

Язык программирования Паскаль

Паскаль - это универсальный язык программирования, позволяющий решать самые разнообразные задачи обработки информации

Паскаль - язык структурного программирования, т.е. на нем с помощью нескольких конструкций можно выразить любые алгоритмы: линейные, ветвление, циклические конструкции.

Структурное программирование эффективно используется для решения различных математических задач, имеющих алгоритмический характер

Основные принципы структурного программирования:

- программирование должно осуществляться сверху вниз;
- весь проект должен быть разбит на модули (подпрограммы) с одним входом и одним выходом;
- допускаются только три основные структуры – линейная, ветвление и цикл;
- недопустим оператор передачи управления в любую точку программы (goto), т.к. нарушается условие чтения программы сверху вниз, *поэтому структурное программирование часто называют программированием без GOTO.*
- комментарии пишутся одновременно с программой.

Достоинства структурного программирования:

- надежность программ (благодаря хорошему структурированию при проектировании, программа легко тестируется и не создает проблем при отладке);
- повышается эффективность программ (структурирование программы позволяет легко находить и корректировать ошибки, а отдельные подпрограммы можно переделывать (модифицировать) независимо от других);
- уменьшается время и стоимость программной разработки;
- улучшается читабельность программ.

На основе стандартного Turbo Pascal создано множество «диалектов» языка, дополненных различными возможностями, такие «диалекты» называют реализациями языка (например, Pascal ABC).

Последние версии языка представляют полную среду для профессионального программирования, обладающую высокими характеристиками:

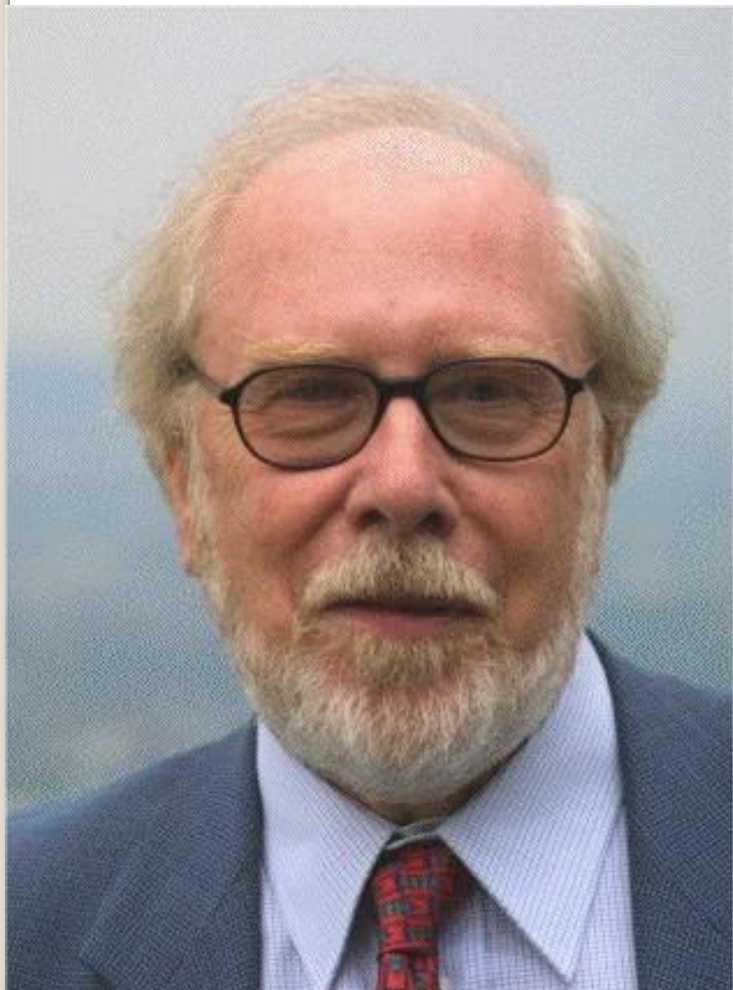
- ✓ Наличие системных библиотечных модулей;
- ✓ Наличие инструментов объектно-ориентированного программирования;
- ✓ Наличие встроенного в интегрированную среду программирования компилятора, обладающего высокой скоростью и позволяющего генерировать оптимизированный код, обеспечивающий быстрое выполнение программ;
- ✓ Наличие интегрированного в среду программирования отладчика, позволяющего выполнить полную проверку переменных, структур данных и выражений по шагам или в заданных точках;
- ✓ Наличие встроенных процедур и функций.

