

80 068
85-4
420352

МИНИСТЕРСТВО РАДИОПРОМЫШЛЕННОСТИ СССР

АКАДЕМИЯ НАУК СССР
ОРДЕНА ЛЕНИНА СИБИРСКОЕ ОТДЕЛЕНИЕ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

УТВЕРЖДЕН

3533847.00058-01 35 02-ЛУ

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ
АВТОМАТИЗАЦИИ ШКОЛЬНОГО УЧЕБНОГО ПРОЦЕССА
(ШКОЛЬНИЦА)

РОБИК

ОПИСАНИЕ ЯЗЫКА

3533847.00058-01 35 02

1985

3533847.00058-01 35 02

АННОТАЦИЯ

В документе описывается язык программирования Робик, являющийся одним из двух входных языков учебной системы программирования, входящей в состав интегрированной программы среды "Школьница".

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

Документ предназначен для всех пользователей языка.

854
420352

СОДЕРЖАНИЕ

- 1. Общие сведения
- 2. Лексика Робика
- 3. Общая структура программы и правила записи предписаний
- 4. Имена и выражения
- 5. Универсальные предписания
- 6. Присваивание, ввод и вывод
- 7. Управляющие конструкции языка Робик
- 8. Исполнители в языке Робик
- 9. Пример исполнителя

85-80068



I. ОБЩИЕ СВЕДЕНИЯ

I.1. Язык Робик предназначен для начального обучения школьников младшего и среднего возраста (8-11 лет) важнейшим навыкам программирования и для формирования основных компонент программистского стиля мышления, имеющих общекультурную ценность.


I.2. Робик является одним из входных языков системы программного обеспечения "Школьника", предназначенной для оснащения школьных вычислительных кабинетов. Все операции надпрограммного уровня (редактирование и трансляция процедур, настройка операционной обстановки, подключение и модификация библиотек и пакетов процедур и т.д.) осуществляются средствами системы "Школьника" общими для всех ее входных языков.





I.3. Первая версия языка была разработана в 1975 году и использовалась в учебной работе харьковской, а затем новосибирской Школьных программистов. В ходе разработки системы "Школьника" язык был заново переработан.

I.4. Робик представляет собой язык-ядро с управляемой синтаксической структурой. Наряду со сравнительно небольшим набором базовых конструкций, в состав языка может быть включено произвольное число исполнителей, описанных средствами системы "Школьника".

I.5. В качестве базового этнического языка для Робика выбран русский язык. В принципе возможно создание версий на основе языков других народов СССР.

I.6. При описании синтаксиса Робика и других входных языков системы "Школьника" используются синтаксические диаграммы. При этом применяются следующие обозначения:

 - начало (слева может быть указано название конструкции, описываемой диаграммой).

-  - конец
-   - терминальный символ или цепочка терминальных символов
-  - нетерминальный символ.

2. ЛЕКСИКА РОБИКА

2.1. В языке Робик различаются следующие логические единицы:

- целые числа
- дробные числа
- тексты
- слова (простые имена и ключевые слова)
- специальные символы (разделители).

2.2. Целые и дробные числа, а также тексты записываются в традиционной форме.

2.3. Слова в Робике записываются в виде последовательности букв и цифр, начинающейся с буквы. Внутри слова допускается использование одиночных подчеркивов (_) для построения сложных имен.

Понятие слова объединяет имена (простые имена по терминологии, принятой в описании языка) и ключевые слова в обычном смысле. Строго говоря, ключевых (зарезервированных) слов в Робике нет, все его предписания записываются в развернутой форме и ключевыми можно считать все слова каждого предписания, за исключением имен.

Ключевые слова в Робике никак не выделяются, поскольку такое выделение нарушило бы естественность и синтаксическую выразительность предписаний. С другой стороны, синтаксическое выделение имен, принятое, например, в языке Лого, противоречит требованию преемственности и, как показывает опыт, плохо усваивается младшими школьниками. По изложенным причинам в Робике принято контекстное распознавание имен.

2.4. Простые и составные специальные символы, используемые в Робике, будут перечислены при описании тех конструкций языка, в которых они используются.

3. ОБЩАЯ СТРУКТУРА ПРОГРАММЫ И ПРАВИЛА ЗАПИСИ ПРЕДПИСАНИЙ

3.1. Программа на Робике - это последовательность директив и процедурных блоков (диаграмма I).

В качестве директивы может быть использовано любое предписание базовой части языка, директива системы "Школьница", не входящая в состав языка, или предписание из МПИ любого включенного в данный момент исполнителя.

Процедурный блок - это описание процедуры или функции. Порядок описания и вызова процедур и функций в Робике описан ниже.

В основном диалоговом режиме каждая директива интерпретируется и, если это возможно, исполняется замедленно после окончания ее ввода и выполнения перевода строки (символ на диаграмме).

3.2. Как видно на диаграмме I, все директивы и процедурные блоки должны начинаться с новой строки. В конце директивы или блока ставится точка с запятой, после которой может быть записан произвольный комментарий вплоть до конца строки.

Формат программы свободный: предписания можно начинать с любой позиции в строке, что позволяет оформлять структурированную запись текста программы. Пробелы и переводы строки могут использоваться в любом месте программы, но не внутри лексем (внутри текстов могут использоваться пробелы, если они являются составной частью текста). Между двумя лексическими единицами, не разделенными контекстно, обязателен по крайней мере один пробел или перевод строки. Количество пробелов и переводов строк между лексемами не ограничивается.

4. ИМЕНА И ВЫРАЖЕНИЯ

4.1. В Робики различаются простые и составные имена (диаграмма 2). Составные имена соответствуют внутренним блокам памяти исполнителей, порядок их употребления описан ниже.

4.2. Простые имена в Робики не описываются. Каждое имя считается известным с момента его первого использования в программе.

