```
class MemManager {
  void register_thread(int num); // called once, before any call to op_begin()
                                 // num indicates the maximum number of
                                 // locations the caller can reserve
  void unregister_thread(); // called once, after the last call to op_end()

  void op_begin(); // indicate the beginning of a concurrent operation
  void op_end(); // indicate the end of a concurrent operation

  bool try_reserve(void* ptr); // try to protect a pointer from reclamation
  void unreserve(void* ptr); // stop protecting a pointer
  void sched_for_reclaim(void* ptr); // try to reclaim a pointer
}
```

FIGURE 19.2 An interface for protecting memory during an optimistic operation.