

Сборка мусора в Java

JSP & Servlets
18 уровень, 3 лекция

ОТКРЫТА

Вспоминаем, что такое сборка мусора в Java

Сборка мусора — это процесс восстановления заполненной памяти среды выполнения путем уничтожения неиспользуемых объектов.

Иногда программист может забыть уничтожить бесполезные объекты, и выделенная им память не освобождается. Расходуется все больше и больше системной памяти, и в конечном итоге она больше не выделяется. Такие приложения страдают от “утечек памяти”.

После определенного момента памяти уже не хватает для создания новых объектов и программа нештатно завершается из-за **OutOfMemoryError**.



Сборка мусора в Java — это процесс, с помощью которого программы Java автоматически управляют памятью. Java-программы компилируются в байт-код, который запускается на виртуальной машине Java (JVM).

Когда Java-программы выполняются на JVM, объекты создаются в куче, которая представляет собой часть памяти, выделенную для них.

Пока Java-приложение работает, в нем создаются новые объекты. В конце-концов некоторые объекты перестают быть нужны. Можно сказать, что в любой момент времени память кучи состоит из двух типов объектов.

- **Живые** — эти объекты используются, на них ссылаются откуда-то еще.
- **Мертвые** — эти объекты больше нигде не используются, ссылок на них нет.

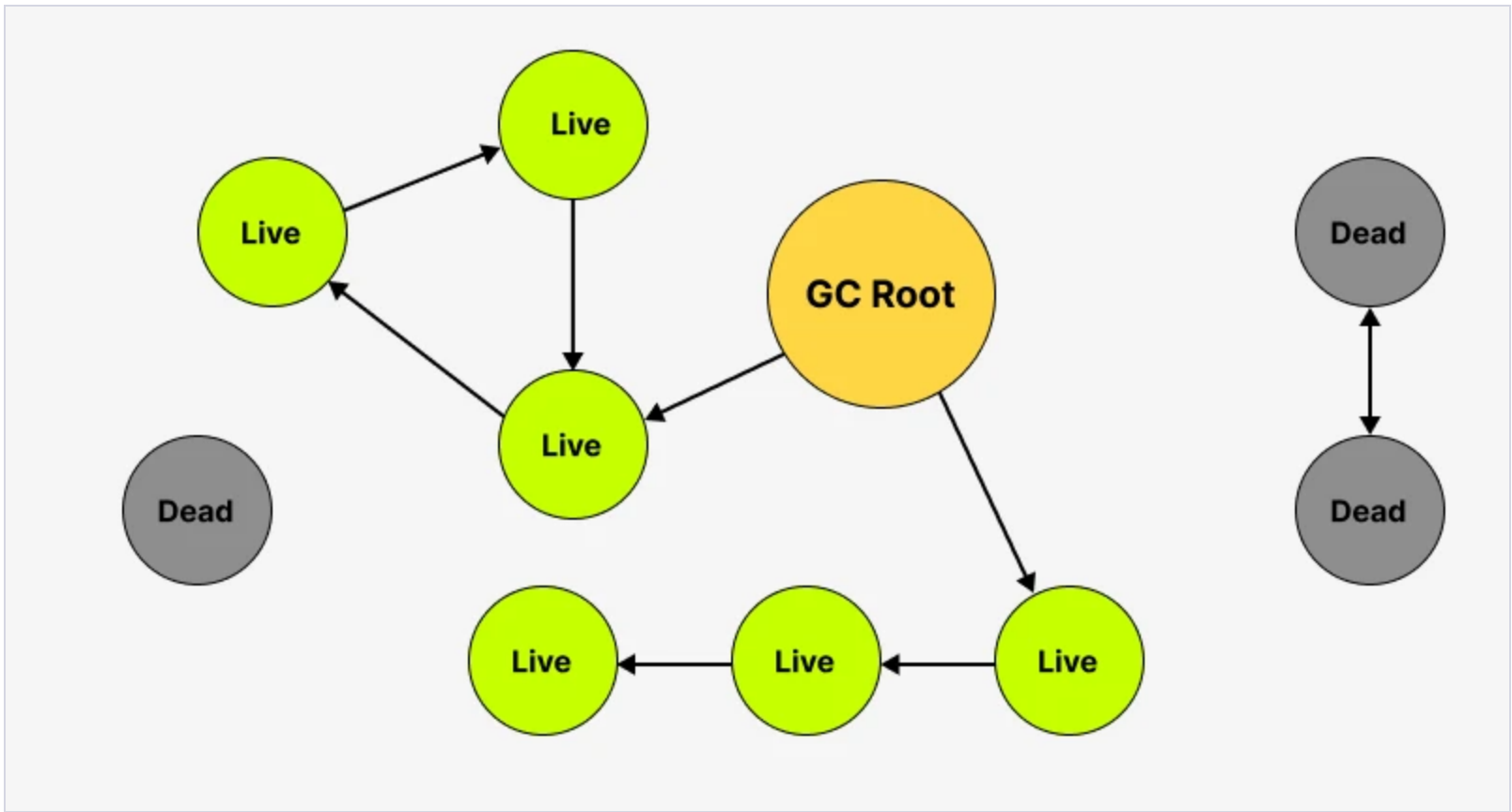
Сборщик мусора находит эти неиспользуемые объекты и удаляет их, чтобы освободить память.

Сборка мусора в Java — **автоматический процесс**. Программисту не нужно явно отмечать объекты, подлежащие удалению.

