

Е 80 067

85-4

42034<sub>2</sub>

МИНИСТЕРСТВО РАДИОПРОМЫШЛЕННОСТИ СССР

АКАДЕМИЯ НАУК СССР

ОРДЕНА ЛЕНИНА СИБИРСКОЕ ОТДЕЛЕНИЕ

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

УТВЕРЖДЕН

3533847.00058-01 35 01-ЛУ

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ

АВТОМАТИЗАЦИИ ШКОЛЬНОГО УЧЕБНОГО ПРОЦЕССА

(ШКОЛЬНИЦА)

РАПИРА

ОПИСАНИЕ ЯЗЫКА

3533847.00058-01 35 01

1985

3533847.00058-01 35 01

АННОТАЦИЯ

В документе описывается язык программирования РАПИРА, являющийся одним из двух входных языков учебной системы программирования, входящей в состав интегрированной программной среды "Школьник".

Производится полное описание синтаксиса языка и краткие сведения о семантике его основных конструкций.

Документ предназначен для всех пользователей языка.

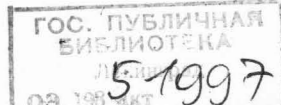
85-4  
42034 а

3533847.00058-01 35 01.

СОДЕРЖАНИЕ

- 1. Общие сведения .....
- 2. Лексика и общая структура программы .....
- 3. Имена и выражения .....
- 4. Присваивание, ввод и вывод .....
- 5. Управляющие конструкции Рапиры .....
- 6. Процедуры и функции .....
- 7. Файлы и модули .....
- 8. Некоторые отладочные средства языка Рапира .....

85-80064



## I. ОБЩИЕ СВЕДЕНИЯ

I.1. Рапира является диалоговым языком высокого уровня, предназначенным для обучения программированию и для решения на ЭВМ небольших прикладных задач. Язык ориентирован в первую очередь на школьников среднего и старшего возраста (II-I7 лет), но может использоваться и другими категориями пользователей.

I.2. Рапира, наряду с Робином, является входным языком системы программирования "Школьник". Как и для языка Робин, операции надпрограммного уровня в Рапире выполняются системными средствами, не относящимися к языку.

I.3. Первая версия Рапиры была разработана в 1978-1979 годах и использовалась в учебной работе со школьниками и студентами. В ходе разработки системы "Школьник" язык был полностью переработан.

I.4. Рапира представляет собой концентрическое семейство языков с последовательно расширяемым набором возможностей. Каждый концентр можно рассматривать как замкнутый язык-оболочку.

В этом документе приводится краткое описание основных особенностей внешнего концентрa канонической (школьной) версии Рапиры.

I.5. В качестве базового этнического языка для Рапиры выбран русский язык. В процессе реализации языка на инструментальной ЭВМ была разработана также английская версия. Ведутся работы над эстонской и грузинской версиями. Возможно создание версий на основе других языков.

## 2. ЛЕКСИКА И ОБЩАЯ СТРУКТУРА ПРОГРАММЫ

2.1. Лексическая структура Рапиры почти не отличается от лексики Робика, описанной в документе 3533847.00058-01 35 02. В Рапире различаются те же лексические единицы: целые и дробные числа, тексты, слова, специальные символы. Форма этих лексем традиционна. Основные отличия лексем Рапиры от Робика заключаются в следующем:

1. Допускается запись дробных чисел в экспоненциальной форме, например  $156.65E12$  означает  $156,65 \times 10^{12}$ .

2. Унарные + и - рассматриваются не как составные части лексем, а как операции.

3. В Рапире используется в несколько раз больше специальных символов, в том числе - составных, чем в Робики.

2.2. Общая структура программы и порядок исполнения отдельных предписаний на Робики и Рапире имеют много общего. Как видно на диаграмме I, программа на Рапире - это последовательность директив, процедурных блоков и описаний имен. Порядок работы с директивами и процедурными блоками такой же, как в Робики.

2.3. В отличие от Робики, допускается произвольное расположение предписаний по строкам (можно записывать несколько предписаний на одной строке и распределять одно предписание на несколько строк. Пробелы и переводы строк могут использоваться в любых местах программы, но не внутри лексем (внутри текстов можно использовать синтаксически значимые пробелы). Между двумя лексемами, не различаемыми контекстно, обязателен по крайней мере один пробел или перевод строки.

Комментарий в Рапире заключается в специальные скобки (диаграмма 2). Комментарий синтаксически эквивалентен пробелу и может быть вставлен между любыми двумя лексемами.

### 3. ИМЕНА И ВЫРАЖЕНИЯ

3.1. Рапира является частично-типизированным языком. Каждое имя, используемое в программе или в процедурном блоке, может быть описано с указанием типа. Тип имени определяет виды значений, которые могут быть присвоены этому имени. Если имя не описано или описано без указания типа, оно по умолчанию относится к универсальному типу. Такому имени можно присваивать значения любых видов. Синтаксис описания имен определяется диаграммой 3.

В диалоговой версии Рапиры описания имен используются в качестве отладочного средства. Предполагается, что в пакетной версии описания должны быть обязательными.

Область действия описаний охватывает главную программу, тот процедурный блок, в котором имя описано, и все вызываемые из него процедуры и функции, в которых не описаны совпадающие с ним по написанию локальные имена и формальные параметры. Таким образом, описания одновременно определяют локализацию имен.

Описания имен и указания типов в Рапире не обязательны, поэтому для оперативного решения небольших задач ее можно использовать в качестве бестипового языка.

3.2. В Рапире различаются следующие виды значений имен:

- целые числа;
- дробные числа;
- тексты;
- множества;
- кортежи;
- записи;
- процедуры;

- функции;
- файлы;
- модули;
- рисунки;
- кадры;
- таблицы;
- пустое значение;

3.3. Пустое значение имеют все имена, отличные от стандартных, которым в данной программе не было присвоено значений. В системе предусмотрено защищенное стандартное имя ПУСТО, которое всегда имеет пустое значение.

3.4. Над любыми парами имен в Рапире определена операция сравнения ТИПА, результат которой - ИСТИНА, если объявленные типы имен совпадают и ЛОЖЬ в противном случае.

Над любыми парами значений (выражений) определена операция сравнения ВИДА, результат которой - ИСТИНА, если виды значений этих выражений совпадают.

3.5. Над числами в Рапире определены следующие операции:

- унарные арифметические: + и -;
- бинарные арифметические: +, -, \*, /, // (деление нацело) - \*\* (возведение в степень);
- операции сравнения: =, /=, <=, >=, <, >;

Все операции имеют обычный математический смысл; допускается сравнение целых и дробных чисел на равенство: условие  $35=35.0$  считается истинным.

3.6. Над текстами определены следующие операции:

- + - конкатенация (слияние текстов), например, "вело"+"сипед" = "велосипед";

ИЗ - проверка принадлежности символа тексту, например, условия "м" ИЗ "микроскоп" истинно;

[ ] - выборка (взятие символа из текста по индексу), например "велосипед" [ 5 ] ="с";

[ : ] - вырезка (выделение участка текста по начальному и конечному индексам), например "крокодил" [ 2:4 ] ="рок";

# - мощность (унарная операция, определяющая число символов в тексте), например, ## "самолет"=7.

./= - сравнение текстов.

