

10

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ
СО АН СССР

А.Г.Зоркольецев, Б.Н.Шувалов

СРЕДСТВА ПОДГОТОВКИ ТЕКСТОВ
В СИСТЕМЕ МИСС

ПРЕПРИНТ 81 - 86



I. Введение

Бесшумные и достаточно быстрые алфавитно-цифровые дисплеи позволили перейти от традиционного построчного редактирования к более удобному и естественному процессу — перемещению подсвечиваемого указателя в определенное место файла и последующей замены, вставки или удаления символа или строки. Программная интерпретация функциональной клавиатуры позволяет получить массу дополнительных возможностей: от самых простых, как движение по предварительно размеченному табулятору или размножение и наложение строк, до максимально сложных, заготавливаемых в виде макроопераций редактирования самим пользователем.

В Институте ядерной физики СО АН СССР для подготовки программ и всевозможной текстовой документации широко используется многотерминальная интерактивная система МИСС, реализованная на ЭВМ М-6000 с различной конфигурацией оборудования /1,2/. В качестве терминалов используются алфавитно-цифровые дисплеи Видеотон-340 и Мера-7953. Вторая версия системы обеспечивает одновременную интерактивную работу более 32 пользователей с небольшим временем ответа, не превышающим одной секунды при максимальных загрузках процессора.

Редактор системы МИСС создан на принципе перемещения дисплейного экрана над текстовым файлом в любом из четырех направлений. При движении вверх или вниз экран можно перемещать построчно, по кадрам, по заданному номеру строки или ее контексту или по заранее расставленным меткам. В процессе редактирования можно выделять фрагменты текста в отдельные файлы или вставлять тексты из других файлов.

На размеры редактируемого файла не накладывается никаких ограничений. Максимальная длина строки выбирается пользователем с учетом его конкретных требований и может быть длиннее ее видимой части. Число строк редактируемого файла произвольно и ограничивается только физическим объемом внешней памяти. В процессе редактирования создается два файла, организованных в виде стеков, элементами которых являются строки переменной длины. При движении по файлу вниз отредактированные строки накапливаются в первом стеке, при движении вверх

- во втором. Если в момент движения вниз второй стек непустой, то строки выбираются из него, в противном случае очередная строка выбирается из исходного файла.

Для расширения основного набора редактирующих функций в редакторе реализованы макросредства, позволяющие создавать различные функции прикладного характера. Под макросом здесь понимается процедура, реализующая новую операцию редактирования на базе основного набора с использованием структурированных языковых средств. Механизм макросов позволяет пользователю самому определить набор необходимых ему редактирующих функций.

Исходный текст таблицы макросов подготавливается как обычный файл с использованием всех возможностей редактора. Подготовленный текст затем компилируется, и интерпретатор, встроенный на пути от клавиатуры терминала к редактору, работает с промежуточным кодом. При реализации компилятора использовался метод правил подстановки Флойда /3/, в виде, похожем на описанный в работе /4/.

Для использования макросов в процессе редактирования необходимо предварительно по команде редактора загрузить с диска желаемую таблицу макросов. В оперативной памяти может находиться только одна таблица. Для исполнения конкретного макроса необходимо из основного режима редактора перейти в режим отработки макроса и нажать клавишу, соответствующую имени, указанному в исходном тексте.

С помощью макросов легко реализуются всевозможные операции контекстного редактирования, подготовки и обработки таблиц с текстовой и числовой информацией, различные наборы ключевых слов для разных языков программирования и т.п.

2. Неформальное описание языка

Основным символом языка является код, вырабатываемый клавиатурой дисплея. При этом язык позволяет оперировать как с внутренним (машинным) представлением кода, так и с внешним (литерным), в точности таким, как он изображается на клавиатуре. Тело процедуры представляет собой некоторую последова-

тельность кодов, каждое вхождение которых при исполнении макроса равносильно нажатию на соответствующую клавишу.

