



>> For our last example, let's look at TriangleListMenuApp. The main method displays a menu of options and then uses a do-while loop with a switch statement to take action based on the user's selection. The menu looks like this, and the options for the users selections are R, to read in a file and create a TriangleList.

P, to print the TriangleList. S, print smallest triangle, rather, smallest perimeter. L, print largest perimeter. T, print total of perimeters. A, add a triangle object. D, delete a triangle object. And Q, quit. So, let's take a look at this example. First, you need to navigate, be sure you're in the Examples folder for the, for this module.

And then you'll find the TriangleListMenuApp folder double click on it. And then in it you'll see the project folder. I'm opening the project folder for this example. And then here's TriangleListMenuApp. Here's the program. And you can look at the UML diagram to see the relationship between these. Notice it says this needs to be recompiled, so let's recompile it.

And so we've got TriangleListMenuApp. It relies on triangle list, which relies on triangle. So here's our the code that actually prints out the, the menu. And we do that ahead of our do loop. And then here we've got the do while, and inside it, a switch. And I'm gonna fold up, fold up the switch, so we can sort of see this in a compact way.

So here we see each case, essentially one for each of the menu items there. So read in the file, print the TriangleList, and so on there. All the way down to default, which if there wasn't a match it just breaks and, and, as does Q. Q breaks out of loop as well, and it goes down and checks to see if if we've got a Q or not.

If we don't have a Q in other words it didn't match, that would be the default case. It would go back, and and loop again. Give the user the the options there. And and go through, through again. So, I'm just gonna run this to give you an idea of what's happening here.

Let me move this up a little bit. I'm gonna hit Run. So here's our list of options and normally, the first thing you'd wanna do is actually read in the file. And so I'm gonna give it an R. And waiting on a file name, and let's give it triangle.dat.



I'm just copying it from the browse tab over here. And we get a message of file read in and TriangleList object created, and now we might want to print it. So I'll enter a P. And that just printed it. The name on the list was nice, small set of triangles and then there, this is the triangles.

It just went through the list, and and printed out each of the triangle there in the list. Might want to find that smallest parameter. So let's give it an S there is the smallest parameter that was in the group. Might wanna find the largest parameter, there is an L.

Let's find the toe parameters, that would be T. Notice each of these, you'll see inside the case statement, is essentially just calling the method in TriangleList that returns this. So next let's do an eight, add a triangle. In fact, let's do a print here to see what we've got there.

And I'm gonna add a triangle, I'll add a 1, 1, 1, that's a nice simple triangle, so let's do an add. So side 1 I'll give it 1, side 2 a 1, side 3 a 1, so the triangle is added, we could check that by printing them out again.

And there we see, 1, 1, 1 was added. And now I'm gonna delete a triangle, let's try that. I'm gonna delete 7, 7, 7. So there's side 1, 2 and 3. Says, it was deleted. And again if we print them, it should be gone and it is. And then finally, we can just quit.

You could actually go back and read in another file. In fact, let's do that. I think this is actually the same data. I'll read in another file. We could print it and so on. So, I think you get the idea of how all of this is going to work.

So I'm going to end this and let's look at a few of the methods that we've got here that were added. In the earlier version of this, we actually did the reading in the main program. But here we're getting the file name in the main program in this particular case of the switch.

But then we're gonna call the read file method that's, that's in the TriangleList file. In fact we'll look at that, here's TriangleList. This is the class and we've, we've actually just moved readFile into it. Where is it there? There it is, right there. And so, it's actually taking in a file name.



And notice that it can throw, this particular method can throw `FileNotFoundException`. Because it's actually doing the reading here and this exception we'll learn later, we'll propagate. And so, the main program actually must call that as, must throw that as well. So main also throws `FileNotFoundException`. So in, in this method here we, we simply open up the scanner on the file and start reading.

We, we read in the file name. That's the first thing in the file. And then we read in side 1, side 2, side 3, create a triangle, add it to the list. Go back and read three more sides. And add a triangle each time, at the end of all that, we create a new `TriangleList` with that list name we read in, and that array list that we, we put them in.

And, here's add a triangle, same sort of thing. Here we're going to actually, it adds a triangle based on the three sides. So that, that's, that's called when we add a triangle, we actually get the three sides. So let's look at adding a triangle. We have the user enter the three sides right here, and then after we get them we, we actually call `addTriangle` with those three sides and it gets added, okay?

To find though, let's look at finding the smallest parameter. So here we just actually call `smallest parameter` method on that list in this particular case. And if we look at `smallest parameter`, let's go there and find that. Let's see, `smallest parameter`, right here. This is gonna do a linear search, and, and this is where we're we set `smallest` to 0 and that shouldn't be in there, of course.

And then we can just looking for it. We set the `smallest` to the first triangle because that will be the `smallest` until we find another. And then we just loop through looking for a smaller one. So if we find that, that `smallest` here is not less than the next one in the list then we actually make that the new `smallest` and, and, and index the list.

So, we're using a while loop here and so on. `LargestPerimeter` same, same type of loop there except this time, we're looking for `largest` and we make the change if we find a new `largest`. So lot of detail here. I'm not gonna go through any more of it but you should take a look at this, this particular example pretty extensive but it should be a good one to study when you're doing your project.

And again the main thing I added here was the fact that we've got a menu, and then a do-while statement, and a switch. And in the switch we're looking at different items that the user enters to, to, to do the work here, to do one of the selected options here.