В операциях вырезки и выборки в качестве индексов могут использоваться любые выражения, значениями которых являются натуральные числа, не превосходящие мощности соответствующего текста. Левый индекс в вырезке не должен превосходить правого.

Эти две операции могут встречаться как слева, так и справа от знака присваивания, позволяя в последнем случае редактировать текст. Длина текста при редактировании не может изменяться: подставляемый текст должен иметь такую же длину, как заменяемый участок. Например, после присваиваний:

"крокодил" -> Зверь; "ол" -> Зверь [2:3];

имя Зверь получит значение "кошкочил". Присваивания

"кош" -> Зверь [2:3]; "а" -> Зверь [2:3];

являются недопустимыми из-за несовпадения длин заменяющего и заменяемого участков текста.

3.7. Составные структуры данных в диалоговой версии Рапиры представлены множествами, записями и кортежами. Каждая из этих структур может содержать произвольное число элементов. Элементом структуры может быть любой объект языка, независимо от вида.

В пакетной версии допускаются типизированные варианты этих

конструкций: все элементы множества и кортежей должны иметь один и тот же тип, а в записях каждый элемент должен иметь определенный тип.

Множество рассматривается как неупорядоченная совокупность попарно различных элементов. Доступ к отдельному элементу множества невозможен. Разрешены только проверка принадлежности элемента множеству, операции над всем множеством и последовательный перебор его элементов при помощи специальной формы цикла.

Запись рассматривается как неупорядоченная совокупность именованных полей. Доступ к элементу записи возможен только через название поля.

Кортеж рассматривается как упорядоченная совокупность произвольных элементов. Доступ к элементу кортежа возможен по индексу.

3.8. Для каждой из перечисленных структур определена соответствующая форма операции формирования, синтаксис которой описан диаграммой 4. Эта операция позволяет построить новую структуру из произвольного количества элементов, представленных значениями выражений. Как видно из диаграммы, для формирования нужной структуры необходимо перечислить через запятую все выражения, значения которых должны стать элементами структуры, и заключить их в специальные скобки, форма которых определяется видом структуры. Простые уголки скобки < > определяют формирование кортежа, фигурные скобки { } или < ж ж > - формирование множества, специальные скобки < ≍ ≍ > - формирование записи. Кроме того, при формировании записи перед каждым элементом должно быть указано название поля, отделяемое от элемента двоеточием.

Например, выражение < 2, "поиск", 5+3, А > определяет формирование кортежа из четырех элементов: числа 2, текста "поиск", чис-

ла 8 и значения имени А. Выражение  $\langle * 2, "слово", 2, 2 * \rangle$  определяет формирование множества из двух элементов: числа 2 и текста "слово".  
 Выражение  $\langle \{ \text{Имя: "Сергей", Фамилия: "Иванов", Год_рождения: 1960} \} \rangle$  определяет формирование записи из трех полей: Имя, Фамилия и Год\_рождения.

3.9. Для всех структур определены следующие операции:

ИЗ - проверка принадлежности;

# - определение мощности (числа элементов);

= и /= - сравнение на равенство и неравенство;

Необходимо отметить, что две записи с различной структурой полей считаются принадлежащими к различным видам.

Два множества считаются равными, если они состоят из одних и тех же элементов, независимо от порядка их записи, например

$$\langle * 1, 5, 5, \langle 1, 2 \rangle, 1 * \rangle = \langle * 5, \langle 1, 2 \rangle, 1 * \rangle$$

Два кортежа считаются равными, если они состоят из одних и тех же элементов, расположенных в том же порядке, например:

$$\langle 1, 5, \langle 1, 2 \rangle \rangle = \langle 1, 2+3, \langle 1, 1+1 \rangle \rangle$$

$$\langle 1, 2, 3 \rangle \neq \langle 3, 2, 1 \rangle$$

Две записи с одинаковой структурой полей считаются равными, если одноименные поля имеют одинаковые значения, например:

$$\langle \{ \text{Имя: "Сергей", Фамилия: "Иванов"} \} \rangle =$$

$$\langle \{ \text{Фамилия: "Иванов", Имя: "Сергей"} \} \rangle$$

$$\langle \{ \text{Номер: 5, Скорость: 10} \} \rangle \neq \langle \{ \text{Тип: 5, высота: 10} \} \rangle$$

3.10. Кроме операций, перечисленных в пп. 3.4, 3.8 и 3.9, над множествами определены операции объединения (+), перечисления (#) и разности (-), имеющие обычный математический смысл.

Аналогично, над кортежами определены такие же операции, как и над текстами, при этом для вложенных кортежей допускается многократная индексация; выражения  $A [2] [3]$  и  $A [2, 3]$  являются семантически

ки эквивалентными и определяют третий элемент кортежа, являющегося вторым элементом кортежа А.

Над записями, кроме общих операций, определена только выборка элемента по названию поля. Например, выражение X. Адрес определяет значение поля Адрес в записи, являющейся значением имени X.

Как и для кортежей, выборка может использоваться с обеих сторон от знака присваивания. Для вложенных записей определена многократная выборка, например: X.Адрес.Квартира .

3.11. Над процедурами и функциями как над значениями имен кроме общих операций, перечисленных в п.5.3.4, определена операция вызова, синтаксис которой описывается диаграммой 5.

3.12. Для других видов значений операции в Рапире не определены. Действия над такими значениями осуществляются при помощи специальных предписаний или процедур.

Сведения о работе с файлами и модулями в Рапире приведены в разделе 7. Работа с рисунками и кадрами осуществляется средствами графической системы "Шага", входящей в состав системы "Школьница". Таблицы как вид значений предусмотрены только для экстраверсивной версии Рапиры и здесь рассматриваться не будут. Семантика пустого значения такая же, как в языке Робик.

3.13. В Рапире, как и в Робике, предусмотрена защита имен. Частичная и полная защита может быть установлена и снята пользователем при помощи соответствующих разновидностей предписаний ВКЛЮЧИТЬ и ВЫКЛЮЧИТЬ, синтаксис которых описан диаграммой 6. Некоторые имена, например, имена процедур и функций во время их исполнения и имя переменной цикла внутри тела этого цикла, имеют абсолютную защиту, которая не может быть снята средствами, доступными пользователю.

3.14. Синтаксис выражения в Рапире описывается диаграммой 7. Как следует из диаграммы, выражение - это последовательность соединенных знаками бинарных операций констант, формирователей, простых имен и выражений в скобках, перед каждым из которых может стоять один или несколько знаков унарных операций, а после - произвольная последовательность операций выборки, вырезки и вызова.

Таким образом, структура выражения в целом традиционна для языков высокого уровня.

3.15. В Рапире приняты следующие приоритеты операций:

1. Выражения в скобках и составляющие выражения в формирователях;

2. Формирование структур;

3. Вычисление фактических параметров функций и индексных выражений;

4. Выборки, вырезки, вызовы функций;

5. Унарные операции;

6. Возведение в степень;

7. Операции  $\ast$ ,  $/$ ,  $//$  (умножение, пересечение, деление, деление нацело);

8. Бинарные  $+$  и  $-$  (сложение, конкатенация, объединение, вычитание, разность множеств).

Описанный порядок в целом соответствует принятому в математике, за тем исключением, что пересечение множеств ( $\ast$ ) получило в Рапире более высокий приоритет, чем объединение ( $+$ ).

