Transcript: TriangleListApp

>> Now let's consider a bit larger example called TriangleListApp, and its associated classes. Essentially, it's going to create a triangle list object from file data, and then print out the triangle list object followed by a summary of its data. Now here's a strategy, we're gonna read the file name from the user, as we've done before.

And then read the list name from the file, that'll be the first thing in the file. And then we'll use a loop to read the rest of the file, which, is sets of three, numbers. Each set of three defines a triangle. And we'll create a triangle object based on each set of three numbers, and add that triangle to an array list of triangles.

So after we've finished reading the file, then we're gonna create a triangle list object based on the name that we read in for the list, and also that array list. And we'll look at the details of triangle list here in just a minute. So after we've got that triangle list created, then we'll just print it, and then we'll print a summary of the triangle list.

So let's take a look at at this program. Now, in the examples folder where we've been finding all of these other examples, up at the top or in there some place, you should see a triangle list app. So, this is a separate folder, let's open it. And then inside you'll see a triangle list app the project file.

So I'm gonna click that. So here we have the project open down below, and it includes a triangle list app, triangle list, and triangle. Now triangle is, is essentially like our previous triangle, although it has a few modifications. In fact let's, let's take a, a quick look at it.

So you see it's got three sides, s1, s2, s3 for a triangle, and there's the constructor for it, the getters. Here's get classification, no change there. Here's perimeter, so we can call the perimeter method, and it returns the sum of the three sides, if you will. We had an equals method that determined if a Triangle t was equal to the triangle and hand, just by comparing the sides.

We've added another equals method that takes in an object, and this object actually might be a triangle. But it can be any other object as well, so in this method we say if object instance of triangle if that's not true. In other words if it's not an instance of triangle then we're gonna return false.

Otherwise we cast obj, that's the parameter that came in. We cast that to triangle, and then set that equal to Triangle t. And then we do the same do the same Boolean expression to see if the corresponding sides are equal. If, if they are, we return true, if they're not we return false.

So if we have an equals with an object parameter, we also should have a hash code method. And we're just gonna return 0 and and this will make Checkstyle happy as well. Checkstyle actually if you, if you override the equals method from for object here, then we need to actually add hash code.

So, with that being done, and then finally the two-string method is simplified. Here we're just printing out the size along with the classification. So, let's look at TriangleList. It's a little more complicated. And what this is now, this is a a, a separate class. So, so we're gonna have three classes in our program, one with main in it, one for TriangleList, and one for Triangle.

And, this particular one, now has two parameters not parameters, but fields. One is the string for the list name, which we talked about reading it. And, the other is an array list of triangle called t list, our triangle list, if you will. And so, those are the only two fields in this.

And there's a constructor, and it takes in a listName, and then also an ArrayList of Triangle. We call it, this tListIn, and then it assigns the, the name read in to listName, and then that ArrayList, a parameter, a T list and it assigns it to T list. It's essentially, the two fields up here.

So, we create a triangle list with the name of the list and, and an array list of triangles. Then we've got get list name. We've got something called num triangles, and this just returns the size of the list. Here's totalOfPerimeters, and this is totalOfPerimeters for the entire set of triangles in the list.

So we've got to have a loop, set index to 0, and total to 0. And then we go through the list, and get each triangle, get the parameter of it, and then add that to total. So each time we go through we're getting the pa- the new parameter of a different triangle and so on.

And when we get through with a list of triangles, we should have the total for the parameters, and then we return that. Now this is returning smallest parameter, this is you know, slightly different. If there's no triangles in the list, we're just gonna return 0. Otherwise we'll set smallest, and smallest is a double here.

We'll set it to the perimeter of the first triangle in the list. In other words, we're getting zero, that'll be the first triangle, if we do a T list type, I get zero. We're getting it's parameter, setting that to smallest, we set index to one. And then we loop through the loop, go through the while loop, as long as index is less than the size of our triangle list.

And we've got smallest and if it's not true that smallest is less than the next one that we're looking at, then that one that we're looking at, becomes the new smallest. If smallest really is, is still small, then we, we leave it that way. But we go through the loop, essentially looking for smallest, each time we find a new small one, it becomes our new smallest.

