Vanit Patel

CS 2400- Data Structure Advance Programming Section 02

Project 1

9-26-24

Section 1: Project Specification & Implementation

The goal of this project was to implement a Bag Abstract Data Type using an array-based structure and to create an application that reads Java keywords from a file and stores time in a bag. The program check is the command-line arguments are Java keywords and it also tests the each of the ArrayBag implementation.

ADT description: The BagInterface<T> contains 9 methods, which are: add(t newEntry), clear(), contains(T anEntry), getCurrentSize(), getFrequencyOf(T anEntry), isEmpty(), remove(), remove(T anEntry), toArray(). The add(T newEntry) type is a boolean and adds a new entry to the bag. The clear() method type is void and it removes all entries from the bag. contain(T anEntry) method type is boolean and it tests whether the bag contains a given entry. getCurrentSize() method type is int and it gets the current number of entries in the bag. getFrequencyOf(T anEntry) method type is int and it counts the number of times a given entry appears in the bag. isEmpty() method type is boolean and it sees whether the bag is empty. remove() method type is T and it removes one unspecified entry from the bag, if possible. remove(T anEntry) method type is boolean and it removes one occurrence of a given entry from the bag, if possible. toArray() method type is T[] and it retrieves all entries that are in the bag.

We implemented the bag ADT using an array-based structure in the ArrayBag class. The array-based bag was chosen because it provides an efficient implementation for managing a moderate number of elements(the Java keywords). The Java keywords were only 53 elements, making it manageable within a fixed-sized array. The array stores the items sequentially. This means operations such as adding, removing, and searching for items are performed in a linear manner.

Section 2: Testing Methodology

To ensure that the ArrayBag class works correctly, I conducted several hardcoded tests in the JavaKeywords class. I first tested the getCurrentSize() method, by telling the user to expect the value 53. The getCurrentSize() method return the value 53, showing that the method works. Next, I checked the toArray() method. The method outputs an array list of all the elements in the bag, showing that the method works. Next, I tested the method contains(). I did this by asking the method to correctly identified whether the bag contains a specific keywords such as "continue." The method outputs True which shows that contains() method works. Next, checked add() method. I added the element "phone" to the bag. To see if the method successfully added "phone" to the bag, I used getCurrentSize() method and contains() method. The getCurrentSize() outputs 54, showing that there is a increase in element in the bag. Then I used contain() method to check if "phone" was in the bag, which it outs true. This shows that add() method works. Next, I checked the remove() method. Since "phone" was the last element added to the bag it should have been removed. Like the previous method, I check if the size of the bag and if the bag still contain the element "phone." The getCurrentSize() method outputs 53 and the contains() method outputs false, showing that remove() method works. Next I tested remove(T anEntry). The removed the element "package" from the bag. I check if the method works by checking if the size of the bag becomes 52 and if the bag does not contain "package." The getCurrentSize() outputs 52 and the contains() outputs false. This means remove(T anEntry) works. Next I checked getFrequency() method. I added the element "boolean" to the bag. The bag contains 2 "boolean" element. The getFrequency() methods outputs 2 which shows that the method works. Finally, I checked if clear() method and isEmpty() method works. I first checked if the current bag is empty. The isEmpty() method returns false. Then I cleared the bag, and checked isEmpty() and getCurrentSize() methods. IsEmpty() returns true and getCurrentSize() return 0. This means that clear() and isEmpty() methods is working.

Section 3: Lesson Learned

In this project, I learned several key lessons. First I learned the importance of Abstract Data Types. Implementing the bag ADT helped to create a reusable design. The interface ensures that the bag can be used in multiple context without needing to know the internal implementation. Second, I learned the importance of Javadoc. In my previous coding project, I never used Javadoc. Lucily, in this project I learned how it allows for better understanding and easier debugging. A good documentation makes it easier to follow the flow during testing. Third, I learned how to hardcode my tests. My creating a specific test case for each method, I was able to identify any errors in my ArrayBag class and with the help of the Javadoc, I can easily debug it. The last thing I learned is the command-line parameter. This the only project, in which I used command-line parameter. At first I was confused when my professor said that he will input the parameters, but after self-learning what command-line parameter is, I found it to be very useful in the real world. Overall, this project showed me the importance of ADT, essecailly the bag ADT, and how to write clear and readable code.