Значением каждого имени может быть объект одного из следующих видов:

- целое число;
- дробное число;
- текст;
- рисунок;
- процедура;
- функция;
- исполнитель;
- модель предметной среды;

Типизация имен в языке не предусмотрена: одно и то же имя может принимать значения различных видов в произвольной последовательности.

Если имени не присвоено никакого значения, то оно имеет специальное пустое значение ПУСТО. В системе предусмотрено соответствующее стандартное имя, всегда имеющее это значение.

4.3. Отдельные имена, используемые в программе, могут иметь полную или частичную защиту. Полностью защищенному имени нельзя присвоить другое значение; попытка такого присваивания рассматривается как ошибка. При попытке присваивания частично-защищенному имени система требует подтвердить необходимость такого присваива-

ния. Присваивание выполняется только в том случае, если такое подтверждение получено.

Частичную защиту имеют имена процедур и функций, а также параметры процедур. Это позволяет избежать случайной потери информации из-за неосторожного выбора вспомогательного имени.

Полную защиту имеют имена исполнителей и моделей предметных сред, некоторые внутренние имена исполнителей, стандартное имя ПУСТО, а также имена всех процедур во время их исполнения.

Средства для управления защитой имен в языке недоступны пользователю-школьнику и могут быть использованы только преподавателем.

4.4. Выражения в Робики имеют упрощенную структуру: в качестве выражения может использоваться имя, число, текст или вызов функции. Трансляция арифметических и текстовых выражений, записанных в традиционной инфиксной форме, не была включена в базовую часть языка.

5. УНИВЕРСАЛЬНЫЕ ПРЕДПИСАНИЯ

5.1. В языке Робик универсальным предписанием называется всякое самостоятельное предложение, которое может быть использовано и в качестве директивы, и в качестве оператора внутри тела процедуры.

5.2. В роли универсальных предписаний могут использоваться простые и составные предписания базового языка (диаграмма 3), также предписания из МПИ отдельных исполнителей.

Простое предписание может быть адресовано определенному исполнителю. В этом случае перед предписанием указывается имя исполнителя, отделяемое от самого предписания двоеточием. Необходимость в такой адресации возникает в тех случаях, когда одновременно оказываются включенными несколько исполнителей, в МПИ которых есть совпадающие предписания.

5.3. В базовой части Робика предусмотрены следующие простые и составные предписания:

- простые: присваивание, ввод, вывод, вызов, включение и выключение;
- составные: ветвление, цикл ПОКА и цикл ВЫПОЛНИТЬ.

Синтаксис и семантика этих предписаний будет рассмотрена в следующих параграфах.

6. ПРИСВАИВАНИЕ, ВВОД И ВЫВОД

6.1. Синтаксис предписаний присваивания, ввода и вывода в языке Робик описывается диаграммами 4-6. На диаграммах хорошо видно, что каждое из этих предписаний имеет несколько семантических эквивалентных форм записи - от развернутых, полностью отражающих семантику соответствующей конструкции, до кратких, близких к традиционным для языков программирования формам. Такая многовариантность позволяет сочетать требования синтаксической выразительности и компактности записи. На первых стадиях изучения языка используется развернутая форма. Эта форма усваивается учащимися путем произвольного запоминания и в дальнейшем служит инструментом вербализованного мышления при составлении и проверке программ, актуализируясь при возникновении затруднений в определении семантики конструкции. Впоследствии учащиеся переходят к сокращенным формам записи, что позволяет избежать громоздких и многослойных конструкций.

Такой прием, который можно назвать снижаемым синтаксисом, широко используется при разработке диалоговых систем для различных категорий пользователей. Эффективность приема в применении к младшим школьникам подтверждена опытом работы новосибирской школы юных программистов.

В качестве примера можно привести различные возможности формы записи предписания присваивания: Полная форма:

ЗНАЧЕНИЕ ВЫРАЖЕНИЯ 287 ПОМЕСТИТЬ В БЛОК ПАМЯТИ С ИМЕНИЕМ Число;

Промежуточная форма:

ЗНАЧЕНИЕ 287 ПРИСВОИТЬ ИМЕНИ Число;

Краткая форма:

287 ПРИСВОИТЬ Число;

Семантика предписания присваивания очевидна.

6.2. Предписание вывода имеет традиционную семантику: значения перечисленных в нем выражений выдаются на экран терминала или на указанное устройство. Вывод производится новой строки, все значения выводятся в естественном формате, вплотную друг к другу. Числа выводятся с фиксированной точкой, тексты - без окаймляющих кавычек. Перед каждым числом автоматически выполняется один пробел.

Вывод рисунка может быть выполнен только после включения графического режима (например, при работе с системой Шпага). При попытке выдать другие значения, не имеющие эквивалентного языкового представления (описания процедур или функций, исполнители, модели предметных сред, пустое значение) на заданное устройство выдается текст (в кавычках) с названием соответствующего вида значения, например, "ПРОЦЕДУРА" или "ПУСТО".

6.3. Способ исполнения предписания ввода зависит от выбранной его формы. Вариант ПРИНЯТЬ СООБЩЕНИЕ используется для ввода текстовых сообщений в естественном виде (без кавычек). В этом случае возможен ввод только одного значения в каждом предписании.

Вариант ПРИНЯТЬ ДАННЫЕ позволяет вводить значения нескольких имен, записанные по общим правилам языка (в частности, тексты - в кавычках). Вводимые значения разделяются запятыми.

В остальном семантика этого предписания традиционна.

7. УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ ЯЗЫКА РОБИК

7.1. Управляющие конструкции языка представлены тремя составными предписаниями: ЕСЛИ, ПОКА и ВЫПОЛНИТЬ, а также средствами для описания и вызова процедур и функций.

Синтаксис составных предписаний и используемой в них конструкции "условие" описывается диаграммами 7-9. Семантика первых двух предписаний очевидна. Предписание ВЫПОЛНИТЬ позволяет повторять некоторые действия заданное число раз. Его исполнение происходит в следующем порядке:

1. Вычисляется значение выражения, входящего в заголовок цикла.
2. Если это значение является натуральным числом, то предписание, образующее тело цикла будет выполнено соответствующее число раз. Если значение этого выражения равно нулю, тело цикла не выполняется ни разу. Все остальные возможные варианты считаются ошибочными.

