

**nerfur**

карма  
**58,5**  
110 голосов

рейтинг  
**0,0**

[Профиль](#)[Публикации \(6\)](#)[Комментарии \(65\)](#)[Избранное \(14\)](#)

20 марта 2011 в 03:30

## Знакомимся с COBOL — ч.2 tutorial

Cobol\*

Ну что же, продолжим наше знакомство дальше, в первой статье мы узнали, из чего состоит COBOL программа, какие правила существуют по ее написанию и запустили нашу первую программу. Пора двигаться дальше, в этот раз мы начнем работать с переменными и совершать первые осмысленные действия с ними.

Итак, вернемся назад к нашей программе.

```
000100 IDENTIFICATION DIVISION.  
000200 PROGRAM-ID. HELLOWORLD.  
000300* --- Это пустая строка. ---  
000400 ENVIRONMENT DIVISION.  
000500 DATA DIVISION.  
000600 PROCEDURE DIVISION.  
000700 BEGIN.  
000800 DISPLAY "Hello World!".  
000900 STOP RUN.
```

Добавим в DATA DIVISION. новые строки, а именно секцию WORKING-STORAGE SECTION. и парочку переменных WS-A и WS-B.

Выглядеть это будет так:

```
000500 DATA DIVISION.  
000510 WORKING-STORAGE SECTION.  
000520 01 WS-A          PIC 999.  
000530 01 WS-B          PIC 9(3).  
000540 01 WS-RESULT     PIC9(6).
```

В PROCEDURE DIVISION. соответственно будут следующие строки:

```
000700 BEGIN.  
000800     DISPLAY "Hello I'm your new calculator!".  
000900     DISPLAY "Please Enter first number from 0 to 999".  
001000     ACCEPT WS-A.  
001100     DISPLAY "Please Enter second number from 0 to 999".  
001200     ACCEPT WS-B.  
001300     DISPLAY "-----".  
001400     DISPLAY " ".  
001500     DISPLAY "Your results are:".  
001600     ADD WS-A TO WS-B GIVING WS-RESULT.  
001700     DISPLAY "Summ is: ", WS-RESULT.  
001800     SUBTRACT WS-A FROM WS-B GIVING WS-RESULT.  
001900     DISPLAY "Subtract is: ", WS-RESULT.  
002000     MULTIPLY WS-A BY WS-B GIVING WS-RESULT.  
002100     DISPLAY "Multiplication is: ", WS-RESULT.  
002200     DIVIDE WS-A BY WS-B GIVING WS-RESULT.  
002300     DISPLAY "Divide is: ", WS-RESULT.  
002400     STOP RUN.
```

Уже традиционно сохраняем это в файл с расширением .cob и запускаем компилятор. (Мы ведь не забыли

добавить недостающие части программы правда? ;-) )

Запускаем удачно собранный исполняемый файл и видим, например:

```
Hello I'm your new calculator!
```

```
Please Enter first number from 0 to 999
```

```
300
```

```
Please Enter second number from 0 to 999
```

```
100
```

```
-----
```

```
Your results are:
```

```
Summ is: 000400
```

```
Subtract is: 000200
```

```
Multiplication is: 030000
```

```
Divide is: 000003
```

300 и 100 ввел я, остальное вывела программа. Результаты говорят сами за себя, а теперь рассмотрим новые строчки кода, которые мы ввели повнимательней.

Начнем с новинок в DATA DIVISION.

000510 WORKING-STORAGE SECTION. - сообщает компилятору о начале секции, описывающей обычные переменные

000520 01 WS-A PIC 999.

000530 01 WS-B PIC 9(3).

000540 01 WS-RESULT PIC 9(6).

С точки зрения смысловой нагрузки эти три строчки идентичны, они определяют три переменных с уровнем 01, следом идет имя переменной (WS-A, WS-B и WS-RESULT), далее идет оператор PIC (который также можно записывать в его более старой форме как PICTURE IS), который устанавливает формат и размер переменной. И вот тут начинается главный сюрприз для тех, кто знаком с другими языками. Размер и формат указывается не количеством выделяемых бит, а количеством максимальных символов.

В нашем случае мы описали две переменных, которые будут содержать 3-х ЗНАЧНОЕ число и одну переменную, которая будет содержать 6-ти ЗНАЧНОЕ число.

