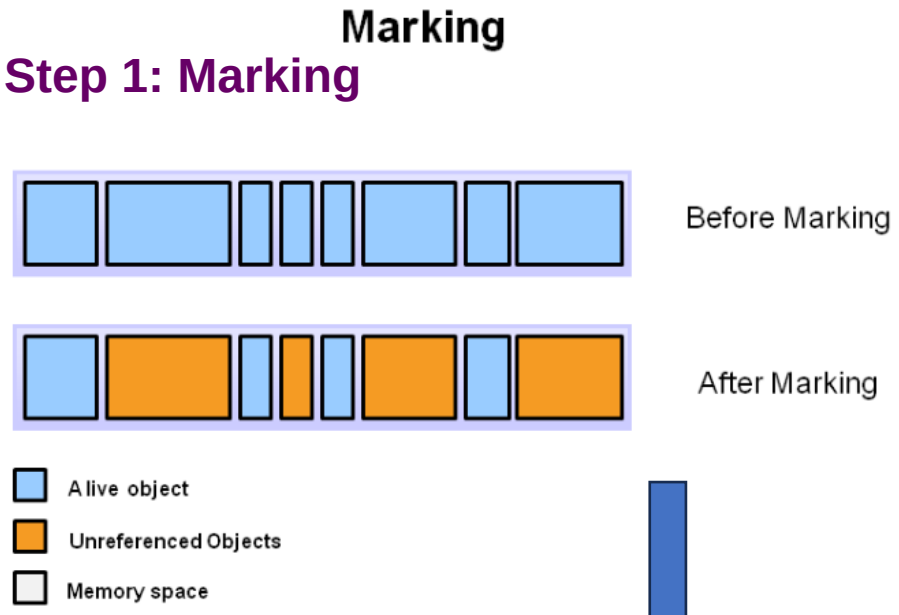
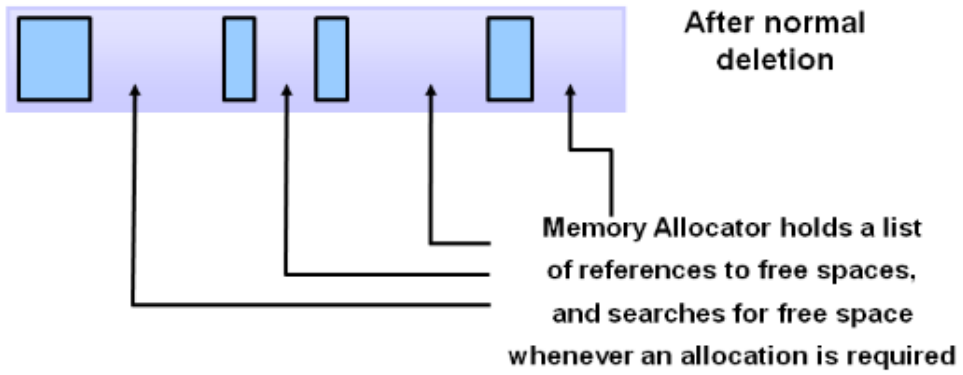
Garbage Collection

