Глава 2. Алгоритмы и основы языка программирования TURBO PASCAL 7.0

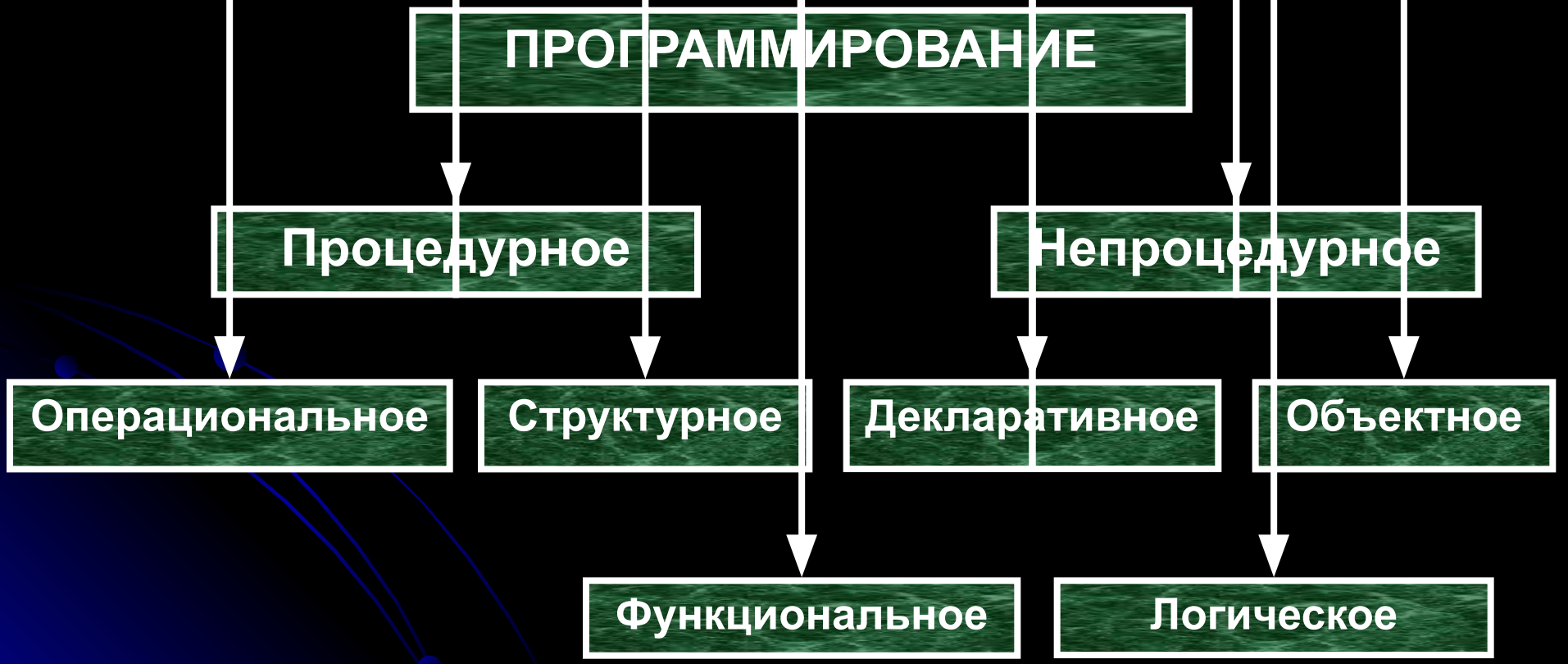
Лекция 7. Введение в Turbo Pascal

2.7.1. Основные понятия программирования
2.7.2. Элементы языка Turbo Pascal
2.7.3. Среда Turbo Pascal

2.7.1. Основные понятия программирования

Классификация языков программирования

Программирование — это раздел информатики, задача которого — разработка программного обеспечения ЭВМ.



2.7.1. Основные понятия программирования

Типы программирования

Процедурное программирование — программа представляет собой детальное описание решения задачи в виде совокупности процедур и функций.

Операциональное программирование — программа представляет собой детальное описание решения задачи на языке, набор команд которого совпадает с системой команд исполнителя (ЭВМ).

Структурное программирование — программа представляет собой детальное описание решения задачи, то есть алгоритм в некоторой специальной записи.

Непроцедурное программирование — программа представляет собой описание решения задачи использованием различных конструкций, отличных от процедур и функций.

Объектное программирование — программа представляет собой множество независимых объектов.

Декларативное программирование — программа представляет собой набор исходных информационных структур, взаимосвязи между ними и свойства результата.

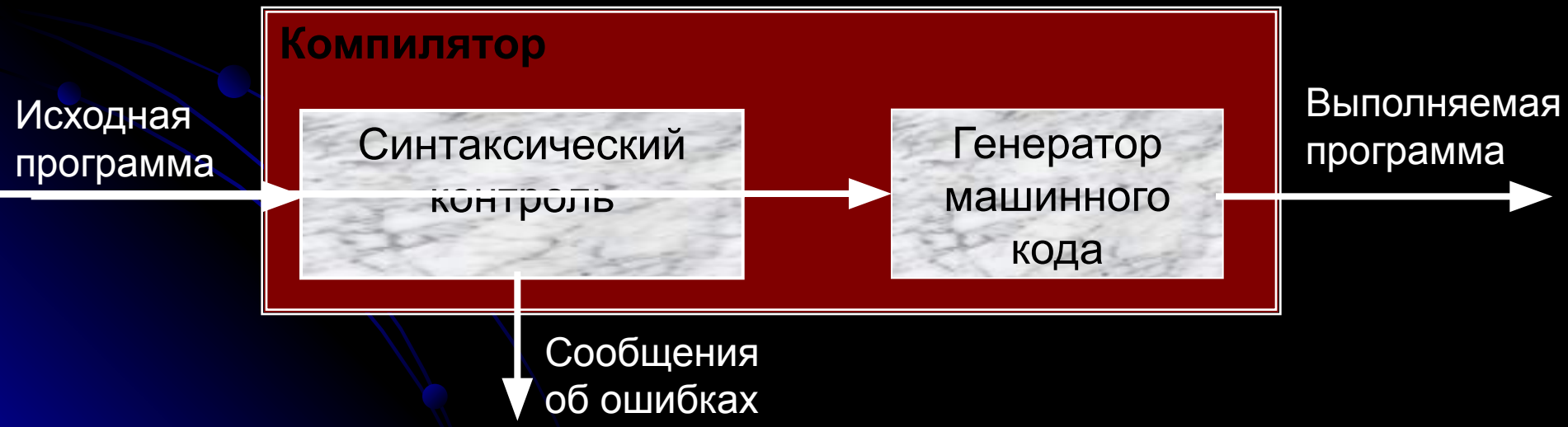
2.7.1. Основные понятия программирования

