

**nerfur**

карма  
**58,5**  
110 голосов

рейтинг  
**0,0**

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

20 марта 2011 в 22:38

## Знакомимся с COBOL – ч.3 tutorial

Cobol\*

К сожалению забыл указать в предыдущей статье несколько важных ключевых слов:

для DIVIDE.

DIVIDE WS-B BY WS-A GIVING WS-RESULT REMAINDER <переменная>.

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

Для любых математических операций:

Также, добавляя ROUNDED после операции мы достигаем того, что результат будет округляться, а не просто “отбрасываться” лишний хвост.

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

А теперь продолжим познавать новое. Исполняя обещание развеять удивление о странном выводе значений переменных мы разберем в этот раз PICTURE IS (он же PIC) и форматы переменных.

И снова мы возвращаемся в DATA DIVISION.

Рассмотрим еще разок строчку с описанием переменной:

01 – уровень переменной;

WS-B – имя переменной;

PIC – ключевое слово, указывающее, что далее идет описание формата переменной, старой формой является PICTURE IS.

9(3) – тип символов в переменной “9” и их количество “3”.

### **УРОВЕНЬ ПЕРЕМЕННОЙ.**

Пользователю для обычных переменных доступны уровни с 01 по 49 включительно и 77. Уровень переменной описывает ее иерархию в группе переменных, чем выше номер, тем ниже переменная в “лестнице”. Например 01 CUSTOMER-RECORD.

05 CUSTOMER-NAME PIC X(10).

05 CUSTOMER-ADDRESS PIC X(30).

Уровень 77 объявляет переменную, как ЭЛЕМЕНТАРНУЮ, то есть у нее не может быть под-переменных.

Уровень 88 объявляет переменную, как conditional, причем в достаточно изощренной и богатой функционально форме, о них подробнее в следующий раз.

Есть и некоторые другие уровни, которые введены либо производителями компиляторов, либо сейчас нам не нужны.

### **ИМЯ ПЕРЕМЕННОЙ.**

Имя переменной может состоять из цифр, букв и “-”, но не должно с “-” начинаться.

Имя переменной должно содержать не больше 30 символов.

### **ОПИСАНИЕ ФОРМАТА ПЕРЕМЕННОЙ.**

Рассмотрим какие описательные символы существуют для переменных в COBOL.

Они делятся на две группы: “обычные” и “редактирующие”.

В обычные входят

9 – одна цифра;

V – знак, отделяющий целую часть от десятичной (может применяться только ОДИН раз в каждой переменной).

S – знак “±”, должен быть самым первым символом в формате переменной (может применяться только ОДИН раз

в каждой переменной).

X – один буквенно-цифровой знак, т.е. цифра, буква, пробел, спец. символы.

A – один буквенный знак или пробел.

Редактирующие символы – благодаря им описываются переменные, которые не будут участвовать в расчетах, но позволяют тонко настроить вид, в котором переменная будет выведена на экран/печать:

Z – заменяет 9'ки с той разницей, что все начальные нули убираются.

\$ — “валютный знак”, указанный для компилятора, по умолчанию, в основном знак доллара, используется ТОЛЬКО как ПЕРВЫЙ символ в описании. Может использоваться только один раз. Его значение устанавливается в ENVIRONMENT DIVISION. с помощью

\* — аналог Z, но начальные нули заменяются на \*.

— (минус) – может указываться как первым, так и самым последним символом, но только один раз. Если число отрицательное, то будет выведен минус, иначе пустое место.

+ (плюс) – аналог “-” (минуса), НО в случае с положительным числом будет выведен “+”, а в случае с отрицательным выводится “-”.

. (точка) – знак “десятичной точки”. На ее месте будет выведена именно “точка”.

, (запятая) – знак “запятой”. Может быть несколько в переменной.

/ (слэш) – знак “слэш”. Может быть несколько. Выводит “слэш”, популярен в датах.

0 (ноль) — “ноль”. Просто на этом месте будет выведен 0.

B (от blank, пустой) — “пробел”.

Но главное, как вы можете понять все эти символы комбинируются и позволяют получать очень гибкие формы и виды. Правило здесь фактически только одно – числовые переменные не должны превышать 18 символов, буквенно-цифровые – 160 символов. Но тут опять же многое зависит от компилятора и его параметров.

