Enhancement Two: Algorithms and Data Structures

Ryne Williams

Department of Computer Science, Southern New Hampshire University

CS 499: Computer Science Capstone

Brooke Goggin

11/18/2023

This artifact is a program that puts dogs and monkeys into separate ArrayLists for rescue work. Each ArrayList contains the animals name and a few other objects relevant to the animal type. The program allows for new dogs and monkeys to be entered into the ArrayList, reserving animals for service, and printing a sorted list of dogs, monkeys, or reserved animals. This artifact was created in IT 145: Foundations in Application Development as the final project for the course.

This artifact was a good option to implement a merge sort algorithm for the printing of the animals, due to the fact that is uses two separate ArrayLists to store the different animal types. This allowed for an improved user experience of the application when viewing the list of animals, as well as being able to showcase my ability to implement a Merge Sort algorithm on an ArrayList data structure. The Merge Sort algorithm has a time complexity of O(nLogn), which gives it an efficiency greater than other sorting algorithms. However, the efficiency begins to drop at the end of the sort for larger datasets.

I believe that I successfully met the course outcomes that I planned to meet for this specific enhancement. This enhancement designed and evaluated computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solutions, while managing the trade-offs involved in design choices. The Merge Sort algorithm will solve a given problem with the best efficiency for a limited list of animals. There are no changes to my outcome-coverage plans.

Implementing a Merge Sort algorithm was a bit tricky at the beginning, as it has been a while since I have had the opportunity to work with sorting algorithms. I wanted to keep the code as simple as possible, which meant using only one sort algorithm for both ArrayLists. This proved to be difficult as the generic method options that I found while researching a solution to the problem all seemed to be incompatible with working with more than one ArrayList. After a good bit of research and looking over Java documentation, I came to the conclusion that I was unintentionally coding the algorithm to sort every object in each ArrayList, which was the reason behind the incompatibility issues and did not fit what I wanted the sorting algorithm to do. I found a way to sort only by the name of the animal in each ArrayList using the “Comparator” Java class. This resolved the issues that I was experiencing with getting the ArrayLists to work with one sort algorithm. Afterward, I removed testing methods for the code and added comments to improve readability and maintenance.