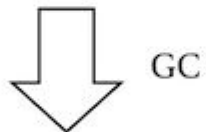


Garbage Collection

GC

 : ALIVE  : DEAD

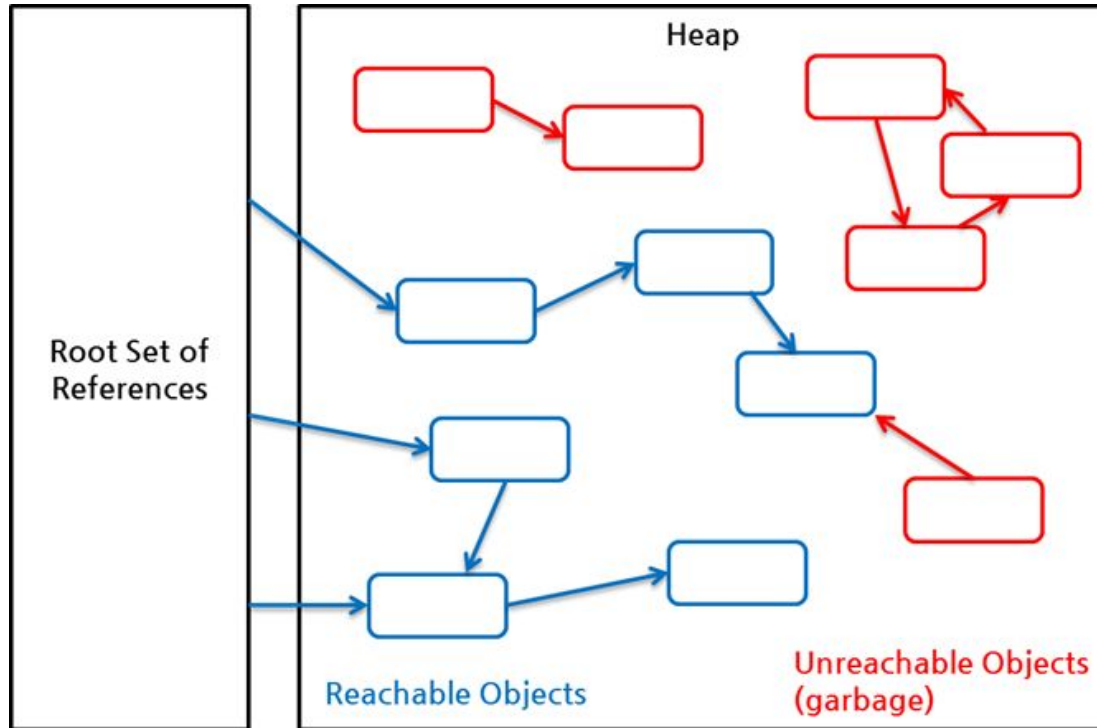


GC의 대상

```
for (int i = 0; i < 10000; i++) {~
    NewObject obj = new NewObject();
    obj.doSomething();
}
```

1. 모든 객체 참조가 null 인 경우
2. 객체가 블록 안에서 생성되고 블록이 종료된 경우
3. 부모 객체가 null이 된 경우, 자식 객체는 자동적으로 GC 대상이 된다.
4. 객체가 Weak 참조만 가지고 있을 경우
5. 객체가 Soft 참조이지만 메모리 부족이 발생한 경우

