Anthony Nguyen

CS 499

4-2 Milestone Three

Enhancement Two

Algorithms & Data Structure

**Artifact Overview**

This artifact was produced in the CS 320 course where I implemented a contact management system that can perform create, retrieve, update and delete operation of Contact information. Its primary function is to manage a collection of `Contact` objects using a `HashMap`. The `HashMap` provides efficient storage and retrieval of contact information, allowing for constant time complexity (O(1)) for operations like adding, updating, deleting, and retrieving contacts. The class also includes methods to update contact information, delete contacts by ID, and handle error cases such as duplicate contact entries or invalid IDs.

**Justification for Inclusion:**

I chose this artifact because it effectively demonstrates my skills and understanding of data structures, particularly the use of hash tables for managing and storing data. The core reason for selecting this item is that it represents my ability to apply algorithms and data structures in practical, real-world scenarios.

**Specifically, the artifact showcases:**

* Efficient Data Handling: The use of the `HashMap` data structure allows for fast lookups, which is crucial in scenarios where large datasets need to be processed quickly.
* Error Handling: The class demonstrates defensive programming by incorporating validation checks for null inputs and duplicate contact IDs, ensuring robust error management.
* Modular Design: Each operation (add, delete, update, retrieve) is implemented as a separate method, following principles of modularity and separation of concerns, which are key software design practices.

Through the enhancements I made, I aimed to further improve the quality of this artifact. I refined the error handling mechanisms and enhanced the update logic to make it more efficient. Additionally, I explored the possibility of using a `TreeMap` as an alternative to `HashMap` for scenarios where contacts need to be sorted in a specific order.

**Alignment with Course Outcomes**:

The enhancements I implemented align directly with several key outcomes of the CS 499 course. First, they demonstrate proficiency in designing and optimizing data structures to efficiently handle information. By leveraging `HashMap` in its current form and exploring alternatives like `TreeMap`, I showcase my understanding of different data structures and their trade-offs in terms of performance.

Additionally, my work on this artifact highlights my ability to manage edge cases and improve error handling, reflecting a deep understanding of algorithm design and defensive coding. The enhancements I made show my attention to creating code that not only works under normal conditions, but is also resilient in handling unexpected inputs.

**Reflection on the Enhancement Process:**

Through the process of enhancing this artifact, I deepened my understanding of how `HashMap` and similar data structures operate. One significant learning moment was identifying how critical it is to account for edge cases, especially when working with data structures like hash maps. Handling collisions or ensuring data integrity during updates and deletions required careful thought and deliberate coding practices.

The challenges I faced primarily involved balancing performance and robustness. While `HashMap` is an excellent choice for most operations due to its constant time complexity, I realized that it doesn't maintain order. For situations where order might be important, using a `TreeMap` could provide better performance, although with a slight decrease in time efficiency (O(log n)), this process gave me insight into selecting the right data structure for specific scenarios.

Lastly, I reinforced my error-handling skills, ensuring that invalid input, such as null values or duplicate contact IDs, is appropriately managed. This taught me the importance of building resilience into code, especially in systems that will manage and manipulate data.

**Conclusion:**

This artifact, along with the enhancements made, clearly demonstrates my growing expertise in algorithms and data structures. By focusing on data structure selection, performance optimization, and robust error handling, I believe I have created a strong example of my skills in this area. These skills are critical to my future career in software development, where efficient data management and reliable code are essential. I look forward to receiving feedback on this artifact and continuing to refine it for my ePortfolio.