Каждая JVM может реализовать собственную версию сборки мусора. Однако сборщик должен соответствовать стандартной спецификации JVM для работы с объектами, присутствующими в памяти кучи, для маркировки или идентификации недостижимых объектов и их уничтожения через уплотнение.

Достижимость объектов

Для того, чтобы признать объект живым — наличия ссылок недостаточно. Все потому, что одни мертвые объекты могут ссылаться на другие мертвые объекты. Именно поэтому нужно, чтобы среди всех ссылок на объект, была хотя бы одна от “живого” объекта.



Сборщики мусора работают с концепцией GC Roots (**корней сбора мусора**) для того, чтоб различать живые и мертвые объекты. Есть 100% живые объекты и от них идут ссылки, которые оживляют другие объекты и так далее.

Примеры таких корней:

- Классы, которые загружаются системным загрузчиком классов.
- Живые потоки.
- Параметры методов, которые выполняются в данный момент и локальные переменные.
- Объекты, которые применяются в качестве монитора для синхронизации.
- Объекты, которые удерживаются из сборки мусора для некоторых целей.
- Сборщик мусора просматривает весь граф объектов в памяти, начиная с этих корней и следуя ссылкам на другие объекты.

Этапы сборки мусора в Java

Стандартная реализация сборки мусора имеет три этапа.

1. Помечаем объекты как живые

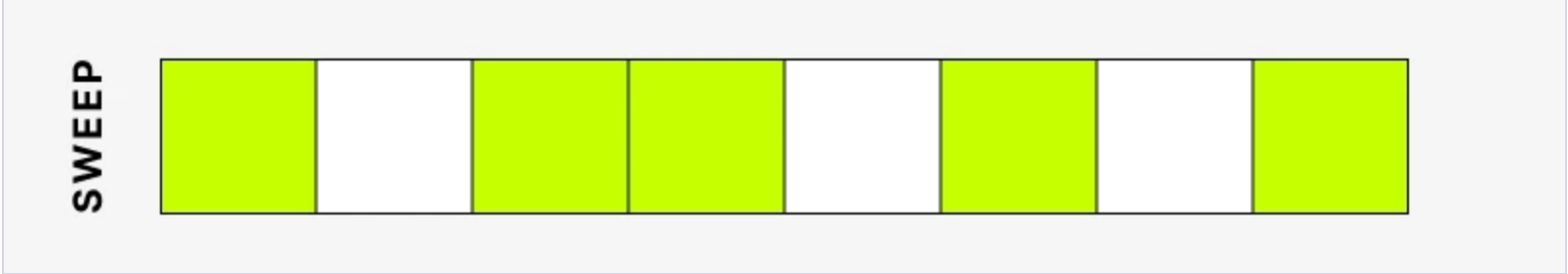
На этом этапе сборщик мусора (GC) должен идентифицировать все живые объекты в памяти путем обхода графа объектов.

Когда он посещает объект, то помечает его как доступный и, следовательно, живой. Все объекты, недоступные из корней GC, рассматриваются как кандидаты на сбор мусора.



2. Зачистка мертвых объектов

После фазы разметки пространство памяти занято либо живыми (посещенными), либо мертвыми (не посещенными) объектами. Фаза зачистки освобождает фрагменты памяти, которые содержат эти мертвые объекты.

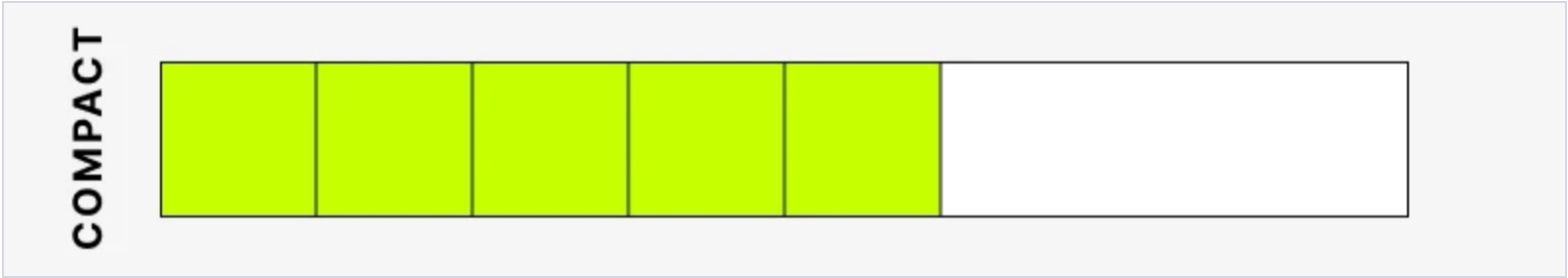


3. Компактное расположение оставшихся объектов в памяти

Не обязательно, чтобы мертвые объекты, которые были удалены на предыдущей фазе, находились рядом друг с другом. Таким образом, ты рискуешь получить фрагментированное (полупустое) пространство памяти.

Но, конечно же, предусмотрев это, есть возможность уплотнить память в момент, когда сборщик мусора удаляет мертвые объекты. Оставшиеся будут располагаться в непрерывном блоке в начале кучи.

Процесс уплотнения облегчает последовательное выделение памяти для новых объектов.



[← Предыдущая лекция](#)

[Следующая лекция →](#)

−

+16

+

Комментарии

популярные

новые

старые

JavaCoder

Введите текст комментария



У ЭТОЙ СТРАНИЦЫ ЕЩЕ НЕТ НИ ОДНОГО КОММЕНТАРИЯ

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский 

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