Поподробнее об описании формата. 9 – говорит компилятору, что переменная будет числовая, количество символов указывает на количество разрядов, естественно, что для больших чисел неудобно писать и считать что-нибудь а ля 9999999999999999 и была сделана форма записи 9(15), то есть в скобках указывается количество раз, которые повторяется символ перед скобками.

Перейдем к PROCEDURE DIVISION.

DISPLAY — как можно понять из программы и ее вывода занимается одной из самых полезных функций в мире. Выводит указанную переменную или текст на экран. Пользовательский текст должен быть заключен в двойные кавычки “”. Можно выводить несколько переменных или текстов, как видно в случае с

001700 DISPLAY “Summ is: ”, WS-RESULT.

главное не забыть запятую между ними.

Также в программе отметились 4 математических функции:

Сложение ADD WS-A TO WS-B GIVING WS-RESULT.

Вычитание SUBTRACT WS-A FROM WS-B GIVING WS-RESULT.

Умножение MULTIPLY WS-A BY WS-B GIVING WS-RESULT.

Деление DIVIDE WS-B BY WS-A GIVING WS-RESULT.

Построены они по одному принципу.

<оператор> значение1 TO/FROM/BY значение2 GIVING переменная.

В качестве значений могут выступать как переменные, так и числа напрямую, например,

ADD 10 TO 20 GIVING WS-RESULT.

Прибавят 10 к 20 и запишут результат (30) в WS-RESULT.

В случае, если GIVING и последующее имя переменной опускается, то результат запишется в переменную идущую ПОСЛЕ TO, FROM, BY – в нашем случае это будет WS-B (но WS-A в случае с DIVIDE).

Маленькое, но важное примечание, есть и немного другие формы записи, например:

ADD 1 2 3 TO WS-B – соответственно вначале сложит 1+2+3, а потом приплюсует это к значению WS-B и запишет результат в WS-B.

Можно опустить и TO – ADD 1 WS-A 2 WS-B сложит все эти значения и запишет результат в WS-B. Несколько переменных можно указать и после TO (к каждой из переменных после TO будут прибавлены значения до TO и результат записан в соответствующую переменную) и после GIVING, тогда в каждую из них будет записан результат. Аналогично можно оперировать и с SUBTRACT.

НО в случае с MULTIPLY и DIVIDE указание нескольких переменных возможно только после GIVING (и то это нужно уточнять для каждого компилятора отдельно).

Помимо этого у DIVIDE есть и вторая форма записи.

DIVIDE WS-A INTO WS-B GIVING WS-C со сменой BY на INTO меняется и порядок переменных, какую именно форму использовать остается выбор за пользователем. Укажу лишь, что:

DIVIDE WS-A INTO WS-B это  $WS-B / WS-A = WS-B$

DIVIDE WS-B BY WS-A GIVING WS-RESULT

это  $WS-B / WS-A = WS-RESULT$

ЭТО ВАЖНО! Вариант с BY НЕ БУДЕТ работать без GIVING. Программа просто не скомпилируется из-за ошибки.

Ну и напоследок маленький “подарок” — наверняка читатель уже задался вопросом “а как же присвоить значение переменной?”. С помощью MOVE

MOVE 10 TO WS-A

или MOVE WS-A TO WS-B

или даже MOVE 10 TO WS-A WS-B

ЭТО ВАЖНО! MOVE принимает несколько переменных только на “выходе”, т.е. после TO.

P.S. И наверняка вас удивляют цифры, которые мы получили на выводе и количество нулей в них? Это мы рассмотрим в следующей статье.

Воронцов “nerfur” Вячеслав. 2011.

**ВАЖНЫЙ UPDATE!** К сожалению забыл указать одно важное ключевое слово для DIVIDE.

DIVIDE WS-B BY WS-A GIVING WS-RESULT REMINDER <переменная>

Используя REMAINDER мы указываем переменную в которую запишется ОСТАТОК от деления.

cobol, программирование, олдскул



👁 2927 ★ 18





nerfur @nerfur

карма  
рейтинг  
58,5 0,0



Бесплатное облако для стартапов

ПОДАТЬ ЗАЯВКУ

Комментарии (51)



max7 20 марта 2011 в 06:58 #

+1

Я как-то наткнулся на короткую заметку [Управление сложностью или почему хорошо программировать — это сложно](#)

Цитирую:

Почему же тогда провалились идеи «менеджер-программист» (COBOL), «бухгалтер-программист» (1C), «аналитик программист» (SQL) и подобные? Ответ хорошо известен. Полезные программы сложны, а для разработки сложных программ требуется уметь прилагать дополнительные усилия, чтобы с этой сложностью бороться, как ещё иногда говорят, управлять сложностью.