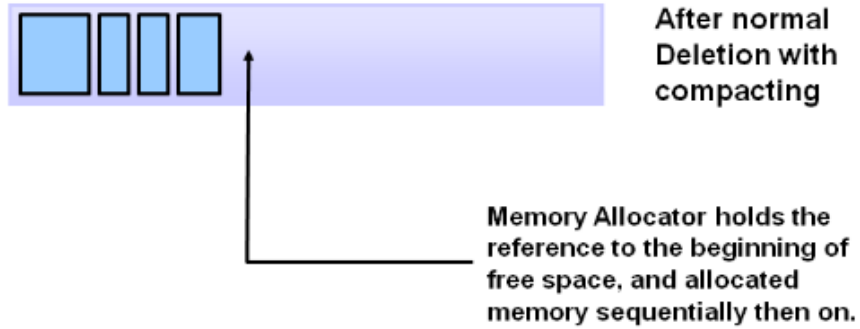
Step 1: Marking



Step 2: Normal Deletion



Step 2a: Deletion with Compacting



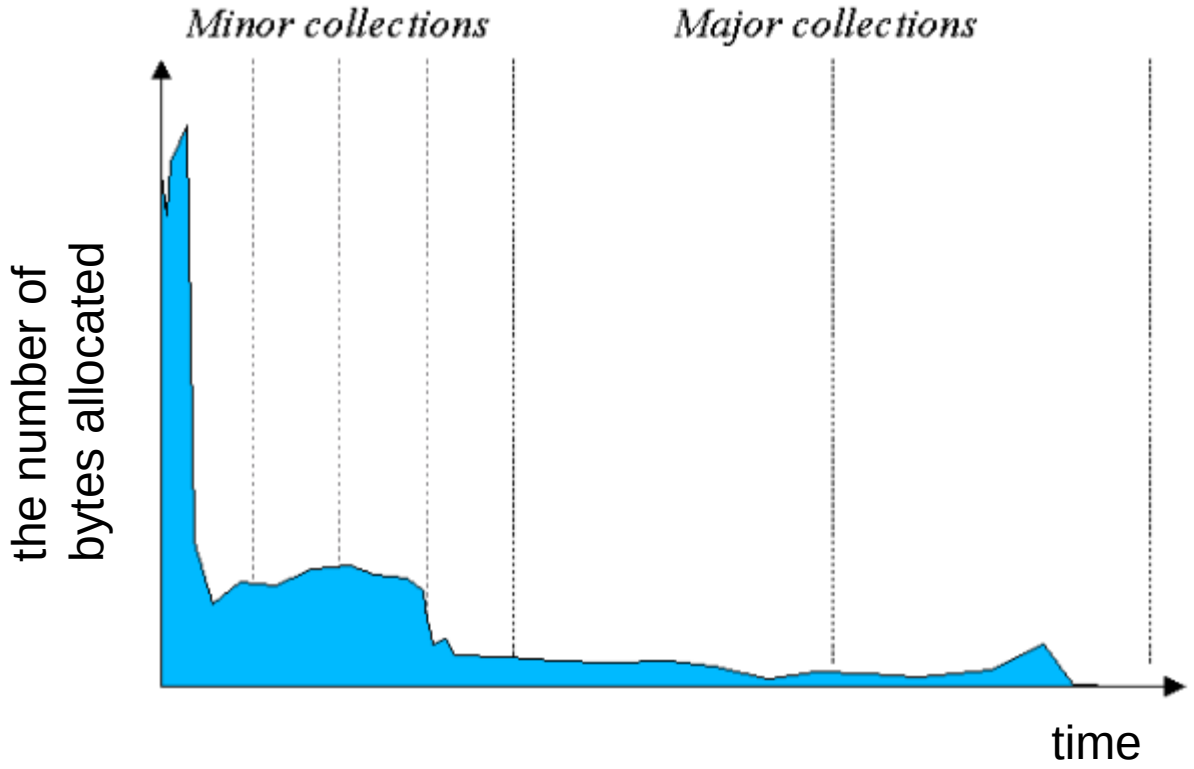
this makes new memory allocation much easier and faster.

Garbage Collection

Problem

- having to mark and compact all the objects in a JVM is inefficient.

