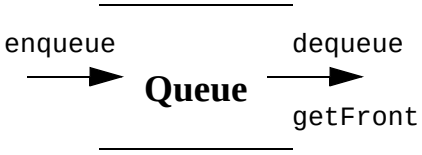


**Figure 6.6** Queue model: Input is by `enqueue`, output is by `getFront`, deletion is by `dequeue`

The basic operations supported by queues are

- `enqueue` – insertion at the back of the line,
- `dequeue` – removal of the item from the front of the line, and
- `getFront` – access of the item at the front of the line

Figure 6.6 illustrates these queue operations. Historically, `dequeue` and `getFront` have been combined into one operation. This is what is done here. `dequeue` gives the front item, and then removes it from the queue.

```
 1 package DataStructures;
 2
 3 import Exceptions.*;
 4
 5 // Queue interface
 6 //
 7 // ******************PUBLIC OPERATIONS*********************
 8 // void enqueue( x )    --> Insert x
 9 // Object getFront( )   --> Return least recently inserted item
10 // Object dequeue( )    --> Return and remove least recent item
11 // boolean isEmpty( )   --> Return true if empty; else false
12 // void makeEmpty( )    --> Remove all items
13 // ******************ERRORS********************************
14 // getFront or dequeue on empty queue
15
16 public interface Queue
17 {
18     void    enqueue( Object x );
19     Object  getFront( ) throws Underflow;
20     Object  dequeue( )  throws Underflow;
21     boolean isEmpty( );
22     void    makeEmpty( );
23 }
```

**Figure 6.7** Interface for the queue