Telling a Random Story

Using and Improving GladLibs

Review

Summary

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

Interactive Transcript

Search Transcript English

0:03
Hi, as you get ready to start the program to create stories from templates, let's summarize what you've learned about the ArrayList class in Java.

0:11
ArrayLists are like arrays. Both are indexable collections, allowing you to access elements with an integer index. ArrayList can grow as elements are added to it. This means you don't need to know in advance how much space to allocate for an ArrayList like you did for an array.

0:30
Like arrays, and individual string elements, indexing in an ArrayList starts with zero. It takes the same amount of time to access the first element of an ArrayList or an array, as it does to access the ten thousandth element. It might help you to think of ArrayLists as a collection of boxes, each addressable with a number.

0:52
To use the ArrayList class, you must import it from the java.util package. You can import just the ArrayList class, or use the asterisk and import java.util.* to gain access to all of the classes in the package, like the random class.

1:08
When you create an ArrayList, you specify the type of elements stored in the ArrayList. Using the angle bracket syntax that Java uses for generic or general elements.

1:18
Here you see an ArrayList that can contain string objects. But ArrayLists can also store integer objects, too. Though the list must store integers or strings, not the same type in one list.

1:33
The integer class allows int values like zero or 57, or negative 352 to be stored in the ArrayList. The integer class automatically converts an int value like 57 into a value stored as an integer object.

1:48
You've seen several methods used in ArrayList objects. The .add method adds an element to the end of an ArrayList. The ArrayList grows as needed. The .size method returns the number of elements stored in an ArrayList. Typically this is the number of elements added via .add. You can write code to access individual elements with and integer index using the .get method and you can change the value stored at a specific index using the .set method.

2:20
ArrayLists are typically processed and accessed using loops. Here's a typical indexing loop that processes each element of an ArrayList. This for loops typically started 0 and loop to less than a size of an ArrayList, which is exactly .size elements. Within the loop, each array element is accessed using the dot get method and the loop index variable.

2:47
When accessing array elements in a loop like this do not call .add or .remove which will change the size that the loop iterates. Typically causing a problem in your algorithm because you will either skip elements or access invalid elements.

3:04
You can also access the elements in an array with an iterable loop, the same kind of loop we used with the edu.duke iterable classes. In an iterable loop, your code indicates the type of value stored in the ArrayList. Your loop takes on each value stored in the ArrayList one at a time, just as with the file resource or image resource classes. You can use this kind of loop when you don't need the index of each ArrayList element but just the element itself. Just as with an indexing loop, do not call .add or .remove with an iterable loop. In this case, Java will generate a run time error if you try.

3:42
ArrayLists are a very useful tool when used properly.

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?