

```

1  public class Backoff {
2      final int minDelay, maxDelay;
3      int limit;
4      public Backoff(int min, int max) {
5          minDelay = min;
6          maxDelay = max;
7          limit = minDelay;
8      }
9      public void backoff() throws InterruptedException {
10         int delay = ThreadLocalRandom.current().nextInt(limit);
11         limit = Math.min(maxDelay, 2 * limit);
12         Thread.sleep(delay);
13     }
14 }

```

**FIGURE 7.5** The Backoff class: adaptive back-off logic. To ensure that concurrently contending threads do not repeatedly try to acquire the lock at the same time, threads back off for a random duration. Each time the thread tries and fails to get the lock, it doubles the expected time to back off, up to a fixed maximum.