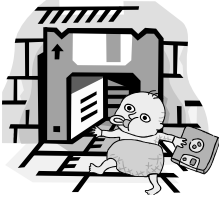


Неформальное введение в язык Pascal



*Усложнять просто,
упрощать сложно.
Закон Мейера*

Компьютерные науки

© М.Л. Цымблер

Содержание

- Историческая справка о языке Pascal
- Пример простейшей программы
- Программные объекты
- Общий вид программы

Неформальное введение в язык Pascal

© М.Л. Цымблер

2

Историческая справка

- Разработан в 1968-71 гг. **Никлаусом Виртом**, профессором Цюрихского института информатики (Швейцария). Первоначальное назначение – *обучение студентов программированию*. Оказался эффективным языком высокого уровня.
- Является одним из самых распространенных языков программирования. Имеет многочисленные диалекты и расширения.
- Послужил основой для языка **Ada** – языка разработки больших программных систем с повышенными требованиями к надежности.



Неформальное введение в язык Pascal

© М.Л. Цымблер

3

Основные свойства Pascal

- Простой синтаксис
- Развитая система типов данных
- Блочная организация программы
- Строгая типизация
 - фиксация типов переменных
 - строгий контроль преобразования типов и доступа к данным в соответствии с их типом
- Отражение в управляющих структурах принципов структурного программирования

Неформальное введение в язык Pascal

© М.Л. Цымблер

4

Пример простейшей программы

```

{ hello.pas
  19-мар-05
  Иванов И.И., МП-101
  Пример простейшей программы. }
program Hello;
begin
    WriteLn("Привет!");
end.
    
```

Комментарий –
спецификация
программного
файла

Заголовок программы

Главный блок

Тело программы

Неформальное введение в язык Pascal

© М.Л. Цымблер

5

Ввод-вывод данных

- Ввод данных с клавиатуры
 - Read(<список переменных>)
 - ReadLn(<список переменных>)
- Вывод данных на экран
 - Write(<список переменных>)
 - WriteLn(<список переменных>)

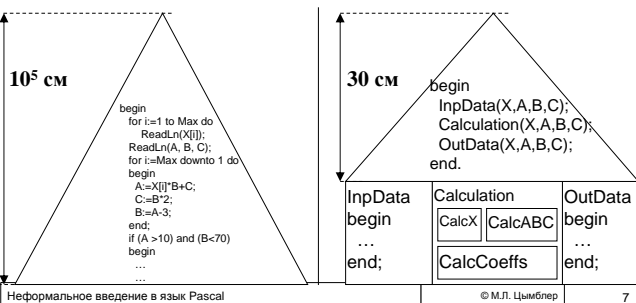
Неформальное введение в язык Pascal

© М.Л. Цымблер

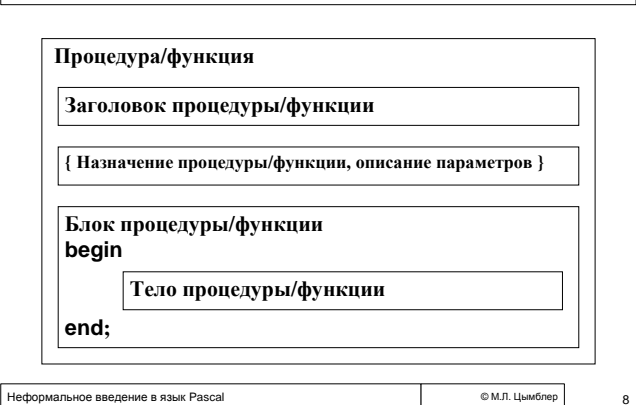
6

Процедуры и функции

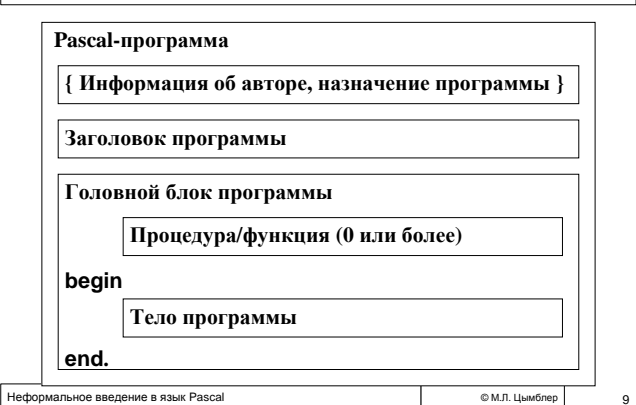
■ Процедуры и функции (подпрограммы) позволяют разделить программу на небольшие, хорошо понятные и читаемые модули.



Структура процедуры/функции



Структура программы с проц./функ.



Пример: программа с процедурами

```

{ myprog2.pas, 19-мар-05
Иванов И.И., МП-101
Пример программы с процедурами и функциями }
Program MyProg2;
var
  A, B, C: Integer; R: Real;
procedure InpData(var A, B, C: Integer);
{ Осуществляет ввод исходных данных с клавиатуры }
begin
  ...
end;
procedure Calculation(A, B, C: Integer; var Result: Real);
{ Выполняет некоторые вычисления с исходными данными }
begin
  ...
end;
procedure OutData(Result: Real);
{ Осуществляет вывод результатов вычислений на экран }
begin
  ...
end;
begin
  InpData(A,B,C);
  Calculation(A,B,C, R);
  OutData(Result: Real);
end.
    
```

Операторы

- Секция кода между **begin** и **end** содержит *операторы*, описывающие действия, выполняемые данной программой, и называется *секцией операторов*.