К символам языка относятся также идентификаторы, константы, служебные слова, знаки операций и разделители. Формат программы свободный, то есть для наглядности ее можно "разбавлять" пробелами и выделять фрагменты программы круглыми скобками. Признаком комментария является звездочка в первой позиции строки.

2.1. Переменные

В языке макросов различаются простые, двойные, строковые и системные переменные. Все переменные являются глобальными для всех макросов, входящих в одну таблицу. Статическое распределение памяти существенно облегчило реализацию языка. Системные переменные характеризуют текущее состояние подсвечиваемого указателя-курсора (номер строки и позиции на экране, номер строки в файле и ее длину, текущий символ) и определяют координаты меток, используемых в редакторе для выделения фрагментов текста. Они доступны только для чтения и имеют выделенные идентификаторы.

Идентификаторы остальных переменных состоят из одной буквы латинского алфавита и одной цифры. Задание типа переменной неявное, по букве идентификатора. Простые переменные начинаются с букв I, J, K, L, M, N, строковые - с буквы S, а остальные буквы латинского алфавита являются идентификаторами двойных переменных. Простые и двойные переменные могут использоваться для хранения литер и различных численных величин. Двойные отличаются от простых тем, что могут хранить числа удвоенной длины. Они предназначены, в основном, для написания макросов, занимающихся элементарной арифметикой. Строковые переменные используются для хранения последовательностей литер. Они имеют фиксированную максимальную длину, а текущая хранится в простых переменных, начинающихся с буквы L и имеющих соответствующий номер.

2.2. Константы

В языке допускаются как численные константы (целые десятичные и восьмеричные), так и литерные. Численные константы могут сопровождаться знаком. Для отличия знака от операции перед ним ставится предупреждающий символ - восклицательный знак. Признаком восьмеричной константы служит литера В, следующая непосредственно за ней. Литерные константы определяют коды, вырабатываемые клавиатурой дисплея. В тексте макроса признаком литерной константы является знак @, за которым следует наименование клавиши. Клавиши именуются согласно их обозначениям на клавиатуре. Сочетания &A,&B,... обозначают код, вырабатываемый клавишами A,B,... на управляющем регистре. В следующей строке приведены примеры правильного написания констант:

```
5 I27B !-2 !+IOB @C @@ @I @LF @&A
```

2.3. Структура макроса

Любой макрос начинается служебным словом MACRO, за которым следует имя макроса, и заканчивается служебным словом END. В качестве имени макроса может быть любой код, вырабатываемый клавиатурой дисплея. Тело макроса представляет собой некоторую последовательность предложений, вырабатывающих коды, интерпретируемые редактором. Предложения макросов состоят из кодов, выражений и операторов. Например, для генерации одного кода достаточно написать его имя с предупреждающим символом # (не надо путать со знаком @, являющимся признаком литерной константы).

2.4. Выражения

Особенностью выражений является отсутствие в них старшинства операций. Все операции выполняются слева направо и последовательность выполнения их нельзя изменить с помощью

скобок. Операндами в выражении могут быть константы и любые переменные. В строковой переменной можно хранить литерное представление числа. Если же строка содержит литеры, отличные от цифр, то значение переменной считается равным нулю. Все арифметические действия целочисленные, диапазон чисел на программном сумматоре и их представление зависит от типа операнда. Если в выражении не используются двойные переменные, то числа находятся в диапазоне от -32768 до +32767. Если же указанные переменные присутствуют в выражении, то результат может находиться в диапазоне от -21474836.47 до +21474836.47.

При умножении и делении чисел двойной длины от второго множителя или делителя берется только его целая часть. При засылке чисел двойной длины в простую переменную и наоборот выполняется преобразование чисел в соответствующее представление. Засылка числа в строковую переменную преобразует его в последовательность литер.

Со строковыми переменными можно оперировать как со стеками. В качестве указателя стека используется длина строковой переменной. По операции PUSH символ из сумматора добавляется в конец строковой переменной и длина ее увеличивается на единицу. По операции POP последний символ убирается из переменной и длина уменьшается на единицу. Символ при этом заносится на сумматор.

