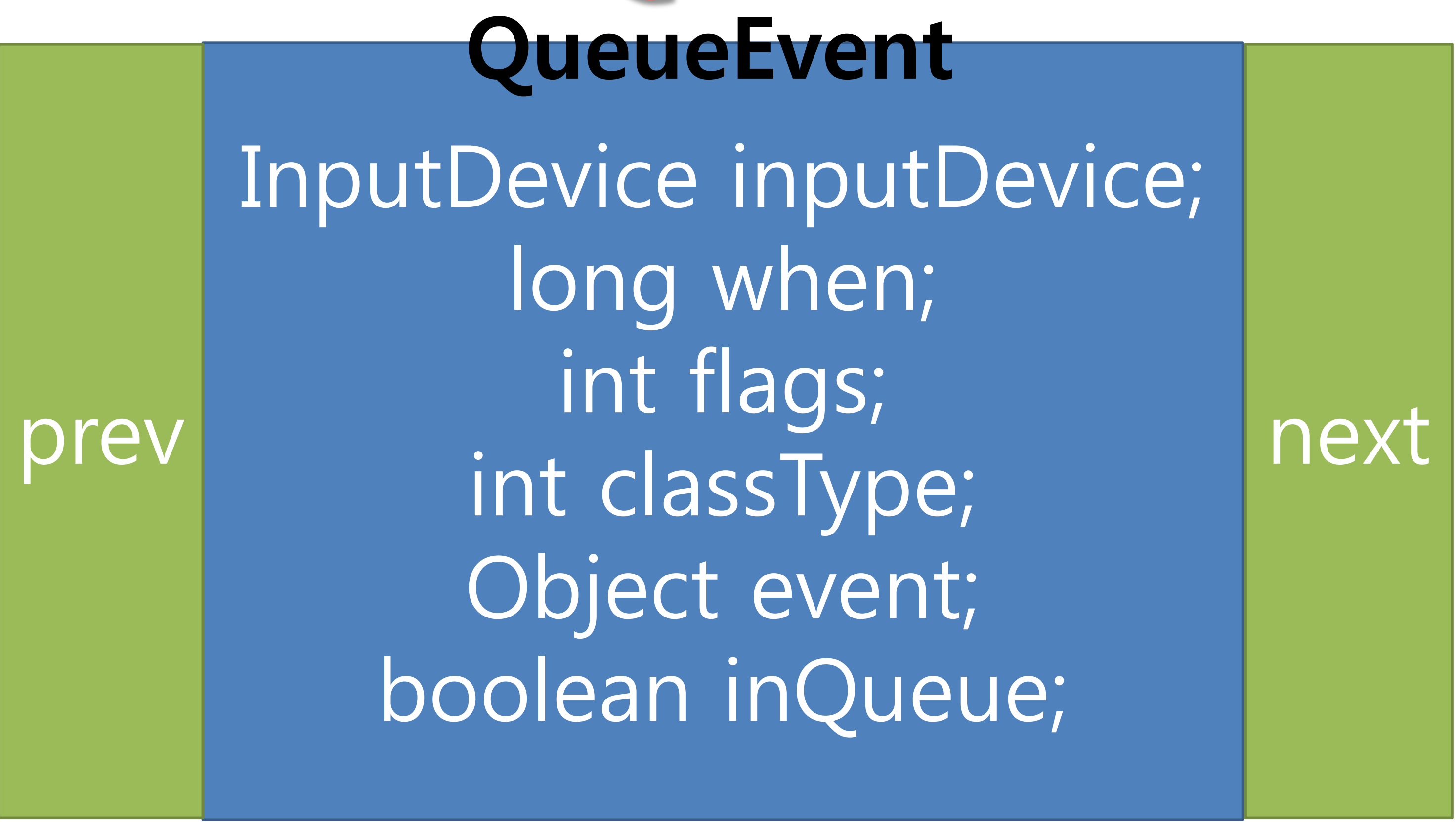
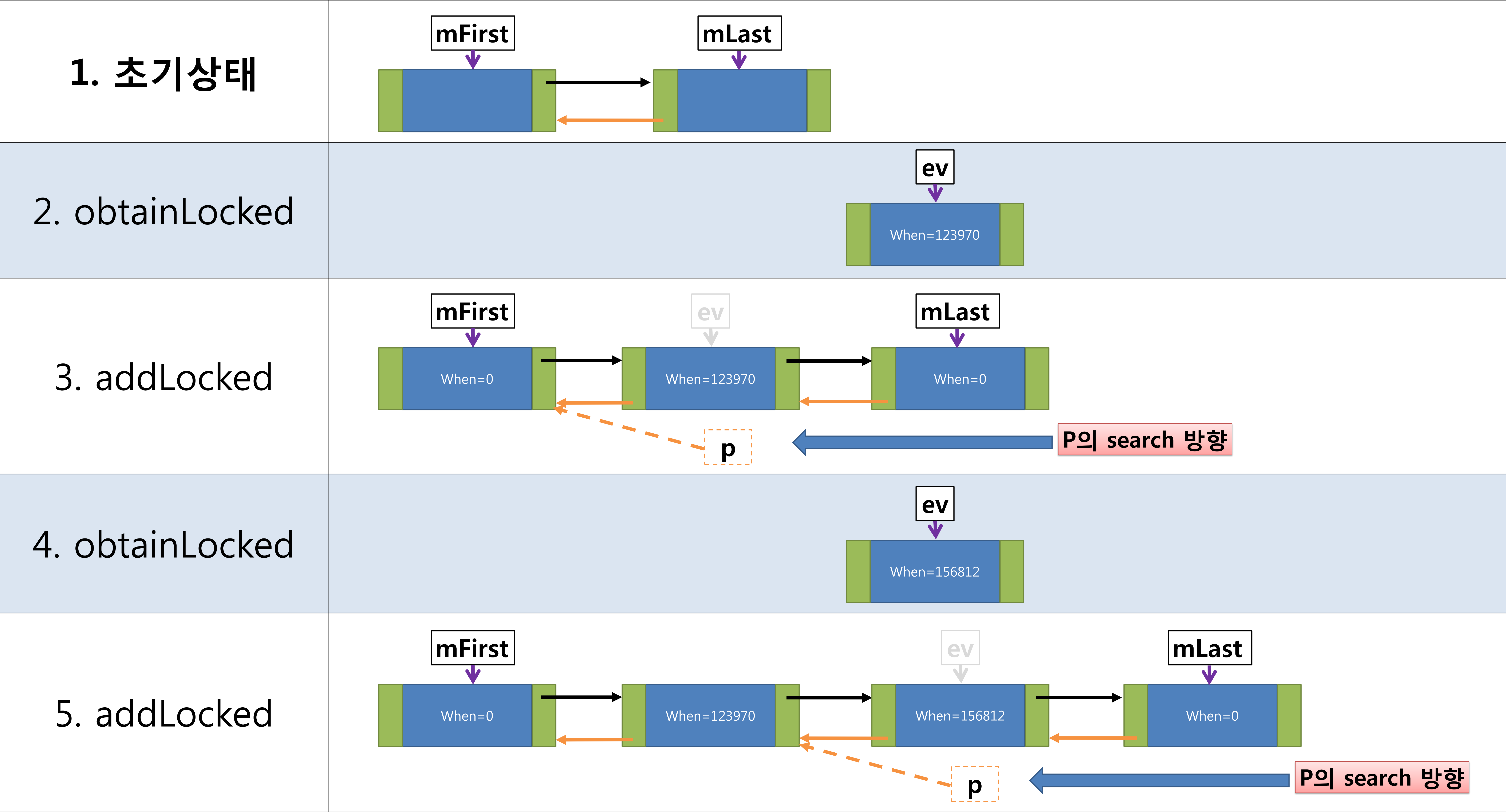


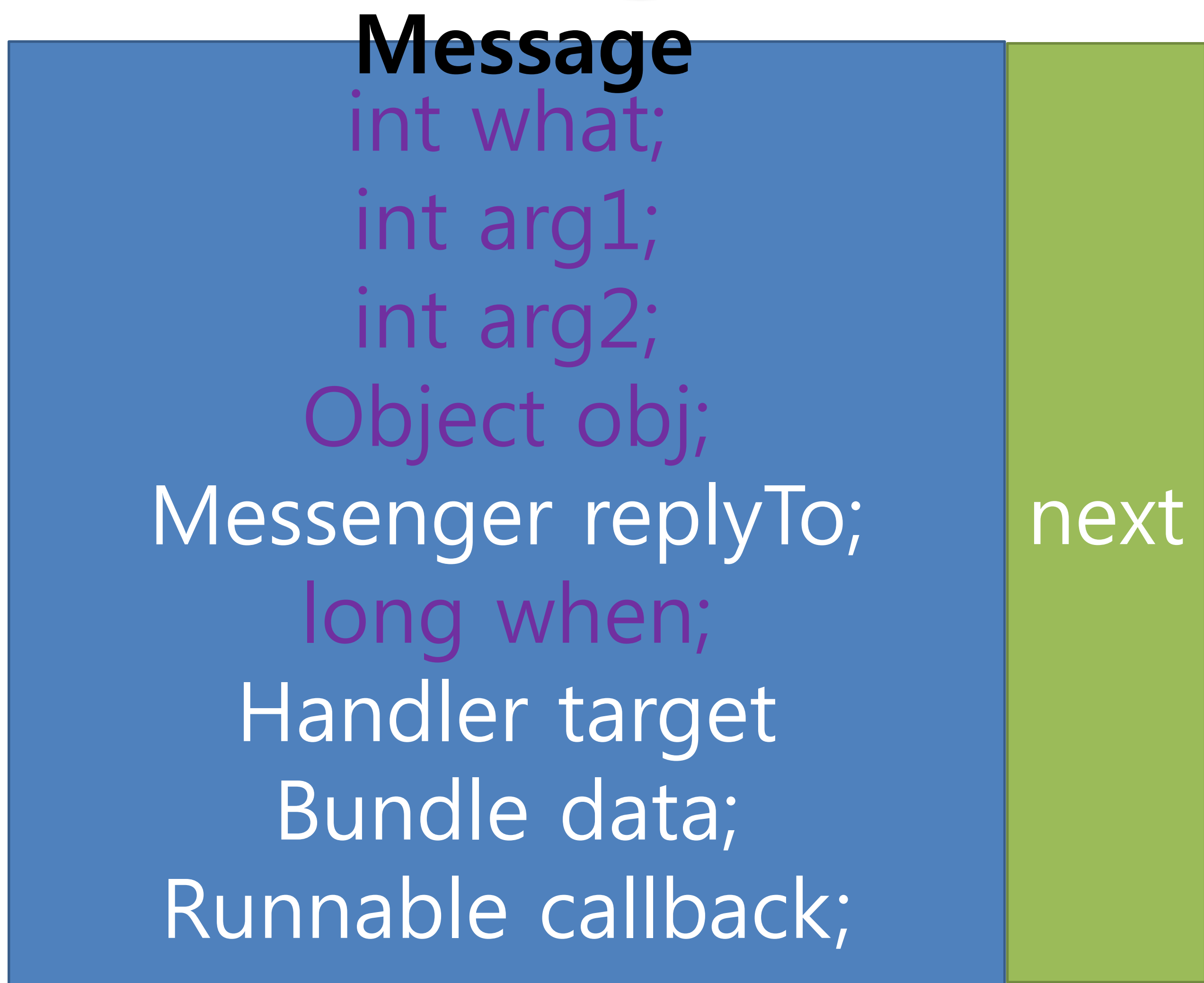
EventQueue분석



1. 항상 10개의 CacheQueue 를 유지함.(초기에 10개 까지는 계속 만들기만하고, 삭제하지 않음.)
2. Event 의 순서는 오래된 Event 부터 왼쪽에 놓인다.
3. 시간이 동일한 Event 가 존재할때는 존재하던 Event 의 오른쪽에 놓인다.



MessageQueue 분석



1. 항상 10개의 mPool을 유지함.(초기에 10개 까지는 계속 만들기만하고, 삭제하지 않음.)
2. Message 의 순서는 when 을 기준으로 한다.
(when 이 p.when 보다 크거나 같으면, while 문을 돌면서 p.when 보다 작을때까지 검색해서, 그곳에 넣는다.)
3. enqueueMessage(Message msg, long when) 에서 두번째 인자(when)가 0이거나, 첫 mMessage가 null이거나, when 이 첫 mMessage 의 when 보다 작으면, MessageQueue의 제일 앞에 놓인다.
4. 시간이 동일한 Message 가 존재할때는 존재하던 Message 의 오른쪽에 놓인다.