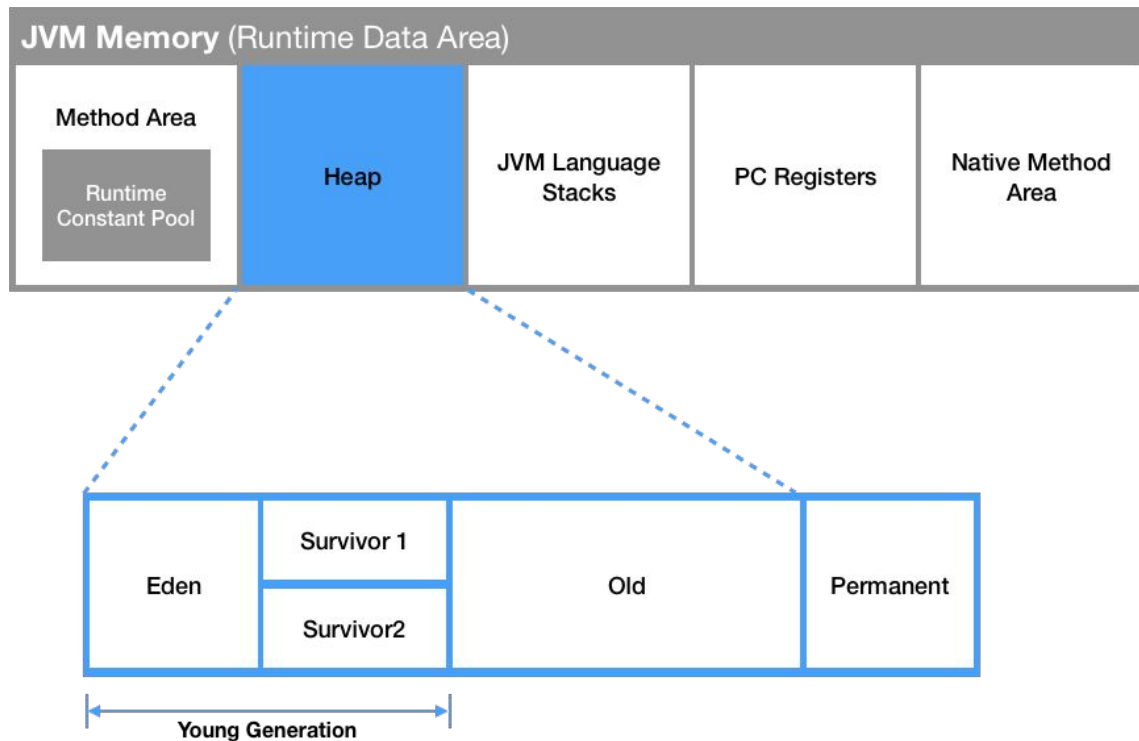
Unreachable Object



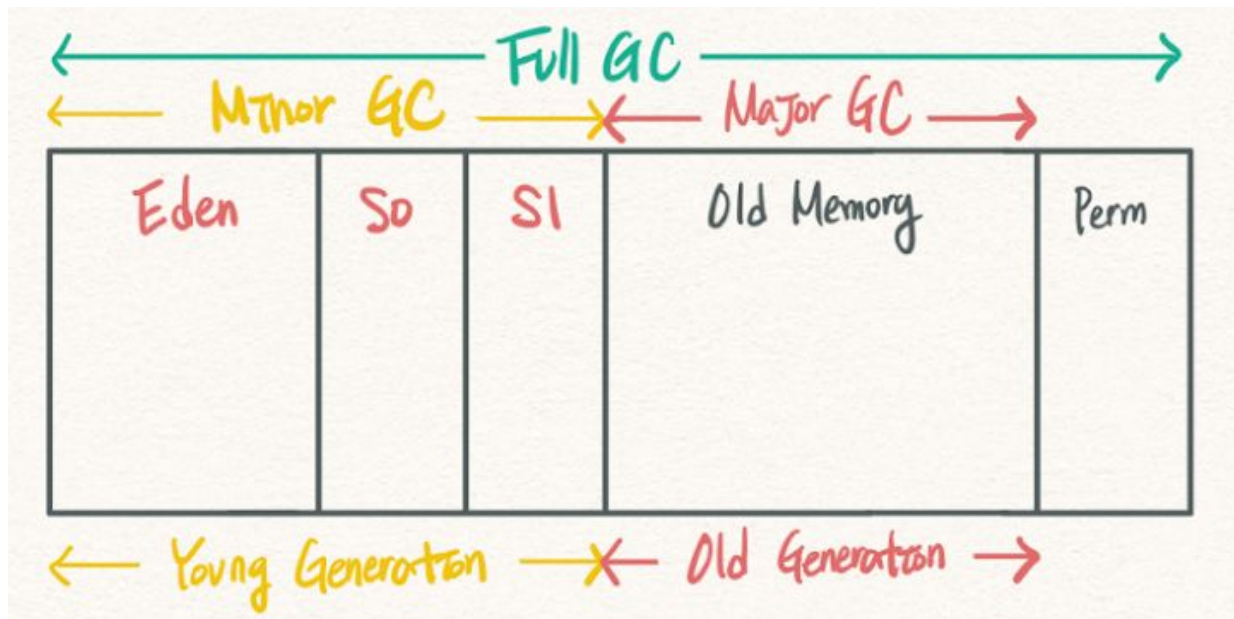
Weak generational hypothesis

1. 대부분의 객체는 금방 접근 불가능 상태(unreachable, garbage)가 된다.
2. 오래된 객체에서 젊은 객체로의 참조는 아주 적게 존재한다.

