Элементы языка Turbo Pascal

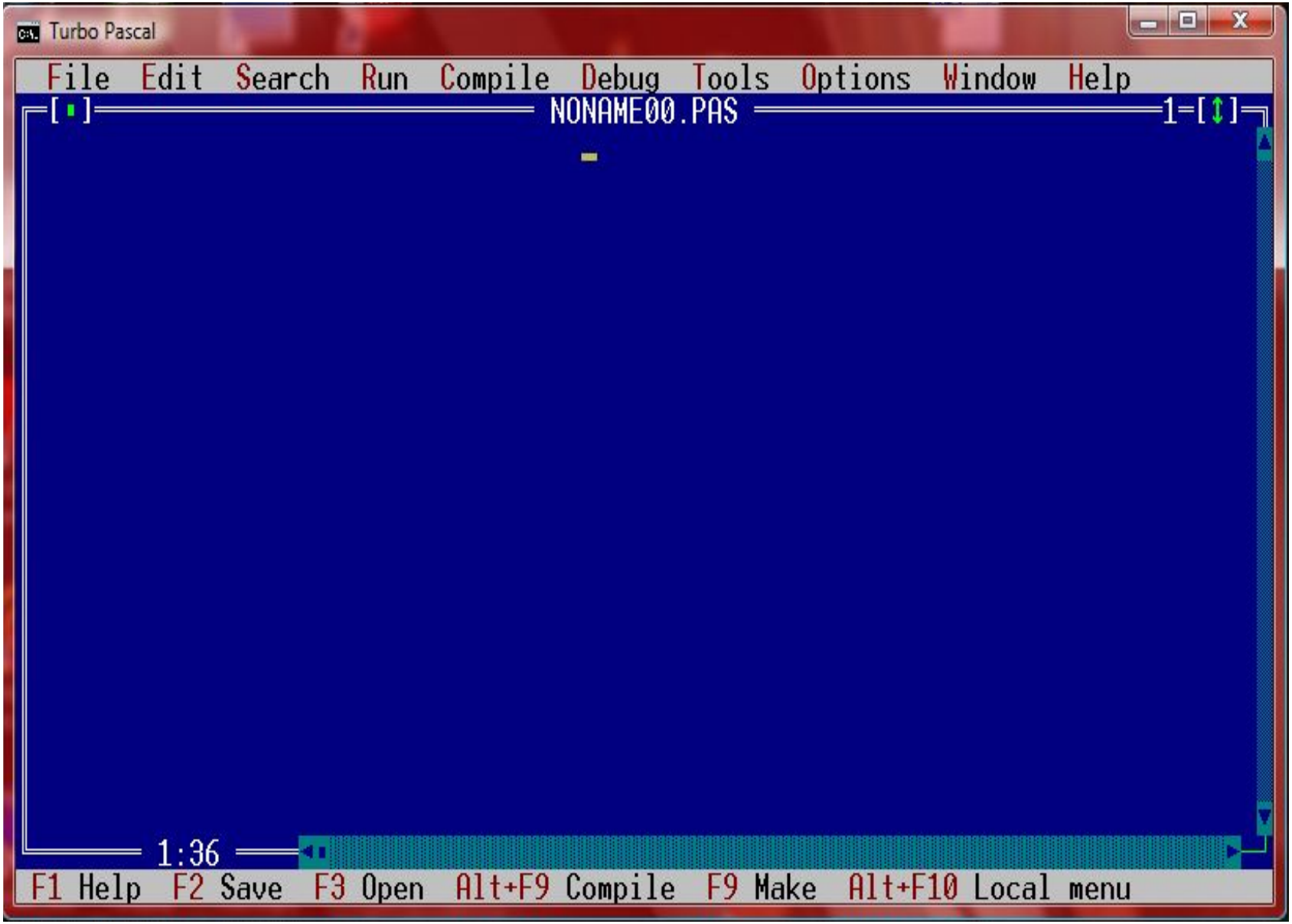
Блез Паскаль (1623 - 1662) –
французский математик, физик,
философ и писатель.
Считал человека трагичным и хрупким существом,
находящимся между двумя безднами –
бесконечностью и ничтожеством
(человек – «мыслящий тростник»).
Все, о чем писал Паскаль,
было глубоко им пережито и выстрадано.
Лучше всего о себе сказал он сам:
«Я только с теми, кто, стеная, ищет истину».

История

- Язык программирования Турбо Паскаль, названный в часть французского математика и философа Блеза Паскаля (1623-1662), разработан в 1968 -1671 гг. Николаусом Виртом, профессором, директором Института информатики Швейцаркой высшей Политехнической школы. Язык Паскаль, созданный первоначально для обучения программированию как систематической дисциплине, вскоре стал широко использоваться для разработки средств в профессиональном программировании.