В Робике отсутствуют рамочные конструкции, за исключением процедур и функций. Поэтому тела циклов и альтернативы в предписании ЕСЛИ образованы одиночными предписаниями. При необходимости использовать в этом контексте несколько предписаний, их требуется объединить в процедуру.

7.2. Аппарат процедур играет в Робике чрезвычайно важную роль. Наряду со своими обычными функциями, процедура в Робике является единственной единицей структурирования программы, единицей локализации имен и единицей хранения программного текста.

Синтаксис описания процедур и функций описывается диаграммами 10-11. В заголовке описаний необходимо указать имя процедуры (функции) и можно перечислить ее формальные параметры и локальные имена.

Параметры и локальные имена локализуются в теле описания, причем параметры считаются частично-защищенными именами. Все локальные имена и параметры в одном описании должны попарно различными. Нелокальные имена, использованные в описании, понимаются в том смысле, какой они имеют в момент вызова процедуры (функции).

В окончании описания функции обязательно должен быть указан результат, вырабатываемый функцией.

7.3. После набора заголовка процедуры или функции система автоматически переключается в специальный режим - режим запоминания текста, общий для всех входных языков системы "Школьника". Порядок работы в этом режиме регламентируется описанием текстового редактора, входящего в состав системы. При нормальном завершении набора текста описание транслируется, полученный объектный код присваивается имени процедуры (функции), текст описания записывается в библиотеку или сохраняется в бефере редактора и система возвращается в основной диалоговый режим, в котором возможно исполнение директив языка Робик.

Таким образом, каждая процедура хранится и транслируется отдельно от остальных; вложенные описания процедур в Робике невозможны.

7.4. Синтаксис вызова процедур и функций описывается диаграммой 12.

Для процедур возможно две формы вызова: с помощью предписания ВЫЗВАТЬ или с помощью круглых скобок, записанных после имени процедуры. Для функций допускается только второй способ вызова. При вызове процедур и функций с параметрами в скобках после имени должны быть указаны фактические параметры, количество которых должно совпадать с количеством формальных параметров. В Робике предусмот-

рен только один способ передачи параметров - по значению - поэтому фактическими параметрами могут быть любые выражения (в смысле Робика, см. раздел 4).

8. ИСПОЛНИТЕЛИ В ЯЗЫКЕ РОБИК

8.1. Аппарат исполнителей составляет наиболее специфическую особенность Робика, определяющую возможность его использования в работе с младшими школьниками.

Каждый исполнитель функционирует в определенной среде (фактически - на модели предметной среды). Среда состоит из конечного набора элементов, которые могут находиться в различных состояниях. Совокупность текущих состояний всех элементов образует обстановку в среде.

Для каждого исполнителя считается известным набор предписаний, которые могут быть им выполнены, т.е. множество предписаний исполнителя (МПИ). Полное описание МПИ определяет функциональные возможности исполнителя и синтаксис его языкового представления.

Синтаксис каждого предписания из МПИ исполнителя описывается при помощи синтаксических диаграмм, то есть так же, как базовые предписания языка Робик.

Семантика предписания определяется теми изменениями, которые происходят с элементами среды при выполнении этого предписания. Для каждого предписания считается заданным условие корректности, определяющее множество обстановок в среде, для которых возможно исполнение предписания. Если это условие нарушено, попытка исполнения предписания рассматривается как ошибка.

Исполнители могут иметь внутренние блоки памяти, содержимое которых изменяется в зависимости от обстановки в среде и от действий, предпринимаемых самим исполнителем. Для обращения к таким блокам используются составные имена, синтаксис которых описывается диаграммой 2. Как указывалось в разделе 4, почти все такие имена являются полностью защищенными.

Работа отдельных исполнителей может изображаться на экране те-

левизионного монитора. В этом случае при описании семантики должна быть задана вся последовательность изменений, происходящих на экране в ходе выполнения каждого предписания.

В составе системы "Школьница" предусмотрен набор стандартных исполнителей, предметная среда которых определяется технической конфигурацией системы. В состав этого набора входят, например, исполнители для выполнения арифметических и текстовых операций, для вычерчивания рисунков и управления внешними устройствами, подключенными к системе.

8.2. Выполнение предписания из МПИ какого-либо исполнителя возможно в программе только в том случае, если предварительно был создан по крайней мере один экземпляр этого исполнителя. Генерация экземпляра осуществляется предписанием ВКЛЮЧИТЬ, синтаксис которого описывается диаграммой 13. Каждый экземпляр исполнителя имеет имя, определяемое предписанием ВКЛЮЧИТЬ и самостоятельный набор внутренних имен. Если имя не указано, то исполнителю присваивается имя, совпадающее с его названием (названием класса). В любой программе может быть создано любое число экземпляров исполнителей каждого класса, функционирующих в рамках одной и той же модели предметной среды. Например, если в системе описан исполнитель под названием МУРАВЕЙ, то по предписаниям

ВКЛЮЧИТЬ МУРАВЕЙ:МАША;

ВКЛЮЧИТЬ МУРАВЕЙ:МИША;

будут созданы два экземпляра исполнителя МУРАВЕЙ с именами МАША и МИША. По предписанию ВКЛЮЧИТЬ МУРАВЕЙ; будет создан один экземпляр такого исполнителя с именем МУРАВЕЙ.

8.3. При включении первого исполнителя, действующего в некоторой среде, система запрашивает (или генерирует, в зависимости от

описания) начальную обстановку в соответствующей среде и начальные значения внутренних имен исполнителя.

При включении последующих исполнителей, функционирующих в той же среде, запрашиваются (генерируются) только значения их внутренних имен.