3.16. Синтаксис условия в Рапире описывается диаграммой 8. Термин "условие" соответствует термину "логическое выражение", используемому в других языках. В Рапире логические значения не могут присваиваться именам, поэтому логические выражения допускаются

только в определенных контекстах, например, в цикле ПОКА и в предписаниях ветвления.

Такое выделение логических выражений введено в язык из методических соображений, чтобы подчеркнуть их семантические отличия от выражений других типов.

Операции сравнения и логические связки, используемые в условиях, имеют традиционную семантику.

Приоритеты операций в условиях:

1. Условия в скобках и выражения, входящие в состав условий;

2. Операции сравнения и проверки принадлежности;

3. НЕ;

4. И;

5. ИЛИ;

#### 4. ПРИСВАИВАНИЕ, ВВОД И ВЫВОД

4.1. Синтаксис предписаний присваивания, ввода и вывода в языке Рапира описывается диаграммами 9-II. Семантика этих предписаний почти не отличается от семантики аналогичных предписаний в языке Робик. Поэтому здесь перечисляются только основные различия между указанными предписаниями в этих языках.

4.2. При сравнении синтаксиса предписаний необходимо учитывать, что несмотря на внешнее сходство структура выражений в Рапире гораздо богаче, чем в Робике, поэтому предписания присваивания и ввода в Рапире позволяют выполнять разнообразные операции над данными различных видов, что в Робике может быть сделано только с применением специализированных исполнителей.

4.3. Предписание вывода в Рапире позволяет выводить не только простые (числа, тексты), но и составные значения (множества, кортежи, записи). Форма вывода совпадает с видом соответствующих формирователей в языке. Аналогично, в режиме ввода данных могут быть введены составные значения.

При выводе текстов и при их вводе в режиме ВВОД ТЕКСТА окаймляющие кавычки не указываются. Тексты, выводимые в составе структур и вводимые в режиме ВВОД ДАННЫХ, окаймляются кавычками.

Если режим ввода не указан, по умолчанию предполагается ВВОД ТЕКСТА.

4.4. Вывод любых значений может производиться с указанием формата: После выводимого выражения через двоеточие указывается число отведенных позиций. Для дробных чисел возможно задание двойного формата: указывается еще и число знаков в дробной части числа. Если значение не умещается в отведенном числе позиций, то оно выводится в свободном формате.

4.5. В соответствии с концентрической структурой Рапиры переключение ввода и вывода на другие устройства может быть выполнено различными способами. Наряду с разовым переключением, осуществляемым так же, как в Робике, при помощи указания нужного режима в заголовке предписания (например: ВЫВОД НА БУМАГУ: А,В), в Рапире возможно переключение всей выводимой или вводимой информации на одно или несколько устройств или файлов. Для этого применяется соответствующая форма предписаний ВКЛЮЧИТЬ и ВЫКЛЮЧИТЬ (диаграмма 6). Например: ВКЛЮЧИТЬ ВЫВОД НА БУМАГУ;



5. УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ РАПИРЫ

5.1. К числу управляющих конструкций Рапиры относятся три разновидности ветвления: ЕСЛИ, ВЫБОР и ВЫБОР-ИЗ, четыре разновидности цикла: ПОКА, ПОВТОР, ДЛЯ-ИЗ и ДЛЯ-ОТ, средства описания и вызова функций и процедур и предписание ВЫХОД для выхода из процедур.

В этом параграфе рассматриваются предписания ветвления и цикла, работа с процедурами и функциями обсуждается в следующем параграфе.

Синтаксис перечисленных конструкций описывается диаграммами 5.12-15. Как видно на диаграммах, все эти предписания имеют рамочную структуру, что позволяет указывать в теле цикла и в альтернативах ветвления произвольное число предписаний без применения искусственных средств для их объединения.

5.2. Семантика простого ветвления (условного предписания) ЕСЛИ-ТО-ИНАЧЕ-ВСЕ вполне традиционна.

Пример:

ЕСЛИ     A > B     ТО  
5 - > A; ВЫВОД: B  
ИНАЧЕ  
5 - > B; ВЫВОД: A  
ВСЕ;

5.3. Как видно на диаграмме 12, предписание множественного ветвления (ВЫБОР ИЗ) состоит из произвольного числа альтернатив, каждая из которых представлена условием, за которым следует одно или несколько предписаний, разделяемых точкой с запятой. Условие отделяется от предписаний двоеточием, альтернативы разделяются знаком !. В конце может стоять конструкция ИНАЧЕ.

Предписание исполняется так. Последовательно, в порядке записи

альтернатив, проверяются указанные в них условия. Если одно из них оказалось истинным, выполняются предписания от соответствующего двоеточия до ближайшего разделителя альтернатив (!), слова ИНАЧЕ или ВСЕ. После этого исполнение предписания заканчивается. Если ни одно условие не было истинным, выполняются предписания между ИНАЧЕ и ВСЕ (если слово ИНАЧЕ использовано). Таким образом, в любом случае выполняется не более одной из предусмотренных альтернатив.

Пример:

ВЫБОР ИЗ

Ветер < = 3: ВЫВОД: "слабо" !  
Ветер < = 7: ВЫВОД: "умеренно" !  
Ветер < = 70, ВЫВОД: "Тревога" !  
ИНАЧЕ ВЫВОД: "Неверно задана скорость"  
ВСЕ;

5.4. Вторая разновидность предписания ВЫБОР исполняется в таком же порядке, но проверяемое условие имеет вид:  $b_0 = b_a$ , где  $b_0$  - значение переключательного выражения, записанного между словами ВЫБОР и ИЗ,

$b_a$  - выражение, записанное в альтернативе (перед двоеточием). Таких выражений может быть несколько (через запятую).

Значение переключательного выражения вычисляется один раз в начале исполнения предписания. В остальном семантика этого предписания соответствует п.5.3.

Пример:

ВЫБОР Оценка ИЗ

5: ВЫВОД: "отлично" !  
4: ВЫВОД: "хорошо" !  
3: ВЫВОД: "средне" !

3533847.00058-01 35 01

1,2: ВЫВОД: "плохо"

ИНАЧЕ ВЫВОД: "ошибка"

ВСЕ;

5.5. Цикл ПОКА (диаграмма I3) имеет традиционную семантику: вначале проверяется условие, указанное в заголовке цикла. Если оно истинно, выполняются все предписания от знака :: (читается "повторять") до слова ВСЕ (тело цикла). Затем вновь проверяется условие и т.д. Если при очередной проверке условие оказалось ложным, исполнение предписания завершается. Если оно было ложным при первой же проверке, то тело цикла не исполняется ни разу.

Пример:

1 -> X;

ПОКА: X < 10 :: ВЫВОД: 2 \* X; X + 1 -> X ВСЕ;

5.6. Цикл ПОВТОР позволяет выполнить некоторый набор предписаний заданное число раз. Он выполняется следующим образом. Вначале вычисляется значение выражения, указанного в заголовке цикла. Оно должно быть неотрицательным целым числом, иначе выдается сообщение об ошибке. Затем тело цикла исполняется указанное число раз.

