

**СИСТЕМА  
МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ  
ДЛЯ ЭВМ «МИНСК-22»**

**Инструкция по пользованию**

Таллин 1970



Научно-исследовательский и проектно-технологический  
институт систем планирования и управления  
в электропромышленности

В.Муст, К.Тинн, Э.Тыгу

СИСТЕМА  
МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ЭВМ  
"МИНСК-22"

Инструкция по пользованию

Таллин 1970

Tartu Ülikooli Raamatukogu  
ARHIIVKOGU

i222880107

## ИІ. ОРГАНИЗАЦИЯ РАБОТ

Общее

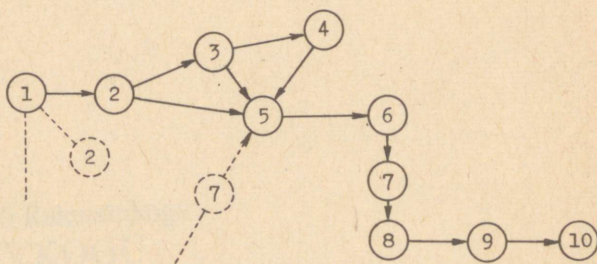
- ИІ.1. Описание проблемы
- ИІ.2. Формулировка задачи
- ИІ.3. Составление алгоритма
- ИІ.4. Программирование и отладка модулей
- ИІ.5. Ввод описаний моделей и подготовка магнитных лент
- ИІ.6. Составление отладочного экземпляра программы
- ИІ.7. Отладка программ
- ИІ.8. Испытание программ
- ИІ.9. Составление блоков загрузки
- ИІ.10. Внедрение

Использование ЦВМ для автоматизации инженерных или экономических расчетов требует, помимо описания алгоритма и составления программы, организации и координирования совместной работы различных специалистов. Эти проблемы могут быть успешно решены, если применяемая система программирования содержит некоторую операционную систему использования ЦВМ, а также средства и методы описания алгоритмов.

Ниже описывается организация работ для системы СМП. Проводимые работы классифицируются следующим образом:

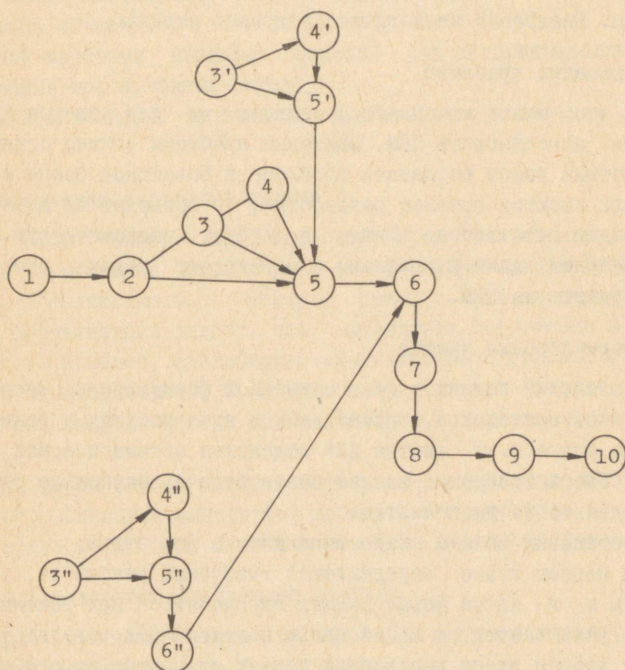
- 1) описание проблемы;
- 2) формулировка задач;
- 3) составление алгоритма задачи;
- 4) программирование и отладка модулей;
- 5) ввод описаний моделей и подготовка магнитных лент моделей;
- 6) составление отладочного экземпляра программы;
- 7) отладка программ;
- 8) испытание программы;
- 9) составление блоков загрузки;
- 10) внедрение.