Язык высокого уровня Turbo Pascal

Программа состоит из операторов (интегрированных команд), каждому из которых может соответствовать несколько машинных операций.

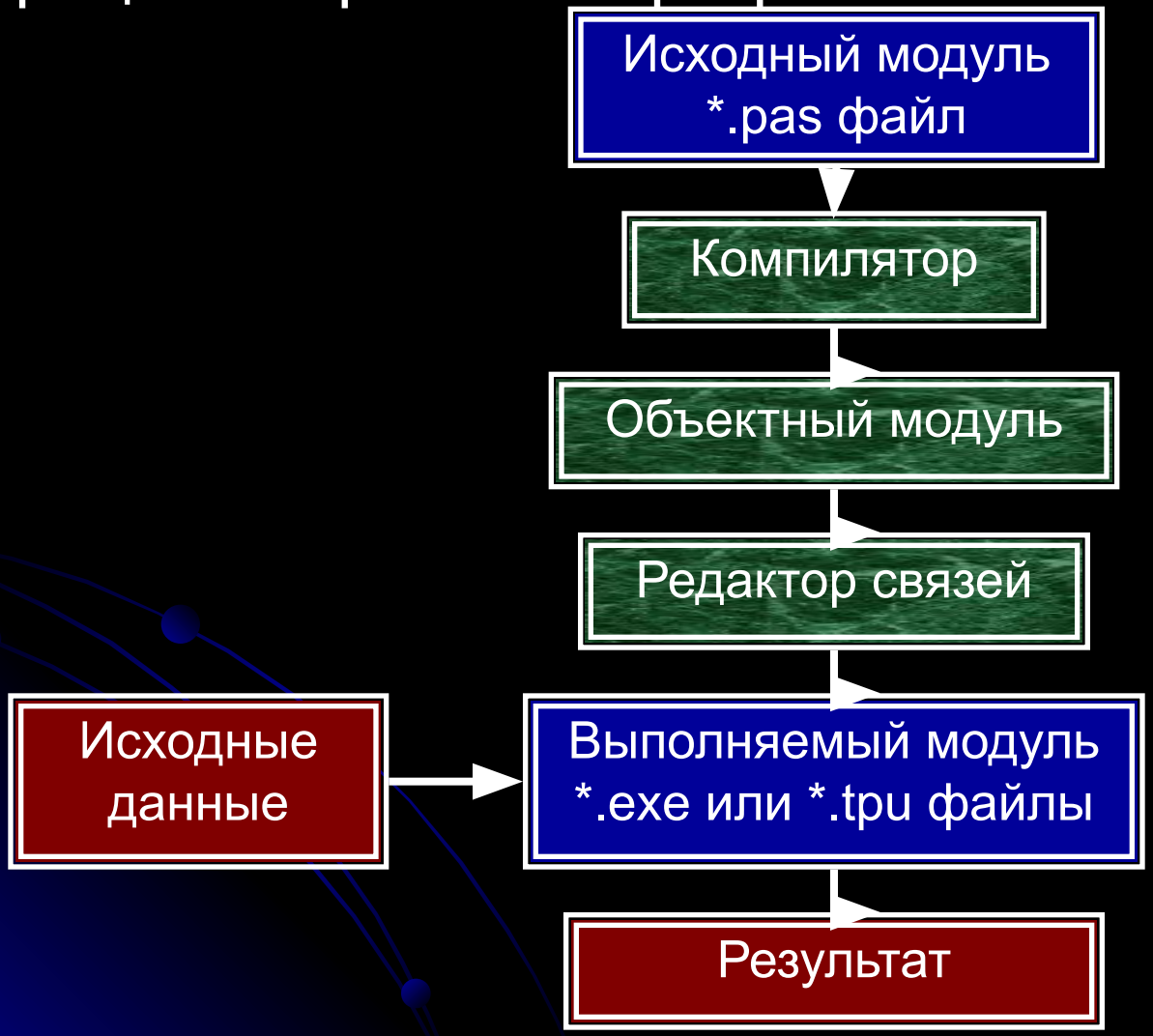
Алгоритмический язык высокого уровня — это система правил для представления данных и описания процесса их обработки.

Компиляция программ



2.7.1. Основные понятия программирования

Процесс обработки программы на языке Turbo Pascal



2.7.1. Основные понятия программирования

Структура программы

ЗАГОЛОВОК ПРОГРАММЫ

Program name

РАЗДЕЛ ОПИСАНИЙ

1. Библиотеки — Uses
2. Метки — Label
3. Константы — Const
4. Типы — Type
5. Переменные — Var
6. Процедуры — Procedure
7. Функции — Function

Begin

РАЗДЕЛ ОПЕРАТОРОВ (ТЕЛО ПРОГРАММЫ)

Операторы присваивания
Операторы ввода-вывода
Оператор безусловного перехода
Составные операторы
Условный оператор
Операторы цикла
Оператор выбора
Процедуры
Функции

End.