2.5. Операторы

Язык обладает минимальным набором управляющих конструкций, обеспечивающих возможность построения структурированных процедур. Условный оператор имеет традиционную форму:

```
IF <условие> THEN <тело> [ELSE <тело>];
```

Квадратные скобки здесь означают, что конструкция может быть опущена. В качестве тела допускается любая последовательность предложений. Кроме обычного булевского выражения в качестве условия можно использовать операцию сравнения строк,

которая выглядит как

[S] COMPARE S
или [S] COMPARE <строка>

где S – строковая переменная. Если левая часть условия отсутствует, то сравнение выполняется с контекстом редактируемого файла. Цикл имеет несколько расширенную по сравнению с классической форму:

[WHILE <условие>] DO <тело> [TIMES <выражение>];

Тело цикла выполняется до тех пор, пока выполняется условие, либо пока число повторений не превысит значения выражения в конструкции TIMES. Возможны случаи, когда тело цикла не выполнится ни разу. При отсутствии конструкций WHILE и TIMES тело цикла будет выполняться бесконечное число раз. Выход из бесконечного цикла возможен по оператору EXIT или по нажатию на специальную клавишу.

В языке есть операции ввода-вывода, позволяющие присваивать переменным литерные и численные значения, полученные с клавиатуры или из редактируемого текста, записывать значения переменных в файл или выводить на экран в виде комментария. Оператор KEYBOARD присваивает простой переменной значение кода, полученное при нажатии на любую клавишу дисплея. На экране это никак не отражается.

С помощью оператора INPUT переменной можно присвоить литерное или численное значение. Характер действий зависит от положения курсора в момент исполнения оператора. Если курсор находится в служебной строке, то переменная получает значение, введенное с клавиатуры дисплея, при этом ввод контролируется визуально. Если же курсор находится в рабочем поле, то введенной последовательностью считается последовательность символов, расположенных с текущей позиции. Признаком конца числа является любой нецифровой символ. Длина строковой переменной определяется по текущему значению соответствующей простой переменной L.

Оператор OUTPUT позволяет вывести текущее значение прос-

той, двойной или строковой переменной, а также любую строку. Значение простой переменной может интерпретироваться или как десятичное число или как код клавиши. Вывод кода равносильен нажатию на соответствующую клавишу и работает одинаково в основном и управляющем режиме редактора. Вывод числа или строки зависит от текущего положения курсора. При выводе в рабочее поле экрана выводимая последовательность литер попадает и в редактируемый файл, а при выводе в служебную строку интерпретируется как комментарий или как параметры команды, если редактор находится в режиме ввода параметров. Отрицательные числа выводятся со знаком, а положительные – без знака.

По оператору CALL может быть сделано обращение к другим макросам из одной таблицы как к процедурам. После отработки вызываемого макроса управление вернется к вызвавшему макросу на конструкцию, стоящую после оператора вызова. Уровень вложенности процедур ограничивается программным стеком.

По оператору SUSP выполнение макроса приостанавливается, после чего с клавиатуры дисплея можно вводить любую последовательность кодов. Для продолжения исполнения макроса достаточно перейти в соответствующий режим.

Естественным завершением макроса является выход его на конец, признаком которого является служебное слово END. Оператор EXIT служит для завершения исполнения до выхода макроса на конец. В любой момент исполнения макроса можно также завершить, нажав на специальную клавишу.

3. Вместо заключения

Большое влияние на выбор языковых средств подготовки текстов оказал проблемный характер их использования и стремление к минимальным затратам по реализации. Целесообразность создания таких средств подтвердил многолетний опыт эксплуатации системы подготовки текстов.