Solution



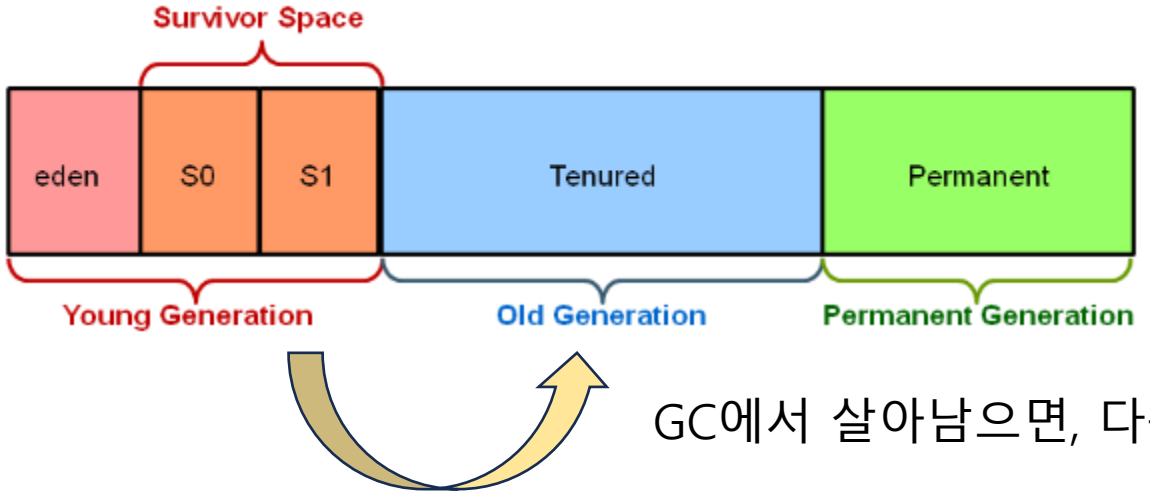
- fewer and fewer objects remain allocated over time
- In fact, most objects have a very short life

Garbage Collection

Stop the World Event

GC 동작 중에 모든 Thread는 중지된다.

JVM Heap



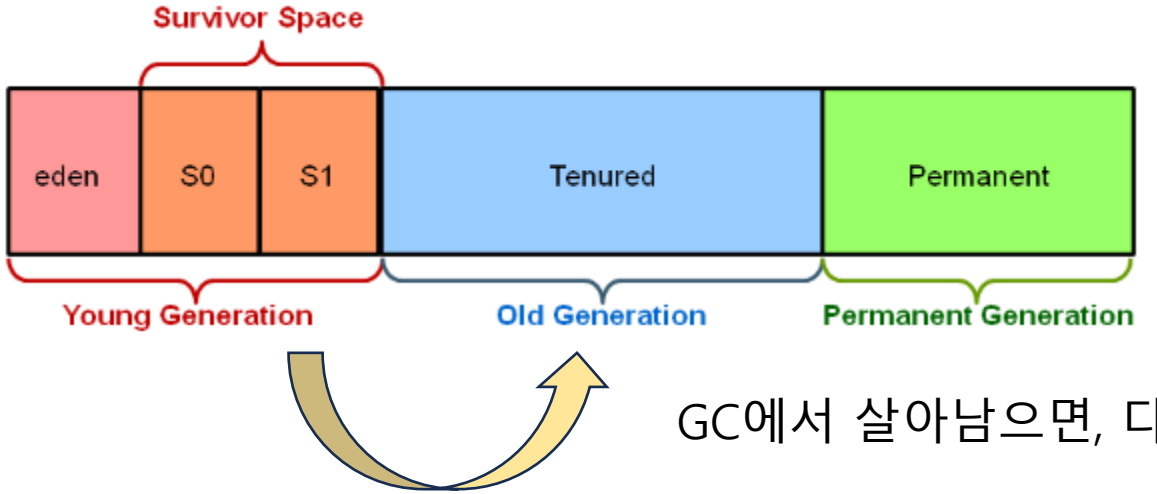
GC에서 살아남으면, 다음 영역으로 이동

Young Generation is

- where all new objects are allocated and aged.
- When the young generation fills up, this causes a ***minor garbage collection***
  Minor collections can be optimized assuming a high object mortality rate.
  A young generation full of dead objects is collected very quickly.
- Some surviving objects are aged and eventually move to the old generation.

