

БИБЛИОГРАФИЯ КУРСА

1. Овсянник В.Н.. Основы программирования на языке Borland Pascal – Учеб. пособие. В 2 ч. – Харьков: Гос. аэрокосмический ун-т “Харьк. авиац. ин-т”, 1999. Ч.1. – 180 с.
2. Овсянник В.Н.. Основы программирования на языке Borland Pascal – Учеб. пособие. В 2 ч. – Харьков: Нац. аэрокосмический ун-т “Харьк. авиац. ин-т”, 2000. Ч. 2. – 180 с.

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

Язык **Borland Pascal (BP)** был разработан профессором Швейцарской Высшей технической школы **Никласом Виртом** и первая публикация о нем появилась в **1971 г.**, который и принято считать **годом рождения языка.**

Свое название язык получил в честь известного французского ученого **Б. Паскаля.**

BP представляет собой **язык высокого уровня**, предназначенный в первую очередь для **первоначального обучения программированию.**

Одной из главных причин **большой популярности языка BP** явилась его интегрированная среда разработки программ **IDE (Integrated Development Environment)**, которая обеспечивает разработчика очень **удобными и высокоскоростными средствами создания и отладки программ.**

IDE для пользователя является **программой-оболочкой**, объединяющей в единое целое следующие компоненты:

- многооконный текстовый редактор;
- высокоскоростной компилятор (компиляция программы за один проход);
- компоновщик (линкер);
- встроенный отладчик;
- справочная система;

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

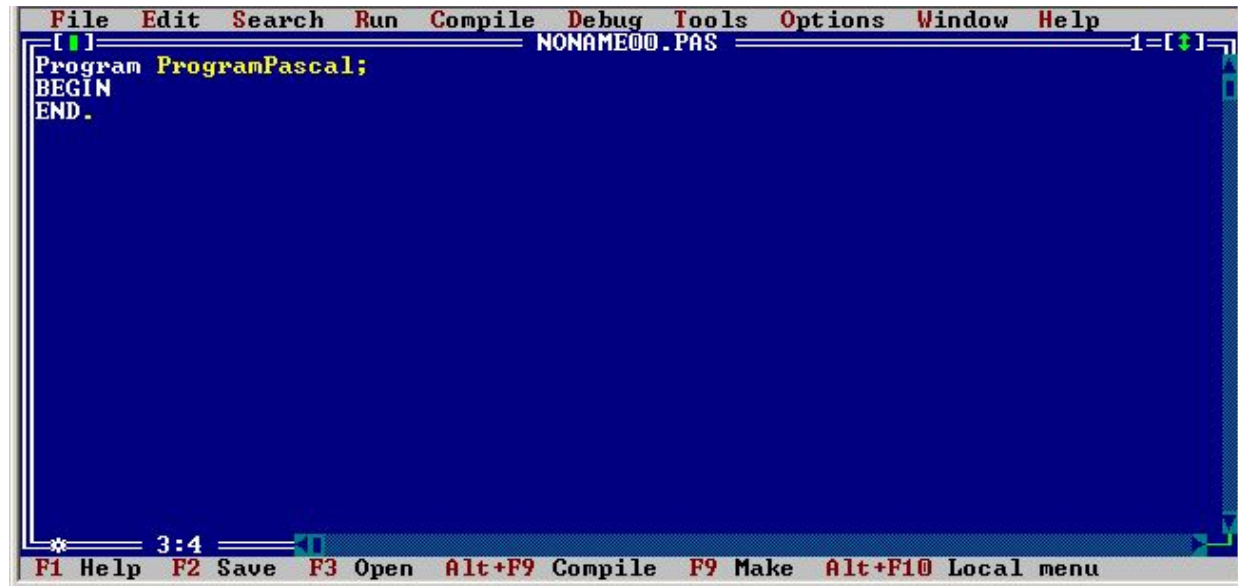
BP имеет хорошие средства для разработки **простых, понятных и надежных программ.**

В программе в машинных кодах каждый оператор представляет собой машинную команду – **элементарную операцию.**

ПРИМЕР ПРОГРАММЫ:

BEGIN

END.

A screenshot of a Turbo Pascal IDE window. The title bar reads "NONAME00.PAS". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The main text area contains the following code:

```
Program ProgramPascal;  
BEGIN  
END.
```

The status bar at the bottom shows "3:4" and various function key shortcuts: "F1 Help", "F2 Save", "F3 Open", "Alt+F9 Compile", "F9 Make", and "Alt+F10 Local menu".

Вместе с тем отсюда следует, что два зарезервированных слова (reserved words) **BEGIN** и **END** являются необходимой принадлежностью программы: **BEGIN** используется для объявления начала тела программы, а **END** с точкой после него – для указания компилятору на конец программы.

Обычно программа включает в себя также зарезервированное слово **Program**, которое позволяет ввести имя программы.

