import java.util.Comparator;

public class OrderQueue<T> extends QueueSecondary<T> {

private static <T> void insertInOrder(Queue<T> q, T x, Comparator<T> order) {

assert q != null : "Violation of: q is not null";

assert order != null : "Violation of: order is not null";

Queue<T> tempQueue = q.newInstance();

while (q.length() > 0 && order.compare(x, q.front()) > 0) {

tempQueue.enqueue(q.dequeue());

}

tempQueue.enqueue(x);

while (q.length() > 0) {

tempQueue.enqueue(q.dequeue());

}

q.transferFrom(tempQueue);

}

@Override

public void sort(Comparator<T> order) {

assert order != null : "Violation of: order is not null";

Queue<T> sortedQueue = this.newInstance();

while (this.length() > 0) {

insertInOrder(sortedQueue, this.dequeue(), order);

}

this.transferFrom(sortedQueue);

}

}