Пример:

1 -> X;

ПОВТОР 10 РАЗ :: ВЫВОД: 2 \* X; X + 1 -> X ВСЕ;

5.7. Цикл ДЛЯ-ОТ соответствует традиционному для большинства языков циклу типа прогрессии. Имя, указанное после слова ДЛЯ, является переменной цикла, выражение после слова ОТ определяет начальное значение, выражение после слова ДО - конечное значение, после слова ШАГ указывается шаг приращения. Если шаг не указан, он принимается равным единице. Значения всех трех выражений должны быть числами (целыми или дробными). Семантика цикла традиционна.

3533847.00058-01 35 01

Переменная цикла в теле цикла имеет абсолютную защиту. После окончания цикла она имеет пустое значение.

Пример:

ДЛЯ X ОТ 1 ДО 10 :: ВЫВОД: 2 \* X ВСЕ;

5.8. Цикл ДЛЯ-ИЗ предназначен для обработки текстов и составных структур, а также файлов соответствующих видов. При исполнении этого предписания переменная цикла поочередно принимает значения каждого элемента соответствующей структуры или файла (или каждого символа текста), после чего исполняется тело цикла. Как и в цикле ДЛЯ-ОТ переменная цикла в теле этого цикла имеет абсолютную защиту, а после его завершения получает пустое значение. Выражение, указанное в заголовке цикла после слова ИЗ, вычисляется один раз перед началом работы цикла.

Пример - подсчет количества букв "А" в тексте с именем Книга:

0 -> Сч;

ДЛЯ Буква ИЗ Книга :: ЕСЛИ Буква="А" ТО Сч+1 -> Сч ВСЕ ВСЕ;

ВЫВОД: "В тексте Книга", Сч, " букв < А >";

## 6. ПРОЦЕДУРЫ И ФУНКЦИИ

6.1. Синтаксическая структура, а также порядок описания и вызова функций и процедур в Рапире и Робики очень близки. Поэтому в этом параграфе перечисляются только отличия процедурного аппарата Рапиры от аналогичного аппарата Робики.

6.2. Как и в Робики, процедурный блок в Рапире является основной единицей локализации имен и хранения программного текста. В дополнение к этому он приобретает свойства основной единицы области действия описаний. Как видно на диаграммах I4-I5, при описании формальных параметров и локальных имен в Рапире, в отличие от Робики, можно указывать тип.

6.3. В заголовках процедур можно указывать способ передачи каждого из параметров. В Рапире используется три разновидности параметров, различающиеся по способу передачи: входные, выходные и возвратные. Эти способы обозначаются символами  $=>$ ,  $<=>$  и  $<=>$  соответственно. Если формальный параметр описан как входной, то соответствующий фактический параметр может быть любым выражением. Фактические параметры, соответствующие двум другим разновидностям формальных, могут быть только именами.

Входные параметры передаются по значению: в момент вызова процедуры значение фактического параметра вычисляется и присваивается соответствующему формальному, на который после этого устанавливается полная защита. Для функций в Рапире, а также для всех видов процедур блоков в Робики этот способ является единственно возможным.

Выходные параметры обрабатываются следующим образом. При вызове процедуры такой формальный параметр получает значение ПУСТО.

После завершения работы процедуры значение формального параметра присваивается соответствующему фактическому.

Возвратный параметр сочетает свойства входного и выходного: при вызове значение фактического параметра присваивается формальному, а после завершения работы процедуры - наоборот.

6.4. Исполнение функции или процедуры в языке Рапиры может быть приостановлено при помощи предписания СТОП (с выходом в специальный режим диалога) или прервано при помощи предписания ВЫХОД.

В режиме приостановки процедуры пользователь может выполнить любое предписание Рапиры, причем все имена сохраняют смысл и значение, которые они имели в момент приостановки. После этого выполнение процедуры может быть продолжено при помощи директивы ПУСК или прекращено при помощи предписания ВЫХОД, использованного в качестве директивы.

6.5. Вызов функции и процедуры в Рапире рассматривается как операция над процедурным значением. Поэтому вызываемая процедура может быть вычислена как значение некоторого выражения непосредственно перед вызовом. Например, предписание

$\langle A, B, C \rangle [ X ] (2, 3);$

позволяет вызвать одну из трех процедур A, B и C, в зависимости от значения имени X, которое должно быть целым числом от 1 до 3. Каждая из процедур должна быть описана с двумя входными параметрами.

6.6. Как в Робики, так и в Рапире, допускаются рекурсивные вызовы функций и процедур. В обоих языках описание и редактирование процедур осуществляется при помощи одних и тех же системных средств.

## 7. ФАЙЛЫ И МОДУЛИ

7.1. Работа системы "Школьница" протекает в операционной среде, основную часть которой составляет система файлов, хранящихся в долговременном запоминающем устройстве с произвольным доступом.

С точки зрения пользователя, файл - это объект языка, значение которого хранится в ДЗУ и сохраняется при выключении или пере-вызове системы. Каждому виду значений, используемому в Рапире, соответствует своя разновидность файлов с таким же названием. Кроме того, в Рапире предусмотрены виды файлов, не имеющие прямых аналогов среди "внутренних" объектов языка: двоичные файлы и специальные файлы других входных языков системы "Школьница", к которым возможен доступ из Рапиры.

7.2. В соответствии с концентрической структурой языка Рапира, работа с файлами может осуществляться несколькими различными способами, последовательно вводимыми на различных концентрических уровнях языка.

1. Всякий объект языка, включая процедуры, может быть целиком скопирован из оперативной памяти во внешнюю или наоборот с помощью предписаний ВЫВОД В ДЗУ и ВВОД ИЗ ДЗУ (диаграммы 10-11).

2. Символьный или графический ввод и вывод может быть переключен на определенный файл при помощи предписаний ВКЛЮЧИТЬ ВЫВОД В ФАЙЛ и ВКЛЮЧИТЬ ВВОД ИЗ ФАЙЛА (диаграмма 6). Однократный обмен с файлом может быть организован с помощью предписаний ВВОД ИЗ ФАЙЛА и ВЫВОД В ФАЙЛ (диаграммы 10-11).

3. Последовательная поэлементная обработка файлов, соответствующих структурным видам значений Рапиры, может быть организована с помощью предписания ДЛЯ-ИЗ ФАЙЛА (диаграмма 13). В этом случае с точки зрения пользователя файл загружается в ОЗУ и обрабатывается поэлементно.

4. Организация традиционных операций с текстовыми файлами (открытие и закрытие, настройка на позицию, возврат к началу и т.д.) осуществляется при помощи специальных предписаний и псевдо-процедур.

5. Управление файловой системой может осуществляться в процессе диалога с текстовым редактором системы "Школьница", используемым, в частности, для описания и редактирования процедур и функций Рапиры. При помощи этого же редактора могут создаваться и редактироваться текстовые файлы.

6. Просмотр каталога файлов может быть выполнен и вне редактора, при помощи соответствующей системной директивы.

7.3. Файлы системы "Школьница" объединены в модули. Модуль - это совокупность файлов различных видов, в том числе - описаний функций и процедур, которые имеют общий набор глобальных переменных.

Создание модуля осуществляется системными средствами, не определяемыми описанием языка Рапира. При этом среди глобальных переменных пакета могут быть описаны закрытые переменные, недоступные пользователю, что позволяет использовать модули для организации расслоенной структуры больших программ.

