

Введение в язык программирования



Паскаль (Pascal) — один из наиболее известных языков программирования, используется для обучения программированию, является основой для ряда других языков.



История

- Язык **Паскаль** был разработан в 1968 - 1969 г. **Никлаусом Виртом**, швейцарским учёным, специалистом в области информатики, одним из известнейших теоретиков в области разработки языков программирования.
- Язык назван в честь французского математика, физика, литератора и философа **Блеза Паскаля**, который создал одну из первых в мире механических машин, складывающую два числа.



История

Pascal

FlamingoText.com

- Первая публикация Вирта о языке датирована 1970 годом.
- Одной из целей создания языка Паскаль Никлаус Вирт считал обучение структурному программированию.
- По мнению Н. Вирта, язык должен был способствовать **дисциплинированию программирования**, поэтому, наряду со строгой типизацией, в Pascal сведены к минимуму возможные синтаксические неоднозначности. Синтаксис (правила) интуитивно понятен даже при первом знакомстве с языком.
- До сих пор Паскаль заслуженно считается **одним из лучших языков для начального обучения программированию**.
- Его современные модификации, такие как Object Pascal, широко используются в промышленном программировании (**среда Delphi**).

Развитие языка Паскаль



- Наиболее популярным решением для персональных компьютеров в 80-е - начале 90 годов стал компилятор и интегрированная среда разработки **Turbo Pascal** фирмы **Borland**. Встроенный компилятор обеспечивал высокую скорость компиляции и высокое качество кода (отсюда приставка Turbo).
- Выпущенная в **1995 г.** как продолжение среды **Turbo Pascal** система программирования **Delphi** стала одной из лучших сред для быстрого создания приложений. Начиная с версии Delphi 7.0, язык Delphi Object Pascal стал называться просто Delphi.

Развитие языка Паскаль



- Наиболее известной свободной реализацией языка Паскаль является **Free Pascal**. Помимо открытости исходного кода, его основным преимуществом является мультиплатформенность, а также **поддержка различных диалектов Паскаля**. На основе FreePascal создана свободная мультиплатформенная **среда Lazarus**, аналогичная среде **Delphi**.
- Однако, бедный и не меняющийся десятилетиями консольный **интерфейс** интегрированной среды **Free Pascal**, мало совместимый с современными интерфейсами рабочих столов операционных систем, всё более отталкивает обучаемых, неправильно формируя у них представление, что Паскаль - устаревший язык.

Развитие языка Паскаль



- В XXI веке растёт спрос на языки программирования, позволяющие писать высокоуровневый код в лёгкой и понятной форме.
- PascalABC.NET – это язык программирования Паскаль нового поколения, сочетающий простоту классического языка Паскаль, ряд современных расширений и огромные возможности платформы Microsoft .NET.
- PascalABC.NET включает бесплатную, простую и мощную среду разработки с подсказками по коду, автоформатированием и образцами кода для начинающих.
- **До сих пор Паскаль заслуженно считается одним из лучших языков для начального обучения программированию.**

Структура программы на языке Паскаль

```
program prog1;           // заголовок программы
Var a,c:real; i,j,k:integer; // блок описания
    s:string;           // переменных
begin                   // начало программы
readln(a);              // чтение данных
readln(i,j,k);         // чтение данных
c:=a*(i+j-k);          // вычисление значения выражения
writeln(c);           // вывод значения на экран
end.                   // конец программы
```

Структура программы

- Заголовок, описание переменных, начало и конец.
- Программа состоит из команд.
- Каждая команда программы заканчивается знаком «;».
- Каждая переменная в программе должна быть объявлена в блоке описания **Var**.
- Исполняемая часть программы начинается с ключевого слова **BEGIN**.
- Далее в программе могут быть любые команды, например вычисления, ввод данных (значений) с клавиатуры, вывод результатов на экран монитора.
- Программа заканчивается ключевым словом **END**.
- *Комментарии к программе пишутся после знаков «//» или «{» и их писать не обязательно.*

Что такое переменная?