Amper 20 марта 2011 в 07:48 # ↑

0

SQL и 1C провалились? а мужики-то и не знают...



max7 20 марта 2011 в 08:04 # ↑

0

Эти «мужики» определённо в школе не учились, и читать не умеют.



Amper 20 марта 2011 в 08:06 # ↑

0

Это вы о чем?



**max7** 20 марта 2011 в 09:28 # ↑**+7**

Все в рамках здравого смысла и до определённого предела, учитывая технические ограничения того времени.

По изначальной задумке отцов-основателей: менеджер использует COBOL для решения своих задач — отсюда такой синтаксис, аналитик использует SQL для получения необходимых ему данных — отсюда такой синтаксис, бухгалтер «автоматизирует» свою работу используя программы в 1С — без комментариев.

И тут как бы внезапно — не осилили! А по этому: COBOL-программист, SQL-программист, 1С-программист, и т.д. И нормальным программистам такое разделение и ограничение не нравится. Про COBOL вообще забыли, SQL ненавидят (особенно синтаксис), над 1С потешаются.

И все на самом деле очень просто: если система требует программирования, то им должны заниматься программисты, а не абы кто, и использовать для этого универсальные инструменты, а не очередной велосипед.

**Amper** 20 марта 2011 в 09:42 # ↑**0**

Не знаю насчет ненависти к SQL, но точно знаю — он не провалился... Он жил, жив и будет жить... Весь корпоративный мир никуда не денется от SQL. SQL востребован сейчас и будет востребован дальше. Тоже самое и с 1С. Да, варится в своей каше, да не популярен и IT-шников, но работает уже многие годы в подавляющем большинстве компаний нашей страны и будет работать дальше, фактически, у 1С нет конкуренции. Провалились идеи «бухгалтер/аналитик/менеджер-программист»? Возможно, но не совсем так... Все вылилось в «программист-бухгалтер/аналитик/менеджер». По сути, реальность такова, что сейчас программист, работающий, например в банке, просто обязан знать банковский/бухгалтерский учет, основы менеджмента, etc... вообще, программист должен знать все о предметной области фирмы, в которой работает. Именно поэтому программист в большинстве известных мне случаев получает больше, например, бухгалтера...

**max7** 20 марта 2011 в 10:29 # ↑**+3**

Я пишу о прошлом, вы категорично о будущем. Я пишу только о языке, о синтаксисе, вы, похоже, о технологии вообще.

«Весь корпоративный мир никуда не денется от SQL. SQL востребован сейчас и будет востребован дальше» — в том же стиле можно говорить и о наркотиках, наркоторговле.

SQL — один из самых ярких провалов в создании специализированных языков. В большинстве случаев запросы генерируются, и иметь их в виде строки которую мог бы написать человек, это то самое, что называют



костыль. Сейчас везде стараются миновать этот этап.

Есть очень много унаследованного и там будет и sql, и фортран, и всё, всё, всё.

1C: я не вижу ни одной серьёзной причины ограничивать используемые в нем языки одним велосипедом, раз уж ограничились виндой, раз уж прикрутили com, можно было использовать [active script](#) .

Здесь я с вами согласен. Сейчас «программист» на самом деле совмещает три «профессии»: «технолог», «собственно программист», и «аналитик». «Технолог» — приходится знать все особенности системы, «собственно программист» — собственно «делать» программу, «аналитик» — приходится понимать бухгалтеров, менеджеров и т.д., потому, что они не хотят понимать программистов. Поэтому минимум тройная зарплата ;-)



**vanxant** 20 марта 2011 в 14:31 # ↑

+3

Вы просто не в курсе, где-то что-то услышали и теперь пересказываете страшилки.

Ничего не мешает написать код на C# или там VB и прикрутить его к 1C. Пожалуйста, цепляйтесь, ядро это позволяет. Только зачем? 1C это очень мощная среда с аспект-ориентированным языком программирования для решения конкретно своей группы задач. С которой она справляется неидеально, но очень и очень хорошо. Если вы думаете, что какой-нибудь SAP R3 или MS Dynamics чем-то существенно лучше, знайте — вас подло обманули.

