

# C# Professional

Сборщик мусора (Garbage Collector)

# C# Professional

После урока обязательно



Повторите этот урок в видео формате на  
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на  
[TestProvider.com](http://testprovider.com)

## Сборщик мусора (Garbage Collector)

# C# Professional

## Garbage Collection

Специальный механизм, называемый *сборщиком мусора* (*garbage collector*), периодически освобождает память, удаляя объекты, которые уже не будут востребованы приложением — то есть производит «сбор мусора».

Сборка мусора была впервые применена Джоном Маккарти в 1959 году в среде программирования на разработанном им функциональном языке программирования Lisp. Впоследствии она применялась в других системах программирования и языках, преимущественно — в функциональных и логических.

# C# Professional

## Garbage Collection

Любая программа использует ресурсы - файлы, буферы в памяти, пространство экрана, сетевые подключения, базы данных и т. п.

В объектно-ориентированной среде каждый тип идентифицирует некий доступный этой программе ресурс.

Чтобы им воспользоваться, должна быть выделена память для представления этого типа.

# C# Professional

## Garbage Collection

Для доступа к ресурсу вам нужно:

1. Выделить память для типа, представляющего ресурс.
2. Инициализировать выделенную память, установив начальное состояние ресурса.
3. Использовать ресурс, обращаясь к членам его типа.
4. Ликвидировать состояние ресурса.
5. Освободить память.

# C# Professional

## Garbage Collection

Сборка мусора (garbage collection) полностью освобождает разработчика от необходимости следить за использованием и своевременным освобождением памяти.

# C# Professional

## Garbage Collection

CLR требует выделять память для всех ресурсов из так называемой управляемой кучи (managed heap).

От кучи исполняющей среды языка C она отличается лишь тем, что разработчику из управляемой кучи удалять объекты не нужно.

Став ненужными приложению, они удаляются автоматически.



# C# Professional

## Garbage Collection

При инициализации процесса CLR резервирует непрерывную область адресного пространства, которая изначально не соответствует никакой физической памяти. Это и есть управляемая куча.

Она также поддерживает указатель, который называется `NextObjPtr`. Он определяет, где в куче будет выделена память для следующего объекта, и изначально указывает на базовый адрес этой зарезервированной области адресного пространства.

# C# Professional

## Garbage Collection

Команда C# new (IL newobj) создает объект.

После получения этой команды CLR:

1. Подсчитывает количество байтов, необходимых для размещения полей типа
2. Прибавляет к полученному значению количество байтов, необходимых для размещения системных полей объекта.
3. Если в управляемой куче достаточно места для объекта, ему выделяется память, начиная с адреса, на который ссылается указатель NextObjPtr, а занимаемые им байты обнуляются.
4. Вызывается конструктор типа, и IL-команда newobj возвращает адрес объекта (также перемещается NextObjPtr).

# C# Professional

## Garbage Collection

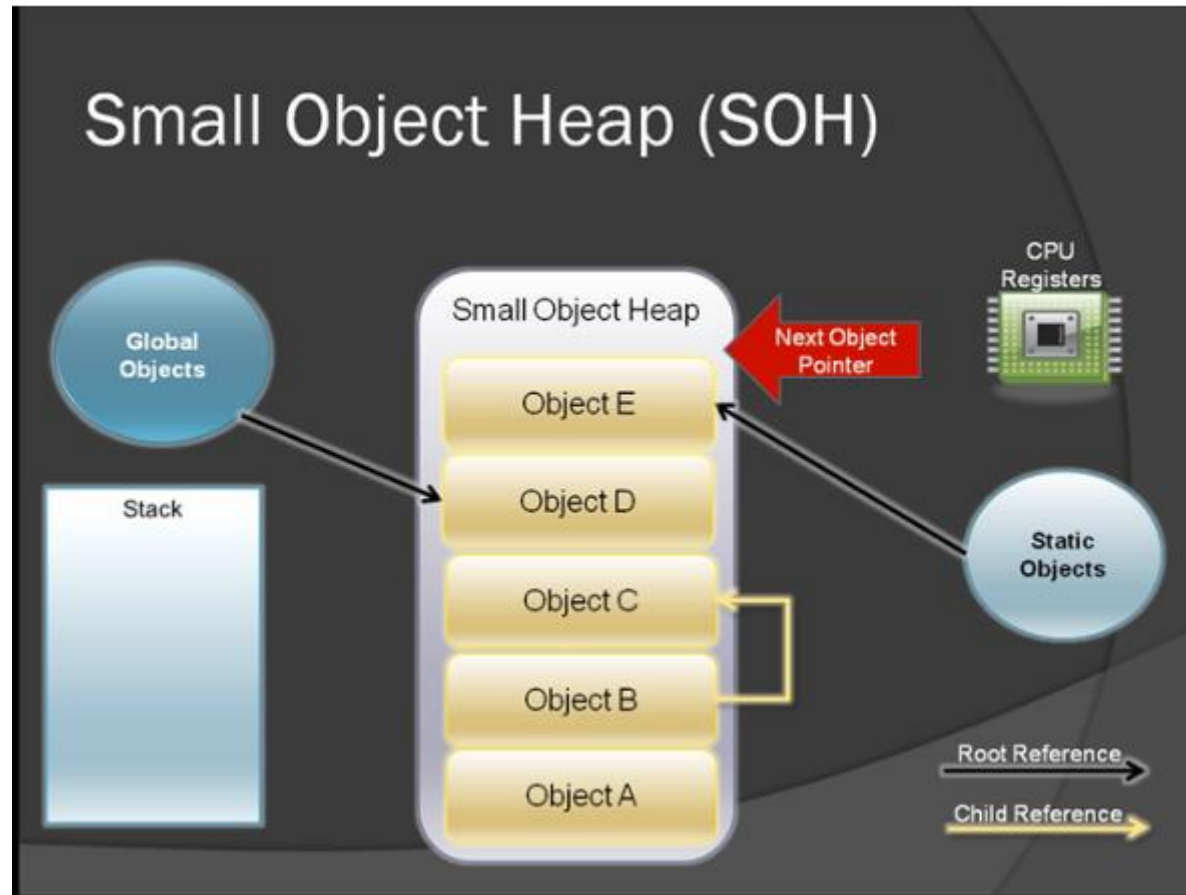
При вызове оператора `new` в области, выделяемой под объект, может не хватать свободного адресного пространства.

Куча выясняет объем недостающей памяти и добавляет байты, необходимые для объекта, к адресу, заданному указателем `NextObjPtr`.

Если результирующее значение выходит за пределы адресного пространства, значит, куча заполнена и следует выполнить сборку мусора.

# C# Professional

## Garbage Collection



# C# Professional

## Garbage Collection

Сборщик мусора проверяет наличие в куче больше не используемых приложением объектов, чтобы освободить занятую ими память.

# C# Professional

## Garbage Collection

Сборщик переходит к этапу сборки мусора, называемому маркировкой (marking).

Он проходит по стеку потока и проверяет все корни маркируя объекты.

После маркировки корня и объекта, сборщик мусора проверяет следующий корень и продолжает маркировать объекты.

Встретив уже маркированный объект, сборщик мусора останавливается.

# C# Professional

## Garbage Collection

Затем сборщик переходит к следующему этапу сборки мусора, называемому сжатием.

Теперь он проходит кучу линейно в поисках непрерывных блоков немаркированных объектов, то есть мусора.

Небольшие блоки сборщик не трогает, а в больших непрерывных блоках он перемешает вниз все немусорные объекты, сжимая при этом кучу.

## Q&A



# Информационный видеосервис для разработчиков программного обеспечения

