```java
public class EliminationArray<T> {
  private static final int duration = ...;
  LockFreeExchanger<T>[] exchanger;
  public EliminationArray(int capacity) {
    exchanger = (LockFreeExchanger<T>[]) new LockFreeExchanger[capacity];
    for (int i = 0; i < capacity; i++) {
      exchanger[i] = new LockFreeExchanger<T>();
    }
  }
  public T visit(T value, int range) throws TimeoutException {
    int slot = ThreadLocalRandom.current().nextInt(range);
    return (exchanger[slot].exchange(value, duration,
            TimeUnit.MILLISECONDS));
  }
}
```

FIGURE 11.7 The EliminationArray<T> class: In each visit, a thread can choose dynamically the subrange of the array from which it will randomly select a slot.