2.7.2. Элементы языка Turbo Pascal

Алфавит

1. Буквы английского алфавита	A, B, C, D, …, X, Y, Z, a, b, c, d, …, x, y, z, _
2. Арабские цифры	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
3. Шестнадцатеричные цифры	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
4. Специальные символы	| + | - | * | | / | = | , | ' | . | : | ; | < | > | | [|] | (|) | { | | } | ^ | @ | $ | # | <> | |<= | >= | := | (* | *) | (. | .)|
5. Пробелы	« », « », « », « »
6. Зарезервированные слова	

2.7.2. Элементы языка Turbo Pascal

Алфавит: зарезервированные слова

Их нельзя использовать в качестве идентификаторов!!!

AND	DIV	GOTO	NIL	RECORD	TYPE
ARRAY	DO	IF	NOT	REPEAT	UNIT
ASM	DOWNTO	IMPLEMENTATION	OBJECT	SET	UNTIL
BEGIN	ELSE	IN	OF	SHL	USES
CASE	END	INLINE	OR	SHR	VAR
CONST	FILE	INTERFACE	PACKED	STRING	WHILE
CONSTRUCTOR	FOR	LABEL	PROCEDURE	THEN	WITH
DESTRUCTOR	FUNCTION	MOD	PROGRAM	TO	XOR

2.7.2. Элементы языка Turbo Pascal

Алфавит: стандартные директивы

Их можно использовать в качестве идентификаторов, но не желательно!!!

ABSOLUTE	FAR	NEAR
ASSEMBLER	FORWARD	PRIVATE
EXTERNAL	INTERRUPT	VIRTUAL

2.7.2. Элементы языка Turbo Pascal

Идентификаторы

Идентификаторы в Турбо Паскале — это имена констант, переменных, меток, типов, объектов, процедур, функций, модулей, программ и полей в записях.

Идентификаторы могут иметь длину не более 63 символов.

Идентификатор всегда начинается буквой, за которой могут следовать буквы и цифры. Пробелы и специальные символы алфавита не могут входить в идентификатор.

Правильные идентификаторы	Неправильные идентификаторы

`a`
`ALPHA`
`MyProgramIsBestProgram`
`date_27_sep_39`
`external`
`_beta`
	`1Program`
	`block#1`
	`My Prog`
	`mod`

2.7.2. Элементы языка Turbo Pascal

Заголовок программы

Program <идентификатор заголовка программы>;

Программа начинается словом Program и содержит объявление имени программы.

Слово Program зарезервировано в Turbo Pascal, т.е. не может использоваться ни в каких иных целях, кроме как для объявления имени программы.

В Turbo Pascal можно опускать объявление имени оператором Program без каких-либо последствий для программы, так как имя программы никак в дальнейшем не используется.

Задание имени программы

```
Program My_First_Program;
```

2.7.2. Элементы языка Turbo Pascal

Библиотеки

Uses <идентификатор модуля 1>[, <идентификатор модуля 2>[, …]];

В Turbo Pascal имеется восемь стандартных модулей, в которых содержится большое число разнообразных типов, констант, процедур и функций. Этими модулями являются SYSTEM, DOS, CRT, PRINTER, GRAPH, OVERLAY, TURBOS и GRAPH3.

Модуль SYSTEM подключается к любой программе автоматически, даже без указания в Uses.

Модули можно создавать в Turbo Pascal самостоятельно.

Подключение библиотек

```
Uses CRT;
```

```
Uses CRT, Printer;
```

2.7.2. Элементы языка Turbo Pascal

Метки

Label <идентификатор метки 1>[, <идентификатор метки 2>[, …]];

Метка в Turbo Pascal — это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него.

В языке Turbo Pascal допускается в качестве меток использование также целых чисел без знака.

Метка располагается непосредственно перед помечаемым оператором и отделяется от него двоеточием.

Оператор можно помечать несколькими метками, которые в этом случае отделяются друг от друга двоеточием.

Описание меток

```
Label L;
```

```
Label
 L10, 20, 1b2;
```

2.7.2. Элементы языка Turbo Pascal

Константы

1. Целые числа.
2. Вещественные числа.
3. Шестнадцатеричные числа.
4. Логические константы.
5. Символы.
6. Строки символов.
7. Конструкторы множеств.
8. Признак неопределенного указателя NIL.

2.7.2. Элементы языка Turbo Pascal

Константы: целые числа

Целые числа записываются со знаком или без него по обычным правилам и могут иметь значение от -2147483648 до +2147483647.

Если целочисленная константа выходит за указанные границы, то она должна записываться с десятичной точкой, то есть определяется как вещественное число.

Объявление целых констант

```
Const
 x1 = 10;
 y = -20;
```

2.7.2. Элементы языка Turbo Pascal

Константы: вещественные числа

Вещественные числа записываются со знаком или без него с использованием десятичной точки и/или экспоненциальной части.

Экспоненциальная часть начинается символом **e** или **E**, за которым могут следовать знаки «+» или «-» и десятичный порядок. Символ e (**E**) означает десятичный порядок и имеет смысл «умножить на 10 в степени».

Объявление вещественных констант

```
Const
 E = -2.7;
 Ya = 154.31;
 Mn = 3.14E5;
 N = -17e-2;
```

2.7.2. Элементы языка Turbo Pascal

Константы: шестнадцатеричные числа

Шестнадцатеричное число состоит из шестнадцатеричных цифр, которым предшествует знак доллара $.