Garbage Collection

JVM Heap



GC에서 살아남으면, 다음 영역으로 이동

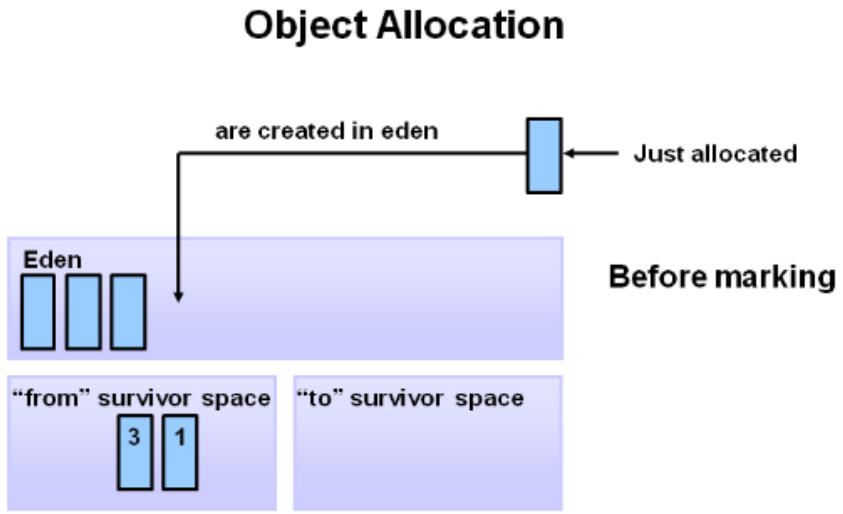
Old Generation

- is used to store long surviving objects.
- Eventually the old generation needs to be collected. This event is called a ***major garbage collection***.

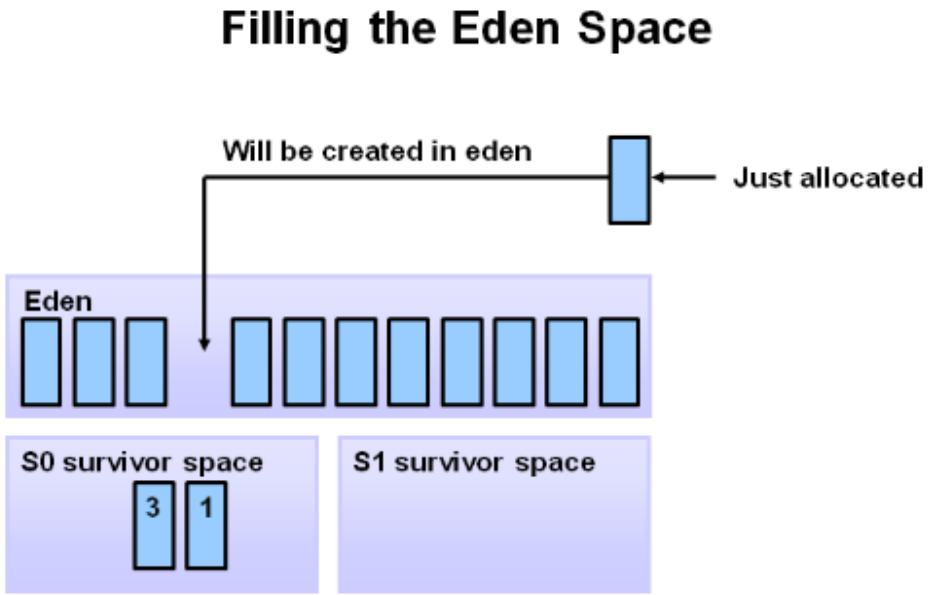
Permanent generation

- contains metadata required by the JVM to describe the classes and methods used in the application.
- is populated by the JVM at runtime based on classes in use by the application.

The Generational Garbage Collection Process

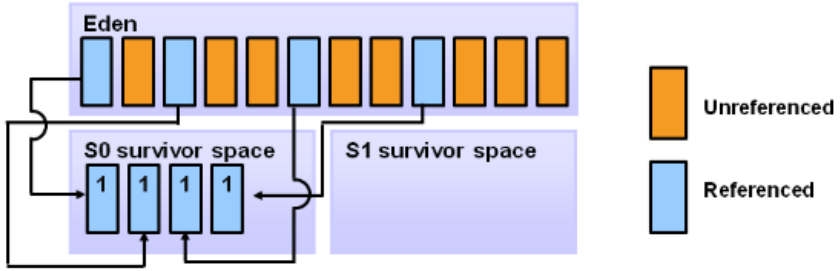


First, any new objects are **allocated to the eden space.**
Both survivor spaces start out empty.



