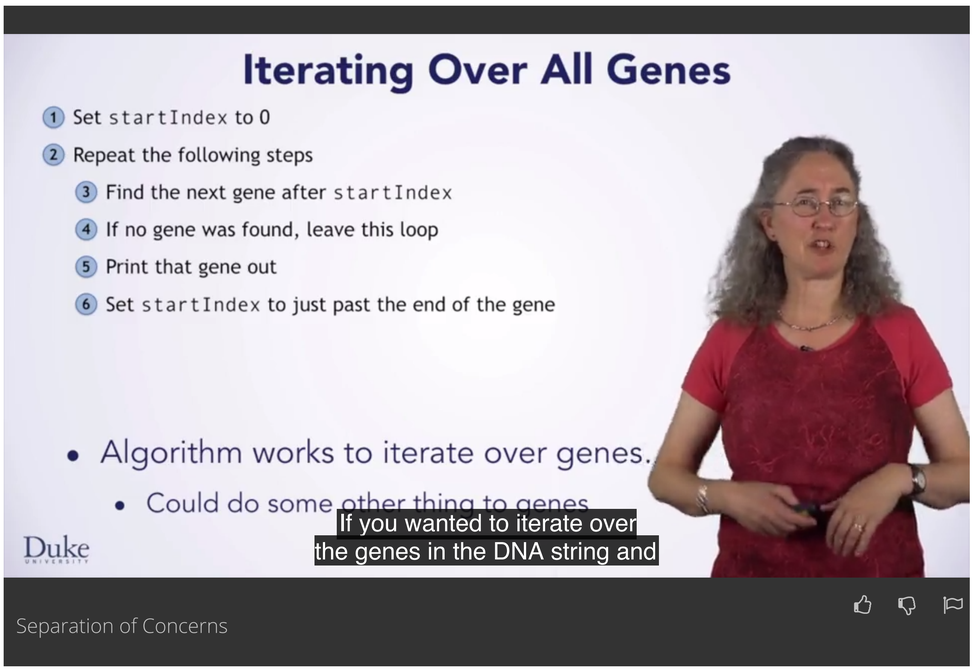

Separation of Concerns

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

Interactive Transcript

Search Transcript English

0:04
Now, you can iterate over all the genes in the DNA string and print them out. If you wanted to iterate over the genes in the DNA string and do something else to them, the algorithm would be pretty similar. In fact, for whatever you want to do with each gene, it would look pretty much like this. The line in blue is the only thing you would change to do whatever you might want to. Maybe printing only those genes that meet some condition, counting genes. Saving them to a file. Building a web page with all of them or anything else you can think of. So if you wanted to do these other things, you might copy the algorithm you have pasted. Paste it into a new method and edit that line to make small changes. This approach works, but it's generally a bad idea. Why? Well, for one thing, copy and paste is error prone. You might forget to change some things that you need to. Even worse, if you find a bug in your original implementation after you have made copies, you need to go fix every copy you made. It is also tedious. You have to go find the method, copy and paste it and change it. This may not be so bad, if you want one variation. But if you want to do five different things, then it's pretty boring.

1:25
And finally, it indicates bad programming design choices. Whenever you find yourself wanting to copy and paste, there's almost always a better approach.

1:35
Let's take a moment to see something that we could improve about this algorithm. Why it would make a lot of work if we leave it as is and copy, paste, change and then we'll understand the motivation for how to make to fix it.

1:51
This is our algorithm to print all the genes in a string.

1:55
We're going to condense it down to a small description at the bottom and then copy and paste, it and change line five to print only the genes with a high CG ratio, and then we'll condense that down to a short description. Now we're going to copy, paste and edit to make several other algorithms to do various things with the genes from our DNA string like printing genes in HTML.

2:24
Writing genes to an output file.

2:29
Counting genes with the codon CGA or whatever else you want to do. All these algorithms are the same, except for the details of what they do to the DNA string. So at first, copy and pasting does not seem like a big deal. Later, we end up with some other DNA data, which list all the genes in a file, one gene per line. We need to do the same sorts of operations on this data too or the algorithm will be slightly different. It will have a for each line in the file loop, then do the same operation for the genes. With our copy and paste approach, we now need to write and test six algorithms. They're pretty similar to each other, so it may not be so hard, but it's tedious error prone work.

3:15
Then if we end of with some other source of data, we're going to have to go make all six algorithms again for that data source. Likewise, if we end up with a new operation that we need to do. We're going to have to write three copies of it, one for each data source. Ick, what a mess. What we would really like to do is redesign out algorithm to use separation of concerns.

3:42
Our initial algorithm does two tasks. One is getting all the genes from some source of data and the other is printing them or whatever else we want to do to each of them. We would like to split these up. By having the algorithms that find the genes, put them into some structure that can hold a list of all of the genes. Then having the algorithms that print the genes, count the genes or whatever else we want to do to the genes. They should operate on that list.

4:12
Now if you need to add some new source of data, you just write a way to get its genes into our list and it automatically works with every processing algorithm you already wrote.

4:25
Likewise, if you need to write a new processing algorithm, it automatically works with every source of data that you already wrote. No copying and pasting is ever needed.

4:36
So, what is this thing that can hold all the genes your algorithms find? We're going to start by using a class from the edu.duke package called StorageResource, which is a simplified way of doing this. Later on, once you have learned a few more concepts, you will transition to using the standard Java.util.ArrayList class, which has similar functionality, but it's a lot more complex. 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?