ВОЗНИКНОВЕНИЕ И НАЗНАЧЕНИЕ ПАСКАЛЯ



Язык назван в честь выдающегося французского математика, физика, литератора и философа Блеза Паскаля, который создал первую в мире механическую машину, складывающую два числа.

ВОЗНИКНОВЕНИЕ И НАЗНАЧЕНИЕ ПАСКАЛЯ



Язык Паскаль был создан Никлаусом Виртом в 1968—1969 годах. Он был опубликован в 1970 году как небольшой и эффективный язык, чтобы способствовать хорошему стилю программирования, использовать структурное программирование и структурированные данные.

Структура программы

Program Primer2; {Заголовок программы}
Var {Объявление раздела переменных}
x,y:Integer; {Собственно описание двух
переменных, тип - целые числа}
BEGIN {Начало программы}
x:=5;
y:=6; {Операторы, присваивающие значение
переменным}
Writeln(x+y); {Оператор вывода на экран с
переводом строки}
END. {Конец программы, все что написано
после точки - игнорируется}

Данные и величины

Совокупность величин, с которыми работает компьютер, принято называть данными.

Различают исходные данные, результаты и промежуточные данные, получаемые в процессе вычислений.

Например:

$$A * X^2 + B * X + C = 0$$

A, B, C – исходные данные (коэффициенты уравнения)

X_1 и X_2 – результат (корни уравнения)

$D = B^2 - 4 * A * C$ - промежуточное значение (дискриминант)

Каждая величина занимает свое определенное место в памяти ПК.

Свойства величины

Любая величина имеет **три основных свойства:**
ИМЯ, ЗНАЧЕНИЕ и ТИП.

Величины подразделяются на **постоянные (Const)** и **переменные.**

Идентификаторы и зарезервированные слова

Идентификаторы - это имена констант, переменных, меток, типов, объектов, процедур, функций, модулей, программ и полей в записях, которые задает программист.

Идентификатором считается любая последовательность латинских букв или цифр, начинающаяся с буквы. Символ подчеркивания "_" - буква.

Например, a1, _h, b123 - идентификаторы, а 1a, ф2 - нет.

Зарезервированные слова (ключевые) служат для оформления конструкций языка и не могут использоваться как идентификаторы:

**and begin case class const else end finalization finally for mod
not program var** и другие

Описание переменных

Переменной называется величина, которая изменяет свои значения в ходе выполнения программы и представляется в алгоритме символическими именами:

X, S2, SUM1

Раздел описания переменных начинается со служебного слова **var**, после которого следуют элементы описания вида

var

a,b,c: integer; - список имен: тип

Имена в списке перечисляются через запятую.

Описание констант

Константы – это данные, значения которых в процессе выполнения программы не могут изменяться.

Константы вводятся в блоке `const`:

const

a=5;

b= π ;

c='значение неизвестно';

В общем виде:

ИМЯ КОНСТАНТЫ = выражение;

Оператор присваивания

Оператор – предписание о выполнении конкретных действий.

ИМЯ := ВЫРАЖЕНИЕ;

(:=) обозначает:

1. вычисляется выражение;
2. полученное значение присваивается переменной.

Формат записи

X := A;

где: **X** – имя переменной;

A - выражение;

:= - знак присваивания.

*Раздел операторов - основная часть программы. Начало и конец раздела операторов программы отмечаются служебными словами **begin** (начало) и **end** (конец).*

Пример последовательного выполнения 4-х команд присваивания, в которых участвуют две величины a и b .