3533847.00058-01 35 01

## 8. НЕКОТОРЫЕ ОТЛАДОЧНЫЕ СРЕДСТВА ЯЗЫКА РАПИРА

8.1. Характерной чертой современных интровертивных языков является наличие большого количества отладочных средств, предназначенных для диалогового использования. Умение пользоваться такими средствами необходимо формировать на этапе обучения программированию, поэтому в Рапире предусмотрена широкая гамма отладочных возможностей и конструкций. В этом параграфе перечисляются важнейшие из них.

8.2. В процессе диалоговой работы может быть включена трассировка вызовов процедур и трассировка присваиваний (прокрутка). Включение и выключение этих механизмов осуществляется при помощи соответствующей разновидности предписаний ВКЛЮЧИТЬ и ВЫКЛЮЧИТЬ (диаграмма 6).

8.3. В языке предусмотрена возможность организовать проверку утверждений о программе, записанных в виде условий на языке Рапира. Такая проверка осуществляется с помощью предписания КОНТРОЛЬ, синтаксис которого описывается диаграммой 16. Если при исполнении этого предписания условие оказывается ложным, формируется выход по ошибке.

8.4. Наряду с контрольными точками в Рапире можно описать набор условий - инвариантов, проверка которых будет выполняться при каждом присваивании. Включение нового инварианта осуществляется по предписанию ВКЛЮЧИТЬ КОНТРОЛЬ, а выключение - по предписанию ВЫКЛЮЧИТЬ КОНТРОЛЬ. Если при некотором присваивании хотя бы один инвариант оказался ложным, формируется выход по ошибке.

8.5. Обработка ошибок и прерываний в Рапире является программируемой. При возникновении ошибки или прерывания система форми-

3533847.00058-01 35 01

рует обращение к стандартной функции, которая осуществляет обработку исключительной ситуации. Эта функция доступна пользователю, что позволяет ему описать на Рапире свои функции для обработки исключительных ситуаций.

8.6. В число отладочных средств входят и некоторые конструкции и механизмы Рапиры, описанные в предыдущих подразделах, в частности:

1. Режим приостановки процедур;
2. Контроль типов имен (для диалоговой версии);
3. Защита имен по записи.

8.7. Отладочные средства, перечисленные в пунктах 8.2-8.4, а также проверка типов имен функционируют только в отладочном режиме, включаемом при помощи предписания ВКЛЮЧИТЬ ОТЛАДКУ (диаграмма 6).

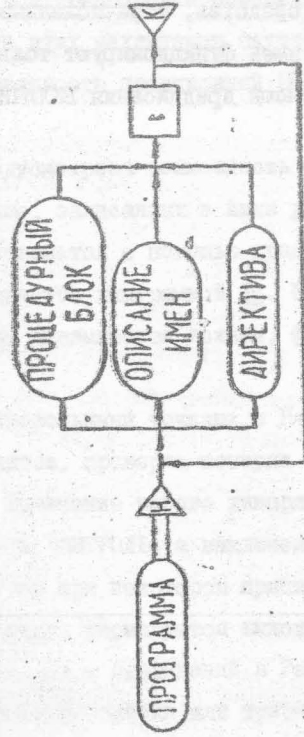
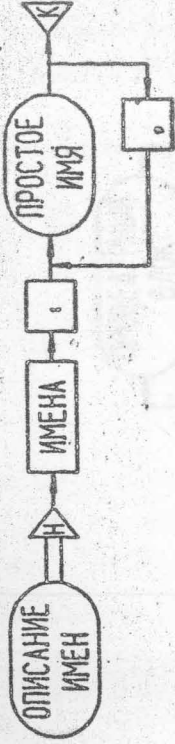


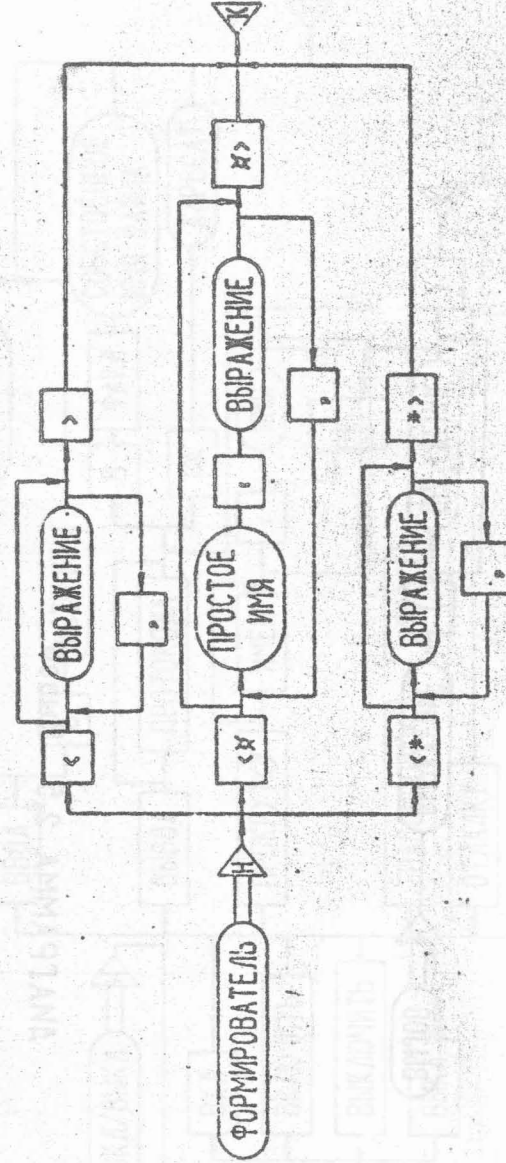
ДИАГРАММА 2.1 Программа.



ДИАГРАММА 2.2 Комментарий.



· ДИАГРАММА 2.3 Описание имен.



· ДИАГРАММА 2.4 Формирователь.