Диапазон шестнадцатеричных чисел — от $00000000 до $FFFFFFFF.

Объявление шестнадцатеричных констант

```
Const
 E = $5AD;
 Y = $10FF;
 M = $209;
```

2.7.2. Элементы языка Turbo Pascal

Константы: логические константы

Логическая константа — это либо слово FALSE (ложь), либо слово TRUE (истина).

Объявление логических констант

```
Const
 Flag = True;
 L = False;
```

2.7.2. Элементы языка Turbo Pascal

Константы: символьные константы

Символьная константа — это любой символ ПК, заключенный в апострофы.

Если необходимо записать собственно символ апострофа, он удваивается: "" — символ ' (апостроф).

Допускается использование записи символа путем указания его внутреннего кода, которому предшествует символ #.

Объявление символьных констант

```
Const
 Z = 'ш';
 C = 'A';
 M = 'o';
 F = '''';
 D = #90;
 Ya = #13;
```

2.7.2. Элементы языка Turbo Pascal

Константы: строковые константы

Строковая константа — любая последовательность символов (кроме символа CR — возврат каретки), заключенная в апострофы.

Если необходимо записать символ апострофа, он удваивается.

Строка символов может быть пустой, т.е. не иметь никаких символов в обрамляющих ее апострофах.

Строку можно составлять из кодов нужных символов с предшествующими каждому коду символами #.

Объявление строковых констант

```
Const
 Str = '';
 C = 'A';
 Dn = 'Понедельник';
 F = 'That''s string.';
 D = #83#121#109#98#11#108;
 Y = 'Ошибка!'#13'Нажмите любую клавишу ...';
```

2.7.2. Элементы языка Turbo Pascal

Константы: конструкторы множества

Конструктор множества — список элементов множества, обрамленный квадратными скобками.

Объявление констант-множеств

```
Const
 A = [1,2,4..7,12];
 Color = [blue, red];
 M = [];
```

2.7.2. Элементы языка Turbo Pascal

Константы выражения

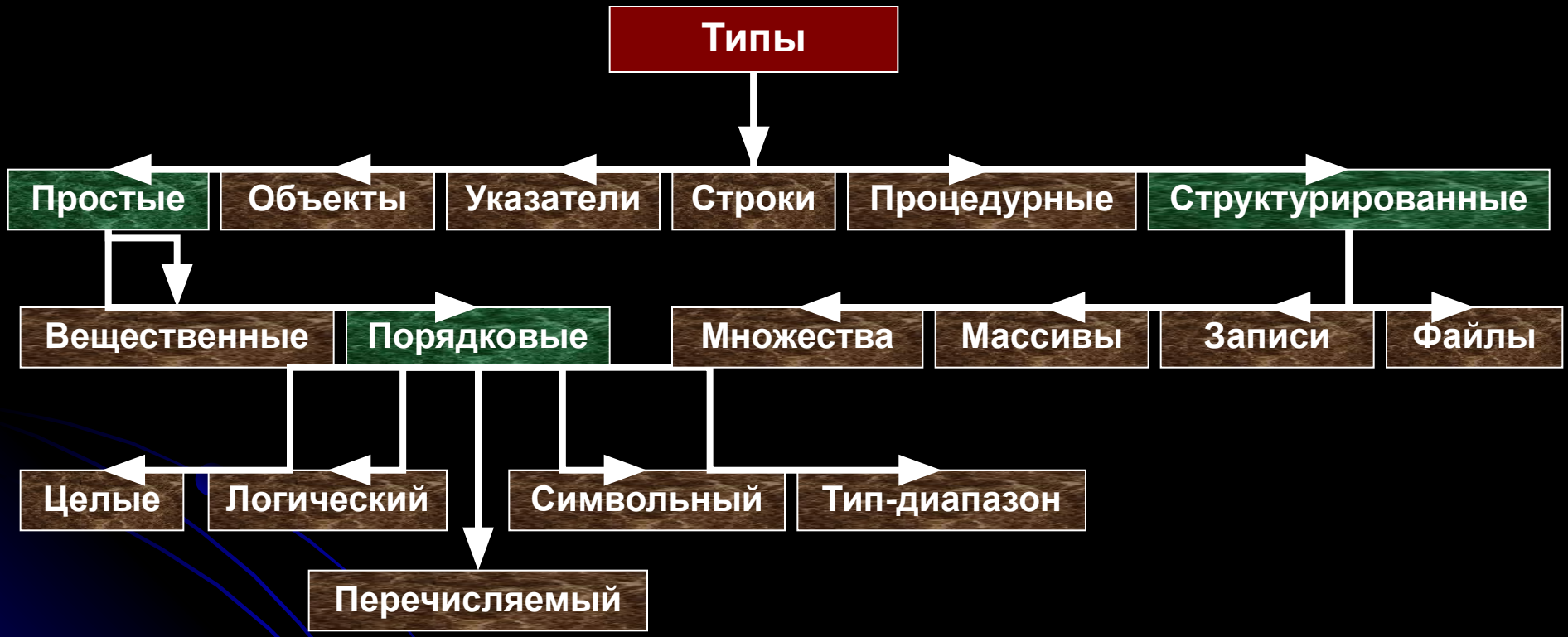
В отличие от стандартного Pascal, в Turbo Pascal разрешается в объявлении констант использовать произвольные выражения, операндами которых могут быть ранее объявленные нетипизированные константы, имена типов и объектов и некоторые функции от них.