Команда	a	b
$a := 1$	1	-
$b := 2 * a$	1	2
$a := b$	2	2
$b := a + b$	2	4

Пример последовательного выполнения команд для замены значений 2-х переменных X и Y . (замена жидкостей в 2-х стаканах)

Команда	X	Y	Z
Ввод x, y	1	2	-
$Z := X$	1	2	1
$X := Y$	2	2	1
$Y := Z$	2	1	1

Три основных свойства присваивания:

- ✓ Пока переменной не присвоено значение, она остается неопределенной;
- ✓ Значение, присвоенное переменной, сохраняется вплоть до выполнения следующего присваивания этой переменной;
- ✓ Новое значение, присваиваемое переменной, заменяет ее предыдущее значение.

Основные типы данных

ТИП	ЗНАЧЕНИЕ	ОПЕРАЦИИ	ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ
ЦЕЛЫЕ	23, -12, 387 и др. целые	+, - , / - арифм. операции <, >, = и др. – операции отношений	Формат с фиксированной точкой
ВЕЩЕСТВЕННЫЕ	2.5, -0.01, 45.0	арифм. операции, опер. отношений	Формат с плавающей точкой
ЛОГИЧЕСКИЕ	True (истина) False (ложь)	лог. операции and (и), or (или), not (нет), операции отношений	1-True 0-False
СИМВОЛЬНЫЕ	A, b, +, *, 8	операции отношений	Коды таблицы ASCII и др.

Типы переменных

Некоторые простые типы:

- 1. Целые типы** (ShortInt, Integer, LongInt, Byte, Word).
- 2. Вещественные типы** (Real, Single, Double, Extended, Comp).
- 3. Логический** (Boolean).
- 4. Символьный** (Char).
- 5. Строковые типы** (String, String [n]).

Тип переменных указывается в разделе «Var». Знаком «*» отмечены наиболее часто используемые типы.

Числовые целочисленные типы

Shortint -128..127

* **Integer** -32768..32767

Longint -2147483648..2147483647

* **Byte** 0..255

Word 0..65535

Типы числовых переменных с запятой (Вещественные типы)

- Real 2.9e-39..1.7e38 (10^{-39} - 10^{38})
- Single 1.5e-45..3.4e38
- Double 5.0e-324..1.7e308
- Extended 3.4e-4932..1.1e4932
- Comp -9.2e18..9.2e18

для хранения данных типа integer используется 2 байт, а для real необходимо 6 байт

Экспоненциальная запись числа. Способ рассчитать величину степени 10 состоит в расчете числа мест, на которые нужно передвинуть десятичный знак, чтобы вернуться к первоначальному числу. Если десятичный знак должен двигаться влево, степень 10 отрицательна. Если десятичный знак должен двигаться вправо, степень 10 будет положительной.

Тип	Диапазон	Число цифр	Память, байт
Real	2.9e-39 ... 1.7e38	11-12	6
Single	1.5e-45 ... 3.4e38	7-8	4
Double	5.0e-324 ... 1.7e308	15-16	8
Extended	3.4e-4932 ... 1.1e493 ?	19-20	10
Comp	-9.2e63 ... (9.2e63)-1 ?	19-20	8

$$12,3 = 1,23 \times 10^1$$

$$123 = 1,23 \times 10^2$$

$$123\ 000\ 000 = 1,23 \times 10^8$$

$$0,123 = 1,23 \times 10^{-1}$$

$$0,0123 = 1,23 \times 10^{-2}$$

$$0,000000000000123 = 1,23 \times 10^{-11}$$

Величина основания 10 часто заменяется буквой E, в частности, в таблицах, созданных компьютером. 'E' стоит перед показателем степени. Вот некоторые примеры:

$$1,25 \times 10^2 = 1,25\ E\ 02$$

$$3,57 \times 10^4 = 3,57\ E\ 04$$

$$1,23 \times 10^{-3} = 1,23\ E\ -\ 03$$

Строковый тип переменной

Строковая переменная – это переменная, которая относится к типу String и содержит в себя некоторый набор текста.

*** String Строка от 0 до 255 символов**

String[10] Строка от 0 до 10 символов
(используется для экономии памяти)

Логический тип переменных

Логической переменной называют переменную булевого значения **Boolean**. Эти переменные могут иметь всего 2 значения **True** или **False**.

* **Boolean** - логический тип, принимает только значение **True (Истина)** или **False (Ложь)**.

* **Char** - символьный тип, используется для ввода нажатой клавиши.

