

Приложение В. Стандартные процедуры и функции UniPascal-я

Приложение содержит описание стандартных процедур и функций в алфавитном порядке.

function ABS(x)

Предназначение: Определение абсолютного значения выражения.

Результат: Того-же типа что и параметр.

Действие: X - арифметическое выражение целого (integer) или вещественного (real) типа.
Результат - абсолютное значение выражения.

function ADDR(var x): pointer;

Предназначение: Получение адреса переменной.

Результат: Типа pointer.

Действие: x - переменная. Результат - указатель, направленный на x. Как и константа nil, результат функции Addr совместим со всеми указателями.

procedure APPEND(var F [, FileName: string]);

Предназначение: Открытие файла для добавления записей вслед за его последней записью.

Действие: Открытие существующего файла для записи с автоматическим установлением в конце файла. Если файл с таким именем не существует, то он создается. Если не указано имя файла, то файловая переменная должна быть связанной с внешним файлом при помощи процедуры REWRITE или OPEN. Если файловая переменная связана с внешним файлом при помощи процедуры RESET, получается ошибка ввода/вывода № 143 в процессе выполнения программы.

function ARCTAN(x): real;

Предназначение: Вычисление арктангенса.

Действие: x - арифметическое выражение вещественного типа. Результат (в радианах) - арктангенс x .

procedure BLOCKREAD(f; var Buffer; count: word [; var Result: word]);

Предназначение: Считывание с файла определенного числа байтов.

Действие: При выполнении BlockRead считываются count (или меньше) байтов с файла f в память компьютера, начиная с первого байта, занимаемого Buffer-ом. Число действительно считанных байтов становится значением параметра Result (если задан). Если не задан параметр Result и было считано число байтов меньше заданного, выводится сообщение об ошибке ввода/вывода (I/O error). Если введена директива {\$I-}, то программист должен сам позаботиться об обработке ошибки.

procedure BLOCKWRITE(f; var Buffer; count: word [; var Result: word]);

Предназначение: Запись в файл определенного числа байтов.

Действие: При выполнении BlockWrite записываются count (или меньше) байтов с памяти (начиная с первого байта, занимаемого Buffer-ом) в файл f. Число действительно записанных байтов становится значением параметра Result (если задан). Если не задан параметр Result и было записано меньше заданного число байтов, выводится сообщение об ошибке ввода/вывода (I/O error). Если введена директива {\$I-}, то программист должен сам позаботиться об обработке ошибки.

function CHR(x: ShortCard): char;

Предназначение: Генерация символа с заданным ASCII-кодом.

Действие: x - целочисленное арифметическое выражение. Результат - символ с ASCII-кодом = x.

procedure CLOSE(var f [, TRUNCATE]);

Предназначение: Закрытие файла.

Действие: f - переменная файлового типа, связанная с типизированным или нетипизированным файлом. Если файл открыт оператором APPEND, RESET, REWRITE или OPEN, то проводятся все необходимые действия для закрытия файла, после чего прерывается связь между переменной и файлом. Если задан параметр TRUNCATE, то до закрытия файла его длина уменьшается до текущей позиции файла.

function COS(x: real): real;

Предназначение: Вычисление косинуса.

Действие: Вычисление косинуса от заданного в радианах параметра.

procedure DEC(var x [: n]);

Предназначение: Уменьшение значения переменной.

Действие: x - переменная ординального типа; n - необязательное арифметическое выражение. При выполнении процедуры переменной x присваивается значение, равное x - n, если n задано, или x - 1, если n не задано.

procedure DELETE(var s: string; ndx, count: ShortCard);

Предназначение: Удаление подстроки из строки.

Действие: Удаление из строки s count символов, начиная с позиции ndx.

procedure DISPOSE(p {, CaseValue});

Предназначение: Удаление динамической переменной.

Действие: p - переменная ссылочного типа, указывающая на динамическую переменную. При выполнении процедуры указываемая динамическая переменная удаляется и занимаемая ею память становится доступной для дальнейшего применения.

Ограничение: Вариантные константы должны быть те-же самые, что и при занятии памяти процедурой NEW.

function EOF((var f)): boolean;

Предназначение: Проверка на достижение конца файла.

Действие: f - переменная типа файла. Если f не задано, подразумевается стандартный входной файл. Результат TRUE, если позиционирование в файле вслед за его последней записью, и FALSE в противном случае.

function EOLN((var f: text)): boolean;

Предназначение: Проверка на достижение конца строки.