Объявление констант-выражений

```
Const
 ln10 = 2.302585092994;
 Ln10R = 1/ln10;
 MaxReal = Maxlnt div SizeOf(real);
```

2.7.2. Элементы языка Turbo Pascal

Структура типов данных



2.7.2. Элементы языка Turbo Pascal

Целые типы

Название	Длина, байт	Диапазон значений
Byte	1	0 ... 255
ShortInt	1	-128 ... +127
Word	2	0 ... 65535
Integer	2	-32768 ... +32767
LongInt	4	-2 147 483 648 ... +2 147 483 647

2.7.2. Элементы языка Turbo Pascal

Логический тип

Название	Значения
Boolean	False, True

Символьный тип

Название	Значения
Char	1 любой символ

2.7.2. Элементы языка Turbo Pascal

Перечисляемый тип

Перечисляемый тип задается перечислением тех значений, которые он может получать. Каждое значение именуется некоторым идентификатором и располагается в списке, обрамленном круглыми скобками.

В Turbo Pascal нельзя использовать кириллицу в идентификаторах, то есть все значения перечисляемого типа не могут содержать русских букв.

Объявление перечисляемого типа

```
Type
 TypeMonth = (jan, feb, mar, apr, may, jun,
        jul, aug, sep, oct, nov, dec);
 Colors = (black, red, white);
 Ordenal= (one, two, three);
```

2.7.2. Элементы языка Turbo Pascal

Тип—диапазон

Тип—диапазон есть подмножество своего базового типа, в качестве которого может выступать любой порядковый тип, кроме типа—диапазона. Тип—диапазон задается границами своих значений внутри базового типа: <минимальное значение>..<максимальное значение>

Объявление типа-диапазона

```
Type
 Digit = '0'.. '9';
 Dig2 = 48..57;
```

2.7.2. Элементы языка Turbo Pascal

Вещественные типы

Название	Размер, байт	Количество значащих цифр	Диапазон десятичного порядка
Real	6	11. . .12	-39. ..+38
Double	8	15. ..16	-324. . .+308
Extended	10	19. . .20	-4951. . .+4932
Comp	8	19. . .20	$-2^{*63}+1$. . .$+2^{*63}-1$

2.7.2. Элементы языка Turbo Pascal

Строковый тип

Тип STRING[N] (строка) в Turbo Pascal широко используется для обработки текстов, где N — максимальное количество символов в строке. Значение N определяется объявлением типа STRING[N] и может быть любой константой порядкового типа, но не больше 255 . Турбо Паскаль разрешает не указывать N, в этом случае длина строки принимается максимально возможной, а именно N=255 .

Объявление строкового типа

```
Type
 ShortString = String[50];
```

2.7.2. Элементы языка Turbo Pascal

Переменные

Переменная — это программный объект, способный принимать некоторое значение. Переменная имеет собственное имя, фиксированный тип и переменное значение.

Тип задает множество значений, которые может принимать переменная данного типа, форму представления ее значений в памяти компьютера и операции над переменной. Тип связывается с именем посредством объявления.

Var <ид. пер. 1>: <ид. типа 1> [; <ид. пер. 2>:<ид. типа 2> [; …]];

Объявление переменных

```
Var
 X, Y : Real;
 Str : String;
 B : Byte;
 I : Integer;
 F : Boolean;
 C, C1 : Char;
```

```
Type
 Colors = (black, red, white);
 Digit = '0'.. '9';
Var
 Col : Colors;
 Dig : Digit;
 STR1, Str2 : String[50];
```

2.7.2. Элементы языка Turbo Pascal

Типизированные константы

Const <идентификатор 1>: <тип 1> = <значение 1>[; <идентификатор 2>: <тип 2> = <значение 2>[; …]];

Типизированными называют константы, значения которых устанавливаются при описании их типа в разделе описания.

Типизированным константам можно присваивать другие значения в ходе выполнения программы, поэтому фактически они представляют собой переменные с начальными значениями.

Объявление типизированных констант

```
Const
 X : Real = 0.5;
 Str : String = 'Привет!';
 B : Byte = 255;
 I : Integer = 1;
 F : Boolean = True;
 C : Char = 'M';
```

```
Type
 Colors = (black, red, white);
 Digit = '0'.. '9';
Const
 Col : Colors = black;
 Dig : Digit = '5';
```

2.7.2. Элементы языка Turbo Pascal

Оператор присваивания

Решение любой задачи представляет собой процесс формирования результатов из заданных правил и исходных данных. Правила формирования результатов задаются на языке Pascal с помощью выражений.

Выражение — это правило (формула) для вычисления нового значения. Выражение строится из операндов, операций и круглых скобок, с помощью которых можно задать требуемый порядок выполнения его операций.

В качестве операндов могут быть константы, переменные и функции.

Форма оператора присваивания:

```
A := B;
```

где A – имя переменной (левая часть оператора присваивания);
B – выражение для вычисления нового значения (правая часть оператора присваивания).

Тип результата выражения должен быть совместим с типом переменной A.

2.7.2. Элементы языка Turbo Pascal

Правила выполнения оператора присваивания

1. Вычисляется выражение правой части оператора присваивания.
2. Результат присваивается переменной левой части оператора: копируется в область оперативной памяти, выделенную переменной левой части оператора присваивания.