And then, when we finish the loop, we return smallest. Here is largest parameter with sides as a string, we're returning this as a string here a little bit different. We set index of largest to o index equals o here. And so as long as index is less than our size, we go through here and get, get the index, get index of largest meaning is parameter, and we check to see if it's less than the new triangle we're getting.

And if it is then that's the new index of largest. We incremented x and we continue through the loop, we only go through here, and assign index of largest if we found a new largest and so on. And then when we get done with the loop, we check to see if tList.size is greater than 0, in other words just if there was anything in it.

If there was something in it, then we get the index of largest, this would be our largest one, it is parameter And then say four sides. And then we get the sides of each one of that triangle. And we return that. Otherwise, we just say there were no triangles in the list.

And then here's the two string for for the list. Now this is the two, two string for the list, not an individual triangle. So we add a, plan feed, and we say in this name, and then new line, and then we go through the list and get each triangle in the list, beginning at 0, as long as we're less than the size there, and we just print that triangle.

Now, since we're concutinating that with line feeds, this is gonna call the two string for that triangle and that's what will get printed. And then we return that. And then summary, this is another method. Pulling this triangle list, dot Java summary of just, prints out the summary information for the list.

So we get the list name, summary four, whatever our list name is, and then we call those specialized methods. So we print out the number of triangles, the smallest perimeter, the largest perimeter in the total of parameters. And we return. So, as you can see, this, this, class here is a little more complicated.

And it uses triangle. But you know, you should take, take some time and go through here and make sure you understand each of these methods.. And then finally we've got a TrianglistApp this is actually quite simple compared to the others, it's it's the driver program if you will and, once we've got you know several pluses in a program which useful to see the UML diagram and.

And that's you can generate that by clicking the low EML symbol here. And this is what it looks like. So we see that we got triangle list app depends on triangle list which depends on triangle. And then triangle list steps also depends uses triangle. So this is how the three classes in our project are all related.

So I'm just gonna close that for now, and let's look at, let's concentrate on triangle list app. So here's TrianglelistApp, and again we're gonna read in this data from the file. So we've got our import statements there, as usual. And in main, it throws file not exception, file not found exception, rather.

And here's our array list of triangle, we'll create a new array list there. And then down here, we'll fill it with triangles and so we ask the user to enter a file name and we get that. And then the line here say t in line numbers line 26 we actually creates scan file.

It's a scanner on that new file. And the, we read the first line in the file and that's gonna be the triangle name. In fact let's look, take a look at triangle.dat here. Here's what we're looking at. So the first line in the file is, is the name of the list.

And so, we're gonna call this a nice, small set of triangles. And then here we've got sets of three numbers that we're gonna read in. And each set of three there represents a triangle. So we'll have a three, four, five, a nine, nine, ten, zero, six, nine. And 7, 7, 7, OK?

So that's, that's what we're gonna read in. And so notice after we read the name, we go into loop. And while triangle, while scanFile rather hasNext, we scan the next double and assign that to side1. And then get the next double, side2. Scan the, the third double And make that side 3, so now we have the, the data for a triangle, so we create a new triangle.

A same triangle, side 1, side 2, side 3, and we add to our array list, we call, my list up here is, is just an array list of triangles. So we add it to it. And then go back and read the next three sides. Make a triangle, add that to the list list, go back and read the next three sides and so on until there's no more data in the file.

After that we close the file, close the scanner which is open on the file. And then we're gonna create our TriangleList object. And so our TriangleList object takes in, the constructor takes in the list name. And then that array list of triangles. And then that gives us the TriangleList.

Now once we have it we can actually print it out. And that calls its toString. Which, of course, prints the name of it and then causes the two strings for each triangle that was the loop we looked at. And then, finally, we say my triangle list dot summary and that calls the summary method, which prints out all of the details for the triangle.

So, let's run this in, in the in the canvas, and we'll see how all that works. We've got a lot of data here we're going to look at. This is just reading in the file name, and then this is the triangle list name, the sides that we're working on in our triangle and so on.

So, I'm going to step step down here and it's gonna ask for a, a file name. And let's see what file we've got here. I think this is triangle.dat, I believe. Yes, triangle.dat. And so I'll just move this over, so we can sort of see all this. So let's hit Enter there.

So now, we've, we've just read in the file name, and we created a new scanner on the file, there it is, right there, and now we're gonna go into create, while we're gonna read in the name here. In fact we should see that there, so here's the name, it came in.