Комментарии

Комментарии - это участки кода, игнорируемые компилятором и используемые программистом для пояснения текста программы.

*{ Это
комментарий }*

(Это
тоже комментарий *)*

Комментарием также считается любая последовательность символов после символов // и до конца строки:

```
var Version: integer; // Версия продукта
```

МАТЕМАТИЧЕСКИЕ КОМАНДЫ

В тексте программы используют следующие команды:

+ Сумма

- Разность

* Произведение

/ Частное (разделить)

$3*(2+4)$ Скобки определяют порядок выполнения действий

Abs(X) Модуль числа

Sqr(X) Квадрат числа

Sqrt(X) Квадратный корень

10 Div 3 Деление нацело (результат = 3)

10 Mod 3 Остаток от деления (результат = 1)

Trunc(x) Целая часть числа

Операторы ввода и вывода

`read (readln)` — ввод значений с клавиатуры(читать);

`read (список переменных);`

`write (writeln)` — вывод результата (или чего-либо) на экран.

`write ('сообщение', список переменных);`

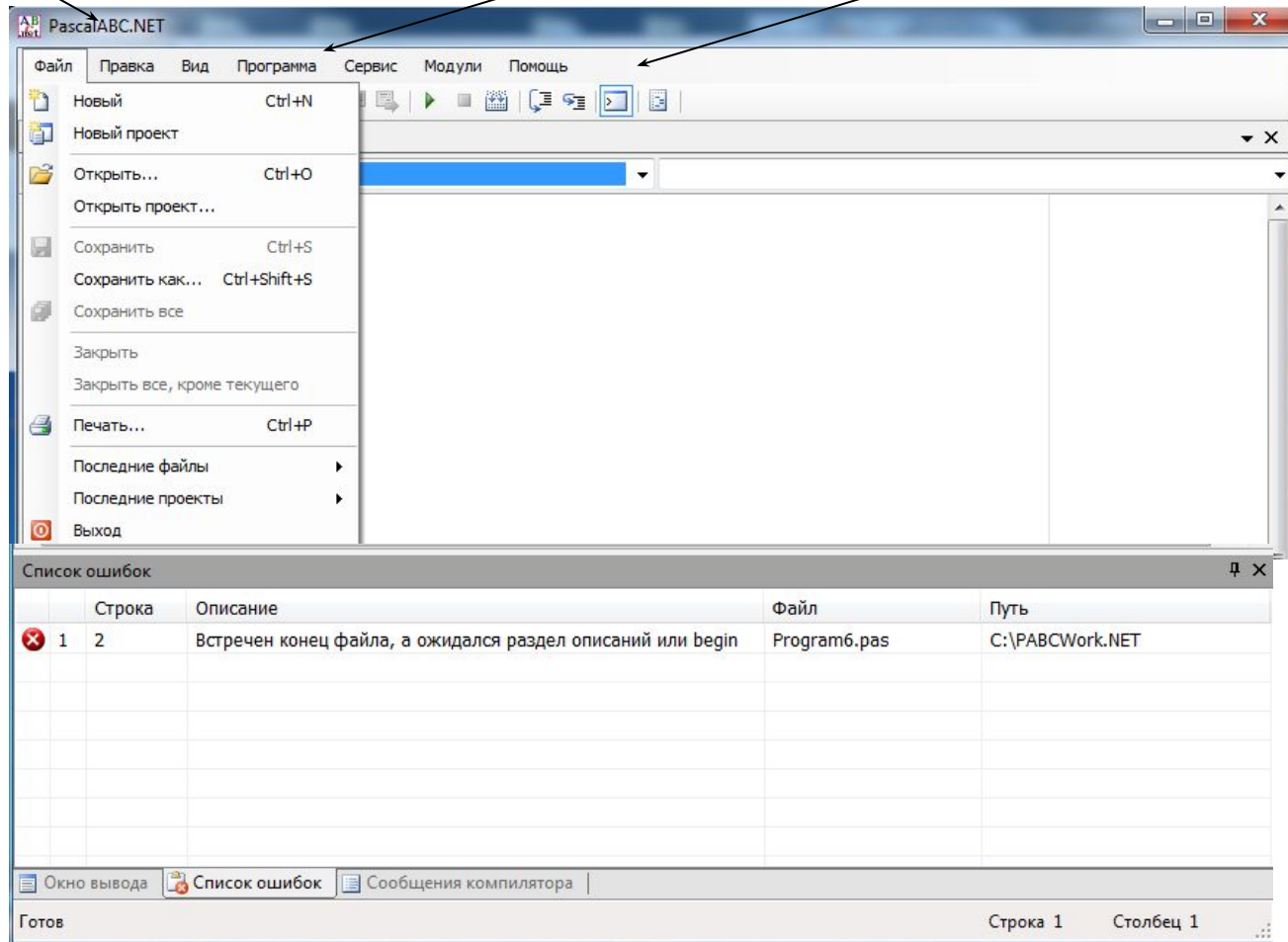
Отличие операторов **`read`** и **`write`** от операторов **`readln`** и **`writeln`** состоит в том, что после выполнения операторов **`readln`** и **`writeln`** курсор переводится на новую строку.