Порядок выполнения работ можно характеризовать следующим сетевым графиком:



Действительные сетевые графики являются более сложными, так как за общей проблемой следует несколько задач (на схеме указаны пунктиром), при этом результаты программирования одной задачи могут использоваться для последующих и т.д. Кроме того, на каждом этапе выявляются ошибки в алгоритмах, в программах, увеличивающие общий объем работ на нескольких этапах. Поэтому, помимо основного сетевого графика, следует составить сетевые графики с исправлениями, если ошибка вызывает изменение содержания работ более чем на одном предыдущем этапе.

Ниже приведен сетевой график с двумя исправлениями для устранения ошибок в алгоритмах.



На каждом этапе должны заполняться документы о состоянии работ, позволяющие ввести коррективы в план сетевого графика, т.е. определить текущий объем на каждом этапе работ и выявить участки, задерживающие успешное выполнение работ по плану.

Приведенная классификация работ аналогична работам по проектированию и изготовлению изделий на заводах.

Действительно, описание проблемы и постановка задач соответствует техническому заданию на проектирование.

Алгоритмизация задач является разработкой технического проекта, программирование до отладки – рабочим проектированием, отладка и испытание программ – экспериментальным изготовлением и испытанием, составление блоков загрузки программ – изменениями к рабочим чертежам. Внедрение аналогично внедрению изделия.

### III.1. Описание проблемы

На этом этапе выясняется, следует ли для решения какой-либо проблемы использовать ЦВМ. Описание проблемы должно содержать характеристики работ по данной проблеме в Советском Союзе и за рубежом. Если имеются готовые разработки, то приводят их характеристики и стадию готовности. Затем исследуют экономическую эффективность решения данной проблемы и определяют задачи, которые требуется решить на ЦВМ.

### III.2. Формулировка задачи

Постановка задачи в математической формулировке несколько отличается от постановок, приводимых в аванпроектах и эскизных проектах на разработку систем или подсистем автоматической обработки данных. Здесь требуется конкретность задачи, насколько это возможно определить на данном этапе.

Постановку задачи можно выполнить в два этапа.

На первом этапе определяется требуемый результат, т.е. какие величины и в какой форме должны определяться при решении задачи, а также указывается, с какой целью получены эти результаты.

На втором этапе постановки задачи стараются строго определить задачу. Этот этап может быть выполнен лишь тогда, когда имеется общий план решения задачи. На этом этапе следует составить схему

упрощенной модели, описывающую решаемую задачу. Для этого определяются:

- 1) исходные данные;
- 2) справочные данные (постоянные массивы).

Требуемые результаты определены раньше. Составленная схема модели должна указывать, каким образом эти величины связаны между собой.

Если имеются предположения о методах решения задачи или ее отдельных частей, то указывают соответствующую литературу.

Для исходных данных, справочных данных и результатов желательно определить их приблизительный объем. Исходные данные и результаты могут быть представлены в таком виде, как они получены при исследовании объекта.

Данный материал является основой для алгоритмизации и программирования поставленных задач.

Постановка задачи выполняется специалистом в области решаемой задачи.

### И1.3. Составление алгоритма задачи

Этот этап является основным перед программированием. От корректности выполнения этого этапа зависит успех дальнейшей работы. Очень часто специалисты в области анализа систем недооценивают важность формализации задачи, что затрудняет выполнение и организацию работ. Конечно, квалифицированные программисты могут решать этот этап самостоятельно, т.е. сами разрабатывать алгоритм решения поставленной задачи. В этом случае автором алгоритма является программист.

В СМП форма записи алгоритмов задается правилами записи алгоритмов (И2). Описание алгоритмов должно быть оформлено так, чтобы этот материал можно было включать в отчеты без переоформления.

### И1.4. Программирование и отладка модулей

Модули программируются согласно И3. Отладка каждого модуля производится отдельно.

