Сборка мусора по поколениям

JSP & Servlets
18 уровень, 4 лекция

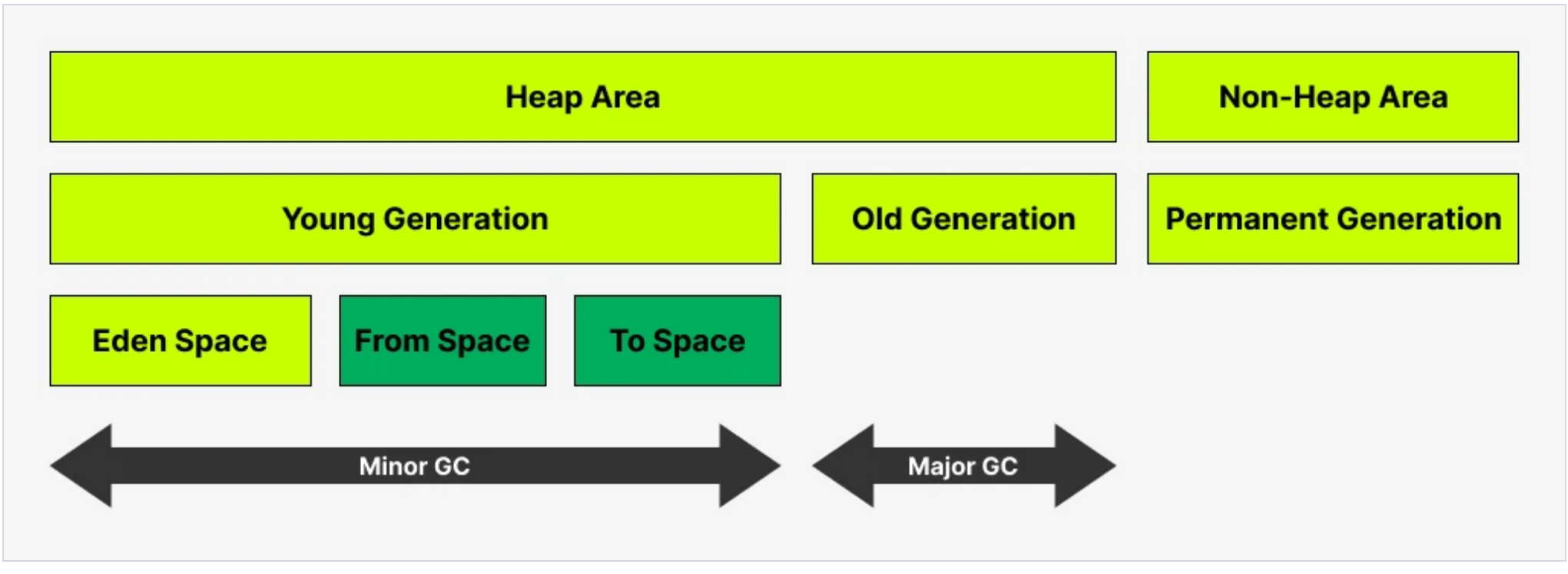
ОТКРЫТА

Работа с поколениями объектов

Java-сборщики мусора реализуют некоторую стратегию сбора мусора поколений, которая умеет классифицировать объекты по возрасту.

Такую необходимость (отмечать и уплотнять все объекты) в JVM можно назвать неэффективной. Так как по мере выделения большого количества объектов их список растет, что приводит к увеличению времени сбора мусора. Эмпирический анализ приложений показал, что большинство объектов в Java недолговечны.

Область памяти кучи в JVM разделена на три секции:



Молодое поколение

Вновь созданные объекты начинаются в молодом поколении. Молодое поколение далее подразделяется на две категории.

- **Пространство Эдема** — все новые объекты начинают здесь, им выделяется начальная память.

  Пространства выживших (FromSpace и ToSpace) — объекты перемещаются сюда из Эдема после того, как пережили один цикл сборки мусора.