Да, когда-то очень давно 1C действительно продвигалась как программа, в которой грамотный бухгалтер сможет подкрутить пару коэффициентов, поставить десяток галочек и настроить учет под свою фирму и свою налоговую. В-общем, так и осталось, но с тех пор добавили еще один слой — конфигурации — в котором программисты и только программисты могут реализовать любой вариант учета, необходимый данной конкретной организации.

Некоторое непонимание между мейнстрим-программистами и 1C-никами связано с тем, что вторые решают конкретные ограниченные задачи и им не нужны новомодные абстракции высокого уровня, такие как IoC и прочий SOLID. Вместо мифической расширяемости в этой сфере нужно думать о многоуровневой отказоустойчивости, идиотоустойчивости и надежности. Именно поэтому намного более продвинутые Java и .Net, которые в теории создавались для решения тех же задач, что и 1C с Динамиксом, на практике используются для чего угодно, но не для создания конкурентов 1C.

С SQL же вообще смешно, никуда он не денется и никакие nosql-системы ему не угрожают. Да, конечно, там где нужно просто хранить пары ключ-значение, NoSQL оправдан, но допустим BerkleyDB используется для этих целей много десятков лет. Всплеск статей на данную тему в последнее время есть просто следствие работы пиарщиков команд, которые создали свой BDB, с ключами и значениями.

В задачах, в которых требуется реализация полноценных теоретико-множественных операций, никакой

posql не справится, и только идиоты будут пытаться реализовать ЭТО самостоятельно поверх posql-движка. В-общем, в программировании есть модные и не очень модные области, но ваши слова о провале это просто смех. Просто посмотрите на обороты оракла и 1С.



**max7** 20 марта 2011 в 15:04 # ↑

0

Я участвовал в создании конфигурации для 8.0. Так что нефиг гнать пургу.  
Где я говорю про «новомодные абстракции» (которым по факту уже пару десятков лет)?  
Где вы у меня увидели posql?  
Свои фантазии оставьте при себе.

«Слухи о вреде наркотиков сильно преувеличенны, просто посмотрите на обороты наркокартелей.»  
Обороты оракла и 1С такой же показатель как и обороты микрософта — нифига не значит. Есть очень много унаследованного, есть некомпетентность, есть решения в которых «программистов не спрашивали», часто хотят «здесь», «сейчас», «подешевле», «что у всех», «что налоговая посоветовала». Часто «программиста» рассматривают как придаток к системе, а не как участника процесса...  
Продолжать можно бесконечно.



**vanxant** 20 марта 2011 в 15:12 # ↑

-1

Нда, ну если вы сравниваете оракл с наркокартелями...



**Amper** 20 марта 2011 в 15:44 # ↑

-2

Простите, но вы несете откровенную пургу, никак не подкрепляя фактами и приводя какие-то странные ассоциации с наркокартелями...  
Вы почему-то называете Microsoft и Oracle (самые успешные IT-компании, бизнес которых завязан в том числе и с SQL) некомпетентными.  
Обоснуйте несостоятельность и провал SQL! Обоснуйте это для нас, которые с ним работают и вполне довольны, которым он кажется вполне логично построенной и удавшейся технологией.  
Вы ошибаетесь по поводу 1С. Для него можно писать на многих языках, пишете и подключайте DLL на C/C++, используйте COM, используйте .NET. Все это вполне возможно в 1С. Конечно для этого придется писать обертки на языке 1С. Но для этого есть куча вполне очевидных и логичных причин.



**max7** 20 марта 2011 в 16:30 # ↑

+2

Где я назвал Microsoft и Oracle некомпетентными? Давайте следовать контексту.  
SQL — это не технология, это язык. Где это он устоялся, если все разработчики баз постоянно его расширяют, часто гарантируя только sql-92? А иногда и того хуже?  
В конфигурации для 1С можно писать только на одном языке — встроенном. Внешние компоненты — на любом. «Внешние компоненты», модули, классы, и т.д. есть везде, это не показатель.

**Amper**

20 марта 2011 в 16:40 # ↑

**-2**

Где я назвал Microsoft и Oracle некомпетентными?

... Обороты оракла и 1С такой же показатель как и обороты микрософта — нифига не значит. Есть очень много унаследованного, есть некомпетентность...

Где это он устоялся, если все разработчики баз постоянно его расширяют, часто гарантируя только sql-92? А иногда и того хуже?

