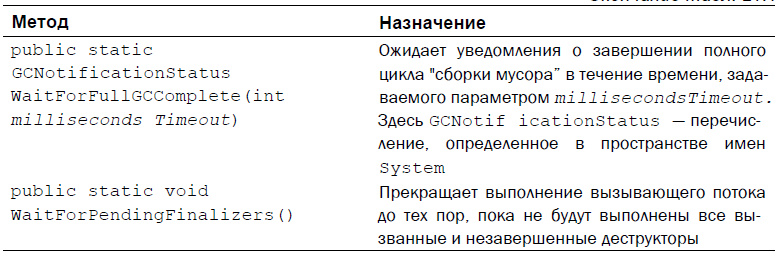
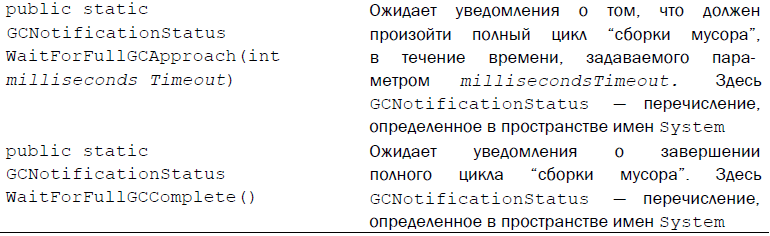
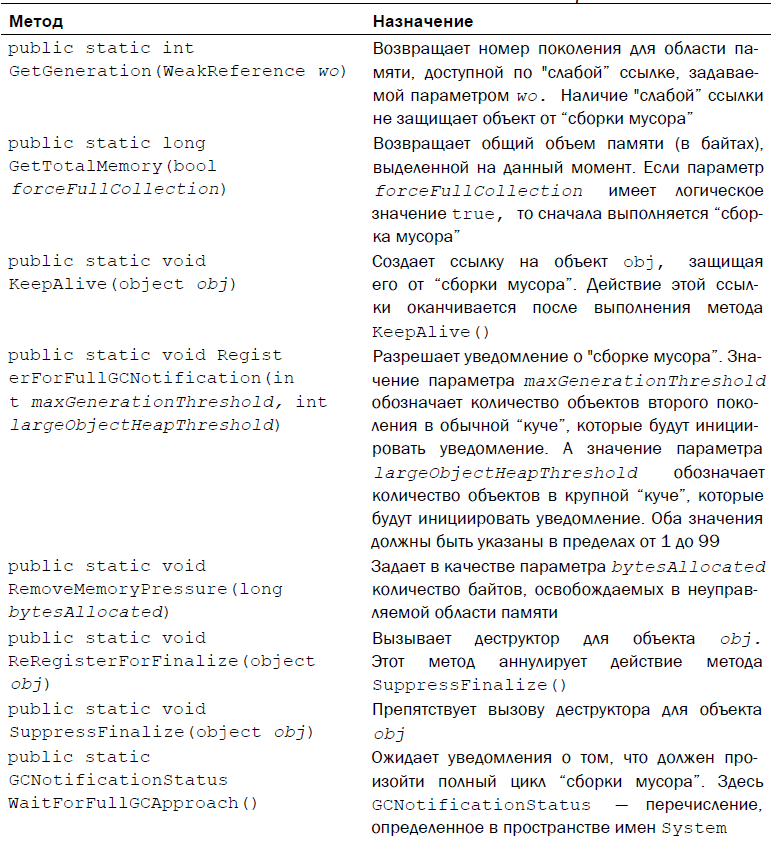
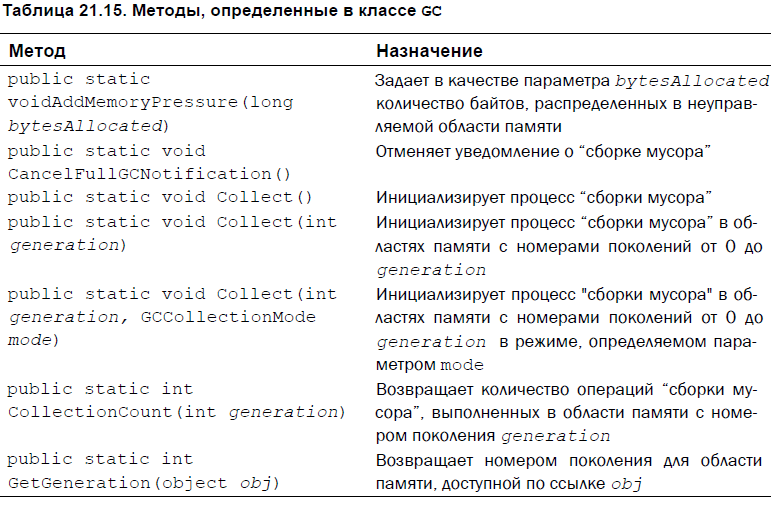
В классе GC инкапсулируются средства *"*сборки мусора*"*. Методы, определенные

в этом классе, перечислены в табл. 21.15.



Кроме того, в классе GC определяется следующее доступное только для чтения

свойство:

public static int MaxGeneration { get; }

Свойство MaxGeneration содержит максимальный номер поколения, доступный

для системы. Номер поколения обозначает возраст выделенной области памяти. Чем

старше выделенная область памяти, тем больше номер ее поколения. Номера поколений

позволяют повысить эффективность работы системы "сборки мусора".

В большинстве приложений возможности класса GC не используются. Но в особых

случаях они оказываются весьма полезными. Допустим, что требуется организовать

принудительную "сборку мусора" с помощью метода Collect() в выбранный момент

времени. Как правило, "сборка мусора" происходит в моменты, не указываемые

специально в программе. А поскольку для ее выполнения требуется некоторое время,

то желательно, чтобы она не происходила в тот момент, когда решается критичная по

времени задача. С другой стороны, "сборку мусора" и другие вспомогательные операции

можно выполнить во время простоя программы. Имеется также возможность

регистрировать уведомления о приближении и завершении "сборки мусора".

Для проектов с неуправляемым кодом особое значение имеют два следующих метода

из класса GC:AddMemoryPressure() и RemoveMemoryPressure(). С их помощью

указывается большой объем неуправляемой памяти, выделяемой или освобождаемой

в программе. Особое значение этих методов состоит в том, что система управления

памятью не контролирует область неуправляемой памяти. Если программа выделяет

большой объем неуправляемой памяти, то это может сказаться на производительности,

поскольку системе ничего неизвестно о таком сокращении объема свободно

доступной памяти. Если же большой объем неуправляемой памяти выделяется с помощью

метода AddMemoryPressure(), то система CLR уведомляется о сокращении

объема свободно доступной памяти. А если выделенная область памяти освобождается

с помощью метода RemoveMemoryPressure(), то система CLR уведомляется о соответствующем

восстановлении объема свободно доступной памяти. Следует, однако,

иметь в виду, что метод RemoveMemoryPressure() необходимо вызывать только для

уведомления об освобождении области неуправляемой памяти, выделенной с помощью

метода AddMemoryPressure().