Jay Li

CMSC204

Professor Tsai

23 June 2024

Learning Experience Assignment 2

Working on this project has significantly enhanced my understanding of fundamental data structures like stacks and queues. By implementing these data structures from scratch in Java, I gained a deeper grasp of their mechanics, particularly how stacks operate on a Last-In-First-Out (LIFO) basis and how queues adhere to a First-In-First-Out (FIFO) principle. This hands-on experience allowed me to appreciate their practical applications and internal workings more effectively than merely studying them theoretically.

Another key learning point was mastering the conversion between infix and postfix notations using stacks and queues. Initially, I had a conceptual understanding from tutorials and videos, but implementing these conversions myself solidified my comprehension. For instance, crafting the logic to handle operators and parentheses correctly in the convertInfixToPostfix method required me to apply stack operations in a methodical manner. This process sharpened my problem-solving skills and deepened my understanding of algorithmic design.

Throughout the project, one of my main challenges was integrating my custom stack and queue implementations into the Notation class effectively. I encountered difficulties with the logical flow, particularly in the convertInfixToPostfix method. Initially, I mistakenly placed the code to transfer operators from the stack to the postfix solution queue inside the main loop, causing repeated InvalidNotationFormatExceptions. Recognizing this error and restructuring the logic to handle right parentheses correctly was a significant breakthrough. It underscored the importance of methodically breaking down problems and validating each step before proceeding, a lesson I will carry forward into future projects.

Looking ahead, I aim to refine my debugging skills further. The time spent troubleshooting issues in the logical flow of the conversion methods highlighted the importance of meticulous testing and step-by-step validation. In future projects, I plan to adopt a more structured approach by breaking down complex tasks into smaller, manageable parts. This approach will ensure clearer logic and reduce the risk of overlooking critical details, ultimately leading to more efficient problem-solving and code development. Overall, this project has been instrumental in honing my Java programming skills, enhancing my understanding of data structures, and preparing me for more intricate software development challenges in the future.