

```
12  public SynchronousDualQueue() {
13      Node sentinel = new Node(null, NodeType.ITEM);
14      head = new AtomicReference<Node>(sentinel);
15      tail = new AtomicReference<Node>(sentinel);
16  }
17  public void enq(T e) {
18      Node offer = new Node(e, NodeType.ITEM);
19      while (true) {
20          Node t = tail.get(), h = head.get();
21          if (h == t || t.type == NodeType.ITEM) {
22              Node n = t.next.get();
23              if (t == tail.get()) {
24                  if (n != null) {
25                      tail.compareAndSet(t, n);
26                  } else if (t.next.compareAndSet(n, offer)) {
27                      tail.compareAndSet(t, offer);
28                      while (offer.item.get() == e);
29                      h = head.get();
30                      if (offer == h.next.get())
31                          head.compareAndSet(h, offer);
32                      return;
33                  }
34              }
35          } else {
36              Node n = h.next.get();
37              if (t != tail.get() || h != head.get() || n == null) {
38                  continue;
39              }
40              boolean success = n.item.compareAndSet(null, e);
41              head.compareAndSet(h, n);
42              if (success)
43                  return;
44          }
45      }
46  }
```

FIGURE 10.20 The SynchronousDualQueue<T> class: enq() method and constructor.