Л И Т Е Р А Т У Р А

1. Б.Н.Шувалов. МИСС - многотерминальная система с разделением времени для подготовки и редактирования текстов. - Препринт ИЯФ 78-60, Новосибирск, 1978.
2. А.Г.Зоркольец, Б.Н.Шувалов. Многотерминальная интерактивная система для ЭВМ М-6000. - Препринт ИЯФ 81-24, Новосибирск, 1981.
3. В.Н.Лебедев. Введение в системы программирования. - Изд. "Статистика", Москва, 1975.
4. Дж.Донован. Системное программирование. - Изд. "Мир", Москва, 1975.

Приложение I: КРАТКОЕ ОПИСАНИЕ КОМАНД РЕДАКТОРА

Редактор системы МИСС вызывается директивой РЕДАКТИРОВАТЬ с указанием имени редактируемого файла. Так как некоторые операции редактирования существенно зависят от максимальной длины строки текста, поэтому при вызове можно настраивать редактор на определенную длину строки. На конкретный дисплей (Видеотон-340 или Мера-7953) редактор настраивается автоматически.

Редактор может находиться в двух режимах - основном и управляющем, в который можно перейти из основного нажатием на клавишу ETX. После вызова редактор находится в основном режиме. На экране дисплея высвечиваются первые 80 символов первых строк файла и две служебные строки - разметка позиций строки и номер строки, на которой находится указатель.

Интерпретация клавиш в основном режиме

Все алфавитно-цифровые коды интерпретируются естественным образом - при нажатии на любую из этих клавиш на экране и в файле появляется соответствующий символ. Коды управления (специальные клавиши или алфавитные на управляющем регистре) интерпретируются следующим образом:

- | | |
|----|--|
| DL | Удаление текущей строки. Все последующие строки перемещаются вверх. Если удаляемая строка является первой на экране и последней в файле, то на ее месте высвечивается предыдущая строка. |
| IL | Образование новой пустой строки над текущей. При этом текущая и все последующие строки перемещаются вниз. Последняя строка с экрана удаляется. |
| DC | Удаление текущего символа. В последней позиции экрана текущей строки высвечивается следующий символ (если строка не помещается на экране) или пробел. |

- IC Вставка пробела в текущую позицию с раздвиганием строки. При этом последний символ в строке теряется.
- TAB Заполнение строки пробелами от текущего положения до ближайшей справа метки табулятора либо до конца строки экрана (если ближайшая метка находится за пределами экрана). При вызове редактор настраивается на стандартную разметку, которую в процессе редактирования можно менять (смотри управляющий режим редактора).
- F^c Перемещение указателя вправо до ближайшей метки табулятора. Если метка находится за пределами экрана, то выполняется смещение текста по экрану влево.
- Q^c Перемещение указателя влево до ближайшей метки табулятора. Если метка находится за пределами экрана, то выполняется смещение текста по экрану вправо.
- HOME Перемещение указателя в первую позицию первой строки экрана. Если он уже находится в этом положении, то перемещение происходит в первую позицию последней строки экрана.
- RETURN Перемещение указателя в первую позицию экрана текущей строки. Если он уже находится в этом положении, то перемещение происходит в последнюю позицию экрана.
- LINE-FEED Образование новой пустой строки ниже текущей. При этом все строки, расположенные выше указателя, перемещаются вверх. Самая верхняя строка с экрана удаляется. Если же экран неполный, то все строки, расположенные ниже указателя, перемеща-

ются вниз.

- ↓ Перемещение указателя вниз на одну строку. Если текущая строка является последней строкой экрана, то все строки высвечиваемого текста перемещаются вверх и в последней строке экрана появляется новая. Если текущая строка является последней строкой файла, то изображение на экране не меняется.
- ↑ Перемещение указателя вверх на одну строку. Если текущая строка является первой строкой экрана, то все строки высвечиваемого текста перемещаются вниз и в первой строке появляется новая. Последняя строка с экрана удаляется.
- ← Перемещение указателя влево на одну позицию в пределах видимой части текущей строки.
- Перемещение указателя вправо на одну позицию в пределах видимой части текущей строки.
- R^c → Размножение части текущей строки справа от указателя. Новая строка располагается на месте текущей, а текущая строка и все строки, расположенные выше, перемещаются вверх. Самая верхняя строка с экрана удаляется.
- R^c ← Размножение части текущей строки слева от указателя.
- S^c → Наложение части текущей строки справа от указателя с предыдущей строкой.
- S^c ← Наложение части текущей строки слева от указателя с предыдущей строкой.

