

```
1 public class BoundedDEQue {
2     RecursiveAction[] tasks;
3     volatile int bottom;
4     AtomicStampedReference<Integer> top;
5     public BoundedDEQue(int capacity) {
6         tasks = new RecursiveAction[capacity];
7         top = new AtomicStampedReference<Integer>(0, 0);
8         bottom = 0;
9     }
10    public void pushBottom(RecursiveAction r){
11        tasks[bottom] = r;
12        bottom++;
13    }
14    // called by thieves to determine whether to try to steal
15    boolean isEmpty() {
16        return (top.getReference() < bottom);
17    }
18    }
19 }
```

FIGURE 16.10 The BoundedDEQue class: fields, constructor, pushBottom(), and isEmpty() methods.