Таблицы и форматы не отлаживаются, а проверяются лишь по контрольной печати, получаемой при трансляции. Программы отлаживаются

на контрольных задачах. При этом контрольная задача содержит не только исходные данные, но и имитацию работы тех модулей, которые используются данным модулем. При отладке программ применяются, как правило, отладочные стандартные программы Прокрутка, Блокировка, Сечение и др. Синтаксические ошибки в описаниях модулей обнаруживает транслятор.

На этом этапе важно вести учет работ по программированию по нижеприведенной таблице.

Запрограммированные модули оформляются и прилагаются к описанию алгоритма. При этом для каждого модуля оформляется:

- 1) аннотация программы с указанием автора программы;
- 2) описание работы программы;
- 3) инструкция использования программы;
- 4) текст программы;
- 5) текст контрольной задачи;
- 6) текст результатов контрольной задачи;
- 7) перфолента программы.

Первые три пункта, а также 5-й не выполняются в том случае, если данный модуль является форматом или структурной таблицей. Пункты 2-й и 3-й не выполняются в случае простых модулей. Текст программы печатается на АЦПУ, а результаты контрольной задачи прилагаются в таком виде, как они печатаются при решении контрольной задачи. Весь материал должен быть оформлен таким образом, чтобы его можно было включить в отчеты без переоформления.

#### И1.5. Ввод описаний моделей и подготовка магнитных лент

Модели описываются согласно И3.7.

При подготовке магнитных лент руководствуются инструкцией И5. Минимальная длина ленты 12 зон.

Описания моделей и модулей следует хранить не менее, чем на двух магнитных лентах и точно учитывать все изменения, вводимые в описания.

Название задачи:  
 Автор алгоритма:  
 Ведущий программист:  
 Тема:

№ п.п.	Предъявление алгоритма						Программиров- ание				Отладка				Изменения				Модуль состав- ляется заново
	Отношение или модуль	Тип	Куда входит	Автор	Дата	Программист	Начало програм- мирова- ния	Конек- програм- мирова- ния	Состав- ление задачи	Перфо- карты модуля	Перфорация новой за- дачи	Отладка	Количество машинного времени (в часах)	Дата окончания задачи	Объем модуля (в командах)	Обнаружена ошибка	Устранена ошибка	Модуль состав- ляется заново	
1																			
2	Отношение или модуль	Тип	Куда входит	Автор	Дата	Программист	Начало програм- мирова- ния	Конек- програм- мирова- ния	Состав- ление задачи	Перфо- карты модуля	Перфорация новой за- дачи	Отладка	Количество машинного времени (в часах)	Дата окончания задачи	Объем модуля (в командах)	Обнаружена ошибка	Устранена ошибка	Модуль состав- ляется заново	
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			

## II.6. Составление отладочного экземпляра программы

СМП работает в динамическом режиме распределения памяти. Для отладки же необходимо знать распределения памяти и вводить временные исправления для устранения ошибок. Эти временные исправления не могут быть внесены в модули или в описания моделей сразу. Поэтому, чтобы не вводить их каждый раз снова, надо составить отладочный экземпляр программы, то есть распределить память автоматически, записать содержимое МОЗУ на магнитную ленту, а затем продолжать работу до тех пор, пока распределение памяти изменится. Новое распределение памяти также записывается на МЛ. В результате получим отладочный экземпляр программы. При отладке вместо распределения памяти происходит чтение с магнитной ленты.

## II.7. Отладка программ

Для отладки программ предусматривается несколько уровней отладки:

- 1) полная прокрутка программы;
- 2) печать выполняемых макрокоманд;
- 3) печать названий и значений входных переменных;
- 4) печать названий и значений выходных переменных;
- 5) печать распределения памяти.

Все названия печатаются в текстовом виде, а значения переменных — в восьмеричной системе, так как тип переменной не указывает, является ли содержимое ячейки кодом, числом с фиксированной запятой и т.д. Форма печати показана на примерах.

При отладке возможны два режима:

- 1) отладка в ходе работы системы;
- 2) отладка на отладочном экземпляре программы.