ВКУСНЫЕ ДОПОЛНЕНИЯ или ЕСТЬ ЖИЗНЬ ПОСЛЕ PIC'a!

Помимо этого, при описании переменной после ее формата могут быть добавлены дополнительные “опции”:

BLANK WHEN ZERO – если переменная равно 0, то вместо нее будет выведено пустое место, а не нули.  
VALUE “значение” — сразу присваивает переменной определенное значение.

Существуют и другие, но упоминать о них сейчас было бы бесполезно.

Итак, теперь давайте просто пощупаем все их в разных видах.

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. VARIABLES-TEST.
000300 AUTHOR. ME.
000400 ENVIRONMENT DIVISION.
000500 DATA DIVISION.
000600 WORKING-STORAGE SECTION.
000700 01 HUMAN-CONTAINER.
000800         05 HUMAN-NAME          PIC A(30).
000900         05 HUMAN-ADDRESS       PIC X(160).
001000 77 SQUARE-METERS            PIC 9(18).
001100 77 SQUARE-PRICE               PIC 9(15)V9(2).
001150 77 FLAT-PRICE                 PIC 9(15)V9(2).
001200 77 TAX-PERCENT                 PIC 9(2).
001300 77 TAX-SUMM                   PIC 9(15)V9(2).
001350 77 NEGATIVE-VALUE             PIC S9(10)V9(6).
001400* --- 0000MG! ---
001500 77 SQUARE-METERS-OUT         PIC Z(17)9.
001600 77 SQUARE-PRICE-OUT         PIC Z(14)9.Z(2).
```

```
001650 77 FLAT-PRICE-OUT-1          PIC Z(3),Z(3),Z(3),Z(3).9(2).
001660 77 FLAT-PRICE-OUT-2          PIC Z(3)BZ(3)BZ(3)BZ(3).9(2).
001700 77 TAX-PERCENT-OUT           PIC Z(2).
001800 77 TAX-SUMM-OUT              PIC Z(14)9.9(2).
001900 77 NEGATIVE-VALUE-OUT-1     PIC +Z(10).9(6).
002000 77 NEGATIVE-VALUE-OUT-2     PIC -Z(10).*(6).
002100* -----
002200 PROCEDURE DIVISION.
002300 BEGIN.
002400 DISPLAY "Please enter Name: ".
002500 ACCEPT HUMAN-NAME.
002550 DISPLAY "Please enter Address: ".
002600 ACCEPT HUMAN-ADDRESS.
002700 DISPLAY "Please enter square meters of flat: ".
002800 ACCEPT SQUARE-METERS.
002900 DISPLAY "Please enter square meter's price: ".
003000 ACCEPT SQUARE-PRICE.
003100 DISPLAY "Please enter percent of tax: ".
003200 ACCEPT TAX-PERCENT.
003300 DISPLAY "Enter any really big NEGATIVE value: ".
003400 ACCEPT NEGATIVE-VALUE.
003500 DISPLAY "-----".
003600 DISPLAY " ".
003700 DISPLAY HUMAN-NAME.
003800 DISPLAY HUMAN-ADDRESS.
003900 DISPLAY "SQUARE-METERS: ", SQUARE-METERS.
004000 MOVE SQUARE-METERS TO SQUARE-METERS-OUT.
004100 DISPLAY "SQUARE-METERS-OUT: ", SQUARE-METERS-OUT.
004200 DISPLAY "SQUARE-PRICE: ", SQUARE-PRICE.
004300 MOVE SQUARE-PRICE TO SQUARE-PRICE-OUT.
004400 DISPLAY "SQUARE-PRICE-OUT: ", SQUARE-PRICE-OUT.
004500 MULTIPLY SQUARE-METERS BY SQUARE-PRICE GIVING FLAT-PRICE.
```