Арифметическое выражение:

Выражение на Turbo Pascal:

```
Z:=(X+Y)/(C-0.5)+(X-Y)/(X*A);
```

2.7.2. Элементы языка Turbo Pascal

Операции

Вид операции	Операции
Унарные	`Not, @`
Мультипликативные	`*, /, Div, Mod, And, Shl, Shr`
Аддитивные	`+, –, Or, Xor`
Отношения	`=, <>, <, >, <=, >=, In`

2.7.2. Элементы языка Turbo Pascal

Правила использования операций с операндами различного типа

Операция	Действие	Тип операндов	Тип результата
not	Отрицание	Логический	Логический
not	Отрицание	Любой целый	Тип операнда
@	Адрес	Любой	Указатель
*	Умножение	Любой целый	Наименьший целый
*	Умножение	Любой вещественный	Exended
*	Пересечение множеств	Множественный	Множественный
/	Деление	Любой вещественный	Extended
div	Целочисленное деление	Любой целый	Наименьший целый
mod	Остаток от деления	Любой целый	Наименьший целый
and	Логическое И	Логический	Логический
and	Логическое И	Любой целый	Наименьший целый
shl	Левый сдвиг	Любой целый	Наименьший целый
shr	Правый сдвиг	Любой целый	Наименьший целый

2.7.2. Элементы языка Turbo Pascal

Правила использования операций с операндами различного типа

Операция	Действие	Тип операндов	Тип результата
+	Сложение	Любой целый	Наименьший целый
+	Сложение	Любой вещественный	Extended
+	Объединение множеств	Множественный	Множественный
+	Сцепление строк	Строковый	Строковый
-	Вычитание	Любой целый	Наименьший целый
-	Вычитание	Любой вещественный	Extended
or	Логическое ИЛИ	Логический	Логический
or	Логическое ИЛИ	Любой целый	Наименьший целый
=	Равно	Любой простой или строковый	Логический
<>	Не равно	Любой простой или строковый	Логический
<	Меньше	Логический	Логический
<=	Меньше или равно	Логический	Логический
>	Больше	Логический	Логический
>=	Больше или равно	Логический	Логический

2.7.2. Элементы языка Turbo Pascal

Вычисление выражений

Порядок

1. Полностью вычисляются выражения в скобках, начиная с самых внутренних.
2. Операции одного ранга (однородные) выполняются последовательно слева направо.
3. При неоднородных операциях последовательность выполнения операций следующая:
   а) значения функций;
   б) одноместные операции;
   в) двухместные операции в порядке их приоритета.

Допускается

1. Не писать + в одноместной операции, т.е. можно писать A := X, а не A := +X;
2. Делать пробелы между операндами, знаками операций и скобками;
3. Перенос выражения на следующую строку, не разрывая идентификатора, ключевых слов и чисел;
4. Использовать лишние скобки для удобства чтения выражений.

Не допускается

1. Пропускать знак * (умножения);
2. При переносе выражения на следующую строку повторять знак арифметической операции.

2.7.2. Элементы языка Turbo Pascal

Арифметические функции

Abs(R: Real): Real Возвращает абсолютное значение аргумента.
ArcTan(R: Real): Real Возвращает арктангенс аргумента.
Cos(R: Real): Real Возвращает косинус аргумента.
Exp(R: Real): Real Возвращает экспоненту аргумента.
Frac(R: Real): Real Возвращает дробную часть аргумента.
Int(R: Real): Real Возвращает целую часть аргумента.
Ln(R: Real) : Real Возвращает натуральный логарифм аргумента.
Pi: Real Возвращает значение числа π.
Sin(R: Real): Real Возвращает синус аргумента.
Sqr(R: Real): Real Возвращает аргумент в квадрате.
Sqrt(R: Real): Real Возвращает квадратный корень аргумента.

2.7.2. Элементы языка Turbo Pascal

Функции преобразования

Chr(X: Byte): Char Возвращает символ с заданным порядковым номером X.

Ord(X) : LongInt Возвращает порядковый номер, соответствующий значению X порядкового типа.

Round(R: Real) : LongInt Округляет значение R вещественного типа до ближайшего целого.

Trunc(R: Real): LongInt Усекает значение вещественного типа до значения типа LongInt путем отбрасывания дробной части.

2.7.2. Элементы языка Turbo Pascal

Процедуры и функции порядкового типа

Procedure **Dec** (var X [; DX: LongInt]) Уменьшает значение переменной X на величину DX, а если параметр DX не задан — на 1.

Procedure **Inc** (var X [; DX: LongInt]) Увеличивает значение переменной X на величину DX, а если параметр DX не задан — на 1.

Function **Odd**(X) : Boolean Проверяет, является ли аргумент нечетным числом.

Function **Pred**(X) Возвращает предшествующее значение аргумента. Тип результата совпадает с типом аргумента.

Function **Succ**(X) Возвращает последующее значение аргумента. Тип результата совпадает с типом аргумента.

2.7.2. Элементы языка Turbo Pascal

Процедуры и функции для работы со строками

Procedure **Delete** (var S: String; Index, Count: Integer)
Удаляет Count символов из строки S, начиная с позиции Index.
Procedure **Insert** (SubS: String; var S: String; Index: Integer)
Вставляет подстроку SubS в строку 5, начиная с позиции Index.
Procedure **Str**(X [: width [: Decimals]]; var S: String)
Преобразует численное значение X в его строковое представление S.
Procedure **Val**(S; String; var X; var Code: Integer)
Преобразует строковое значение S в его численное представление X. Параметр Code —содержит признак ошибки преобразования (0 — нет ошибки).
Function **Concat**(S1 [, S2,...,SN]): String
Выполняет конкатенацию последовательности строк.
Function **Copy**(S: String; Index, Count: Integer): String
Возвращает подстроку из строки S, начиная с позиции Index и длиной Count символов.
Function **Length**(S: String): Byte
Возвращает текущую длину строки S.
Function **Pos**(SubS, S: String): Byte
Возвращает позицию, начиная с которой в строке S располагается подстрока SubS (О — S не содержит SubS).