Действие: f - переменная типа текстового файла. Если f не задано, подразумевается стандартный входной файл. Результат TRUE, если позиционирование в файле после последнего символа строки - т.е. на маркер EOLN, или вслед за его последней записью - т.е. EOF(f)=TRUE, и FALSE в противном случае.

procedure EXCL(var s: set; x)

Предназначение: Выключение элемента из множества.

Действие: s - переменная множественного типа, а x - элемент, принадлежащий базовому типу множества. Эта процедура исключает заданного элемента x из множества s , т.е. **EXCL**(s , x) эквивалентно $s := s - [x]$;

procedure EXIT(ProcedureName);

Предназначение: Вызывает выход из процедуры (функции).

Действие: Вызывает принудительно выход из процедуры (функции), имя которой задано параметром. При выполнении этой процедуры, выполняются все завершающие части процедур, из которых надо выйти. Если указанная параметром процедура (функция) не активна, сообщается об ошибке в процессе выполнения процедуры EXIT.

function EXP(x: real): real;

Предназначение: Вычисление экспоненты.

Действие: x - арифметическое выражение вещественного типа. Результат - экспонента x , т.е. $EXP(x) = e^x$.

function FILEPOS(var f): LongInt;

Предназначение: Получение текущей позиции в файле.

Действие: f - файловая переменная. Результат типа LongInt. Если текущая позиция соответствует началу файла, то FilePos равно 0, а если соответствует концу файла ($EOF(f)=true$), то $FilePos(f) = FileSize(f)$.

Ограничения: Файл должен быть нетекстовым и открытым.

function FILESIZE(var f): LongInt;

Предназначение: Получение размера (длины) файла.

Действие: f - файловая переменная. Результат типа LongInt. Результат - число компонент файла.

Ограничения: Файл должен быть нетекстовым и открытым.

procedure FILLCHAR(var Src; count: Natural; ch: char);

Предназначение: Заполнение непрерывной области памяти.

Действие: непрерывная область памяти с начальным адресом Src заполняется символом ch , повторенным $count$ раз.

procedure FILLWORD(var Src; count: Natural; W: word);

Предназначение: Заполнение непрерывной области памяти.

Действие: непрерывная область памяти с начальным адресом Src заполняется словом W , повторенным $count$ раз.

function FRAC(x: real): real;

Предназначение: Получение дробной части вещественного числа.

Действие: возвращает в качестве результата вещественного типа дробную часть вещественного арифметического выражения x , заданного в качестве параметра.

procedure FREEMEMWORDS(p: pointer; WrdSz: Natural);

Предназначение: Освобождение занятой памяти.

Действие: p - переменная ссылочного типа, указывающая на динамическую переменную длиной в $WrdSz$ слов. При выполнении процедуры указываемая динамическая

переменная удаляется и занимаемая ею память становится доступной для дальнейшего применения.

procedure GETDATE(var Year, Month, Day: word);

Предназначение: Получение текущей даты.

Действие: Переменным Year, Month, Day присваиваются значения текущего года, месяца и дня, соответственно. Year может быть от 1980 до 2099, Month - от 1 до 12, а Day - от 0 до 31. Если переменная Day получит значение 0, значит в операционной системе не установлена текущая дата.

procedure GETMEMWORDS(var p:pointer; WrdSz: Natural);

Предназначение: Резервирование памяти.

Действие: p - переменная ссылочного типа. В памяти занимает область размером в WrdSz слов. Начало области возвращается в p. Внимание! Нельзя предполагать, что память, занятая двумя последовательными выполнениями процедуры, образует непрерывное поле памяти.

procedure GETTIME(var Hour, Min, Sec, s100: word);

Предназначение: Получение текущего времени.

Действие: Переменным Hour, Min, Sec и s100 присваиваются значения текущего часа, минуты, секунды и сотой секунды, соответственно. Hour может быть от 0 до 23, Min и Sec - от 0 до 59, а s100 - от 0 до 99.

procedure HALT((ExitCode: Natural));

Предназначение: Прекращение выполнения программы.

Действие: ExitCode - необязательный параметр и, если не указан, подразумевается 0. Выполнение программы прекращается.

function HI(x: word): byte;

Предназначение: Получение старшего байта слова.

