

GOTO — Lurkmore

«10 SIN
20 GOTO HELL »

— *Роботы о сабже.*

10

GOTO (*go you*, *бдлкдр. goto*) — слитное написание английского *go to*, что переводится как «*иди к...*» либо «*иди на...*», оператор безусловного перехода к указанной после него метке/строке программы. Причина лютых срачей между программистами. Споры на тему «*дозволено ли GOTO или нет*» (можно ли **ТАКОЕ** употреблять в программах) идут с самого начала существования [структурного программирования](#).

При ненадлежащем использовании в коде этот самый код становится [непонятен](#) даже создателю. При **годном** использовании может **СИЛЬНО** облегчить код и исключить использование более сложных конструкций. Собственно срач подразумевает [первый случай использования](#).



Сферичнее только 10 GOTO 10

20 GOTO 60

30 Срач

«Как пользоваться оператором GOTO? Подскажите, пожалуйста. А то друг подарил книгу — самоучитель C++, а страница с goto — вырвана.»

Вы удивитесь, но первый троллинг GOTO возник ещё в 1968 году, когда [Эдсгер Дейкстра](#) написал высер «Go To Statement Considered Harmful» («Оператор GOTO считается опасным», точнее «О вреде оператора GOTO», ещё точнее — «взгляд на GOTO, как на вредный оператор», а изначально — «A Case Against the Goto Statement», [заменённое](#) редактором журнала [Communications of the ACM](#) Никлаусом Виртом на название с популярным уже тогда «Considered Harmful». Оба названия можно перевести как «[Доводы против оператора GOTO](#)»; в них самих не ненависть или утверждение о всеобщей нелюбви к GOTO, а только указание на то, что в статье будет критика). Прочсть [его](#) крайне рекомендуется — там тонны лулзов (несмотря на то, что переводил явно надмозг). Чего только стоит первое же предложение: «За многие годы я утвердился во мнении о том, что квалификация программистов — функция, обратно зависящая от частоты появления операторов go to в их программах».

Еще более доставляет тот факт, что фраза «[Considered Harmful](#)» сама по себе стала мемом. Так, ответ на статью Дейкстры назывался «'GOTO Considered Harmful' Considered Harmful», ответ на неё назывался «'„GOTO Considered Harmful“ Considered Harmful' Considered Harmful?» и [так далее](#) (всего 78 (!) статей с данной фразой).

Вообще, в двух словах критика GOTO выражается очень просто: «Хуй поймешь, в каком порядке выполняется этот код». Отладка кода, щедро усыпанного GOTO, напоминает распутывание клубка ниток, причем состоящего из разных кусочков ниток, которые надо собрать в определенном порядке. Кстати, согласно этому оператору, данную статью следует читать начиная с первого раздела, а потом перейти к шестому, и если вы это читаете, не прочтя шестой, то вы ламер, лол.(пруф линк)(факты=}лулзы)goto...

Кстати, если хотите на своем опыте понять проблему GOTO, попробуйте добавить в эту статью новый раздел, [сохранив](#) корректность всех GOTO-ссылок.



[IRL](#)

Так вот, упомянутый во втором разделе язык Basic подвергся отдельной критике Дейкстры: «Студентов, ранее изучавших Бейсик, практически невозможно обучить хорошему программированию. Как потенциальные программисты они подверглись необратимой умственной деградации». А ещё он назвал копирования отечественными компостроителями архитектуры IBM «Величайшей диверсией Запада против СССР», и, ЧСХ, отчасти оказался прав!

Вернемся к GOTO. Любопытный факт: несмотря на массовую нелюбовь, он есть в большинстве современных языков программирования. И [выстрелить себе в ногу](#) с его помощью могут многие, под здоровый смех более просветленных [быдлокодеров](#). Но, что интересно, оператор GOTO имеется в исходниках весьма популярных программ, вроде ядра [линукса](#) или гугловского «[Андроида](#)», более того, сам [Линус одобряет](#) использование GOTO в языках программирования не таких дурацких, как Pascal. Почему? Потому что существуют редкие случаи, когда его использование несколько упрощает код. Другое дело, что использование его вне этих случаев немедленно приводит к полному пиздецу. Впрочем, случаи этого в последнее время не так уж часты — вероятно, по той причине, что многие ньюфаги, не знавшие раннего Бейсика, просто *не знают* о том, что такой оператор существует (для [PHP](#)-быдла он, как правило, и [не нужен](#), потому что там редко встретишь сколь-либо сложные алгоритмы).