Примеры операторов

```

A:=B+C;      { Присваивание значения }
Calculate(L, N);{ Вызов (активация) процедуры }
if X<2 then  { Условный оператор }
  A:=X*Y;
begin       { Составной оператор }
  X:=3;
  Y:=4;
end;
while not EOF(InFile) do{ Оператор цикла }
begin
  Readln(InFile, Line);
  Process(Line);
end;
    
```

Выражения

- В состав операторов могут входить *выражения*. Выражение состоит из *операций* и *операндов* (то, над чем выполняются операции).

Неформальное введение в язык Pascal

© М.Л. Цымблер

13

Примеры выражений

-X
 A+B
 (C mod 7 – D div 8) * 5
 Z/3.14
 Flag <> Error
 5<2
 not (A or B and C)

Неформальное введение в язык Pascal

© М.Л. Цымблер

14

Лексемы

- *Лексема* – это наименьшая единица языка, имеющая самостоятельный смысл. Лексемы, в частности, используются для построения операндов и операций в выражениях.
- Различаются следующие *классы* лексем:
 - *специфические символы*
 - *зарезервированные (ключевые) слова*
 - *метки*
 - *идентификаторы*
 - *числа*
 - *строковые константы.*

Неформальное введение в язык Pascal

© М.Л. Цымблер

15

Примеры лексем

```
begin end { Резервированное слово }
)      { Спецсимвол }
:=     { Спецсимвол }
Calculate { Идентификатор }
123    { Число }
$123   { Число }
1E-23  { Число }
'To be or not to be?' { Строковая константа }
'Madam I'm Adam' { Строковая константа }
```

Типы, переменные, константы

- *Переменная* хранит значение, которое может изменяться. Каждая переменная должна иметь определенный *тип*.
- *Тип переменной* определяет множество значений, которые может принимать данная переменная.
- При описании *константы* ей назначается значение, которое никогда не может меняться.
- При описании *типизированной константы* указывается ее тип и начальное значение, которое может быть впоследствии изменено. Типизированную константу можно рассматривать как переменную с начальным значением.

Пример программы

```
Program Example;
const
  A=12;      { Константа А никогда не изменит свое значение }
  B: Integer = 3; { Типизированная константа В получает начальное значение }
var
  X, Y: Integer; { Переменные X И Y – целого типа (Integer) }
  J: Real;      { Переменная J – вещественного типа (Real) }
begin
  X:=1;        { Переменной X присваивается значение }
  Y:=2;        { Переменной Y присваивается значение }
  X:=X+Y;      { Значение переменной X изменяется }
  B:=52;       { Типизированная константа В получает новое значение }
  J:=0.05;     { Переменная J получает значение с плавающей точкой }
end.
```

Модули Unit

- Программа на Turbo Pascal может использовать блоки кода из отдельных модулей, называемых *unit*.
- Подобно программе, модуль **unit** имеет заголовок, в котором определяется его имя, и головной блок, содержащий тело модуля между зарезервированными словами **begin** и **end**.
- Если программа использует модуль **unit**, то его имя указывается в предложении *uses* перед головным блоком программы.

Неформальное введение в язык Pascal

© М.Л. Цымблер

19

Пример использования модуля unit

```
{ myprog3.pas, 19-мар-05
Иванов И.И., МП-101
Пример программы с модулем unit. }
Program MyProg3;
uses Ucalc; { Подключение модуля unit }
var
  A, B, C: Integer;
  R: Real;
procedure InpData(var A, B, C: Integer);
{ Осуществляет ввод исходных данных }
begin
  ...
end;
procedure OutData(Res: Real);
{ Осуществляет вывод результатов }
begin
  ...
end;
begin
  InpData(A,B,C);
  Calculation(A,B,C, R);
  OutData(R);
end.
```

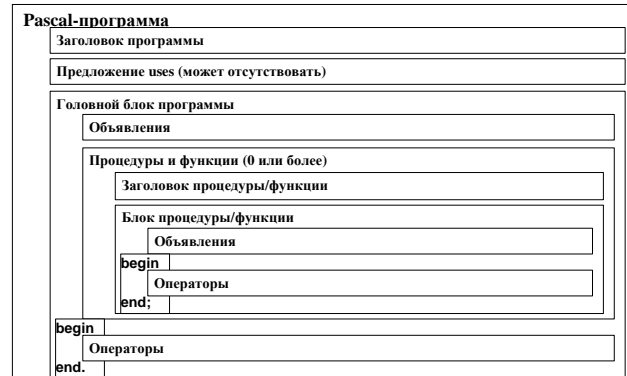
```
{ ucalc.pas, 19-мар-05
Петров П.П., МП-102
Пример модуля unit. }
unit Ucalc;
interface
procedure Calculation(A, B, C: Integer; var
Result: Real);
{ Выполняет некоторые вычисления с
исходными данными }
implementation
procedure Calculation(A, B, C: Integer; var Result:
Real);
{ Выполняет некоторые вычисления с
исходными данными }
begin
  ...
end;
procedure CalcX(A, B, C: Integer; var X: Real);
{ Выполняет вспомогательные вычисления }
begin
  ...
end;
...
end.
```

Неформальное введение в язык Pascal

© М.Л. Цымблер

20

Общий вид Pascal-программы



Неформальное введение в язык Pascal

© М.Л. Цымблер

21

Вложенность блоков

- **Блок может содержать другой блок:**
процедуры и функции могут включать в себя вложенные объявления процедур или функций.