Интерфейс программы PascalABC.NET

Строка заголовка

Строка меню

Панель инструментов



Правка	Вид	Программа	Сервис
	Отменить		Ctrl+Z
	Восстановить		Ctrl+Shift+Z
	Вырезать		Ctrl+X
	Копировать		Ctrl+C
	Вставить		Ctrl+V
	Найти...		Ctrl+F
	Найти далее		Ctrl+L
	Заменить...		Ctrl+R
	Перейти к строке...		Ctrl+G
	Выделить все		Ctrl+A
	Закладки		

Вид	Программа	Сервис	Модули
	Очистить вывод		Ctrl+Del
	Нижняя панель		F5
	Окно вывода		
	Список ошибок		
	Сообщения компилятора		
	Результат поиска символов		
	Локальные переменные		
	Просмотр выражений		
	Перейти назад		Ctrl+-
	Перейти вперед		Ctrl+Shift+-

Программа	Сервис	Модули	Помощь
	Выполнить		F9
	Выполнить без связи с оболочкой		Shift+F9
	Параметры командной строки...		
	Завершить		Ctrl+F2
	Компилировать		Ctrl+F9
	Перекомпилировать все		Ctrl+Shift+F9
	Шаг без входа в подпрограмму		F8
	Шаг с входом в подпрограмму		F7
	Шаг с выходом из подпрограммы		F10
	Идти к курсору		F4

Сервис Модули Помощь

Настройки...

Настройки

- Общие
- Редактор
- Опции компиляции
- Intellisense

Общие

Язык

Русский

Показывать ошибки

Первую синтаксическую и первую семантическую

Делать паузу после завершения консольной программы

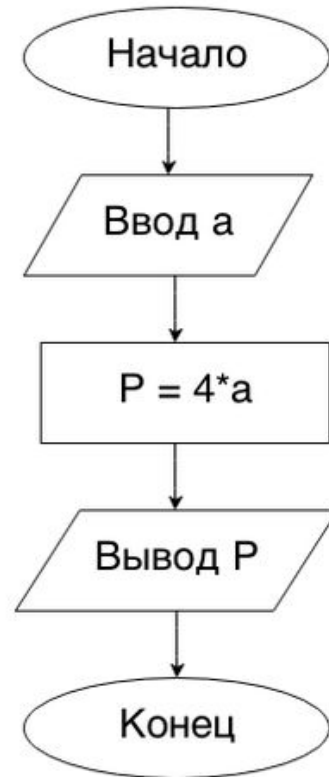
Сохранять файлы если компиляция прошла успешно

Помощь

- Справка F1
- Документация по .NET
- Коротко о главном
- Примеры
- Изучаем PascalABC.NET
- О программе

Задача L_1. Дана сторона квадрата a . Найти его периметр $P = 4 \cdot a$.

```
program L_1;  
var  
  a, P: real;  
begin  
  write('Введите a:');  
  readln(a);  
  P := 4 * a;  
  write('P=', p);  
end.
```



Задача L_2. Дана сторона квадрата a . Найти его площадь $S = a^2$.