Все это, однако, не мешает периодически выходить новым статьям, доказывающим или опровергающим то, о чем Дейкстра и его оппоненты написали полвека назад. Под любой такой статьей немедленно начинается спор между GOTO-фагами и GOTO-ненавистниками. Вторые обычно аргументируют «GOTO нельзя, потому что GOTO нельзя, и вообще можно ту же задачу решить без него», а первые осторожно замечают, что иногда это все-таки можно (если не прыгать вверх по программе), к тому же в некоторых языках не хватает адекватных заменителей, а на этих языках по-прежнему пишется код. Например, выход из многовложенных циклов на Си — можно извращаться с return-ами или делать флаги выхода, но, бля... А можно и не извращаться: по правилам структурного программирования break и вложенные циклы строго запрещены, потому что это сводит читаемость кода (и, как следствие, его работоспособность) к нулю. Про бесконечный цикл while(true) вообще молчим. [Правильный](#) вариант — использование функций, возвращающих false в случае, если нужно срочно выйти из цикла. Немного соображающий в кодинге тролль может вызвать немало ненависти, выступая за ту или другую сторону.

Но надо сказать, что современные языки с GOTO все-таки стараются ограничить его применение — так, обычно нельзя так просто прыгать между функциями. Так что GOTO [уже не тот](#), что был раньше. Но мы помним!

40 На самом деле

[На самом деле](#), GOTO — это *единственный* оператор, который используется для управления выполнением кода этими вашими центральными процессорами. Для всякого ЦП есть в сущности два способа управлять кодом: условный оператор (если да, то выполнить следующую команду, иначе выполнить последующую) и перейти на другой участок кода (GOTO). Всё. А что вы хотели, [машина Поста, десу!](#) И все эти ваши процедурное программирование, функциональное программирование, ООП, ААП, замыкания, циклы и прочая синтаксическая ересь — после компиляции превращается в сотни, тысячи, миллионы GOTO (а точнее, в JMP, JE, JLE, JGE, JNE, *etc.*), я гарантирую это! Несогласные goto 30

50 GOTO 80

60 Суть

Оператор GOTO — древнейший оператор. Используется он для безусловного перехода из одной точки кода в другую. До появления процедур и функций был по сути единственным способом запустить один и тот же код несколько раз. В [x86-ом ассемблере](#) этому оператору соответствует JMP.



Последствия использования

Люди, начинавшие программировать на раннем Basic'e, безусловно помнят, что каждая строчка на нем нумеровалась (10, 20 и т. д.). Соответственно, написав **GOTO 10**, можно было скануть в самое начало программы. Или ещё куда-нибудь. Многие ранние версии Бейсика позволяли вычисляемый GOTO (хоть от функции RND или INPUT), а иногда в таком лапидарном виде: 10 RUN. В более современных языках появились символные метки, так что стало возможным писать **goto nahuy**; или ещё что-то в этом роде.

70 GOTO 30

80 См. также

- [Рекурсия](#)
- [Пункт А](#)
- [Пункт Б](#)
- [Григорий Зельднер](#)

90 Ссылки

- [GoTo Жив!](#)

100 END ELSE GOTO 10



Языки программирования

++i + ++i 1C AJAX BrainFuck C Sharp C++ Dummy mode Erlang Forth FUBAR
God is real, unless explicitly declared as integer GOTO Haskell Ifconfig Java JavaScript LISP
My other car Oracle Pascal Perl PHP Prolog Pure C Python RegExp Reverse Engineering
Ruby SAP SICP Tcl TeX Xyzy Анти-паттерн Ассемблер Быдлокодер
Выстрелить себе в ногу Грязный хак Дискета ЕГГОГ Индусский код Инжалид дежице
Капча КОИ-8 Костыль Лог Метод научного тыка Очередь Помолясь Проблема 2000
Программист Процент эс Рекурсия Свистелки и перделки Спортивное программирование
СУБД Тестировщик Умение разбираться в чужом коде Фаза Луны Фортран Хакер
Языки программирования