Ну расширяют, и? это показатель провала? Но тем не менее бизнес Oracle построен на SQL. Они ничего не понимают?

В конфигурации для 1С можно писать только на одном языке — встроенном. Внешние компоненты — на любом. «Внешние компоненты», модули, классы, и т.д. есть везде, это не показатель.

По вашему лучше зоопарк языков программирования в одной платформе?

Я так и не услышал от вас доводов в сторону провала ни SQL ни 1С.

**max7**

20 марта 2011 в 17:39 # ↑

**+1**

Там энтер потерялся, это разные абзацы.

В технологическом плане Oracle не авторитет. В плане продавливания своих продуктов через «администрации» — «впереди планеты всей». Вы сказали, что язык устоялся, я сказал, что это не так. Каким боком здесь оракл, который использует модификацию языка sql как интерфейс к некоторым из своих продуктов?

Выбор всегда лучше, вас выбирать никто не заставляет, пишите на «встроенном», а Я хочу на «своём». Как программист Я к 1С никогда и близко не подойду, даже если буду с голоду умирать. И даже не по тому, что платформа плоха, как платформа 1С более-менее ничего. Но сложившиеся «стандарты» и представления... Я не Дон Кихот, чтоб с мельницами бороться.

Доводов вполне достаточно. Он не соответствует решаемым задачам. При декомпозиции задачи до уровня sql, часто получается слишком трудный для реализации и поддержки монстр. Не сложный, а именно трудный.



**ononokomati** 20 марта 2011 в 16:30 # ↑

+3

Все эти «можно прикрутить» тут не к месту.

Речь про явно избыточный якобы «человекопонятный» синтаксис COBOL и, например, ABAP. Отчего он именно такой. Предполагалось, что любой бухгалтер или экономист сможет открыть исходник программы на этих языках, прочитать все эти «DIVIDE WS-B BY WS-A GIVING WS-RESULT» как будто по-английски и модифицировать программу под свои нужды.

На деле же программы — программистам, пользу — пользователям. И куча накладных расходов на ужасный синтаксис.

«Провал» тут относится именно к этой затее.



**nerfur** 20 марта 2011 в 10:32 # ↑

+4

А может быть это потому, что у нас пользователи, те же бухгалтера, аналитики и менеджеры зачастую не то, что «программировать», а кнопку Пуск самостоятельно найти не могут благодаря изощренной форме образования и влиянию «корпораций зла», играющих на человеческих слабостях и лени? (да да, я верю в некоторые теории заговора! ;-)

И я сильно сомневаюсь, что SQL или 1С создавался для того, чтобы сами бухгалтера на них писали, COBOL же так вообще создавался как легко читаемый для «управленца» язык, чтобы программист (а никто в 1959 году и не мог помышлять о том, чтобы финансист сам что-то там «кодил» на мэйнфрейме!) понимали друг друга легко и помоему со своей целью язык справлялся и во-многом справляется.

А по поводу универсальных инструментов... спорно, очень спорно) Почему же до сих пор не создали суперуниверсальный? )

**shogunkub**

20 марта 2011 в 10:44 #

**+1**

Насчёт 1С — в 6.0 именно так и было, бухгалтеры писали сами. Немногие конечно, но было такое. Вот только возможностей с гулькин нос, и язык кривой до ужаса. А увеличение возможностей языка бухгалтеры осилить не смогли, и появились те самые «программисты-бухгалтеры».

**nerfur**

20 марта 2011 в 10:49 #

**0**

с историей 1С и SQL к сожалению слабо знаком, поэтому и выразил сомнения)

404 **amarao** 20 марта 2011 в 08:06 #

**0**

В своё время я помню в гвайде по юниксу (ред бук) Неймет сильно высмеивал программистов на коболе. Я думал, для меня это будет абстрактной конструкцией, с которой я никогда не столкнусь.

А потом я столкнулся. Ближайшим аналогом «программиста на коболе» является «программист на 1С».

И отношение к коболу ровно такое же. Это штука для кодирования бизнес-логики, от которой сводит зубы у любого уважающего себя программиста.

**nerfur**

20 марта 2011 в 10:23 #

**+2**

Но ведь никто и не предлагает кодировать на ней нечто отличающееся от бизнес-логики, да он для этого и не предназначен в принципе, нет, конечно, на нем можно и текстовый редактор, а может и плеер написать, а может и что-то заслуживающее восторженного «аххх!»... но зачем же? Язык доказал свою пригодность, мало того, множество программистов его искренне любят и продолжают работать на нем, видимо они себя не уважают, но это их выбор и выбор тех, кто когда то его у себя применил... и «оно» продолжает работать и не кашлять...