8.4. Уничтожение экземпляра исполнителя производится по предписанию ВЫКЛУЧИТЬ, синтаксис которого описывается диаграммой 14. Если имя выключаемого исполнителя не указано, будут выключены все включенные в данный момент исполнители. При выключении последнего исполнителя, функционирующего в некоторой среде, информация об обстановке в этой среде стирается из памяти.

8.5. Если одновременно включено несколько экземпляров одного и того же исполнителя, или если у включенных в данный момент экземпляров различных исполнителей совпадают предписания из МПИ, каждое предписание может быть адресовано определенному исполнителю. Способ адресования описан в п. 5.2. Если адресование не применяется, предписание будет исполнено всеми исполнителями, в МПИ которых оно входит, в произвольном порядке.

8.6. При работе с исполнителями могут быть использованы все конструкции базовой части языка Робик, в том числе составные предписания и процедуры. Без применения исполнителей, очевидно, на Робике могут быть составлены только самые тривиальные программы.

9. ПРИМЕР ИСПОЛНИТЕЛЯ

9.1. В качестве примера рассмотрим описание исполнителя ОБХОДЧИК, предусмотренного для наглядного изложения свойств управляющих конструкций языка. С точки зрения пользователя-школьника этот исполнитель предназначен для проверки стыков на участках железных дорог. Он может двигаться вперед и назад по железнодорожным путям на заданное расстояние, проверять исправность стыков и отмечать (например, окрашивать в красный цвет) отдельные стыки.

9.2. Среда для исполнителя обходчик представляет собой железнодорожную линию (фактически, разумеется, модель такой линии). Линия с двух сторон ограничена тупиками. Через равные интервалы расположены стыки, часть которых может быть неисправна.

9.3. Обстановка в среде определяется числом стыков, расстоянием между ними, положением неисправных стыков и текущим положением ОБХОДЧИКА. Обстановка отображается на экране телевизионного монитора (см. ниже).

Начальная обстановка может генерироваться случайным образом или запрашиваться из специального файла, организованного преподавателем.

- 9.4. Исполнитель имеет следующие внутренние имена:
ВПЕРЕДИ (возможные значения: "тупик", "свободно", "занято")
СЗАДИ (возможные значения те же);
СТЫК (возможные значения "исправен", "неисправен").

Значение последнего имени определяется состоянием стыка, на котором остановился ОБХОДЧИК. Если он остановлен не на стыке, значение этого имени - ПУСТО.

9.5. МПИ исполнителя ОБХОДЧИК описывается диаграммами 15.

Условия корректности предписаний:

для предписания ВПЕРЕД: ОБХОДЧИК.ВПЕРЕДИ="свободно"

для предписания НАЗАД: ОБХОДЧИК.СЗАДИ="свободно"

для предписания ОТМЕТИТЬ СТЫК: ОБХОДЧИК.СТЫК/=ПУСТО

9.6. Рассмотрим следующую задачу. ОБХОДЧИК установлен на первом стыке линии неизвестной длины (на экране изображено 9 стыков). Расстояние между стыками - 24 метра. Требуется проверить все стыки, отметить неисправные и вернуть ОБХОДЧИКА в исходное положение.

Программа на языке Робик для решения этой задачи может выглядеть так:

ВКЛЮЧИТЬ ОБХОДЧИК;

ЗАПОМНИТЬ ПРОЦЕДУРУ ПРОВЕРКА_СТЫКА;

ЕСЛИ ОБХОДЧИК.СТЫК="неисправен" ТО ОТМЕТИТЬ СТЫК;

ВПЕРЕД 24 М;

ЗАКОНЧИТЬ;

ПОКА ОБХОДЧИК.ВПЕРЕДИ="свободно" ПОВТОРЯТЬ ПРОВЕРКА_СТЫКА(;

ПОКА ОБХОДЧИК.СЗАДИ="Свободно" ПОВТОРЯТЬ НАЗАД 24 М;

ВЫКЛЮЧИТЬ ОБХОДЧИК;