$$5x^2 + x = 0$$

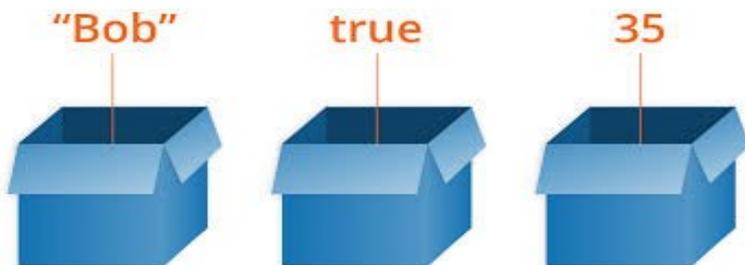
переменная

$$sq = 4$$

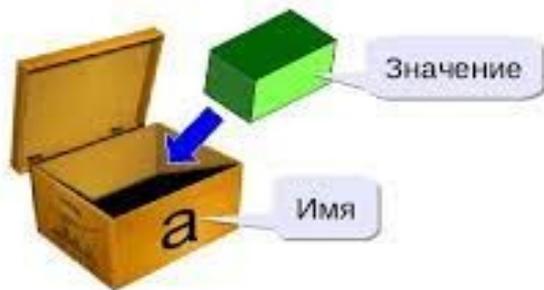
имя переменной операция присваивания данные



Что такое переменная в программировании?



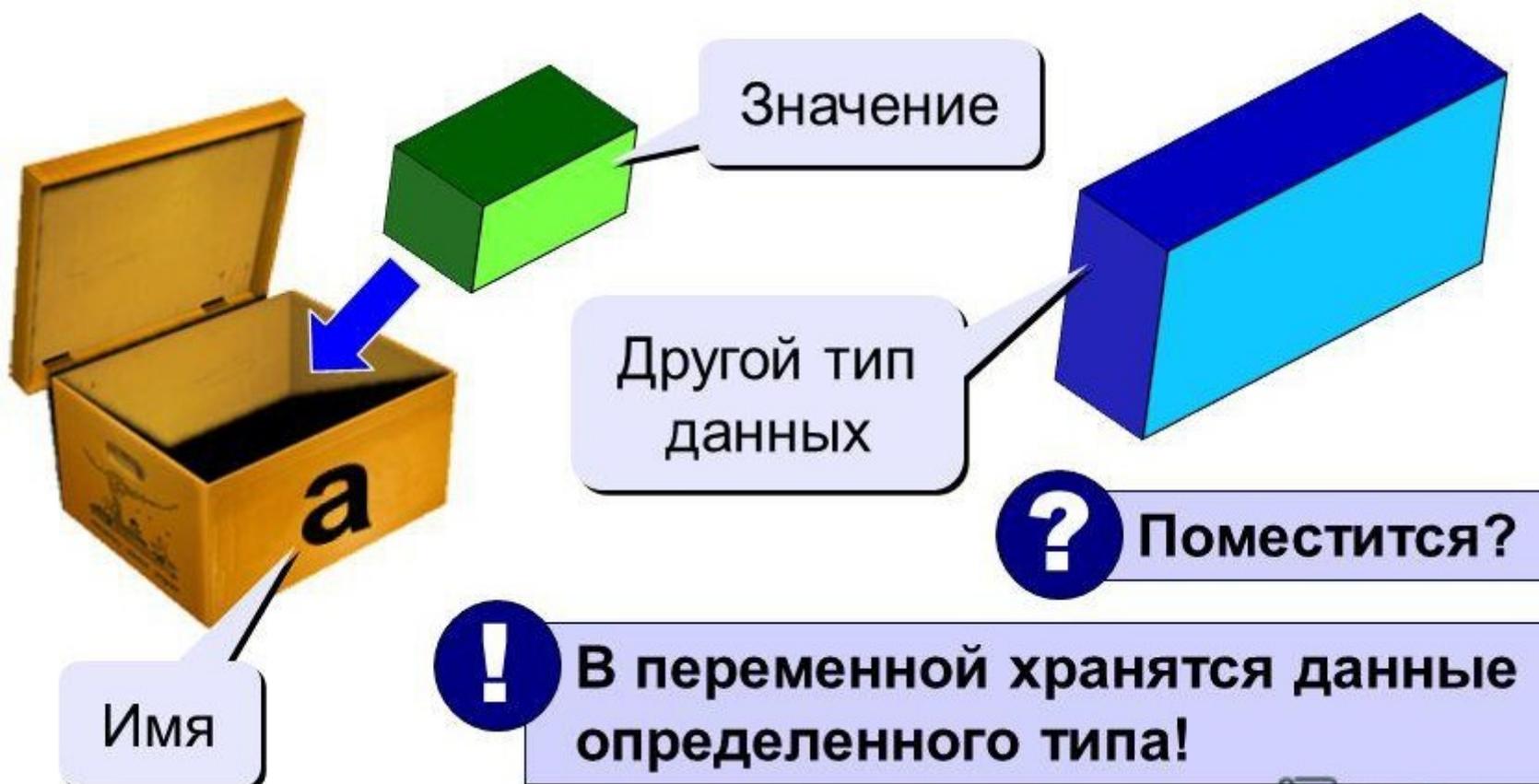
Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Ячейка (ячейки) в памяти компьютера, в которую можно помещать некоторые значения, имеющая определенный адрес с которым связано **Имя переменной.**

Понятие переменной

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имя переменной

Может содержать:

- латинские (большие или малые) буквы;
- цифры и другие символы.