2.7.2. Элементы языка Turbo Pascal

Операторы ввода вывода

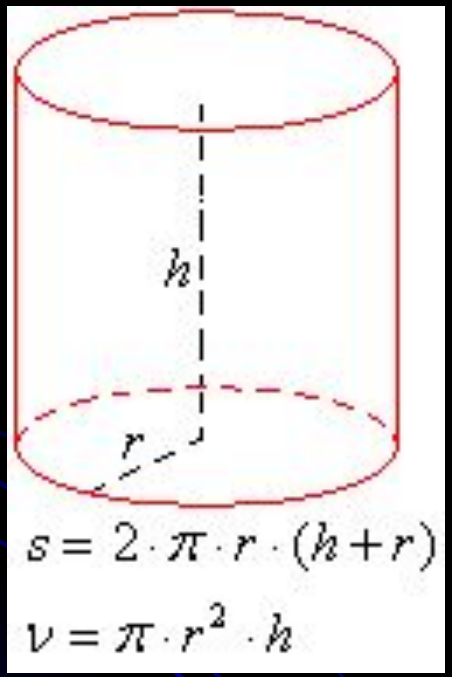
Оператор	Назначение	Пример вызова
Read(*список-ввода*)	Ввод данных (клавиатура)	`Read(a, b, c)`
Readln(*список-ввода*)	Ввод данных, пропуск маркера конца строки	`Readln(a, b, c)` `Readln`
Write(*список-вывода*)	Вывод данных (экран)	`Write('n = ', n:p)` `Write('n = ', n:p:q)`

Writeln(*список-вывода*)	Вывод данных, вывод маркера конца строки	`Writeln('n = ',n:p)`
		`Writeln('n = ',n:p:q)`
		`Writeln`

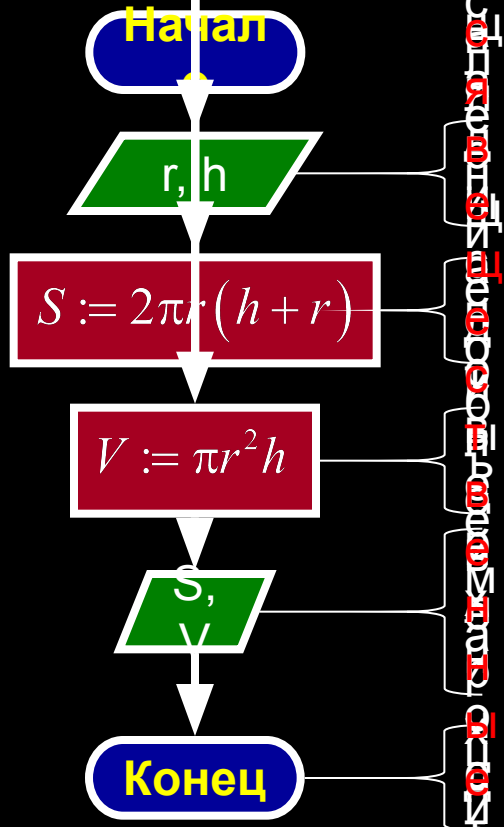
2.7.2. Элементы языка Turbo Pascal

Пример элементарной программы

Задача: Написать программу вычисления площади поверхности и объема цилиндра (высота и радиус цилиндра вводятся с клавиатуры).



Алгоритм



2.7.2. Элементы языка Turbo Pascal

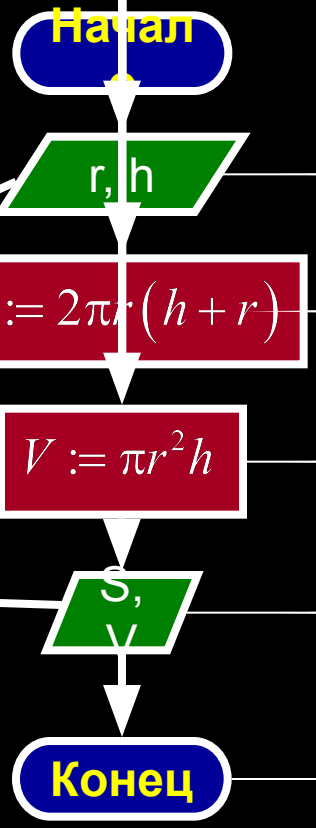
Пример элементарной программы

Задача: Написать программу вычисления площади поверхности и объема цилиндра (высота и радиус цилиндра вводятся с клавиатуры).

Программа

```
Var
  R, H, S, V : Real;
Begin
  Readln(r, h);
  S:=2*PI*r*(h+r);
  V:=PI*SQR(r)*h;
  Writeln(S, V);
End.
```

Алгоритм



2.7.2. Элементы языка Turbo Pascal

Пример элементарной программы

Задача: Написать программу вычисления площади поверхности и объема цилиндра (высота и радиус цилиндра вводятся с клавиатуры).

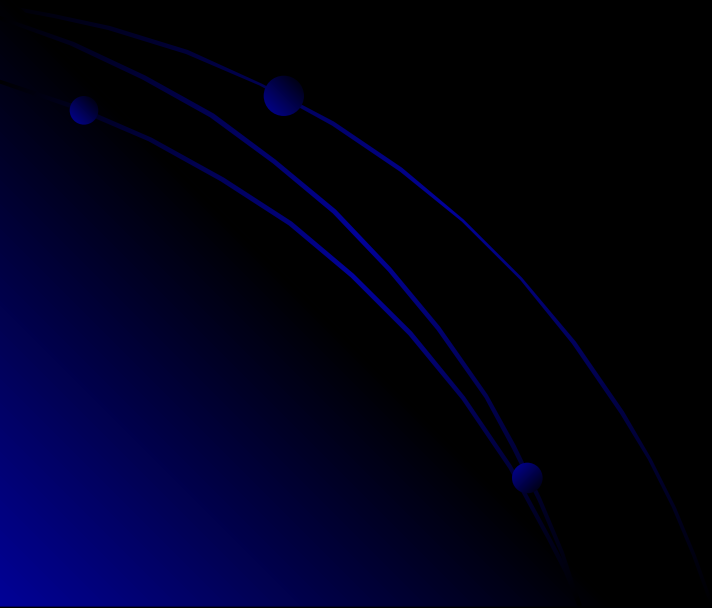
Программа

```
Var
  R, H, S, V : Real;
Begin
  Write('Введите радиус цилиндра r=');
  Readln(r);
  Write('Введите высоту цилиндра h=');
  Readln(h);
  S:=2*PI*r*(h+r);
  V:=PI*SQR(r)*h;
  Writeln('Площадь поверхности цилиндра S=',S);
  Writeln('Объем цилиндра V=',V);
End.
```