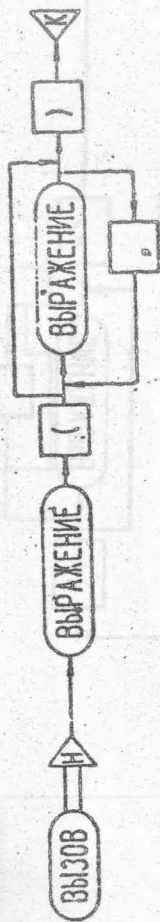


ДИАГРАММА 2.5 Вызов

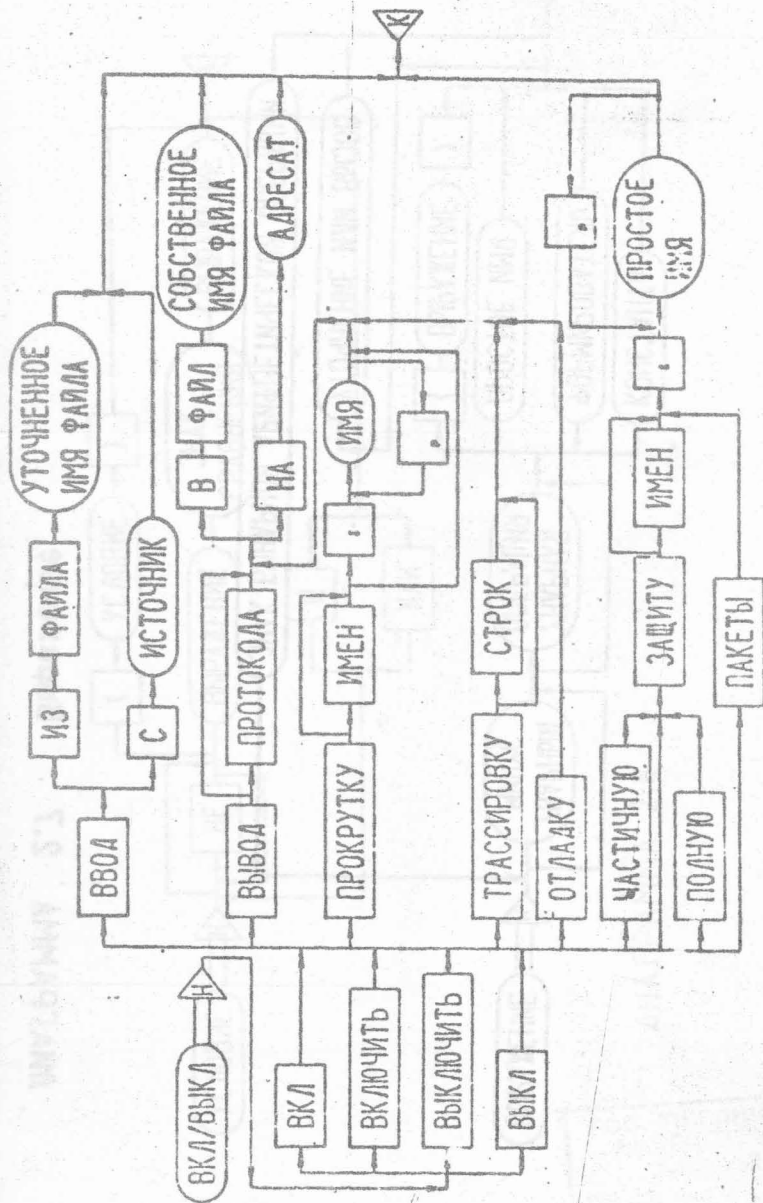


ДИАГРАММА 2.6 Вкл/выкл.



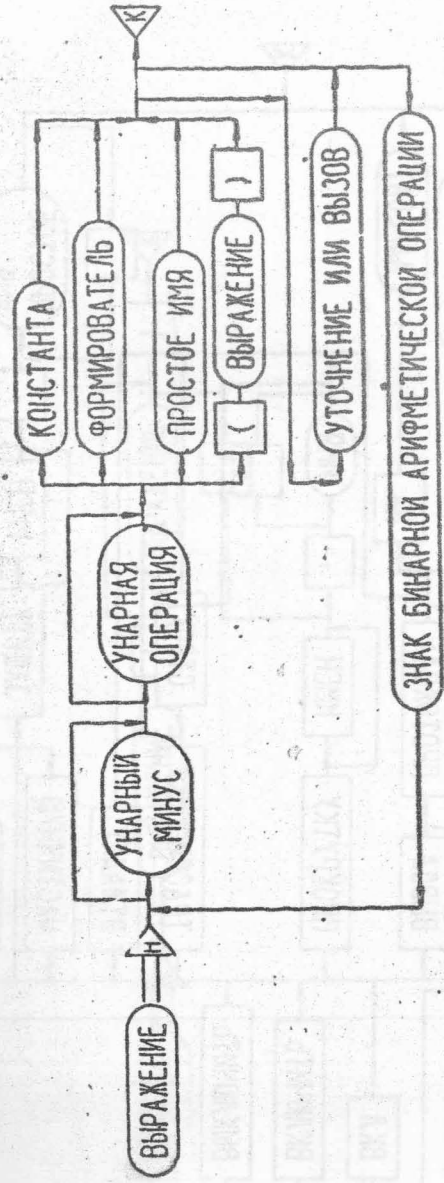


ДИАГРАММА 2.7 Выражение

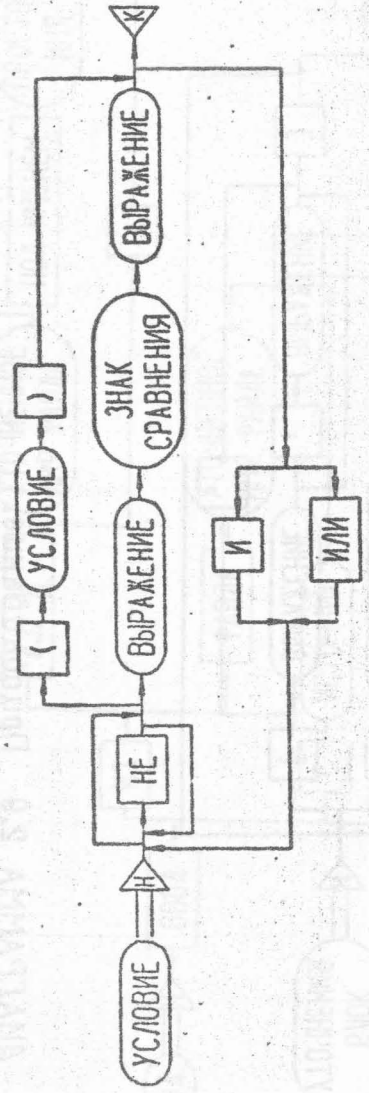


ДИАГРАММА 2.8

ГОС. ПУБЛИЧНАЯ  
 БИБЛИОТЕКА  
 Ленинград  
 09 108 акт

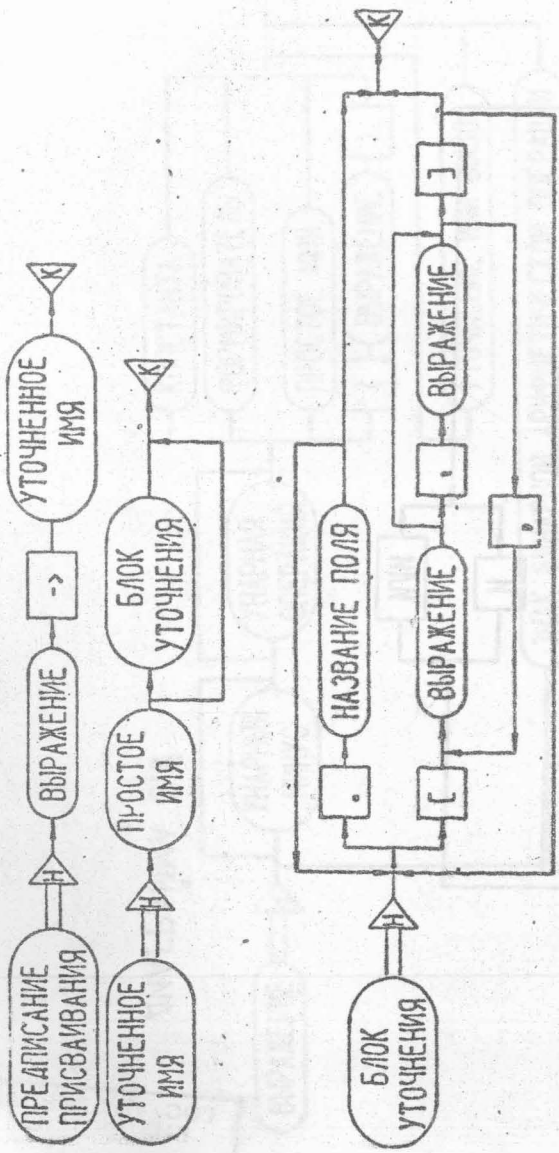


ДИАГРАММА 2.9 Присваивание.

