```
// First In First Out
public class Queue {

  // INSTANCE VARIABLES
  private Node first;
  private int size;

  // CONSTRUCTORS
  public Queue() {
    this.first = null;
  }

  // PUBLIC METHODS
  public boolean enqueue(String content) {
    boolean enqueued = false;
    if (this.size == 0) {
      this.first = new Node(content);
      this.size++;
      enqueued = true;
    }
    else {
      Node temp_node = this.first;
      while (temp_node.getNext() != null) {
        temp_node = temp_node.getNext();
      }
      temp_node.setNext(new Node(content));
      this.size++;
      enqueued = true;
    }
    return enqueued;
  }
  public boolean dequeue(){
    boolean dequeued = false;
    if (this.size > 0) {
      this.first = this.first.getNext();
      this.size--;
      dequeued = true;
    }
    return dequeued;
  }
  public boolean isEmpty() {
    boolean is_empty = true;
    if (this.size > 0) {
      is_empty = false;
    }
    return is_empty;
  }
  public int size() {
    return this.size;
  }
  public String peek() {
    return (this.first != null) ? this.first.toString() : "[queue is empty]";
  }
}
```