Действие: x - арифметическое выражение с размером в 1 слово. Результат типа byte и представляет собою старший байт параметра.

function HIWORD(x: LongWord): word;

Предназначение: Получение старшего слова двойного слова.

Действие: x - арифметическое выражение с размером в 1 двойное слово. Результат типа word и представляет собою старшее слово параметра.

procedure INC(var x [, n]);

Предназначение: Увеличение значения переменной.

Действие: x - переменная ординального типа; n - необязательное арифметическое выражение. При выполнении процедуры переменной x присваивается значение, равное старому, увеличенное на n, если n задано, или на 1, если n не задано.

procedure INCL(var s: set; x)

Предназначение: Включение элемента в множество.

Действие: s - переменная множественного типа, а x - элемент, принадлежащий базовому типу множества. Эта процедура включает заданного элемента x в множество s, т.е. **INCL(s, x)** эквивалентно **s := s + [x]**;

procedure INSERT(const Src:string; var s:string; ndx:ShortCard);

Предназначение: Вставка строки в другую строку.

Действие: Строка Src вставляется в строку s с позиции ndx.

function INT(x: real): real;

Предназначение: Получение целой части вещественного числа.

Действие: Возвращает в качестве результата вещественного типа целую часть вещественного арифметического выражения x, заданного в качестве параметра.

function IORESULT: integer;

Предназначение: Получение кода ошибки при последнем выполнении операции ввода/вывода.

Действие: Результатом функции является код завершения операции ввода/вывода. Код = 0 - операция осуществлена успешно.

Внимание! Для выполнения функции должна быть задана директива {\$I-}.

function LENGTH(const s: string): ShortCard;

Предназначение: Получение длины строки.

Действие: Результатом функции является длина заданной в качестве параметра строки.

function LN(x: real): real;

Предназначение: Вычисление натурального логарифма.

Действие: x - арифметическое выражение вещественного типа. Результат - натуральный логарифм x. $\ln(x) = \log_e X$

function LO(x: word): byte;

Предназначение: Получение младшего байта слова.

Действие: x - арифметическое выражение с размером в 1 слово. Результат типа byte и представляет собою младший байт параметра.

function LONGSWAP(w: LongWord): LongWord;

Предназначение: Обмен значениями байтов двойного слова.

Действие: В двойном слове попарно обмениваются значениями байты по схеме 1 <-> 4, 2 <-> 3.

function LOWORD(x: LongWord): word;

Предназначение: Получение младшего слова двойного слова.

Действие: x - арифметическое выражение с размером в 1 двойное слово. Результат типа word и представляет собою младшее слово параметра.

procedure MARK(var p: pointer);

Предназначение: Пометка состояния динамической памяти.

Действие: При выполнении процедуры в переменной ссылочного типа p помечается состояние динамической памяти (адрес первого свободного байта). Переменная p дальше следует использовать в качестве параметра процедуры RELEASE для удаления всех динамических переменных, созданных после пометки. Переменная p не должна изменяться после вызова процедуры MARK вплоть до выполнения процедуры RELEASE.

function MAX(type): type;

Предназначение: Получение максимального значения ординального типа.

Действие: Результатом функции является максимальное значение из множества значений, принадлежащих данному ординальному типу.

function MAXAVAIL: natural;

Предназначение: Получение объема наибольшего непрерывного блока свободной памяти.

Действие: Результат процедуры - объем в словах наибольшего непрерывного блока свободной памяти.

function MEMAVAIL: natural;

Предназначение: Получение полного объема свободной памяти.

Действие: Результат процедуры - объем в словах всех блоков свободной памяти - т.е. полный объем свободной памяти.

function MIN(type): type;

Предназначение: Получение минимального значения ординального типа.

Действие: Результатом функции является минимальное значение из множества значений, принадлежащих данному ординальному типу.

procedure MOVE(const Src; var Dest; Nbytes: Cardinal);

Предназначение: Перемещение непрерывной области памяти.

Действие: Начиная с адреса Src, Nbytes байтов перемещаются в переменную Dest.

procedure MOVEWORDS(const Src; var Dest; Nwords: Cardinal);

Предназначение: Перемещение непрерывной области памяти.

Действие: Начиная с адреса Src, Nwords слов перемещаются в переменную Dest.

function MSGLIMIT: natural;

Действие: В качестве результата возвращает число символов строк экрана.

procedure NEW(var p {, CaseValue});

Предназначение: Создание новой динамической переменной.