```
004600  DISPLAY "FLAT-PRICE: ", FLAT-PRICE.  
004700  MOVE FLAT-PRICE TO FLAT-PRICE-OUT-1 FLAT-PRICE-OUT-2.  
004800  DISPLAY "FLAT-PRICE-OUT-1: ", FLAT-PRICE-OUT-1.  
004850  DISPLAY "FLAT-PRICE-OUT-2: ", FLAT-PRICE-OUT-2.  
004900  DISPLAY "TAX-PERCENT: ", TAX-PERCENT.  
005000  MOVE TAX-PERCENT TO TAX-PERCENT-OUT.  
005100  DISPLAY "TAX-PERCENT-OUT: ", TAX-PERCENT-OUT.  
005200  DISPLAY "TAX-SUMM: ", TAX-SUMM.  
005300  MOVE TAX-SUMM TO TAX-SUMM-OUT.  
005400  DISPLAY "TAX-SUMM-OUT: ", TAX-SUMM-OUT.  
005500  DISPLAY "NEGATIVE-VALUE: ", NEGATIVE-VALUE.  
005600  MOVE NEGATIVE-VALUE TO NEGATIVE-VALUE-OUT-1 NEGATIVE-VALUE-OUT-2.  
005700  DISPLAY "NEGATIVE-VALUE-OUT-1: ", NEGATIVE-VALUE-OUT-1.  
005800  DISPLAY "NEGATIVE-VALUE-OUT-2: ", NEGATIVE-VALUE-OUT-2.  
005900  STOP RUN.
```

И что же мы получаем в итоге.

Please enter Name:

Slava

Please enter Address:

Moscow 15

Please enter square meters of flat:

44

Please enter square meter's price:

1234123

Please enter percent of tax:

20

Enter any really big NEGATIVE value:

-123123213213

-----