When the eden space fills up,
a minor garbage collection is triggered.

GC 시작!

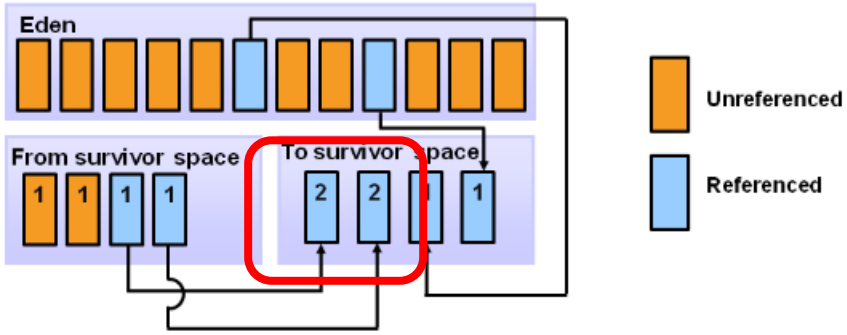


Referenced objects are moved to the first survivor space.
Unreferenced objects are deleted when the eden space is cleared.

살아 남으면, Survive 영역으로 이동

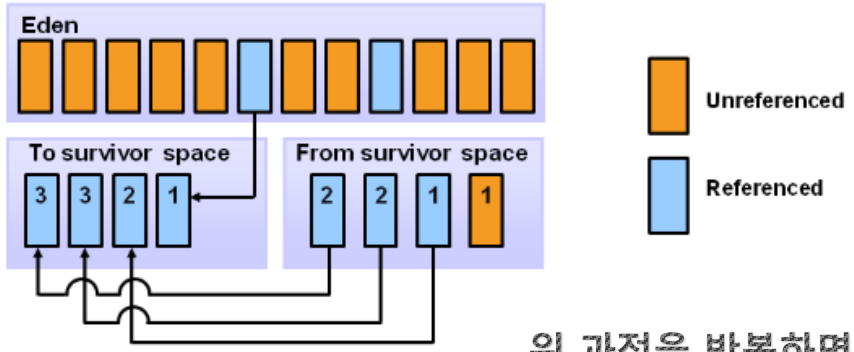
The Generational Garbage Collection Process

Object Aging



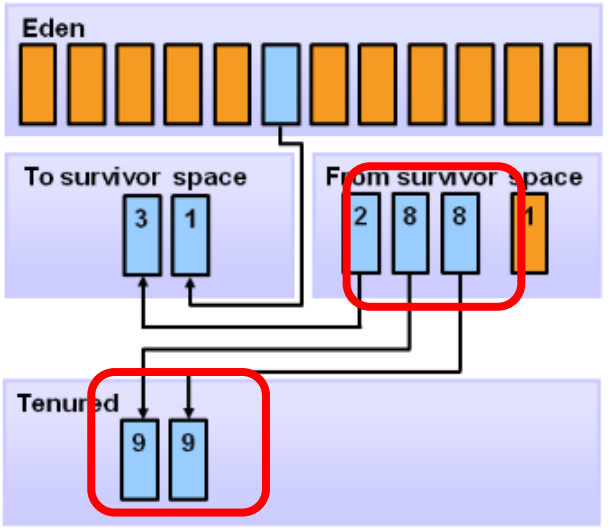
S0에서도 살아남은 object는 S1으로 이동하면서 count 증가!

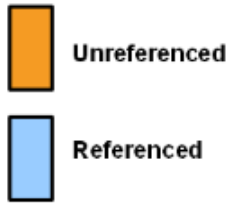
Additional Aging



위 과정을 반복하면서 계속 생존하면, count 증가

Promotion





계속 생존하면, 결국 Old Generation으로 이동한다.

Promotion