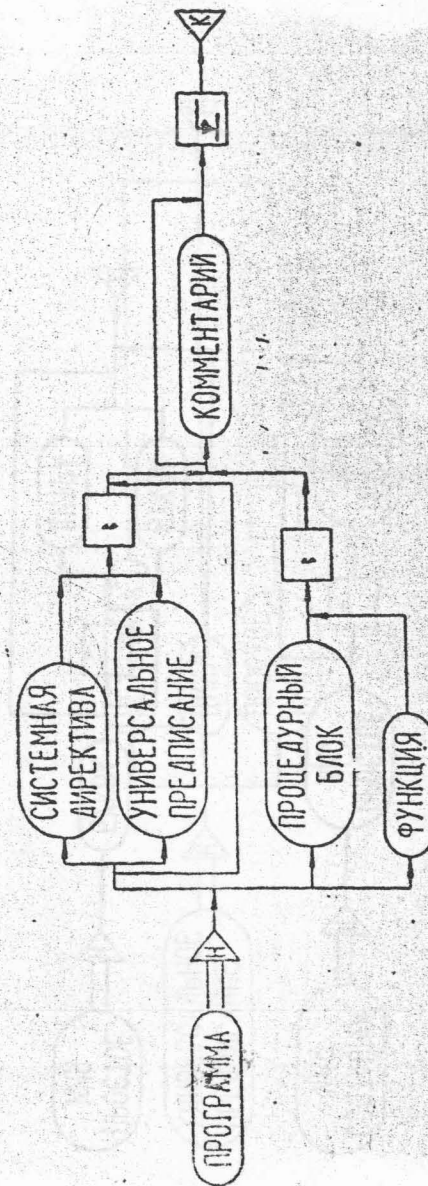


ДИАГРАММА 1.1

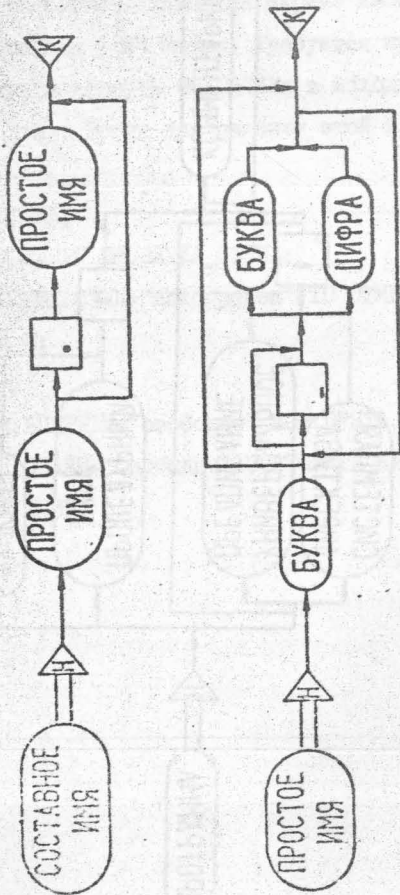


ДИАГРАММА 1.2

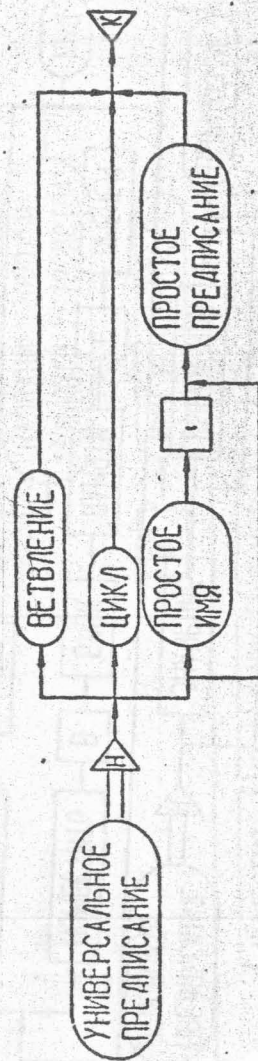


ДИАГРАММА 1.3