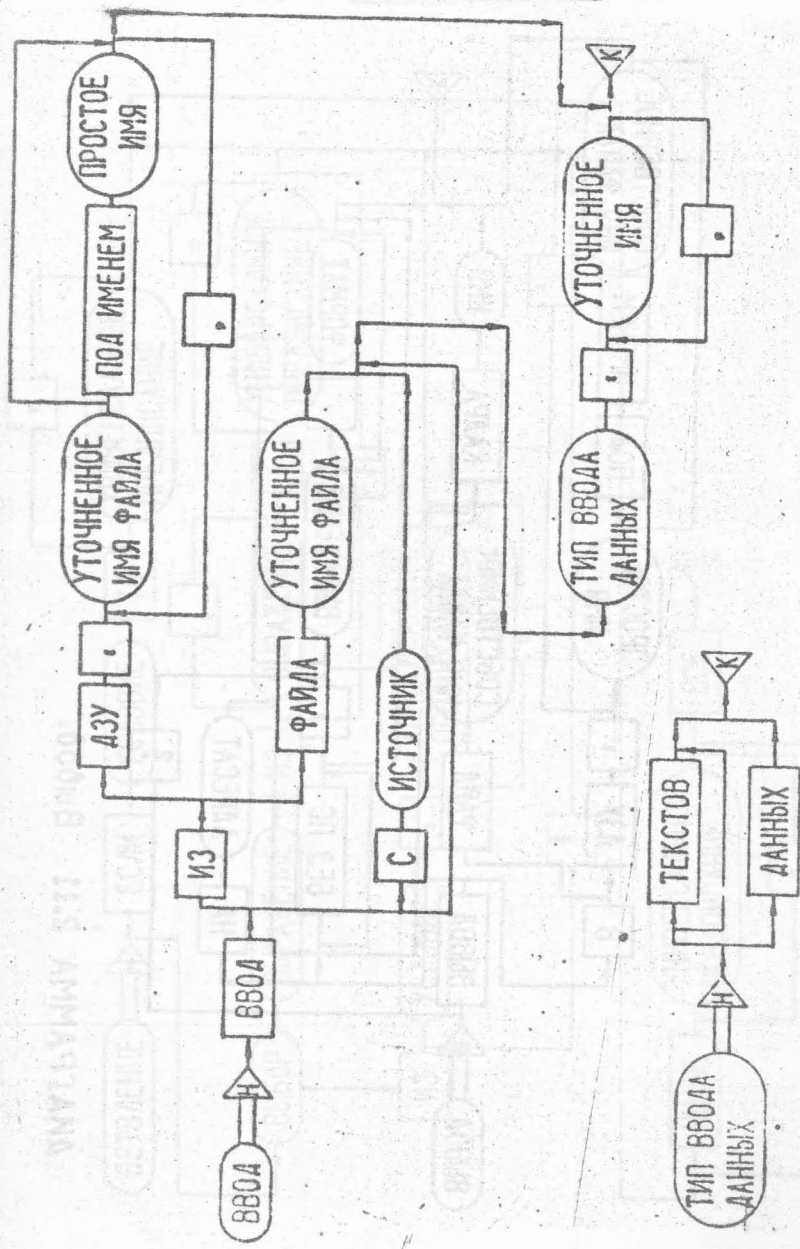


ДИАГРАММА 2.10 Ввод.

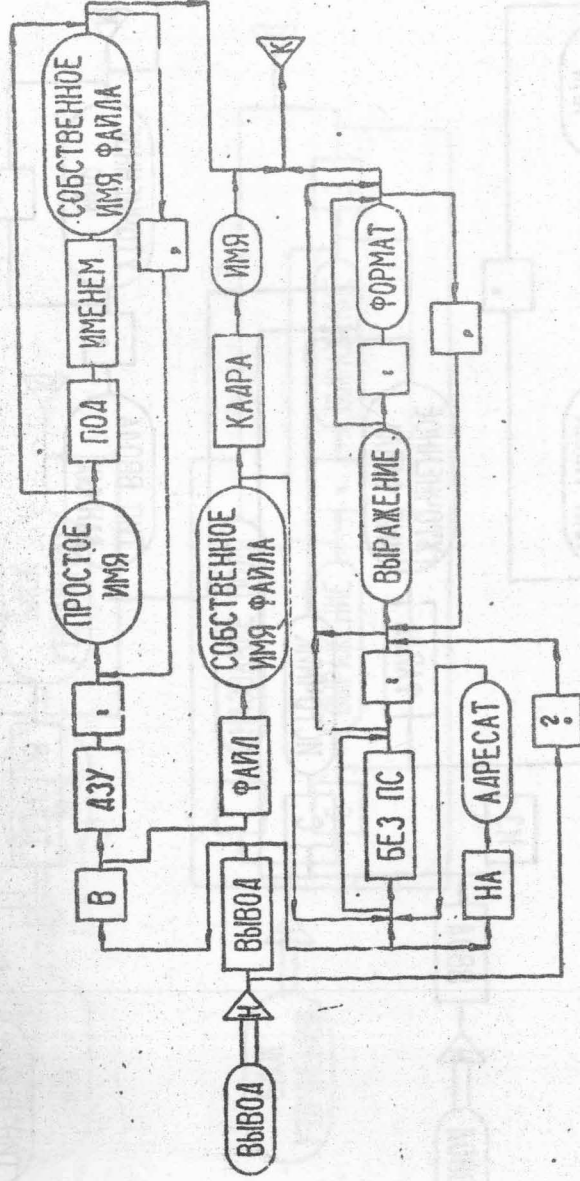


ДИАГРАММА 2.11 Выбор.