Действие: Создается новая динамическая переменная и ее адрес заносится в переменной ссылочного типа p. Вариантные константы должны следовать в порядке нарастания вложенности вариантных частей.

function ODD(x: LongInt): boolean;

Предназначение: Проверка на нечетность.

Действие: Результат TRUE, если параметр нечетное число.

procedure OPEN(var f [, Fname: string [, TEMPORARY]]);

Предназначение: Открытие файла для ввода/вывода.

Действие: Открытие для ввода/вывода внешнего файла с именем, заданным переменной Fname. Если файл с таким именем не существует, то он создается и открывается. Если не указано имя внешнего файла, файловая переменная f должна быть уже связанной с внешним файлом при помощи другой активизации OPEN. В таком случае первая запись файла становится текущей. Если задан параметр TEMPORARY, открывается файл с уникальным именем. В таком случае Fname должна быть переменной типа STRING (с максимальным спецификатором длины не меньше 80), которая при активизации OPEN должна

содержать путь, где будет создан файл, а после выполнения процедуры fname будет содержать имя созданного файла.

function ORD(x): integer;

Предназначение: Получение очередного номера в множестве значений заданного типа.

Действие: x - параметр ординального типа. Функция возвращает номер x в множестве значений типа, которому принадлежит.

function PARAMCOUNT: natural;

Предназначение: Счет числа параметров, введенных с командной строки.

Действие: Возвращает число параметров, введенных с командной строки операционной системы в момент запуска программы. Если программа вызвана без параметров, результат функции - 0.

procedure PARAMSTR(n: natural; var s: string);

Предназначение: Получение значения параметра, введенного с командной строки.

Действие: Получение в строке S значения N-ого по очереди введенного с командной строки операционной системы параметра. Если N = 0, то получается имя программы.

function POS(const SubStr, Src: string): ShortCard;

Предназначение: Поиск в строке подстроки.

Действие: В строке Src осуществляется поиск строки SubStr и, если она найдена, возвращается номер позиции, с которой она начинается. Если в Src не содержится SubStr, то результатом является 0.

function PRED(x)

Предназначение: Получение предыдущего элемента множества значений заданного типа.

Действие: x - выражение ординального типа. Результат функции того-же типа и является предыдущим по отношению заданного элемента множества значений типа.

function RANDOM[(x: Cardinal)]

Предназначение: Получение псевдо-случайного числа.

Действие: Если не задан параметр, результат - вещественное число в интервале $0 \leq \text{Random} < 1$. Если задан параметр, то результат - целое число в интервале $0 \leq \text{Random}(x) < x$

procedure RANDOMIZE(x);

Предназначение: Инициализация генератора случайных чисел.

Действие: Если эта процедура не была выполненной, программа при каждом вызове процедуры RANDOM будет получать очередное число постоянного ряда случайных чисел. В некоторых случаях это необходимо. Чтобы инициализировать новый ряд псевдослучайных чисел, необходимо выполнить RANDOMIZE без параметров. Если задан параметр, первым числом ряда чисел будет значение параметра.

procedure READ([f,] v1 [, v2, ... vn]);

Предназначение: Ввод с типизированного файла.

Действие: f - файловая переменная. Если она не задана, используется стандартный входной файл. Если f - текстовый файл (или не задано) v1, ..., vn должны быть переменными типа integer, cardinal, longint, char или ихние диапазоны, а так-же

real или string. С файла считываются символы и преобразуются во внутренние формы представления соответствующего типа. Если *f* - типизированный файл, то *v1*, ..., *vn* должны иметь тип компонент этого файла. Для каждого *v* считывается необходимое количество байтов.

procedure READLN([f,] [v1, ..., vn]);

Предназначение: Ввод с файла с последующим пропуском символов до конца строки.

Действие: Процедура применима только для текстового файла. Она эквивалентна по действию процедуре Read с теми самыми параметрами. После ввода последнего заданного параметра, символы до конца строки файла пропускаются и подготавливается ввод следующей строки.

procedure RELEASE(p: pointer);

Предназначение: Восстановление состояния динамической памяти.

Действие: Динамическая память, занятая после вызова процедуры MARK освобождается и соответствующие динамические переменные удаляются.

procedure RESET(f [, Fname: string]);

Предназначение: Открытие файла для ввода.

