

Цикл — это управляющая конструкция, предназначенная для организации многократного исполнения набора инструкций.

Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).

Итерация – это один проход цикла.

`goto` (англ. *go to* — «перейти к») — оператор безусловного перехода к определённой точке программы, обозначенной номером строки либо меткой.

В 1968 году, в своей статье «Обоснование пагубности оператора `goto`», Эдсгер Дейкстра отметил, что качество кода обратно пропорционально количеству `goto`, используемых в коде. Дейкстра утверждал (это просто его мнение), что: корректность кода без `goto` доказать легче

код с операторами `goto` трудно форматировать

`goto` влияет на логическую структуру программы

применение `goto` препятствует оптимизации, выполняемой компилятором

`goto` усложняет анализ кода

На практике применение оператора `goto`, приводит к нарушению принципа хода алгоритма строго сверху вниз.

Сторонники `goto` выступают за осторожное применение оператора, при определенных условиях.

Большинство аргументов против `goto`, не говорит о полном отказе от этого оператора, а предостерегает от неразборчивого его использования.

Хорошее программирование не означает исключение всех `goto`.

Стремление к коду без `goto` не должно быть целью.

Десятилетия исследований оператора `goto` не смогли подтвердить его вредоносность, а теоретические и экспериментальные доводы, выдвигаемые против этого оператора, оказались не убедительными.

И наконец, операторы `goto` входят во множество современных языков.

Microsoft чаще использует `goto` при автогенерации кода, когда этот код не предназначен для восприятия и редактирования человеком. Корректность такого кода определяется корректностью создающего его инструмента.

Использование `goto` – это вопрос религии.

Цикл, с предусловием `while` – это цикл, который выполняется до тех пор, пока условие удовлетворяет истинности. Условие проверяется до выполнения тела цикла. Если изначально условие не удовлетворяет истинности, то тело цикла `while` ни разу не выполнится.

Цикл, с постусловием `do-while` – это цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело `do-while` выполняется хотя бы один раз.

Цикл со счетчиком `for` – это цикл, в котором переменная – счетчик итераций цикла, с определенным шагом, изменяет свое значение до заданного конечного значения.

Блок выражений цикла `for`, содержит три выражения:

`for` (начальное-выражение; условное-выражение; выражение-цикла) { тело цикла } или в другой нотации это звучит так:

`for` (инициализация; условие; модификация) { тело цикла }

В теле цикла `for`, разрешено изменение значения начального выражения (т.е. счетчика итераций).

Циклы с выходом из середины, отсутствуют в C#, но такой цикл можно смоделировать при помощи любого существующего цикла и оператора досрочного выхода `break` или оператора безусловного перехода `goto`.

Досрочный выход из цикла. Команда досрочного выхода из цикла `break`,

применяется тогда, когда необходимо прервать выполнение цикла, в котором

условие выхода еще не достигнуто. Работу цикла есть смысл прерывать, если например, при выполнении тела цикла, обнаруживается ошибка, после которой дальнейшее выполнение цикла не имеет смысла.

Оператор досрочного выхода из цикла `break`, применяется только для того цикла в котором он непосредственно находится.

Пропуск итерации. Оператор пропуска итерации `continue`, применяется, когда необходимо пропустить все команды до конца тела цикла.

Неструктурные средства безусловных переходов: `goto`, `break`, `continue`. С точки зрения структурного программирования, команды досрочного выхода из цикла и продолжения итерации считаются избыточными. Нужно стараться моделировать их действия чисто структурными средствами – условиями и циклами.

С точки зрения Эдсгера Дейкстры (это просто его мнение), сам факт использования в программе неструктурных средств, будь то оператор безусловного перехода `goto`, или одной из его специализированных форм – операторов `break` и `continue`, является свидетельством недостаточно проработанного алгоритма решения задачи.

Несмотря на свою ограниченную полезность и возможность замены на другие языковые конструкции, команды пропуска итерации и, особенно, досрочного выхода из цикла в отдельных случаях оказываются полезны, именно поэтому они сохраняются в C# и других современных языках программирования.

Совместные циклы - `foreach`. Эта разновидность циклов будет детально рассмотрена на 15 уроке курса C# Базовый.

Вложенные циклы – это циклы, организованные в теле другого цикла.

Вложенный цикл в тело, другого цикла, называется внутренним циклом. Цикл, в теле которого существует вложенный цикл, называется внешним.

Полное число исполнений внутреннего цикла, всегда равно произведению числа итераций внутреннего цикла на произведение чисел итераций всех внешних циклов.

Одна из проблем, связанных с вложенными циклами – это организация досрочного выхода из них. Решений у этой проблемы несколько, одна из них – использовать оператор безусловного перехода `goto`.

Циклы с несколькими охраняемыми ветвями. Цикл Дейкстры и «Паук».

Цикл Дейкстры состоит из одной или нескольких ветвей (охраняемых выражений),

каждая, ветвь представляет собой пару из охраняющего условия и охраняемой команды.

Цикл «Паук» - это модифицированный цикл Дейкстры с явными условиями выхода.

Бесконечным циклом называется цикл, написанный таким образом, что условие выхода из него никогда не выполняется.

Бесконечный цикл `while` выглядит так: `while (true) { тело цикла }`

Бесконечный цикл `do-while` выглядит так: `do { тело цикла } while (true)`

Бесконечный цикл `for` выглядит так: `for (; ;) { тело цикла }`

В написании программ, решающих реальные задачи, бесконечные циклы, как правило, используются очень редко и являются одним из источников неустойчивой работы программы. Например, бесконечные циклы есть смысл использовать в многопоточном программировании, в потоках контролирующей работу других потоков.