R^c - R на управляющем регистре

- U^c Поменять местами текущую строку и предыдущую — движение строки вверх. Для строк, находящихся за пределами экрана, эта и последующая команда не работают.
- D^c Поменять местами текущую строку и последующую — движение строки вниз.
- A^c Установить метку А. По данной и следующей команде запоминается текущее состояние указателя (номер строки файла и номер позиции в строке) и высвечивается в последней строке экрана.
- B^c Установить метку В. Метки А и В используются при обработке команды СОЗДАТЬ ФАЙЛ и должны предшествовать ей (смотри управляющий режим).
- G^c Переход в режим выполнения макроса. Любой введенный после этого код интерпретируется как имя макроса. Если указанного макроса нет в текущей таблице или вообще отсутствует таблица макросов, то никаких действий не происходит. Ввод таблицы макросов смотри в управляющем режиме.
- ETX Переход в управляющий режим.

Управляющий режим редактора

Управляющий режим предназначен для ввода и обработки команд, требующих для исполнения более продолжительного интервала времени или дополнительного ввода параметров. В управляющий режим можно попасть из основного нажатием на клавишу ETX. При этом указатель перемещается в последнюю строку экрана вслед за номером строки. После этого любой полученный код интерпретируется как команда редактора. Если команда требует дополнительных параметров, то признаком конца ввода

является нажатие на клавишу LINE-FEED.

- A^c Выход из редактора без сохранения результатов редактирования.
- E^c Нормальное завершение с сохранением результатов редактирования.
- RETURN Возврат в основной режим. Указатель возвращается на то место, в котором была нажата клавиша ETX. Эту команду можно использовать для выяснения номера текущей строки текста, а также для возвращения в основной режим из состояния ввода параметров команд.
- TAB Разметка табулятора. Необходимо ввести через запятые в порядке возрастания желаемые метки табулятора. После ввода указатель возвращается на старое место. При неправильном вводе используется стандартная разметка.
- ↓ Перемещение по тексту на экран вниз. Если последняя строка файла находится на экране, то он остается без изменений.
- ↑ Перемещение по тексту на экран вверх. Если первая строка текста находится на экране, то он остается без изменений.
- Перемещение по тексту вправо до конца строки. Если последняя позиция строки находится на экране, то он не изменяется.
- ← Перемещение по тексту влево до начала строки. Если первая позиция строки находится на экране, то он остается без изменений.
- S^c Поиск строки по номеру. Необходимо ввести номер

строки, и через некоторое время на экране появится новый текст, указатель при этом будет находиться на искомой строке. Если указанный номер больше номера последней строки файла, то высвечиваются последние строки файла.

D^c Поиск строки по контексту вниз. Необходимо ввести контекст (последовательность символов), и при успешном поиске указатель будет установлен на первый символ контекста. Если контекст не будет найден, то высвечиваются последние строки файла.

U^c Поиск строки по контексту вверх. Команда аналогична предыдущей. Если контекст не будет найден, то на экране высвечиваются первые строки файла. Поиск контекста всегда начинается с начала текущей строки.

W^c Создать файл. Для создания файла необходимо предварительно метками А и В указать координаты создаваемого файла (см. основной режим). После высвечивания сообщения необходимо ввести имя создаваемого файла. При всех некорректных действиях выводится соответствующее диагностическое сообщение.

R^c Вставить файл. Файл с указанным именем вставляется после текущей строки, начиная с позиции, на которой находился указатель. После отработки команды редактор переходит в основной режим и на экране высвечиваются последние вставленные строки.

N^c Наложить файл. Файл с указанным именем накладывается на редактируемый файл, начиная с позиции, на которой находился указатель. Если строка выходит за пределы редактируемого файла, то она обрезается.