Систему программирования Турбо Паскаль называют ***интегрированной:***

- Множества накладывающихся окон;
- Поддержка мыши, меню, диалоговых окон;
- Многофайловый редактор;
- Расширенные возможности отладки;
- Полное сохранение и восстановление среды разработки.


Turbo Pascal
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
1:36
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

F1	Вызов справки
F2	Сохранить программу на диске
F3	Открыть файл с текстом программы
At+ F3	Закрыть окно редактирования
Alt+F5	Просмотр результата работы программы
Ctrl+F5	Изменение размеров окна(Shift + стрелки)
F6	Вызвать следующее окно для редактирования
Shift+F6	Вызвать предыдущее окно редактирования

F10	Вход в главное меню
Ctrl+F9	Запуск программы на выполнение
F3	Открыть файл с текстом программы
At+ X	Выход из PASCAL
Alt+F5	Просмотр результата работы программы
Ctrl+ Break	Прерывание исполнения программы
Esc	Закрытие диалогового окна или окна меню
Alt+ цифра	Переход в окно редактирования с таким номером (от 1 до 9)

Клавиши редактирования текста

Shift + стрелки	Выделить фрагмент текста
Shift+Del	Удалить выделенный текст в буфер
Ctrl+Ins	Скопировать выделенный текст в буфер обмена
Shift+ Ins	Вставить текст из буфера обмена
Ctrl+Del	Удалить выделенный текст, не помещая его в буфер обмена.
Ctrl+Y	Удалить строку, в которой находится курсор.
Alt+Back Spase	Отменить последнее действие редактирования.

Алфавит ЯП PASCAL

1. Латинские буквы – большие и маленькие;
2. Буквы кириллицы - большие и маленькие;
3. Цифры – от 0 до 9;
4. Знаки операций - +, -, /, *,=,<, >, <>, <=, >=;
5. Разделительные знаки – ( ) [ ] { } ; . , ‘ _% & # и т. д.

Служебные слова

Служебные слова – предназначены для написания команд. В Турбо-Паскале есть несколько служебных десятков слов, которые программисту нельзя использовать в качестве имен переменных. Такими словами являются:

and	case	procedure
file	function	until
not	or	do
string	type	if
else	const	program
begin	goto	var
for	to	downto
of	mod	in
then	div	while

Идентификаторы

Правила создания идентификаторов

1. Состоит из строчных или прописных латинских букв, цифр и знака подчеркивания «_».
2. Начинается с буквы или знака подчеркивания «_».
3. Не может быть служебным словом.
4. Длина не должна превышать 127 символов
5. Желательно, чтобы идентификатор отображал смысл переменной.

Правильные идентификаторы:

Temp_ x1 _33name

_1_2_3 My_Variable

Неправильные идентификаторы:

Temp- 1x 33name1_2_3

My Variable

План работы при создании и отладке новой прогаммы

1. Открыть новое окно редактирования для ввода программы **Файл ⇨Новый.**
2. Набрать текст новой программы.
3. Сохранить текст программы на диске **Файл ⇨Сохранить** и указать путь и имя файла.
4. Запустить программу на выполнение **Ctrl+F9.**
5. Если есть ошибки, то исправить их.
6. Просмотреть результат выполнения программы **Alt+F5.**
7. Сохранить правильную программу на диске F2.

Раздел описания меток.

- Метка состоит из имени и следующего за ним двоеточием. Именем может служить идентификатор или число. Раздел описания меток начинается зарезервированным словом label, за которым следуют имена меток, разделенными запятыми. В конце последнего имени ставиться точка с запятой.
- Формат:

Label < имя, ..>

- **ok10;**

- После записи метки в разделе операторов, следует двоеточие, показывающее компилятору, что идентификатор используется как метка:

```
Label
M1, M2;                    { описание  метки}
begin
   …
  M1:< оператор >  {использование M1 в разделе
   операторов}
   …
  M2:< оператор >  {использование M2 в разделе
   операторов}
end
```

Раздел описания констант

- В разделе описания констант производится идентификация констант постоянных значений. Раздел начинается зарезервированным словом **const,** за которым следует ряд выражений, присваивающих идентификаторам постоянные числовые или строковые значения. Выражения присвоения отделяются друг от друга точкой с запятой.
- Формат: **const <идентификатор> = <значение>**