В имени нельзя использовать &, %, буквы русского алфавита и пробелы;

Имя должно начинаться с буквы.

Примеры правильных имен: ab, c12, q1w, school_3

Примеры неправильных имен: &adc, %dfg, дгвт и т.д..

Типы данных

Количество ячеек в памяти компьютера, которые отводятся для записи переменной зависит от **Типа переменной (типа данных)**.

В Паскале существует несколько стандартных типов данных:

INTEGER (Целые) — для работы с целыми числами, каждая переменная **занимает в памяти 2 байта**.

REAL (Вещественные) - для работы с дробными числами, каждая переменная **занимает в памяти 6 байтов**. При записи в программе целая часть числа отделяется от дробной **точкой** «.», то есть: 12.5, -123.45 и т.д.

STRING (Строковые или текстовые) — для работы со строкой текста. Длина строки не превышает 255 символов (букв).

Команда присваивания

Для записи присваивания используется знак «:=»

Например: **a:=b*(c+d)/0.5**

Слева от знака присваивания должно стоять имя переменной, справа — число или арифметическое выражение.

В выражении могут быть использованы скобки.

Знаки арифметических операций:

+ - сложение

- - вычитание

***** - умножение

/ - деление

Операторы ввода и вывода данных

Вывод – это процесс передачи данных после обработки из оперативной памяти на внешнее устройство.

`Write('ПРИВЕТ')` – выводит слово ПРИВЕТ

`writeln('ПРИВЕТ')` – выводит слово ПРИВЕТ с новой строки

`write(c)` — выводит на экран монитора значение переменной `c`.

`writeln(a,b,c)` - выводит на экран монитора значение переменных `a`, `b` и `c` новой строки, числа выводятся без пробелов.

`writeln(a,' ', b)` — выводит на экран монитора значение двух переменных с новой строки. Между числами будет пробел.

`write('c=',5)` – выводит на экран: `c=5`

Вывод данных

- В случае использования процедуры **write**, после вывода последнего элемента списка на экран, курсор останется в той же строке.
- Процедура **writeln**, после завершения вывода, переводит курсор на следующую строку.

Правила записи параметров процедуры вывода:

- Список вывода разделяется запятыми.
- Список вывода может содержать переменные, константы и выражения.

Форматный вывод данных

Под вывод любого значения можно задать **формат поля вывода** - количество позиций, в которых будет размещаться величина (указывается сразу после выводимого значения через двоеточие):

- **Пример: `Writeln(a:5, b:5)` – на вывод чисел отводится 5 позиций. Если $a=10$ и $b=7$ на экране будет: 10 7 (числа разделены тремя пробелами).**

Для вещественных чисел (не целых, имеющих дробную часть, **например 2.5**) в формате поля вывода предусматривается возможность указания количество выводимых позиций и отводимых **под дробную часть** числа (десятичная точка также занимает одну позицию).

- **Пример: `writeln('a = ', a:2, ' ', 10 + sqrt(a):2:4, ' ', pi:4:2);`**

Ввод данных

Ввод данных — это процесс передачи исходных данных от внешнего устройства в оперативную память.

read(x,y) — вводит с экрана монитора в программу значения двух переменных в соответствии с их типом

readln(z, t, p) — вводит с экрана монитора в программу значения трех переменных в соответствии с их типом.

Процедура **readln** аналогична процедуре **read**, единственное отличие заключается в том, что после считывания последнего значения переменной из списка, происходит перевод следующего ввода данных в начало новой строки.

Пример

Пусть необходимо ввести в программу 4 целых числа.

- Дадим переменным имена — **a, b, c, d.**
- **В программе:**
- В блоке описания переменных запишем:
- **var a, b, c, d:integer;**

Пример

- В программе после **begin** запишем:
- **readln(a,b,c,d);**
- С клавиатуры переменные должны быть введены в соответствии с типом (целые числа данных **a, b, c, d** через пробел, например: **4 6 10 -2**
- После ввода значений необходимо нажать кнопку **<Enter>** — числа будут введены в программу и их значения будут присвоены переменным **a, b, c, d** (**a=4, b=6, c=10, d=-2**).
- вводимые значения должны соответствовать типам переменных, в противном случае возникает ошибка (например, переменная **b** имеет тип **integer**, а вводится вещественное число, например – **2.5**).