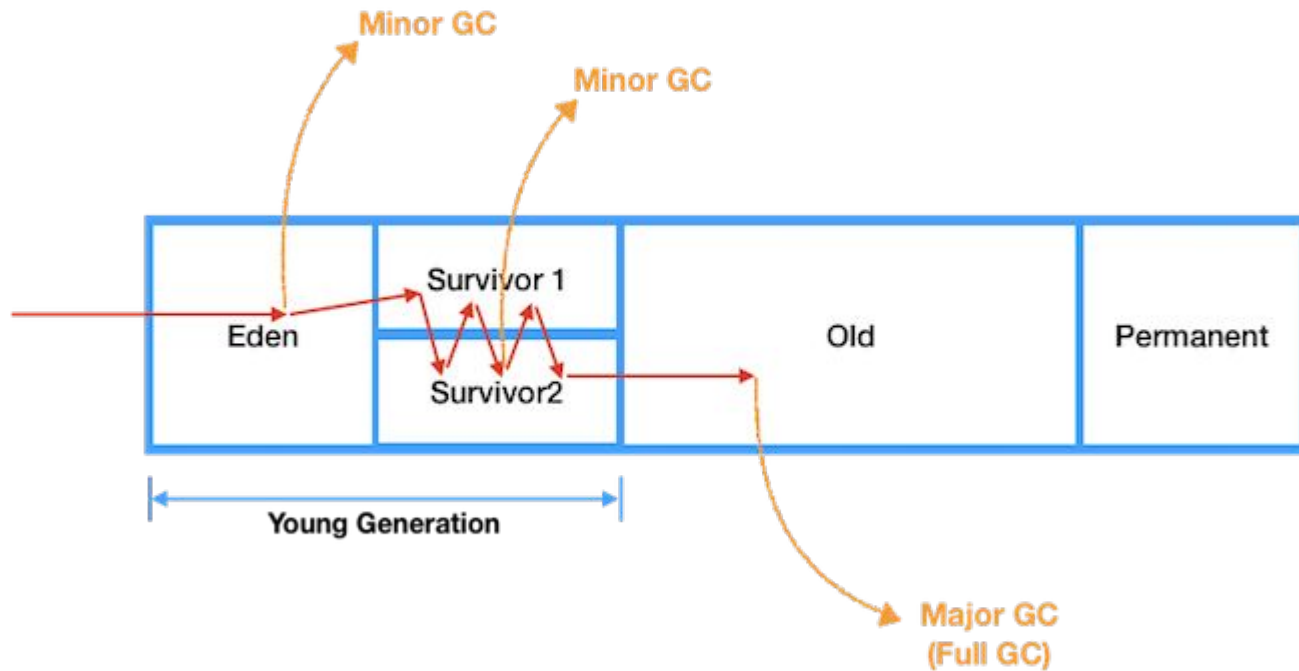
JVM 메모리 구조



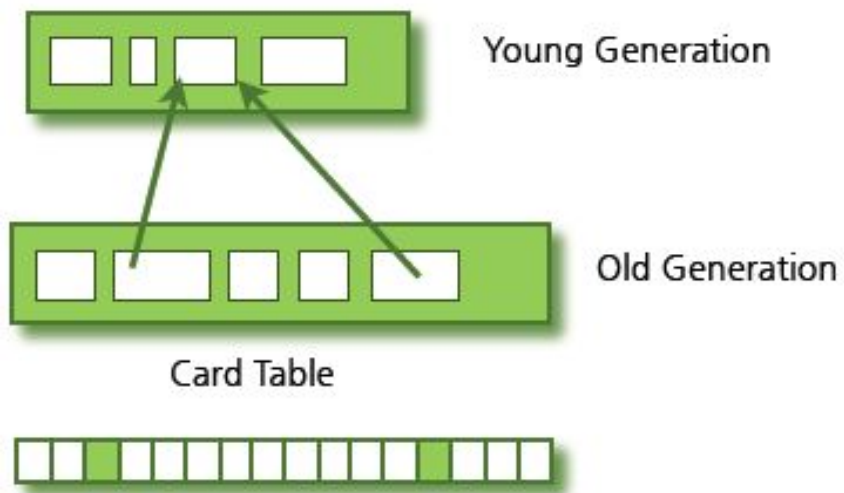
Young & Old 영역



객체의 이동



Card Table



Thank you

<https://taxol1203.github.io/technology/Garbage-Collection/#young-%EC%98%81%EC%97%AD%EA%B3%BC-old-%EC%98%81%EC%97%AD>