**karter**

20 марта 2011 в 10:54 #

**+1**

«P.S. И наверняка вас удивляют цифры, которые мы получили на выводе и количество нулей в них? Это мы рассмотрим в следующей статье.» — Интрига на протяжении всего цикла статей?) Отличный способ заставить с нетерпением ждать следующего выпуска :) Еще раз спасибо.

**nerfur** 20 марта 2011 в 10:55 #

↑ Ну куда ж без интриги)

+1

**Gorthauer87** 20 марта 2011 в 11:21 #

+3

Мдя... всегда любил лаконичный С подобный синтаксис, здесь же просто мясо какое-то. В каком-то плане даже 1С и тот покрасивше выглядит.

Одного не понимаю, язык бизнес-ориентированный, то есть ориентирован не на программистов-математиков, а на менеджеров, которые чуть-чуть смыслят в коде, но какого черта он такой непонятный? Смотрю на код как идиот! Количество избыточной информации на строчку кода просто зашкаливает.

**bitterman** 20 марта 2011 в 14:15 # ↑

0

не знаю, кому как, но мне после текста этой программы-калькулятора непонятным осталось только объявление переменных. Кобола, естественно, не знаю.

**nerfur** 20 марта 2011 в 14:22 # ↑

0

Проблему с объявлением переменных мы уже решаем) Ждите сегодня)

**Gorthauer87** 20 марта 2011 в 15:04 # ↑

0

А после бейсика меня воротит от подобной нумерации строк. Понимаю еще оно было оправдано для перфокарт, но для человека это не читабельно. Не удивлюсь, если там и GOTO такой же, как в бейсике.

Дейкстра все такие ЯП не зря ругал.

Хотя COBOL конечно важную роль сыграл, он показал, каким НЕ ДОЛЖЕН БЫТЬ язык программирования, хорошо, что авторы насобирали на нем кучу граблей и развитие двинулось дальше.

**bitterman** 20 марта 2011 в 15:07 # ↑

+1

ну, COBOL, как собравший больше всех реально написанного (а не общедоступного) кода, показал, каким не должен быть ЯП.

Windows — какой не должна быть ОС.

какие ещё примеры распространённого того, чего не должно было быть? PHP вон тоже не должен быть :-)

**Gorthauer87** 20 марта 2011 в 15:16 # ↑

0

---

Жигули

**nerfur**

20 марта 2011 в 15:27 # ↑

0

Не нужны не Жигули, а те люди, которые поддерживают условия при которых новые Жигули разваливаются раньше советских... и собственно не развиваются толком (утрировано конечно и пафосно, но мне кажется это более точно выражает)

**Gorthauer87**

20 марта 2011 в 15:46 # ↑

0

Вот тоже можно и про Кобол сказать, он вообще не развивается, можно сказать он уже мумифицировался. Это всё-таки не лисп, который до сих пор жив.

**nerfur**

20 марта 2011 в 15:56 # ↑

0

последний стандарт COBOL 2002 года) В разработке следующий, компиляторы (тем более «ентерпрайз») развиваются и пишутся, код вертится и жужжит, и где здесь мумификация? Да, я согласен, к сожалению COBOL-разработчики не кричат на каждом углу, хотя стоило бы возможно, но это не признак мифической смерти языка, который обрабатывает миллиарды транзакций в день )

**Gorthauer87**

20 марта 2011 в 16:11 # ↑

+1

О да... это же как раз они решали проблему 2K.

**nerfur**

20 марта 2011 в 16:52 # ↑

0

да, именно, и почему то когда все вспоминают про нее, никто не помнит что когда эти программы писались экономили на всем, включая первые две цифры года, а теперь умножьте 2 на несколько млн или млрд записей... были ли эти гигабайты носителей тогда? Так что проблема 2000 это проблема не языка и программистов, а больше поздно зачесавшихся руководителей, которые неожиданно вспомнили (да и то сами ли)

**nerfur**

20 марта 2011 в 15:25 # ↑

+1

Это тот самый «классический», как я писал в первой части, вид кода, современный кобол и к отступам не так требователен и большие буквы ему не обязательно нужны, и нумерация, да даже некоторые разделы можно выкинуть,

НО... лично я считаю, что даже для хобби лучше научиться «как надо», чтобы понимать что написано в старом, чтобы «прочувствовать» всю «глубину» и, наконец, мне это просто нравится, в какой то мере это заставляет концентрироваться на коде и приучает к аккуратности.

