Lesson 3: ArrayLists

45 minutes

Overview

Why would I use an `ArrayList` instead of an array?

Students explore the `ArrayList` class and learn how to declare and initialize an `ArrayList` and add elements. Students identify scenarios where an `ArrayList` is more suitable and differentiate between a dynamic data structure and a static data structure. Students then practice creating an `ArrayList` and adding values to the list.

Standards

Full Course Alignment

CSA Conceptual Framework

▶ **VAR-2** - To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure wihtout creating individual variables for each value.

Agenda

Warm Up (10 minutes)
- **Storing User Input in a List**

Activity (30 minutes)
- **The ArrayList Class**
- **Using ArrayLists**

Wrap Up (5 minutes)
- **Revisiting Storing User Input**
- **Assessment: Check for Understanding**
- **AP Classroom Topic Questions**

Objectives

Students will be able to:

- Describe the structure of the `ArrayList` class
- Use methods in the `ArrayList` class to obtain the size and add elements

Preparation

- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **U6L3 Extra Practice** - Handout

Vocabulary

- **ArrayList** - A class that represents a resizable list
- **Mutable** - The ability to change after initialization
- **dynamic data structure** - a data structure that grows and shrinks as needed
- **static data structure** - a data structure that is fixed in size

Teaching Guide

Warm Up (10 minutes)

Storing User Input in a List

Remarks

> We have worked with 1D and 2D arrays in different ways to store and manipulate multiple items or values. One common use of lists is to store and use data that a user enters. Let's look at an example.

Do This: Have students write pseudocode for the problem.

Discuss: Use the Hold That Thought strategy to discuss the prompt.

- *What if we don't know how many orders the owner needs to enter into the program? What would we need to change in our solution?*

Discussion Goal: Students share ideas for modifications, such as asking how many items they are entering first and using that value to initialize the array. Students realize that since the size of 1D and 2D arrays cannot change after initialization, they need to get the number of items from the user first.

Activity (30 minutes)

The ArrayList Class (15 minutes)

Remarks

> It turns out that Java has a class that solves this exact problem! Today, we will learn about the `ArrayList` class and practice creating `ArrayList`s to store elements in your programs.

Do This: Review the lesson objectives.

Group: Place students in pairs.

Do This: Direct students to Level 1 on Code Studio to investigate the program with a partner. Students make the changes to the program as prompted.

1 **Investigate: ArrayLists**

Discuss: Click through the animated slide to display the prompts.

- *What do you notice about the code in this program?*
- *What do you wonder about the code in this program?*

Discussion Goal: Students notice that an `ArrayList` stores a list of values and that the type is `Integer` instead of `int`. Students also notice that the syntax is different for an `ArrayList` than a 1D or 2D array and that the `add()` methods add elements to either the end of or at a specific location in the `ArrayList`. Students may wonder why they would use an `ArrayList` instead of a 1D array.

Teaching Tip

Ask students to predict the size of the `ArrayList` before and after the values are added. Have students identify how initializing an `ArrayList` is different from initializing a 1D or 2D array, including that they do not need to specify the size of the `ArrayList` like they did for a 1D or 2D array.

Display: Show the video – *The ArrayList Class*.

Do This: Click through the animated slide to define *dynamic data structure* and *static data structure*.

Do This: Define `ArrayList` and mutable and identify the package that contains the `ArrayList` class.

Do This: Click through the animated slide to explain the constructor for an `ArrayList`.

Do This: Click through the animated slide to demonstrate the `add()` and `size()` methods.

Remarks

> Because `ArrayList` is a class, it has the same three parts that we've seen in other classes: instance variables, constructors, and methods. Let's take a look at some of the source code for the `ArrayList` class.

Discuss: Click through the animated slide to discuss the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What is the purpose of each instance variable?*
- *What is the relationship between an array and an* `ArrayList`*?*

Discussion Goal: Students identify that the `DEFAULT_CAPACITY` instance variable is the default size of an `ArrayList`, the `size` instance variable is the number of elements in the `ArrayList`, and the `data` instance variable is a 1D array. Students suggest that the relationship between an `ArrayList` and an array is a has-a relationship and an `ArrayList` stores data in the array instance variable.

Do This: Explain the purpose of each instance variable:

- The `DEFAULT_CAPACITY` constant
- The `size` instance variable
- The `data` instance variable

Teaching Tip

The code segment shown is the source code for the `ArrayList` class, which includes components not covered in the AP CSA framework. Acknowledge questions but avoid addressing topics such as implementing interfaces or the `transient` keyword. If students would like to explore them further, consider doing so as an extension activity.

Using ArrayLists

Remarks

> An `ArrayList` is useful when we are working with an unknown amount of data or we need the flexibility to add elements at specific locations or expand the size of the list.

Do This: Direct students to Level 2 on Code Studio to complete Levels 2, 3, and 4. Students complete a Check for Understanding on Level 2, then continue to Level 3 to declare and initialize an `ArrayList`. On Level 4, students complete a choice level to add elements to an `ArrayList`.

2-4 **Using ArrayLists**

2 3 4

Wrap Up (5 minutes)

Revisiting Storing User Input

Discuss: Use the Retrieve-Pair-Share strategy to discuss the prompt.

- *How could we use an* `ArrayList` *to solve this problem?*

Discussion Goal: Students suggest using an `ArrayList` to store the user input and using the `add()` method to add each input to the end of the list. Students may also suggest asking the user for a specific location to add the input to the list at a given index.

Do This: Review the concepts covered in this lesson.

Display: Key Vocabulary

Assessment: Check for Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.

5 **Check for Understanding**

AP Classroom Topic Questions

To assign questions from the AP Classroom Question Bank that align with this lesson, create a custom quiz in AP Classroom by searching the Question Bank for the Essential Knowledge statements listed at the top of this lesson plan. You can find instructions and video demonstrations to do this on **AP Central**.