- Например:

```
Const
   MaxInd:= word=100;
  {типизированная константа}
   Name =’Петя ’;
  {строковая константа}
   Code = $124;          {константа –
  шестнадцатеричное значение}
   N =10;
   Pi= 3.14159265;
```

Раздел описания типов данных

- Тип данных может быть описан либо непосредственно в разделе описания переменных, либо определяться идентификатором типа. Стандартные типы не требуют описания.
- Раздел описания типов данных начинается зарезервированным словом **type**, за которым следуют одно или несколько определений типов (integer, real, char, boolean и т.д), разделенных точкой с запятой.
- Формат:

type < имя типа >=<значение типа>

- Например:

```
type
   LatLetter=( ‘ A ’  . . ’ z ’);
   Days = 1 . . 31;
   Mart = array   [1 . . 10]  of integer;
```

- Каждое описание задает множество значений и связывает с этим множеством некоторое имя типа.

Раздел описания переменных

- **К**аждая встречающаяся в программе переменная должна быть описана. Описание обязательно предшествует использованию переменной. Раздел описания переменной начинается зарезервированным словом **var**, затем через запятую перечисляются имена переменных и через двоеточие следует их тип и точка с запятой.
- Формат :

Var

<идентификатор,..> : <тип>;

- Например:

```
var
   {описание раздела переменных}
  A,B, Proizved: integer;
   {переменные A,B, Proizved – целые}
  X, h,  sum: real;
  D,  l,  r:  string;
```

Раздел описания процедур и функций

- В этом разделе размещаются тела подпрограмм. *Подпрограммой* называется программная единица, имеющая имя, по которому она может быть вызвана из других частей программы. В языке Паскаль роль подпрограмм выполняют процедуры и функции.
- Для описания подпрограмм используются зарезервированные слова **procedure и function**, которые записываются в начале подпрограмм.

Раздел операторов

- Это основной раздел программы. Раздел операторов начинается словом **begin**, далее следует оператор языка. Завершает раздел зарезервированное слово **end.**

- Операторы выполняются строго последовательно в том порядке, в котором они записаны в тексте программы в соответствии с синтаксисом и правилам пунктуации.

- **Например:**

```
begin                                   {начало программы}
   Write('введите значение целого числа A>');
  {ввод запроса на экран}
   Readln(A);
   Write('введите значение целого числа B>');
   Readln(B);
      P:=A*B;                           {вычисление
  переменной P}
   Write('произведение чисел ',A,' и ',B' = ', P);
  {вывод ответа}
end.                                    {конец
  программы}
```

КОММЕНТАРИЙ

- Не выполняются программой, а служат для пояснения отдельных ее частей.
- В текст программы комментарии могут быть включены в фигурных скобках {это комментарий} или в круглых скобках в сопровождении символа «*» - (*это тоже комментарий *)

Общий вид программы:

```
program (имя программы);
label  (список меток);
const (список постоянных значений);
type (описания сложных типов данных);
var (описания данных программы);

begin (начало программного блока)
   выполняемые операторы,
   реализующие заданный алгоритм
end  (конец программы)
```

Программа на Паскале состоит из двух частей (разделов): описания используемых данных и операторов по их преобразованию. Вторая часть (раздел) также называется программным блоком (или разделом выполняемых операторов).

Команда присваивания

Переменная := Выражение;

A:=3*4.8;

Su:=X+X*4.78;

C:=C+1;

Между всеми элементами выражения должны быть знаки операций.

3x ⇨ 3*x

Аргументы функций должны быть заключены в ():

sinx ⇨ sin(x)

Команда ввода

Read (Список переменных);

Readln (Список переменных);

Примеры:

Read **(I, j);**

Readln **(k);**

При выполнении команды *Read* или *Readln* выполнение программы останавливается и компьютер ждет, пока пользователь не введет с клавиатуры нужное количество значений для переменных.

Вводятся только значения для переменных.

Ввод заканчивается нажатием клавиши ENTER.

Readln **отличается от *Read* тем, что после его выполнения автоматически осуществляется переход на следующую строку.**

Команда вывода

Write (Список выражений);

Writeln (Список выражений);

Значения выражений сначала вычисляются, затем выводятся на экран. После выполнения команды *Writeln* следующая команда ввода или вывода начинает свою работу с новой строки.

Примеры:

Пусть

i=1, j=2, k=3

l=4, m=5, n=6

После выполнения команд:

Write (l, j);

Writeln (k);

Write (l, m, n)

На экране получим:

123

456

Команда ввода

Формат вывода

Для того, чтобы числа не «слипались при выводе на экран, можно указать компьютеру сколько позиций необходимо выделить для данной переменной. Это делается так:

Writeln (x:8,y:5)