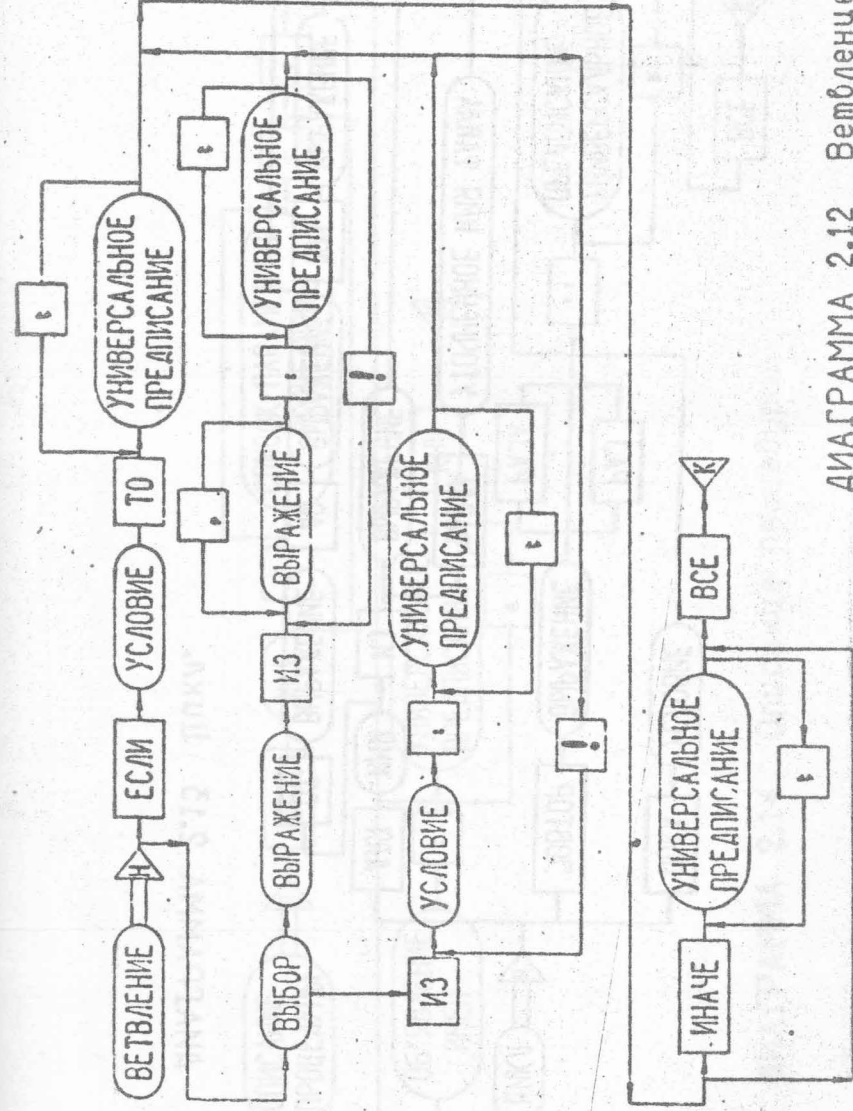


ДИАГРАММА 2.12 Ветвление

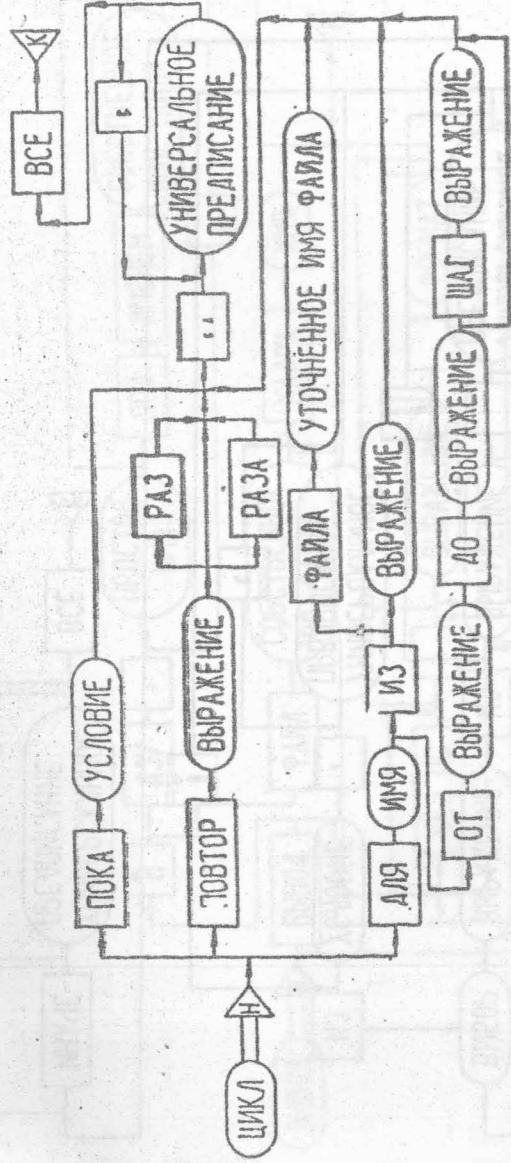


ДИАГРАММА 2.13 Цикл.

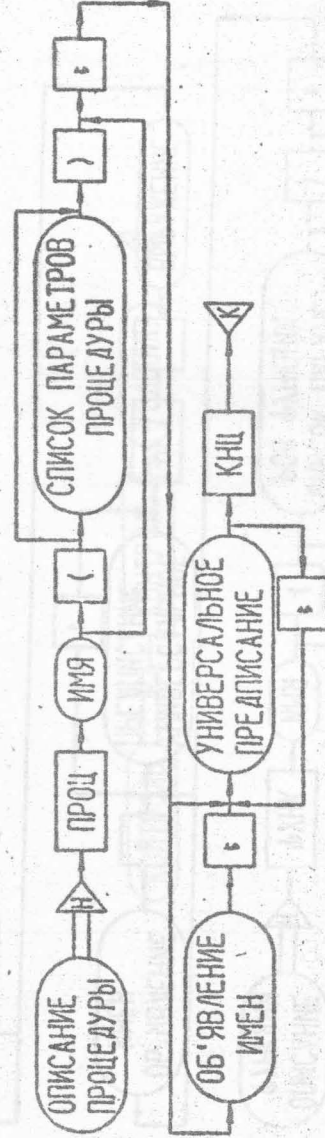


ДИАГРАММА 2.14 Описание процедуры.

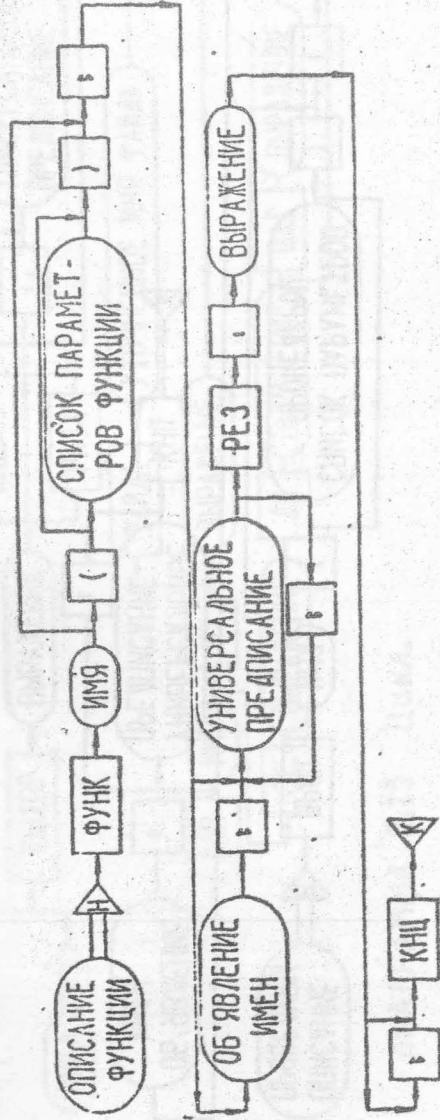


ДИАГРАММА 2.15 Описание функции.



ДИАГРАММА 2.16 Контроль

### ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм	Номера листов (страниц)				Всего л.ст.ов (страниц) в докум.	№ документа	Входящий № сопро- водительно- го докум. и дата	Подп.	Де- те
	Изменен- ных	Заменен- ных	новых	аннулиро- ванных					

Д-Т-1522 прт 6/III 88 Подписано в печать 13/III 88  
 Формат бумаги 60x84/16 Бумага писчая. Ротапринт.  
 Объем 2,5 п.л. Тираж 250 экз. Заказ № 56 Бесплатно.  
 Ротапринт НИИПОТСО АГН СССР. 103062 Москва, Длин п., д. За.