**bitterman**

20 марта 2011 в 18:53 # ↑

0

именно. как и с goto — лучше понять пользу, чем пытаться себя от вреда обезопасить :-)

**zanudische**

20 марта 2011 в 11:22 #

+3

Этот ваш кобол омерзительно многословен. До рвоты. Желаю его авторам (если еще живы) кодить на Malbolge вручную.

**bitterman**

20 марта 2011 в 14:16 # ↑

0

прикрутить сниппеты и нормально будет, только успевай tab нажимать )

**Gorthauer87**

20 марта 2011 в 15:05 # ↑

0

А к мозгу тоже сниппеты прикрутишь, когда это потом читать и модифицировать понадобится?  
Подобные ЯП называют ещё иногда write-only

**bitterman**

20 марта 2011 в 15:09 # ↑

0

Многословность в любом случае можно так или иначе решить редактором. Когда в бизнес-логике на С стоит не тот символ не в том месте — мозг alarm не включает. Можно и поподробнее, лишь бы оно на самом деле делало то, что написано.

**nerfur**

20 марта 2011 в 15:29 # ↑

0

Я смотрю на многих таких писал? ) А простите, что-нибудь еще про Дейкстру кроме той цитаты знаем? )

**Gorthauer87**

20 марта 2011 в 15:45 # ↑

0

Во write only можно любой ЯП превратить, но есть такие, где по другому нельзя, например, brainfuck. Cobol же, судя по примерам, очень быстро превращается в плохочитаемый студень. Современным ЯП в этом плане до него далеко. Можно сказать, что наряду с бейсиком и отчасти Фортраном, Кобол был одними из наиболее больших грабель в



развитии программирования.

И вообще

«Проекты, предлагающие программирование на естественном языке, гибельны по своей сути.»(с) Дейкстра

Про алгоритм его имени я в курсе ;)

**nerfur**

20 марта 2011 в 15:52 # ↑

0

Лично для меня плохочитаемый студень это C++ и Java, которые я в прямом смысле плохо переношу даже просто глядя на них (к С как ни странно отношусь намного лучше).

Кобол, бейсик и фортран — одни из наибольших грабель? Оставлю без прямых комментариев... В конце концов мое мнение, что язык нужно судить не по мифической и субъективной красоте слога, а по тому, что он в этот мир принес, в качестве пользы... Теоретические же рассуждения о красоте и читаемости... в конце концов... что вы вообще делаете в таком случае в этом топике? пришли потроллить? )

А уж сравнивать с эзотерикой, это совсем мне кажется кощунство.

**Gorthauer87**

20 марта 2011 в 16:07 # ↑

-1

>Кобол, бейсик и фортран — одни из наибольших грабель?

Вот именно, что грабель.

>В конце концов мое мнение, что язык нужно судить не по мифической и субъективной красоте слога, а по тому, что он в этот мир принес, в качестве пользы...

Это уже история. Да и в конце концов то, что они выявили собой огромные проблемы тоже польза и огромная. Но тем не менее, эти языки были ну уж очень далеки от совершенства.

**shogunkub**

20 марта 2011 в 16:22 # ↑

+2

Да вы просто слишком много требований выдвигаете к языку образца 59-го года, в конце концов С почти на 15 лет моложе, было бы удивительно, если бы он был хуже Кобола :)





**Klaus** 20 марта 2011 в 11:34 #

0

А раз блг называется «Старое железо» будут ли примеры работы с коболом на старом железе?



**nerfur** 20 марта 2011 в 11:42 # ↑

0

а что именно вы ждете? Историю из жизни? Или некий конкретный код?



**Klaus** 20 марта 2011 в 13:21 # ↑

0

да что угодно. Я просто очень люблю ретрокомпьютинг и было бы интересно всякое



**Amper** 20 марта 2011 в 12:54 #

0

Кобол обычно критикуется за многословность и громоздкость, поскольку одной из целей создателей языка было максимально приблизить конструкции к английскому языку. **(До сих пор Кобол считается языком программирования, на котором было написано больше всего строк кода).**

ого, не знал...