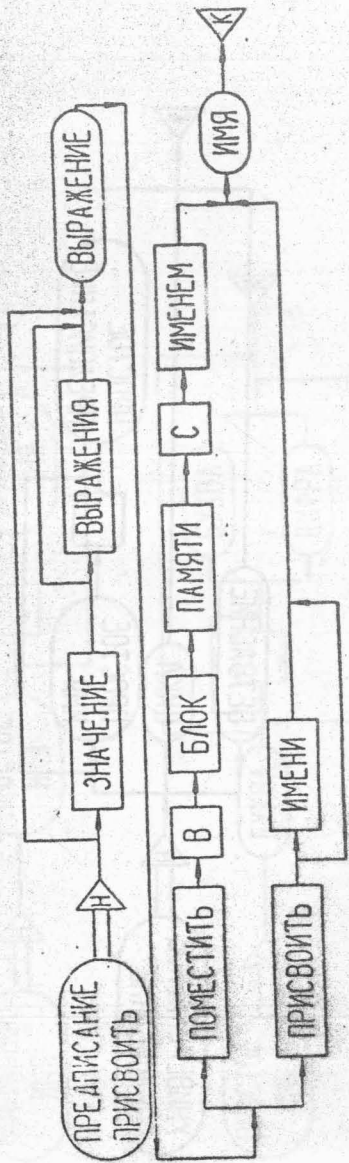


ДИАГРАММА 1.4

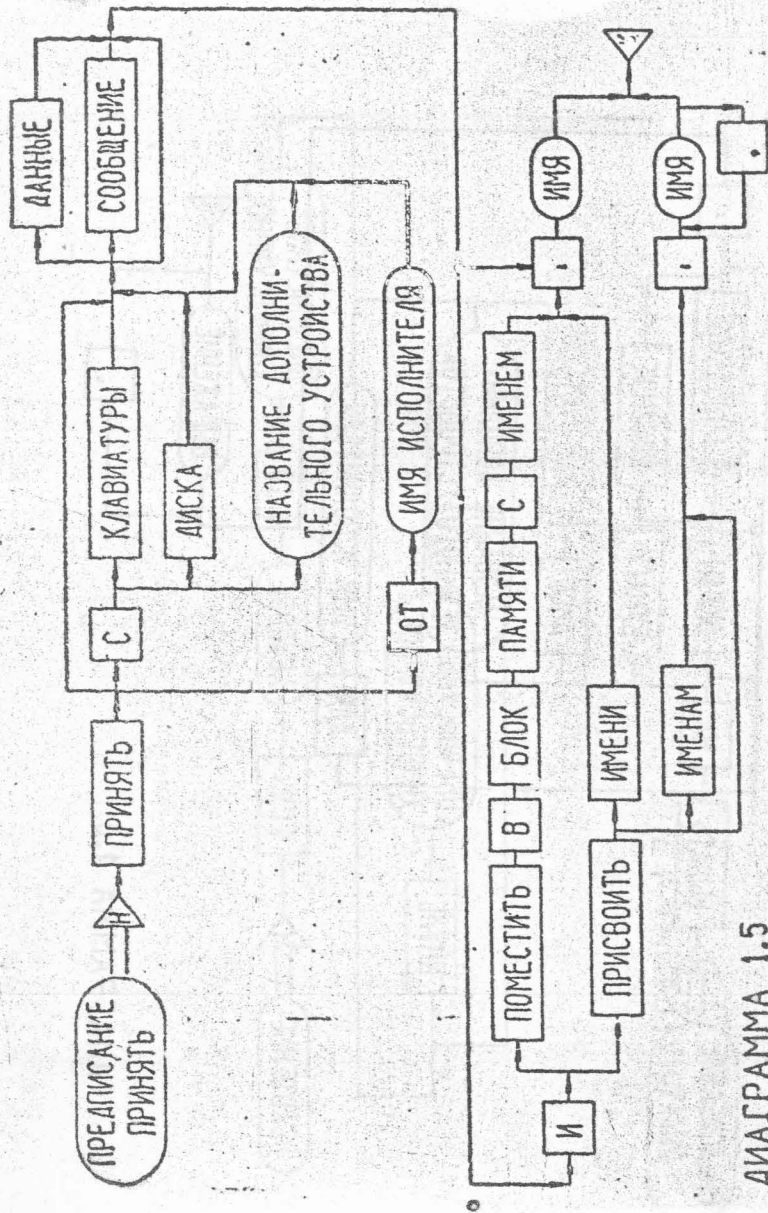


ДИАГРАММА 1.5

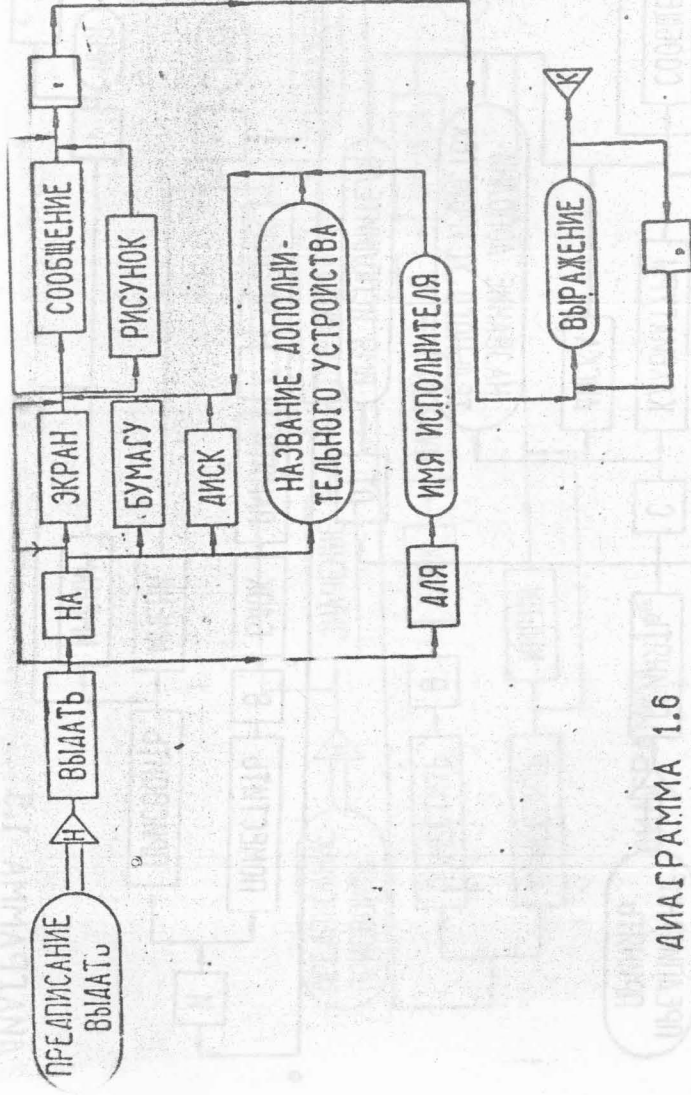


ДИАГРАММА 1.6

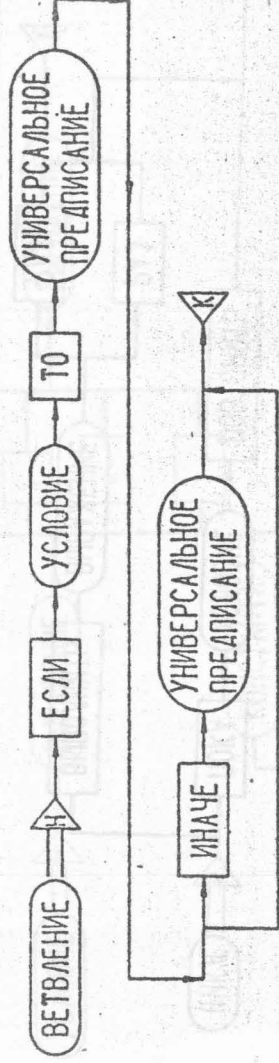