Процесс, когда объекты собираются в мусор из молодого поколения, называется малым событием сборки мусора.

Когда пространство Эдема заполнено объектами, выполняется малая сборка мусора. Все мертвые объекты удаляются, а все живые — перемещаются в одно из оставшихся двух пространств. Малая GC также проверяет объекты в пространстве выживших и перемещает их в другое (следующее) пространство выживших.

Возьмем в качестве примера следующую последовательность.

1. В Эдеме есть объекты обоих типов (живые и мертвые).
2. Происходит малая GC — все мертвые объекты удаляются из Эдема. Все живые объекты перемещаются в пространство-1 (FromSpace). Эдем и пространство-2 теперь пусты.
3. Новые объекты создаются и добавляются в Эдем. Некоторые объекты в Эдеме и пространстве-1 становятся мертвыми.
4. Происходит малая GC — все мертвые объекты удаляются из Эдема и пространства-1. Все живые объекты перемещаются в пространство-2 (ToSpace). Эдем и пространство-1 пусты.

Таким образом в любое время одно из пространств для выживших всегда пусто. Когда выжившие объекты достигают определенного порога перемещения по пространствам выживших, они переходят в старшее поколение.

Для установки размера молодого поколения можно воспользоваться флагом **-Xmn**.

Старшее поколение

Объекты, которые живут значительное время (например, большую часть времени жизни программы) в конечном итоге становятся старшими объектами – долгожителями. Оно также известно как штатное поколение и содержит объекты, которые долгое время оставались в пространствах выживших.

Пороговое значение срока службы объекта определяет, сколько циклов сборки мусора он должен пережить, прежде чем будет перемещен в старшее поколение. Процесс, когда объекты отправляются в мусор из старшего поколения, называется основным событием сборки мусора.

Для установки начального и максимального размера памяти кучи можно воспользоваться флагами **-Xms** и **-Xmx**.

Так как Java задействует сборку мусора по поколениям, то, чем больше событий сборки мусора переживает объект, тем дальше он продвигается в куче. Он начинает в молодом поколении и в конечном итоге заканчивает в штатном поколении, если проживет достаточно долго.

Чтобы понять продвижение объектов между пространствами и поколениями, рассмотрим следующий пример:

Когда объект создается, он сначала помещается в пространство Эдема молодого поколения.

Как только произойдет малая сборка мусора, живые объекты из Эдема перемещаются в пространство FromSpace. Когда происходит следующая малая сборка мусора, живые объекты как из Эдема, так и из пространства перемещаются в пространство ToSpace.

Этот цикл продолжается определенное количество раз. Если объект все еще “в строю” после этого момента, следующий цикл сборки мусора переместит его в пространство старшего поколения.

Постоянное поколение и мета-пространство

Метаданные, такие как классы и методы, хранятся в постоянном поколении. JVM заполняет его во время выполнения на основе классов, используемых приложением. Классы, которые больше не используются, могут переходить из постоянного поколения в мусор.

Для установки начального и максимального размера постоянного поколения вы можете воспользоваться флагами **-XX:PermGen** и **-XX:MaxPermGen**.

Мета-пространство

Начиная с Java 8, на смену пространству постоянного поколения (PermGen) приходит пространство памяти MetaSpace. Реализация отличается от PermGen — это пространство кучи теперь изменяется автоматически.

Это позволяет избежать проблемы нехватки памяти у приложений, которая возникает из-за ограниченного размера пространства PermGen в куче. Память мета-пространства может быть собрана как мусор, и классы, которые больше не используются, будут автоматически очищены, когда мета-пространство достигнет максимального размера.

‹ Предыдущая лекция

Следующая лекция ›

JavaCoder

Введите текст комментария

Ant Уровень 25 — 25 декабря 2022, 15:37

[Сборка мусора в Java: что это такое и как работает в JVM]
Полная статья

Ответить 0

MICRO_MVP_10011 Уровень 37 — 1 декабря 2022, 20:56

голодные игры)

Ответить 0

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