Quá trình thực hiện đồ án

1. Thiết lập lớp Node
   1. Mục đích:

Lớp Node được dùng để tạo thành các phần tử cấu thành nên cấu trúc dữ liệu dạng danh sách liên kết (DSLK).

* 1. Cấu trúc:
     1. Các thuộc tính:
* T data: là thuộc tính kiểu Generic dùng để lưu dữ liệu của 1 node
* Node<T>? next: là thuộc tính tham chiếu đến node kế tiếp của node đang được xét, thuộc tính này có thể null trong trường hợp đây là 1 node độc lập hoặc có thể hiểu đây là 1 DSLK có duy nhất 1 phần tử.
  + 1. Các phương thức:
* Node() và Node(T data): các hàm khởi tạo
* T Data{} và Node<T> Next{}: hàm getter và setter của các thuộc tính
* Boolean hasNext(): trả về true nếu node hiện tại có tham chiếu đến 1 node kế tiếp, ngược lại sẽ trả về false.

1. Thiết lập lớp cQueue
   1. Mục đích

Đây là lớp dùng để tạo ra các data object dạng danh sách hàng chờ với nguyên tắc, đối tượng được nạp vào hàng chờ trước sẽ được lấy ra trước.

* 1. Cấu trúc
     1. Các thuộc tính
* int count: dùng để đếm số phần tử trong hàng chờ
* Node<T>? head: node ở đầu danh sách
* Node<T>? tail: node ở cuối danh sách
  + 1. Các phương thức
* int Count {}: hàm getter của thuộc tính count
* cQueue(): hàm khởi tạo của queue
* void Enqueue(T value): hàm thêm 1 giá trị vào queue. Giá trị được thêm vào sẽ được “đính” vào 1 node mới và nối tiếp node mới đó vào tail của queue, đồng thời node mới đó sẽ đóng vai trò làm tail mới của queue.
* void Dequeue():

1. Thiết lập lớp cStack
2. Thiết lập lớp PolishNotationBuilder
3. Thiết lập lớp Calculator
4. Thiết lập các lớp phụ trợ