ДИАГРАММА 1.7

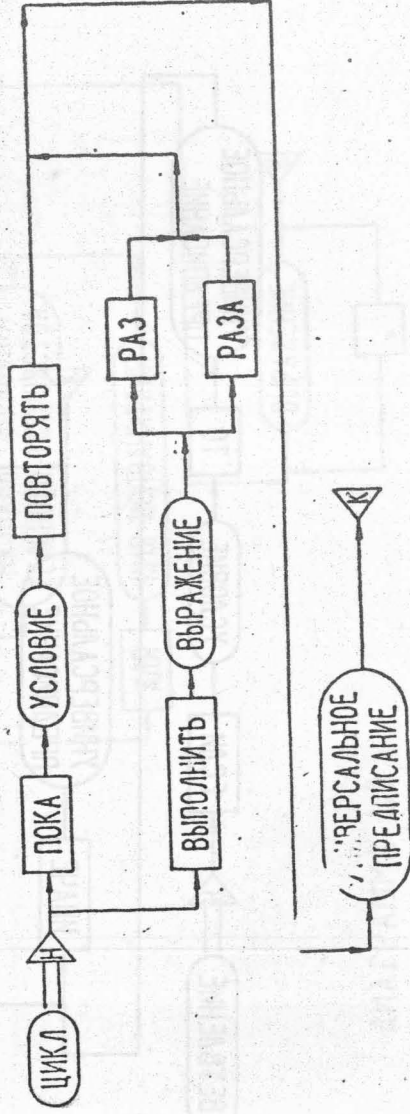


ДИАГРАММА 1.8

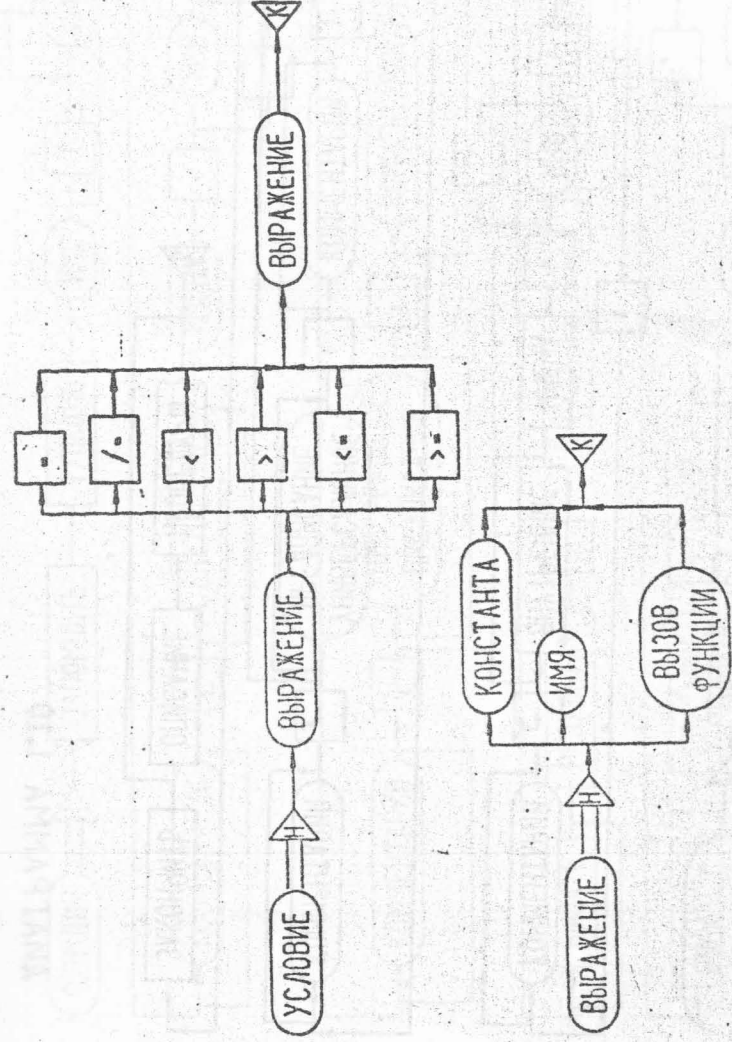


ДИАГРАММА 1.9

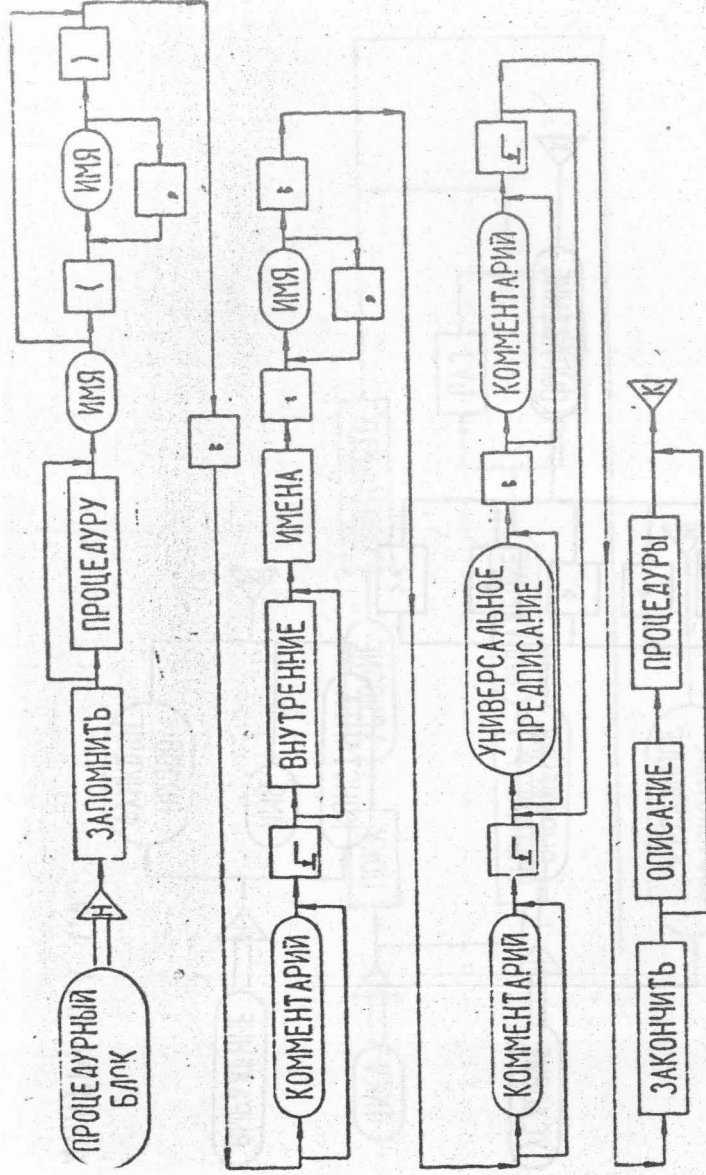


ДИАГРАММА 1.10

