

JAVA NOTES

DATA STRUCTURES AND ALGORITHMS

Terry Marris July 2001

10 QUEUE INTERFACE

10.1 LEARNING OUTCOMES

By the end of this lesson the student should be able to

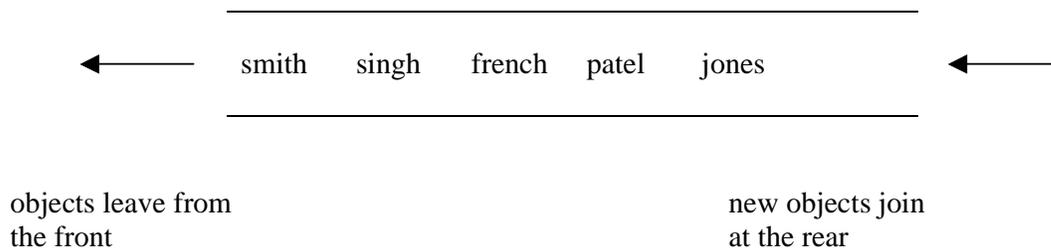
- describe the characteristic features of a queue
- explain the queue interface
- show how a queue may be used to solve simple problems

10.2 INTRODUCTION

In the last three lessons we looked at the stack data structure. Now we begin our study of another data structure: queues.

Queues are used for simulating and managing queuing problems such as patients waiting to see consultants, traffic flow controlled by a system of traffic lights management and aeroplane landings and take offs at a busy airport.

A queue is what you think it is: objects join the rear of the queue and leave it from the front.



A queue is a linear data structure.

There are several kinds of queue. A priority queue organises its objects in some kind of order according to need, not on a first-come first-served basis. Yet another queue allows objects to leave from anywhere in the queue, but only the object at the front can be served. We shall look at just the simplest, most basic kind of queue: objects join the rear and leave from the front.

10.3 THE QUEUE INTERFACE

The standard queue operations are *join(Object)*, *leave()*, *retrieve()* and *isEmpty()*.

```
/* Queue.java
   Terry Marris  28 July 2001
*/

public interface Queue {
    /* defines the standard queue operations
       join(Object), leave(), retrieve() and isEmpty().
    */

    public boolean isEmpty();
    /* If this queue contains no objects, true is returned.
       If the queue contains objects, false is returned.
    */

    public String join(Object obj);
    /* If there is space in this queue, adds the given object to
       the rear of this queue and returns "success".
    */

    public String leave();
    /* If this queue is not empty, removes the object at
       the front of this queue and returns "success".
       If this queue is empty, returns "failure - queue empty".
    */

    public Object retrieve();
    /* If this queue is not empty, returns the object at
       the front of this queue.
       If this queue is empty, returns null.
       In both cases, the contents of this queue remain
       unchanged.
    */
}
```

10.4 REVIEW

10.5 FURTHER READING

LEWIS & LOFTUS *Java Software Solutions* pp 506

In the next lesson we see how to implement a queue using an array.

10.6 EXERCISES

1 List and describe the standard queue operations.