L^c Поиск метки. Необходимо ввести символ А или В. После успешного поиска указатель будет находиться в месте последней установки соответствующей метки.

T^c Загрузить таблицу макросов. После вывода сообщения ввести имя файла с таблицей макросов.

O^c Высветить список команд основного режима. Выход в основной режим из данной и последующих команд по нажатию клавиши RETURN.

ETX Высветить список команд управляющего режима.

Q^c Высветить список системных макросов.

K^c Высветить описание текущей таблицы макросов.

Приложение 2: СИНТАКСИС ЯЗЫКА МАКРОСОВ

В описании синтаксиса приняты следующие обозначения:

- в угловые скобки заключены нетерминальные символы;
- все правила имеют следующий вид:

$\langle \text{нетерминал} \rangle ::= \text{строка}$

где "строка" - непустая последовательность терминальных и нетерминальных символов;

- для сокращенной записи множества правил с одинаковой левой частью используется вертикальная черта, обозначающая альтернативу;
- в квадратные скобки заключены конструкции, которые могут быть опущены;
- круглые скобки также являются метасимволами и используются для сокращенной записи правил, имеющих в правой части одинаковые последовательности символов;
- для получения из сокращенной формы записи всего множества правил необходимо раскрыть все альтернативы, при этом с круглыми скобками поступают так, как в обычном арифметическом выражении.

$\langle \text{макрос} \rangle ::= \text{MACRO} \langle \text{имя макроса} \rangle ; \langle \text{тело макроса} \rangle \text{ END}$
 $\langle \text{имя макроса} \rangle ::= \# \langle \text{код} \rangle$
 $\langle \text{код} \rangle ::= \langle \text{управляющий} \rangle \mid \langle \text{литера} \rangle \mid \langle \text{литера} \rangle$
 $\langle \text{управляющий} \rangle ::= \text{DL} \mid \text{IL} \mid \text{DC} \mid \text{IC} \mid \text{TAB} \mid \text{RETURN} \mid \text{ETX} \mid$
 $\text{HOME} \mid \text{LF} \mid \text{CU} \mid \text{CD} \mid \text{CR} \mid \text{CL} \mid \text{ERASE}$
 $\langle \text{тело макроса} \rangle ::= \langle \text{элемент макроса} \rangle \mid$
 $\langle \text{тело макроса} \rangle \langle \text{элемент макроса} \rangle$
 $\langle \text{элемент макроса} \rangle ::= \# \langle \text{код} \rangle \mid \langle \text{выражение} \rangle \mid \langle \text{оператор} \rangle$
 $\langle \text{выражение} \rangle ::= \langle \text{операнд} \rangle \mid \langle \text{выражение} \rangle \langle \text{операция} \rangle \langle \text{операнд} \rangle$
 $\langle \text{операнд} \rangle ::= \langle \text{константа} \rangle \mid \langle \text{переменная} \rangle$
 $\langle \text{константа} \rangle ::= \langle \text{численная} \rangle \mid \langle \text{литерная} \rangle$
 $\langle \text{численная} \rangle ::= [! \langle \text{знак} \rangle] \langle \text{целое} \rangle [B]$
 $\langle \text{знак} \rangle ::= + \mid -$
 $\langle \text{литерная} \rangle ::= @ \langle \text{код} \rangle$
 $\langle \text{переменная} \rangle ::= \langle \text{простая переменная} \rangle \mid \langle \text{системная переменная} \rangle \mid$