2.7.2. Элементы языка Turbo Pascal

Задание для самостоятельной работы

Задача: Написать программу пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах.



2.7.2. Элементы языка Turbo Pascal

Задание для самостоятельной работы

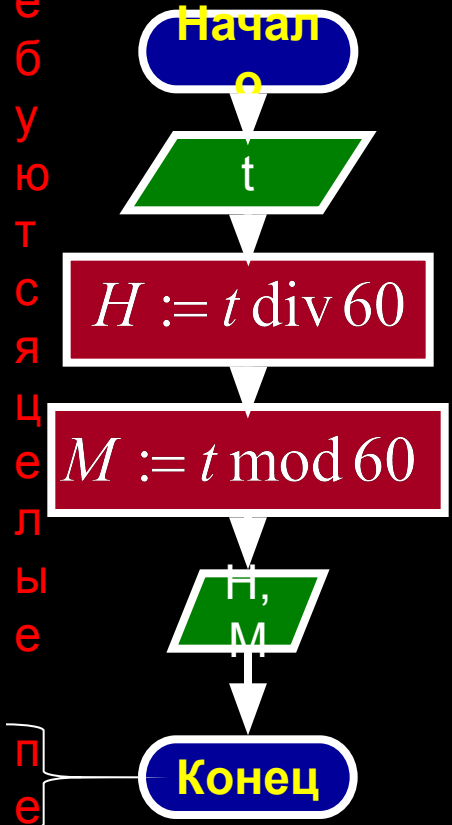
Задача: Написать программу пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах.

Требуются целые переменные

Программа

```
Var
  t, H, M : Integer;
Begin
 Write('Введите время в минутах t=');
 Readln(t);
 H:=t div 60;
 M:=t mod 60;
 Write(t, ' минут это ');
 Writeln(H,' часов ',M,' минут.');
End.
```

Алгоритм



2.7.3. Среда Turbo Pascal

Интерфейс программы

2.7.3. Среда Turbo Pascal

Система меню

File (файл) — действия с файлами и выход из системы.
Edit (редактировать) — восстановление испорченной строки и операции с временным буфером.
Search (искать) — поиск текста, процедуры, функции или места ошибки.
Run (работа) — прогон программы.
Compile (компилировать) — компиляция программы.
Debug (отладка) — отладка программы.
Tools (инструменты) — вызов вспомогательных программ (утилит).
Options (варианты) — установка параметров среды.
Window (окно) — работа с окнами.
Help (помощь) — обращением справочной службе.

2.7.3. Среда Turbo Pascal

Команды, передаваемые среде из редактора

F1 — получить справку.
F2 — записать файл из окна редактора на диск.
F3 — прочитать файл с диска в окно редактора.
F4 — исполнить до курсора (выполнить опцию RUN/GO TO CURSOR).
F5 — распахнуть окно на весь экран или вернуть ему прежние размеры.
F6 — активизировать следующее окно.
F7 — проследить процедуру (выполнить опцию RUN/TRACE INTO).
F8 — пропустить процедуру (выполнить опцию RUN/STEP OVER).
F9 — компилировать программу (выполнить опцию COMPILE/MAKE).
F10 — перейти в главное меню.
Ctrl—F2 — сбросить режим отладки.
Ctrl—F7 — добавить выражение в окно отладки.
Ctrl—F8 — переключить контрольную точку.
Ctrl—F9 — выполнить компиляцию и прогон программы.
Ctrl—Del — очистить буфер редактора.
Ctrl—Ins — копировать блок в буфер редактора.
Alt—X — выйти из Турбо Паскаля.
Alt—F5 — показать окно программы.
Shift—Del — перенести блок из окна редактора в буфер.
Shift—Ins — копировать буфер в окно редактора.

2.7.3. Среда Turbo Pascal

Команды работы с блоками

Ctrl-K B — пометить начало блока.
Ctrl-K K — пометить конец блока.
Ctrl-K T — пометить в качестве блока слово слева от курсора.
Ctrl-K P — напечатать блок.
Ctrl-K C — копировать блок, начиная с позиции курсора.
Ctrl-K K — переместить блок.
Ctrl-K H — убрать выделение блока цветом. Повторное использование **Ctrl-K H** вновь выделит блок.
Ctrl-K Y — удалить блок.
Ctrl-K R — читать блок из дискового файла.
Ctrl-K W — записать блок на диск.
Ctrl-K I — сместить блок вправо.
Ctrl-K U — сместить блок влево.

2.7.3. Среда Turbo Pascal

Команды удаления/вставки

Ctrl-V или **INS** — включить/отключить режим вставки.
Ctrl-N — вставить строку.
Ctrl-Y — удалить строку.
Ctrl-H или **Backspace** — стереть символ слева от курсора.
Ctrl-G или **DEL** — стереть символ над курсором.
Ctrl-T — стереть слово справа от курсора.
Ctrl-Q Y — стереть остаток строки справа от курсора.