**nerfur** 20 марта 2011 в 13:10 # ↑

+1

мало того, до сих пор ооочень большая часть всей «денежной» информации в мире проходит через ПО на Коболе, оценивают от 60 и выше %.



**bitterman** 20 марта 2011 в 14:31 #

+1

Хм, а ведь OpenCOBOL транслирует в С. Значит, на ARM'е заработает. Хы ;-) )




**nerfur** 20 марта 2011 в 14:40 # ↑

0


Насколько я помню есть и сборка под ARM, во всяком случае, что-то такое я слышал и регулярно в IRC пролетают слова об идеях/реализациях на различных мини-«киосках». Не углублялся в эту тему правда.

Только зарегистрированные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

## Что обсуждают

Сейчас	Вчера	Неделя	Месяц	
				
Как «Змейка» может познакомить с ООП: сложная концепция простыми словами				9
Adobe выпустила экстренное обновление для Flash Player				8
Представляем 3CX Phone System v14 SP1 с множеством новых функций				15
Composer — the right way				135
Скандал вокруг Volkswagen и роль программистов в нём				21

## Самое читаемое

Сейчас	Вчера	Неделя	Месяц	
				
Скандал вокруг Volkswagen и роль программистов в нём				4k
Памятка евангелиста PostgreSQL: критикуем MySQL грамотно				3k
HAProxy исполнилось 1.6				4k

Удалённое исполнение системных команд по запросу через сокеты на Python 3 или как я сайты скачивал

7k

Frida-node или немножко странного кода

2k

## Лучшее на Geektimes

- 1 Газ из дров: замена газа в котле + электричество.  $1\text{кг}=2.2\text{м}^3$  газа или  $1\text{кг}=1\text{кВт}\cdot\text{час}$  при получении электроэнергии (27)
- 2 Измерение электроэнергии. Часть 1. Измерение напряжения и тока (2)
- 3 USPS прекращает отслеживание регистрируемых международных почтовых отправлений (13)
- 4 Lenovo, Microsoft, HP, Dell и Intel запустили первую в своей истории совместную масштабную рекламу (22)
- 5 Базе данных по фильмам IMDb исполнилось 25 лет (1)

[Все публикации](#)[Популярные хабы](#)[Компании](#)

## Лучшее на Мегамозге

- 1 Переход из разработчиков в руководители: советы и книги по теме
- 2 Оператор Tele2 влез в миллиардные долги, чтобы запуститься в Москве (5)
- 3 Как действительно создавать софт, который действительно работает? (1)

**4** [Стивен Балмер купил 4% Twitter \(1\)](#)

**5** [Бесплатное продвижение. Основные методы и их работоспособность](#)

[Все публикации](#)

[Популярные хабы](#)

[Стартапы](#)

## Вакансии на «Моём круге»

### **PHP разработчик**

Полный рабочий день

Программист Python и PyQt

Полный рабочий день

Android-developer (разработка Android-приложений, связанных с ГИС-системами)

Новосибирск • Полный рабочий день

### **PHP-программист**

Неполный рабочий день

Frontend Developer

Москва • Полный рабочий день

Android Senior Developer

Москва • Полный рабочий день

iOS Senior Developer

Москва • Полный рабочий день

Senior iOS Engineer

Сан-Франциско и район залива • Полный рабочий день

разместить вакансию  
Frontend Engineer

[все вакансии](#)

Сан-Франциско и район залива • Полный рабочий день

Frontend-разработчик meteor.js/react.js/node.js

Одесса • Полный рабочий день

## Заказы на «Фрилансим»

IOS разработчик с опытом создания инструментов обработки видео

18.10.2015 • 0 откликов

Разработать дизайн шаблона сайта

18.10.2015 • 11 откликов

Натяжка psd на wordpress

18.10.2015 • 3 отклика

IntelliJ IDEA + Spring Framework + Gradle

18.10.2015 • 0 откликов

Сделать прозрачный фон на фотографиях

18.10.2015 • 4 отклика

Настроить VPS под работу с Wordpress Multisite

17.10.2015 • 3 отклика

Бот для Telegram

17.10.2015 • 6 откликов

Приложение для такси ios/android

18.10.2015 • 6 откликов

все заказы

Создать промежуточный сервер для обработки структурированных данных

17.10.2015 • 3 отклика

Настроить Windows Server 2012 R2 IIS

17.10.2015 • 3 отклика