$\langle \text{двойная переменная} \rangle \mid \langle \text{строковая переменная} \rangle$
 $\langle \text{простая переменная} \rangle ::= \langle \text{идентификатор простой} \rangle \langle \text{цифра} \rangle$
 $\langle \text{идентификатор простой} \rangle ::= I \mid J \mid K \mid L \mid M \mid N$
 $\langle \text{двойная переменная} \rangle ::= \langle \text{идентификатор двойной} \rangle \langle \text{цифра} \rangle$
 $\langle \text{идентификатор двойной} \rangle ::= A \mid B \mid C \mid D \mid E \mid F \mid G \mid H \mid O \mid P \mid$
 $Q \mid R \mid T \mid U \mid V \mid W \mid X \mid Y \mid Z$
 $\langle \text{строковая переменная} \rangle ::= S \langle \text{цифра} \rangle$
 $\langle \text{системная переменная} \rangle ::= \text{CC} \mid \text{SF} \mid \text{PS} \mid \text{SS} \mid \text{LSI}$
 $\text{XA} \mid \text{YA} \mid \text{XB} \mid \text{YB}$
 $\langle \text{операция} \rangle ::= + \mid - \mid * \mid / \mid \backslash \mid = \mid$
 $\text{IOR} \mid \text{AND} \mid \text{XOR} \mid \text{PUSH} \mid \text{POP}$
 $\langle \text{оператор} \rangle ::= \langle \text{условный} \rangle \mid \langle \text{цикл} \rangle \mid \langle \text{ввод} \rangle \mid \langle \text{клавиатура} \rangle \mid$
 $\langle \text{вывод} \rangle \mid \langle \text{вызов} \rangle \mid \langle \text{прервать} \rangle \mid \langle \text{завершить} \rangle$
 $\langle \text{завершить} \rangle ::= \text{EXIT}$
 $\langle \text{прервать} \rangle ::= \text{SUSP}$
 $\langle \text{вызов} \rangle ::= \text{CALL} \langle \text{имя макроса} \rangle$
 $\langle \text{клавиатура} \rangle ::= \text{KEYBOARD} \langle \text{простая переменная} \rangle$
 $\langle \text{ввод} \rangle ::= \text{INPUT} (\langle \text{пусто} \rangle \mid : \mid \langle \rangle \langle \text{простая переменная} \rangle \mid$
 $\langle \text{двойная переменная} \rangle \mid$
 $\langle \text{строковая переменная} \rangle)$
 $\langle \text{вывод} \rangle ::= \text{OUTPUT} (\langle \text{пусто} \rangle \mid : \mid \langle \rangle \langle \text{простая переменная} \rangle \mid$
 $\langle \text{двойная переменная} \rangle \mid$
 $\langle \text{строковая переменная} \rangle \mid$
 $" \langle \text{строка} \rangle ")$
 $\langle \text{строка} \rangle ::= \langle \text{литера} \rangle \mid \langle \text{строка} \rangle \langle \text{литера} \rangle$
 $\langle \text{условный} \rangle ::= \text{IF} \langle \text{условие} \rangle \text{ THEN} \langle \text{тело} \rangle [\text{ELSE} \langle \text{тело} \rangle] ;$
 $\langle \text{тело} \rangle ::= \langle \text{тело макроса} \rangle$
 $\langle \text{условие} \rangle ::= \langle \text{составное условие} \rangle \mid$
 $[\langle \text{строковая переменная} \rangle] \text{ COMPARE}$
 $(\langle \text{строковая переменная} \rangle \mid " \langle \text{строка} \rangle ")$
 $\langle \text{составное условие} \rangle ::= \langle \text{простое условие} \rangle \mid$
 $\langle \text{составное условие} \rangle \text{ LOR} \langle \text{простое условие} \rangle \mid$
 $\langle \text{составное условие} \rangle \text{ LAND} \langle \text{простое условие} \rangle$
 $\langle \text{простое условие} \rangle ::= \langle \text{выражение} \rangle [\langle \text{отношение} \rangle \langle \text{выражение} \rangle]$
 $\langle \text{отношение} \rangle ::= = \mid / = \mid > \mid > = \mid < \mid < =$
 $\langle \text{цикл} \rangle ::= [\text{WHILE} \langle \text{условие} \rangle] \text{ DO} \langle \text{тело} \rangle [\text{TIMES} \langle \text{выражение} \rangle] ;$