При решении данной задачи воспользуемся функцией **sqr**.

```
program L_2;
```

```
var
```

```
  a, s: real;
```

```
begin
```

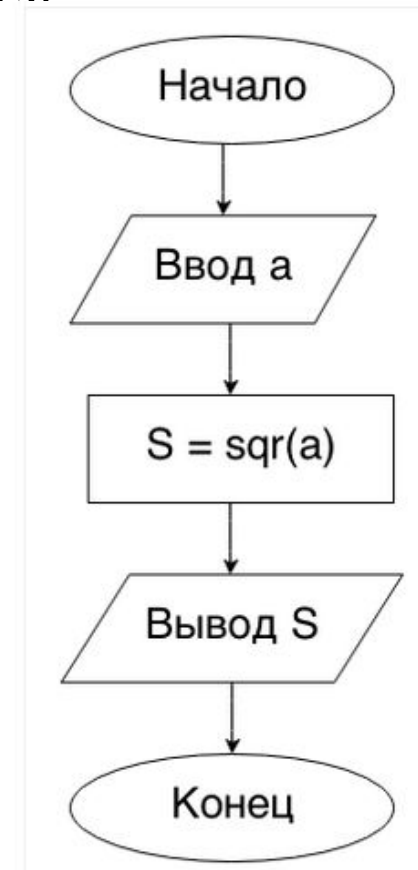
```
  write('Введите a: ');
```

```
  readln(a);
```

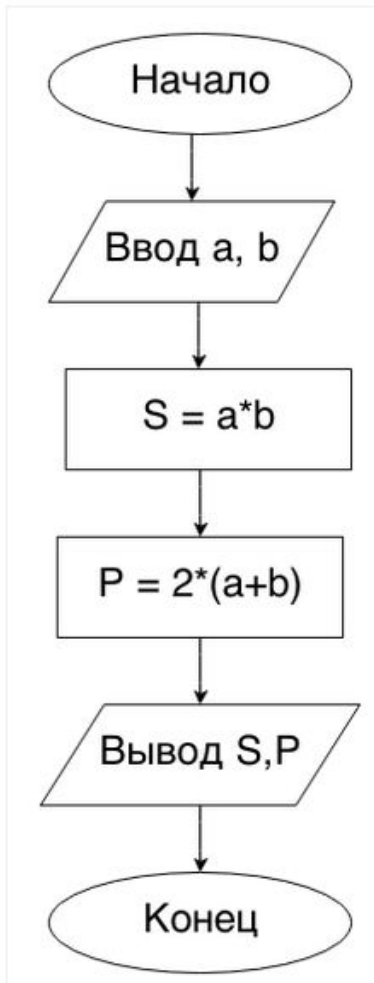
```
  s := sqr(a);
```

```
  write('Площадь квадрата -- ', s);
```

```
end.
```



Задача L_3. Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.



```
program L_3;  
var  
  a, b, S, P: real;  
begin  
  write('Введите a: ');  
  readln(a);  
  write('Введите b: ');  
  readln(b);  
  s := a * b;  
  p := 2 * (a + b);  
  write('Площадь прямоугольника -- ', s, ');  
  write('Периметр прямоугольника -- ', p);  
end.
```

$$z = \frac{\pi \cdot H}{6} (p + H^2), \quad \text{где } p = 3(a^2 + b^2)$$

Program Lin;

var

a,b,p,z,h:real;

Begin

write('введите число a=');

readln(a);

write('введите число b=');

readln(b);

write('введите число h=');

readln(h);

p:=3*(sqr(a)+sqr(b));

z:=pi*h/6*(p+sqr(h));

writeln(z);

end.

albina-ivlieva@yandex.ru

Правила оформления отчета

Отчет должен быть предоставлен в текстовом документе.

- ✓ **Имя файла: Иванов_12 ИС_лин.**
- ✓ **В документе должен быть колонтитул с Фамилией и номером группы студента, выставлена нумерация страниц.**
- ✓ **Блок-схема алгоритма.**
- ✓ **Скопированный в среде программирования код программы.**
- ✓ **Скопированный в среде программирования результат решения задачи.**

Отчет должен быть оформлен согласно требованиям:

- ✓ **Шрифт TNR 12, междустрочный интервал – одинарный, интервал перед и после абзаца – 0.**