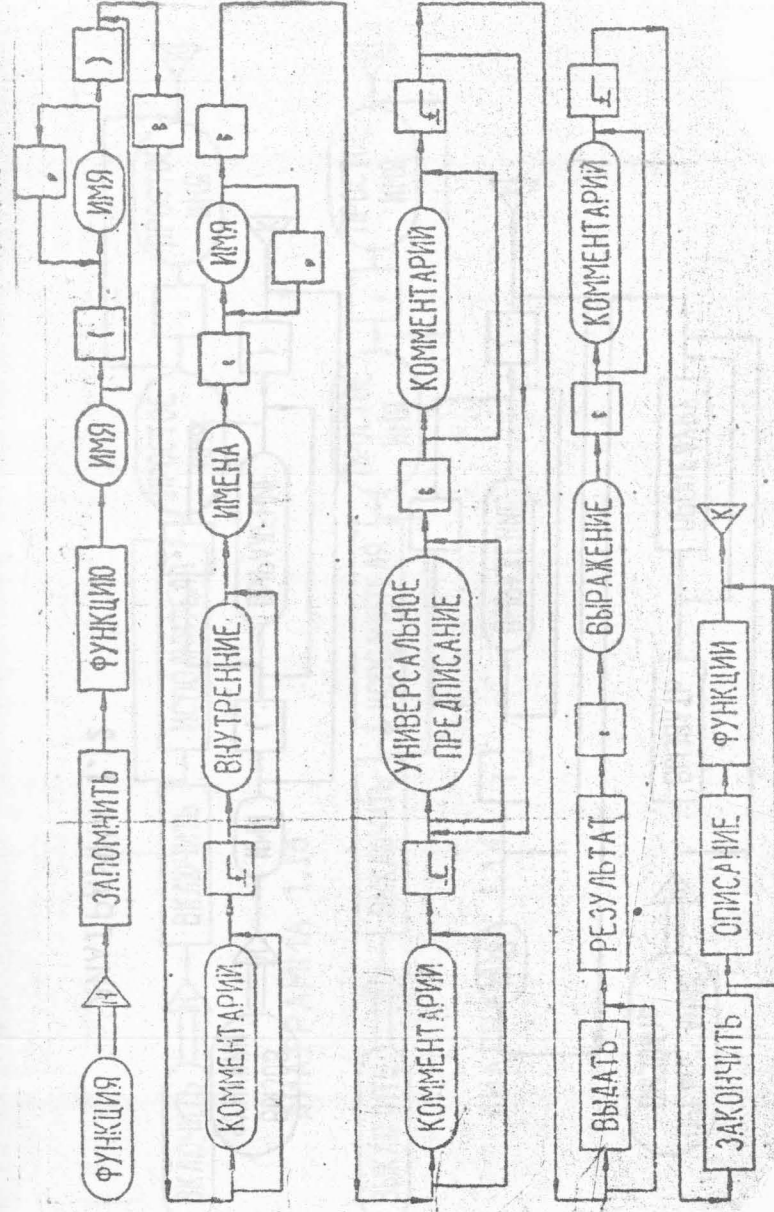


ДИАГРАММА 1.11

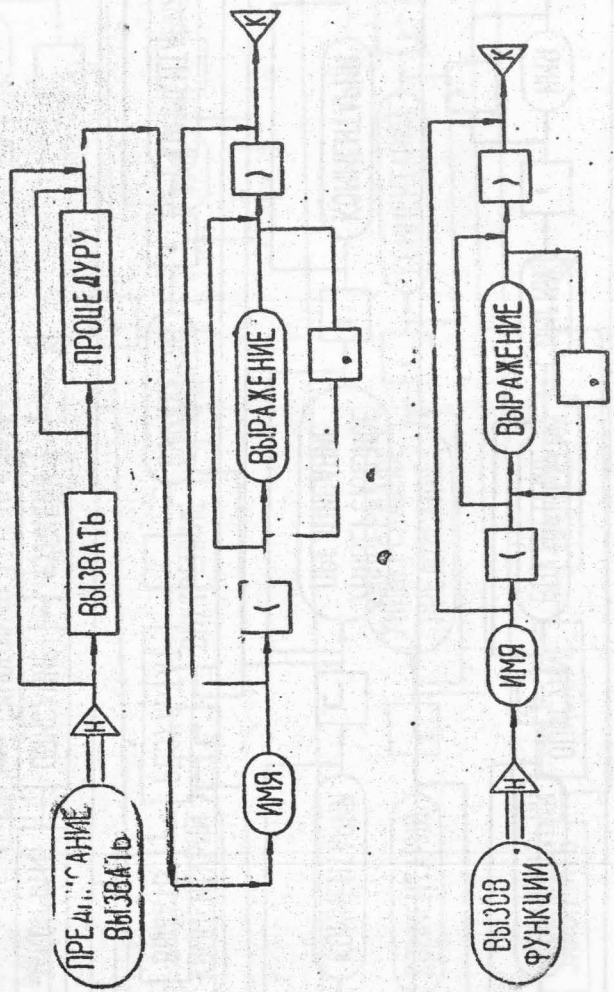


ДИАГРАММА 1.12

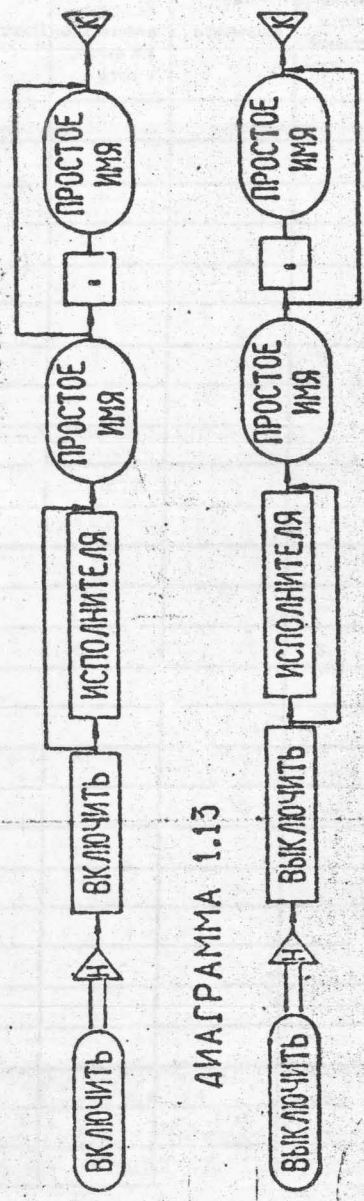


ДИАГРАММА 1.13

ДИАГРАММА 1.14