Действие: Открытие только для ввода внешнего файла с именем, заданным переменной Fname. Попытка вывода в файл приводит к ошибке ввода/вывода. Если не задано имя файла, файловая переменная *f* должна быть уже связанной с внешним файлом при помощи OPEN или RESET. В этом случае предпринимаются необходимые действия для того, чтобы следующий ввод был с начала файла.

procedure RETURN([x]);

Предназначение: Прекращает работу текущей процедуры.

Действие: Прекращается работа текущей процедуры и управление передается вызвавшей ее процедуре. Если текущая процедура имеет завершающую часть (EXIT:), вызов процедуры RETURN эквивалентен оператору GOTO EXIT, если RETURN предшествует метке EXIT. Если RETURN используется внутри функции, в качестве параметра нужно задать результат, который возвращается функцией. Если укажется только (), то результат будет таким, каким он был при предыдущем присваивании значения идентификатору функции.

procedure REWRITE(var f[, Fname: string [, NEW | TEMPORARY]]);

Предназначение: Создание нового внешнего файла.

Действие: Создание нового внешнего файла с именем, заданным переменной Fname. Попытка ввода с файла приводит к ошибке ввода/вывода. Если внешний файл с этим именем уже существует, он уничтожается. Если имя не задано, то файловая переменная *f* должна быть уже связанной с внешним файлом при помощи OPEN, APPEND или REWRITE. В таком случае предпринимаются необходимые действия для того, чтобы следующий вывод был с начала файла. Если необходимо сделать файл пустым, после этого выполнения используйте процедуру TRUNCATE. Если задан параметр NEW и существует старый файл, он не уничтожается и сообщается об ошибке. Если задан параметр TEMPORARY, файл открывается аналогичным способом, применяемым и процедурой OPEN с параметром TEMPORARY.

function ROUND(x: real): LongInt;

Предназначение: Преобразование вещественного числа в целое путем округления.

Действие: Значение арифметического выражения вещественного типа округляется и получается целое число.

procedure SEEK(var f; NewPos: LongInt);

Предназначение: Изменение позиции файла.

Действие: Указатель позиции файла получает значение NewPos. Первый элемент файла имеет номер позиции 0. Если NewPos больше FileSize(f) и файл открыт процедурой RESET (т.е. только для ввода) то возникает ошибка ввода/вывода. Если файл открыт и для вывода, то его длина увеличивается.

Ограничения: Процедура не работает с текстовыми файлами.

function SIN(x: real): real;

Предназначение: Вычисление синуса.

Действие: Вычисление синуса заданного в радианах параметра.

function SIZEOF(x {, CaseValue}): Cardinal;

Предназначение: Определение длины параметра в байтах.

Действие: x может быть переменной или типом. Если тип с вариантными записями (или переменная такого типа), то возможно перечислить вариантные константы таким-же способом, как и в NEW и DISPOSE.

function SQR(x);

Предназначение: Вычисление квадрата значения параметра.

Действие: x - арифметическое выражение целого или вещественного типа. Результат функции того-же типа.

function SQRT(x: real): real;

Предназначение: Вычисление квадратного корня.

Действие: Результат функции - квадратный корень значения параметра.

function SUCC(x)

Предназначение: Получение следующего элемента множества значений заданного типа.

Действие: x - выражение ординального типа. Результат того-же типа и представляет собою следующий за элементом, определенным параметром, в множестве значений заданного типа.

function SWAP(w: word): word;

Предназначение: Обмен значениями младшего и старшего байта одного слова.

Действие: Старший и младший байт одного слова обмениваются значениями.

function TRUNC(x: real): LongInt;

Предназначение: Преобразование вещественного числа в целое число путем удаления дробной части.

Действие: $\text{trunc}(r) = \text{round}(\text{int}(r))$;

procedure TRUNCATE(var f);

Предназначение: Усечение файла.

Действие: Процедура усекает файл до его текущей позиции.

function UPCASE(ch: char): char;

Предназначение: Преобразование буквы в прописную букву.

Действие: Если значением параметра функции является строчная буква латинского алфавита ('a'..'z'), результатом функции будет соответствующая прописная буква. В противном случае, результатом функции будет тот же самый символ, который задан параметром.

function WORDSIZEOF(x {,CaseValue}): Cardinal;

Предназначение: Определение длины параметра в словах.