Can look at that, it was small there it is nice small list of triangles that was the name and we're sitting right here on three that's gonna be the next number we read. In fact, let's scan over a little bit there. So we're about to read in 345, and and let's do that.

So we're entering our loop. So I'm just gonna step, and there we're scanning the file, and that side one came in. Or, and this is nex-, we're getting next double there, and that came in as, side two you see, and there's side three, and now we create a new triangle.

There's a new triangle, so 3, 4, 5, and now we're going to add it to the list, so we add it to the list and now we see that and the list. And we just keep stepping through this, adding triangles. There’s a second one, here’s third one coming in.

This is our 069 which when we check its category, it’s not gonna be a triangle. Since it's got a zero side there. And finally, maybe finally, reading in another one. Let's see this should be the last, it was the last one. There was seven, seven, seven came in.

We're at our, return and new line, and so, we're going to close the file And now we're gonna create the triangle list. And there it is, it's a nice small set of triangles, and this is the array list with all the triangles in it. Essentially, it's the same as my list up here, since that's what we created it from.

And now when we print out, I'm just going to move this over here, we're going to print out my triangle list. It's, went to the toString. We've gotta a break point in the toString, that's why it's stepped in. So I'll just step through that. And, this is all the toString building it actually.

And then finally, we're gonna step out. There we go, and we, and we printed that that entire my triangle list. And then finally, we're gonna print the summary. And we can step in here too. Let's step in here and we'll see how we're gonna build that summary. So we're gonna call how many there are there.

In fact, let's just move that off to the side. And we can a, actually open result and see this get built here. So, there's summary, and then we just called number of triangles, and it's four. We're about to call smallest parameter, and there it is 12. And there's largest parameter 28 with it's size.

Here's total parameter, and then we are about to return this string here, the result. So I am just going to move this out of the way and close it. And we will see that printed out. When we, when we click step on this statement, it gets printed out.

Okay, so that's and one more step, and we end there, okay. So that's that, that's a triangle. Let's look at one more data file here. I'm gonna go back and let's look at 0. We'll run it with, there's no data in here. So let's, let's, let's see how that works.

I'm gonna close that, but we'll run it with, let me just copy that, triangle_0.dat. So let's close that file. And, we'll end this, and let's run it again. And let's step, this will go much quicker since there's no data.

>> Oops, waiting for input. Waiting for the file there.

Let's give it this triangle_0.dat, we open a file and then we're gonna scan in the create the scanner. And then this is us reading the file for the file name, so we just read the file name in there and it essentially says there's no, no triangles in here.

That's what it said up there on top, empty, empty list if you will. And it, notice after it, there is no data. This is just all, no characters there. And so the file is not gonna have any data. And so it just skips to the bottom there, it closes and, and goes on there.

In fact, we'll just run this one out, and we had an empty set of triangles there. So one more set of data, this is triangle_1. Here we've written a nice set of triangles. And I've, and I've actually got the data in sets of three on each line. And since we're just reading next double each time, it's gonna go through here and, and get all these, so let's read triangle 1..

We'll close this and just run this again. We should be able to see this go step, step, waiting on the file name. Got our new scanner. Now we've read in, and now we see, here these lines here, and they're not, they're only separated by spaces. And then we got, a line feed and and return character, and then the next set of three.

And so, so that, that's what we're gonna see there. And this is going to work just fine. There they go. And if we get all of them in, we read in one, and so now we read in the next three. We're just reading in next double, and this is, this is to show you that reading in next double like this, with the end, we're going to create a triangle.

And we go back up here, and notice we're at the end of a line here, but we're gonna try to get the next double. And when we do it, it just reads passed this carriage return and a new line character, and, and goes all the way to zero.

So we just read passed it there, and, and so on. Again, easy to use this, using next, double next ent and so on. So eventually, we finished this and we're going to print things out. There we go. That all printed out, and then, whoops, we went into two string again.

I'm going to make this break point go away and step out, and then just step. There we go. And there's our summary data printed. Okay, this has been a pretty involved example, but you know, it uses lot of if statements, and of course, lots of loops, especially in those methods in the triangle list.

So be sure to take a look at those and, and make sure you understand how, how those work. And then of course we read n a file and it pretty simple. We could read it in, per line or we could actually read the data in all on a separate line.

So we'll wrap our module, with the next little segment.