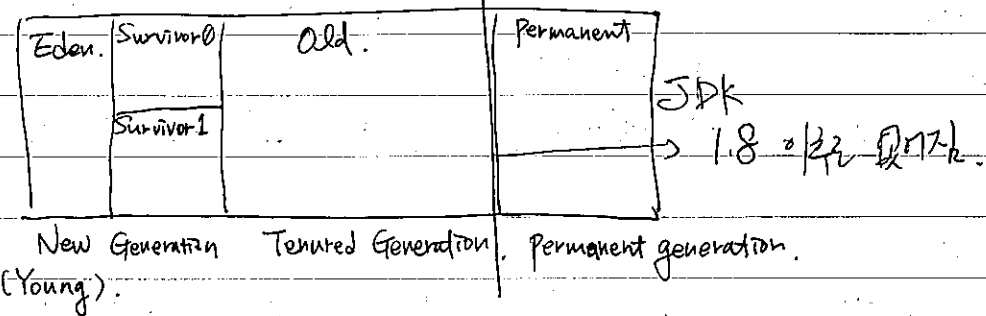


## < Garbage Collection >



- New (Young) Generation : 새롭게 생성된 객체가 위치. New를 이용하여 객체가 생성되면 Eden 영역에 위치하며, GC가 발생하면 살아남은 객체는 Survivor 영역으로 이동.
  - Eden 영역이 가득차게 되면 GC가 발생. (minor gc).
  - 살아남은 객체는 Survivor0 영역으로 이동.
  - 다음 minor gc 때 eden에서 살아남은 객체는 Survivor1 영역으로 이동.
  - 원래 survivor0에 있던 참조 객체들은 survivor1로 이동. (survivor0은 clear).
  - (→ 반드시 하나의 survivor 공간은 clear 됨.)

- Old Generation : ~~일정~~ 특정 age threshold를 넘은 객체들은 old generation으로 이동. old generation이 가득차면 major gc가 발생하며 garbage를 회수하고 compact 하여 memory를 확보.
  - 시간이 오래 걸리고 수행중 process가 중지된다. (stop-the-world)
  - GC 실행 스레드 이외의 스레드는 모두 멈춘다. → 프로그램 실행중

- GC의 종류 : Serial GC - 작은 메모리와 CPU일때 유리    메모리를 바꿀수 없음
- Parallel GC - 많은 자원을 쓸때 유리

- GC 튜닝 JVM 옵션
  - Xms : JVM 시작시 heap 크기
  - Xmx : 최대 heap 크기.
  - XX: NewRatio : New와 Old 비율
  - XX: EdenSurvivorRatio : Eden/Survivor 비율
  - XX: +UseSerialGC
  - XX: +UseParallelGC
  - XX: ParallelGCThreads = value

## < Generic >

- 타입을 파라미터화 하여 컴파일시 구체적인 타입이 결정되게 하는 것.
  - since Java 1.5
  - 컴파일시 강한 타입체크 가능! 예기치 못한 런타임 에러 방지.
  - 형변환에 의한 성능 저하 방지.

- 비밀의 생성에는 왜 generic을 사용할 수 없는가?
  - new 연산자는 비final 필드를 동적으로 생성하므로

## < Java의 특징 >

- 이식성이 높다 → JVM. 각 JVM은 해당 플랫폼에 맞게 수정.
- 객체 지향 언어
- Garbage collector.
- 클래스 동적 로딩.