Действие: x может быть переменной или типом. Если тип с вариантными записями (или переменная этого типа), то возможно перечислить вариантные константы таким-же способом, как и в NEW и DISPOSE.

procedure WRITE([f,] v1 [, v2, ... vn]);

Предназначение: Вывод в типизированный файл.

Действие: f - файловая переменная. Если она не задана, используется стандартный выходной файл.

Если f - типизированный файл, то v1, ..., vn должны быть типа компонент этого файла. При том любое v записывается в файл в его внутреннем представлении.

Если f - текстовой файл, v1, ..., vn - переменные или выражения типа INTEGER, CARDINAL, LONGINT, CHAR или их диапазоны, а так-же REAL, PACKED ARRAY OF CHAR или STRING. Все данные преобразуются с внутренних форм в символы и тогда выводятся. Любой из параметров v1, ..., vn записывается в одной из следующих трех форм: v, v:w, v:w:f. Здесь через v обозначается параметр, через w и f обозначаются арифметические выражения типа INTEGER, при том третья форма допустима только для параметров вещественного типа. Когда параметр w задан, им определяется размер поля (число позиций), в котором должен быть записан параметр. Если размер поля не задан, подразумевается 0, которым обозначается, что можно использовать столько позиций, сколько нужно. Если параметр w задан и его абсолютное значение меньше размера символьного представления параметра v, то его значение игнорируется (т.е. используются столько позиций, сколько необходимо). Если абсолютное значение параметра w больше числа позиций, необходимых для представления v, то: если значение w положительное, символьному представлению v предшествуют столько пробелов, что вместе с символами представления v занимали бы w позиций (т.е. делается выравнивание вправо); если значение w отрицательное, пробелы записываются за символами представления v (т.е. делается выравнивание влево).

LONGINT и все его диапазоны (INTEGER, CARDINAL, ... тоже диапазоны LONGINT) преобразуются в символьном десятичном представлении (без ведущих нулей).

CHAR Отпечатывается символ (возможно и пробелы, если $w < -1$ или $1 < w$).

CHAR ARRAY Одномерный упакованный массив символов (PACKED ARRAY [xx..xx] OF CHAR) отпечатывается так, как была бы отпечатана строка, содержащая все его элементы.

STRING Отпечатываются последовательно все символы строки.

LONGWORD и его диапазоны BYTE и WORD преобразуются в символьном шестнадцатеричном представлении (с ведущими нулями), т.е. число записывается в шестнадцатеричной системе счисления с 8, 2 и 4 символами, соответственно.

REAL При параметре v вещественного типа, w является длиной поля, в котором должно быть отпечатано значение. Если параметр f не задан, вещественное число отпечатывается в экспоненциальном виде: **sd.ddddE±ee**, где s - знак минуса при

отрицательном числе или пробел при положительном; **d** - цифра мантиссы вещественного числа, число цифр мантиссы равно $w-6$; \pm - знак '+' или '-' в зависимости от знака порядка (экспоненты); **ee** - десятичное представление порядка. Если задан параметр *f*, то число отпечатывается с *f* цифрами за десятичной точкой, а в остальных ($w-f$) позициях отпечатываются его целая часть, точка и знак числа. Незначащие ведущие нули перед целой частью и незначащие нули за дробной частью не отпечатываются (за исключением первого 0 дробной части). Если заданное число позиций для дробной части меньше необходимого, число округляется при отпечатывании. Минимальное значение *w* при отпечатывании в экспоненциальном виде - 8 (символьный вид - **sd.dE \pm ee**). Если заданное значение меньше 8, используется 8. Максимальное значение параметра *w* - 15 (символьный вид - **sd.dddE \pm ee**).

Примеры: `writeln(124, ', ', byte(124):4, ', ', word(124): -6);`
отпечатает следующее (символом '<' обозначается пробел):
124, .7с, 007с.

```
writeln('UniPascal': -16, '-', 'v1.60': 14);
UniPascal.....v1.60
```

```
writeln(12.34:12, ', ', 1.234:10:5, ', ', 1.0:8:3, ', ',
1.512:8:0, ', ', 1.455:8:2, ', ');
.1.23400E+01, .1.234, .1.0, .2, .1.46;
```

procedure WRITELN([f,] [v1, ..., vn]);

Предназначение: Вывод в файл с последующим переходом на новую строку.

Действие: Процедура применима только для текстового файла. Она эквивалентна по действию процедуре Write с теми самими параметрами. После вывода последнего заданного параметра осуществляется переход на новую строку.