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

Имя программы, как и другие вводимые пользователем символические имена, может состоять **только из букв латинского алфавита, символа подчеркивания и цифр.**

Язык **ВР** не накладывает строгих ограничений на способ записи программы:

- операторы программы могут начинаться с любой позиции строки;
- в тексте допускаются пустые строки;
- вводимые программистом идентификаторы могут быть составлены из строчных или заглавных букв;



```
File Edit Search Run Compile Debug Tools Options Window Help
[ ] NONAME00.PAS 1=[↑↓]
PROGRAM ProgramPascal;begin End.
```

- в одной строке программы может быть записано несколько операторов;



```
File Edit Search Run Compile Debug Tools Options Window Help
[ ] NONAME00.PAS 1=[↑↓]
Program ProgramPascal;BEGIN END.
```

- один оператор может занимать произвольное число строк.

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ



```
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS 1=[↓]
PROGRAM ProgramPascal; <the name of program>
begin End.
```

В этой программе использованы **комментарии** для пояснения ее текста.

Комментарии в **ВР** – это **любой текст, заключенный в фигурные скобки**.

Комментарии игнорируются компилятором и не оказывают никакого влияния на выполнение программы.

Программа, написанная на **ВР**, состоит из **лексем и разделителей**.

Лексемы – это минимальные значимые единицы текста в программе, состоящие из разрешенных символов.

Они представлены следующими категориями:

- идентификаторы (символические имена);
- числовые, символьные и строковые константы;
- метки;
- специальные символы, в том числе знаки операций.

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

Лексемы должны быть отделены друг от друга одним или несколькими разделителями.

Разделитель представляет собой пробел, символ табуляции, комментарий или символ начала следующей строки (ASCII-код 13).

BP использует следующие подмножества набора символов кода ASCII как разрешенные символы:

- буквы – буквы английского алфавита от A до Z и от a до z;
- цифры – арабские цифры от 0 до 9;
- символ подчеркивания (ASCII-код 95).
- специальные символы (всего 22):

+ - * / = < > [] . , () : ; ^ @ { } \$ # ‘

Следующие пары символов являются **неделимыми**, т.е. между ними **нельзя** поставить разделитель, и имеют специальное назначение:

- знак присвоить := ;
- знаки операций отношения <> <= >= ;
- указатель диапазона .. (две точки);
- начало и конец комментария или директивы компилятора (* *);
- эквивалентные квадратным скобкам пары символов (.) .

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

Идентификаторы выступают в качестве имен констант, типов, переменных, процедур, модулей, программ и полей в записях.

Для составления идентификаторов можно использовать:

- латинские буквы заглавные **A..Z** и строчные **a..z**;
- цифры **0..9**;
- символ подчеркивания **_**.

Идентификатор может иметь любую длину, однако только первые его 63 символа являются значимыми.

Идентификатор должен **начинаться с буквы или символа подчеркивания** и не может содержать пробелов.

Идентификатор может **состоять из одной буквы или одного или нескольких символов подчеркивания**, но не может начинаться с цифры.



```
File Edit Search Run Compile Debug Tools Options Window Help
[ ] NONAME00.PAS 1=[ ]
read READ Read rEAD
a MyFunction _Read_File_From_Disk Work_2_
```

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

Все идентификаторы можно разделить на две основные группы:

- зарезервированные **слова и стандартные директивы ВР;**

and	Exports	mod	Shr
array	file	nil	String
asm	for	not	Then
begin	function	object	To
case	goto	of	Type
const	if	or	Unit
constructor	implementation	packed	Until
Destructor	in	procedure	Uses
Div	inherited	program	With
Do	inline	record	While
downto	interface	repeat	xor
else	label	set	var
end	library	shl	

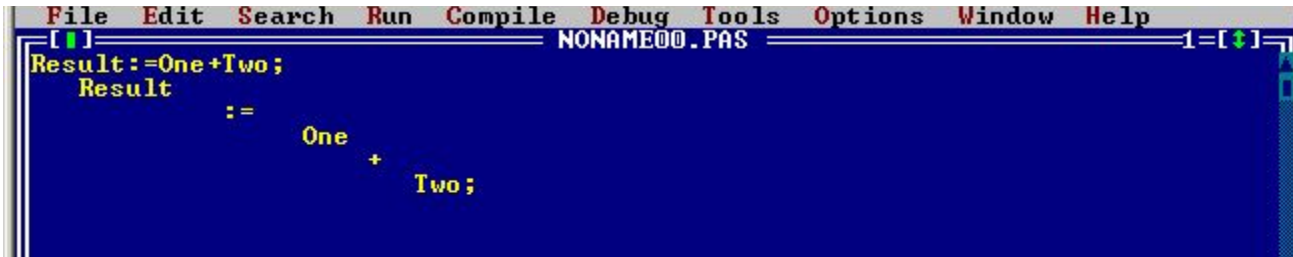
- пользовательские идентификаторы.

ИНФОРМАЦИОННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ.

ОБЩИЕ ПОНЯТИЯ

ПРАВИЛА ЗАПИСИ ПРОГРАММЫ:

- в целях повышения читабельности программы рекомендуется набирать строки, которые помещаются на экран монитора целиком, т.е. длиной до **80 СИМВОЛОВ**;
- в одной строке программы может быть записано любое число операторов или один (длинный) оператор может быть записан в нескольких строках;



```
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
Result:=One+Two;
Result
:=
One
+
Two;
```

- при записи текста программы для пояснения ее операторов или других целей можно (и нужно!) использовать комментарии;



```
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
S<sum>:=A<1 summand>+B<2 summand>
(*sum*)
<It's a long comment
which occupies two lines>
```