```
Slava
Moscow 15
SQUARE-METERS: 000000000000000044
SQUARE-METERS-OUT:                44
SQUARE-PRICE: 000000001234123.00
SQUARE-PRICE-OUT:                1234123.00
FLAT-PRICE: 000000054301412.00
FLAT-PRICE-OUT-1:                54,301,412.00
FLAT-PRICE-OUT-2:                54 301 412.00
TAX-PERCENT: 20
TAX-PERCENT-OUT: 20
TAX-SUMM: 0000000000000000.00
TAX-SUMM-OUT:                    0.00
NEGATIVE-VALUE: -3123213213.000000
NEGATIVE-VALUE-OUT-1: -3123213213.000000
NEGATIVE-VALUE-OUT-2: -3123213213.000000
```

Как видите вывод переменных “форматированных” и сырых отличается и настраивается.

Естественно это не просто ваше право, а обязанность поиграться с переменными, посмотреть как они будут себя вести.


А между прочим, если вы внимательно смотрели вывод своей программы, то могли заметить очень неприятную “мелочь” — значения, которые не влезли в переменную, если у вас этого не произошло, то попробуйте поэкспериментировать.


До следующей Встречи! Впереди будет еще много интересного)

Воронцов “nerfur” Вячеслав. 2011.

cobol, программирование, олдскул

 +4 

 2474  10



**nerfur** @nerfurкарма  
58,5рейтинг  
0,0

## Комментарии (6)

**Gorthauer87** 22 марта 2011 в 00:14 #

0

Черт и жили же люди как-то до изобретения отступов в коде!!! Полотна кода просто мозг поедают, вообще близко не похоже на традиционные ЯП, видимо тут и склад ума нужен не программисткий с современной точки зрения, чтобы на этом что-то писать и понимать.

ЗЫ

Технарь, который не слишком то горит желанием разбираться в бизнес-процессах, экономике и прочих финансах.

**nerfur** 22 марта 2011 в 08:45 # ↑

+1

не туда ответил) хотя может и опера глючит...

**jtootf** 22 марта 2011 в 20:28 # ↑

0

Разукрашивание кода для облегчения восприятия всегда можно переложить на IDE.

А с непривычки оно и в CL одни скобки, и J обфусцирован, и в Haskell зигохистоморфные препроморфизмы повсюду.

**nerfur** 22 марта 2011 в 08:15 #

+2

Вот тут поспорю, как раз кто кто, а КОБОЛ был МАКСИМАЛЬНО требователен к отступам, вспомните «зоны» из первой статьи, по идее тут все очень и очень красиво, НО тег «коде» в редакторе хабра съедает лишние пробелы, а современные компиляторы уже



намного менее требовательны) Кстати, любой совет в решении этой проблемы будет ОЧЕНЬ кстати)

ЗЫ приношу извинения за текст и особенно название тега, пишу с мобильного)  
ЗЫЫ Кстати, хабру очень не хватает мобильной версии... или я просто отстал от жизни и в этом?

 **rasa** 17 августа 2013 в 14:37 #

0

Почему не завернете код в соответствующие теги? Все переносы строк скушались.

 **nerfur** 17 августа 2013 в 17:59 # ↑


+1

спасибо, что сообщили, он и был завернут, в code, но видимо обработку поменяли, поменял на pre

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

---

## Что обсуждают

Сейчас	Вчера	Неделя	Месяц	
				
Детективная история — Что общего между auto.ru и brazzers?				50
Сетевые технологии в высшем образовании. Печальный опыт поиска молодых специалистов				65
Продуктивность по почте: Как создавался Evernote для Outlook				2
Что там у ReactOS?				40
NikariCP — самый быстрый пул соединений на java				49

## Самое читаемое

Сейчас	Вчера	Неделя	Месяц	
Детективная история — Что общего между auto.ru и brazzers?				41k
Сетевые технологии в высшем образовании. Печальный опыт поиска молодых специалистов				3k
Русские хакеры взломали Dow Jones и завладели инсайдерской информацией				9k
Балансировка 70 тысяч запросов в секунду на HighLoad++				693
Что там у ReactOS?				22k

## Лучшее на Geektimes

- 1 Ракетные «сосиски» и «сардельки» или проклятие неуниверсальности (17)
- 2 Вероятность в квантовой механике. Откуда она взялась и как это упростить для понимания (6)
- 3 Российские светодиодные лампы (19)
- 4 История радиуса детектируемости цивилизаций (10)
- 5 Мастера DIY-жанра: рельсотрон — собственными руками (26)

Все публикации    Популярные хабы    Компании

## Лучшее на Мегамозге

- 1 [Как мы участвовали в крупнейшей выставке всего за 36 500 руб \(2\)](#)
- 2 [Некоторые интересные факты и инсайты с конференции «Электронные деньги и платежи»](#)
- 3 [Анализ проектов ФРИИ \(7\)](#)
- 4 [Пять уровней пирамиды эффективности \(2\)](#)
- 5 [Изменить одно слово в СТА-кнопке и увеличить конверсию: миф или реальность? \(4\)](#)

**Все публикации**

**Популярные хабы**

**Стартапы**

## Вакансии на «Моём круге»

Системный Инженер (Dev Ops) / Разработчик

Москва • Полный рабочий день

Руководитель команды по автоматизации тестирования

Москва • Полный рабочий день

Разработчик мобильных приложений (Android)

Москва • Полный рабочий день

Product Marketing Manager

Казань • Полный рабочий день

Разработчик JavaScript (Node.js)

г. Пермь • Полный рабочий день

[все вакансии](#)

Senior Ruby on Rails разработчик

Полный рабочий день

.NET Разработчик

Москва • Полный рабочий день

**Senior iOS Engineer**

Сан-Франциско и район залива • Полный рабочий день

Senior Android Engineer

Сан-Франциско и район залива • Полный рабочий день

Frontend Engineer

Сан-Франциско и район залива • Полный рабочий день

## Заказы на «Фрилансим»

Разработка электроники для квест комнаты

20.10.2015 • 0 откликов

Разработка модуля MODX Revolution для работы с API Яндекс.Вебмастер

20.10.2015 • 0 откликов

Доработка текущего веб-проекта и добавление нового функционала

18.10.2015 • 18 откликов

Лэндинг+Директ, тематика "Аквапанели кнауф"

19.10.2015 • 1 отклик

---

Разместить заказ

все заказы

19.10.2015 • 0 откликов

Сверстать html-страницу из PSD макета

19.10.2015 • 8 откликов

**Написать статью про Windows 10 IoT Enterprise**

19.10.2015 • 0 откликов

Сверстать небольшой интернет-магазин на Opencart

19.10.2015 • 5 откликов

Перевод с русского на английский 500 символов общая тематика

19.10.2015 • 1 отклик

Postgresql master-slave

19.10.2015 • 2 отклика