Отладка в этих режимах требует различной организации работ. В первом случае обнаруженные ошибки должны быть исправлены в модулях до следующей отладки. Во втором случае исправления временно вносятся в отладочный экземпляр программы, а модули исправляются независимо от отладки. Если объем задачи большой, следует применять отладку на отладочном экземпляре. Последняя может быть эффективна в случае, если потом все исправления учитываются. Учет ошибок производится в следующей форме:

№ п.п.	Идентификатор модуля с ошибкой	Кем обнаружена	Когда	Исправлено в отла- дочном экземпляре (номер ячейки) содержание	В чем заключается ошибка (для исправ- ления алгоритма)	Примечание

### ИІ.8. Испытание программы

Испытание программы заключается в решении ряда реальных задач. По результатам проверки оценивается качество данной разработки, а также приемлемость к внедрению. На этом этапе появляется ряд необходимых изменений. Эти исправления вносятся в программы и алгоритмы. Результаты проверки оформляются протоколом испытаний.

### ИІ.9. Составление блоков загрузки

В СМП содержатся различные отладочные режимы, которые облегчают отладку, но снижают скорость решения задач. Составление блоков загрузки программ предусмотрено для увеличения быстродействия работы системы. На этом этапе происходит объединение модулей и исключение отладочных режимов. Полученные программы выводятся на перфоратор и при внедрении поблочно вводятся в ЦВМ.

### ИІ.10. Внедрение

Внедрение происходит при участии авторов алгоритма и ведущего программиста.

## И2. ОПИСАНИЕ АЛГОРИТМОВ

Общее

- И2.1. Содержание описания алгоритма
- И2.2. Аннотация
- И2.3. Блок-схема
- И2.4. Исходные данные и результаты
- И2.5. Описание модели
- И2.6. Схема модели
- И2.7. Описание переменных
- И2.8. Описание отношений

Настоящая инструкция определяет форму записи алгоритмов инженерных и экономических задач, принятую в СМП.

Программирование и отладка алгоритмов, записанных по настоящей инструкции, производится программистами без участия автора алгоритма. Участие последнего необходимо только в том случае, если обнаруживаются ошибки в алгоритме.

Основным средством описания структуры алгоритмов являются блок-схемы. Каждый блок в блок-схеме должен быть описан подробно в самой блок-схеме или в прилагаемой к блок-схеме модели. Последняя содержит сведения о переменных и отношениях между переменными, используемыми в алгоритме.

Каждому алгоритму соответствует одна или несколько моделей, на которые ссылаются в блок-схеме алгоритма. Описываемый алгоритм рассматривается как управляющая программа, которая задает очередность выполнения операторов, реализующих заданную модель отношения. Некоторые действия над данными могут описываться непосредственно в блок-схеме, а простые алгоритмы могут описываться без ссылки на модель. В таком случае описание модели отсутствует.

## И2.1. Содержание описания алгоритма

В описание алгоритма входят:

- аннотация;
- блок-схема алгоритма;
- описание исходных данных и результатов алгоритма;
- описание модели (или моделей).

## И2.2. Аннотация

Аннотация должна содержать название алгоритма, перечень исходных данных и результатов названия моделей, на которые ссылаются в блок-схеме, назначение алгоритма, описание метода вычислений или содержания алгоритма, сведения об авторе, номер темы и библиографические ссылки на использованные публикации.

## И2.3. Блок-схема

Блок-схема состоит из блоков основных действий и разветвлений. Блоки основных действий обозначаются прямоугольниками, а бло-

ки разветвлений - прямоугольниками с закругленными вершинами. На блок-схеме должны быть указаны слова НАЧАЛО, КОНЕЦ и каждый внутренний ОСТАНОВ (с указанием причины останова).

И2.3.1. Основные действия бывают следующие:

а) использование отношения, для которого указан его идентификатор и идентификатор модели;

б) ВЕЛГОЛ - программа (на ВЕЛГОЛ-3), содержащая только операторы присваивания, циклического присваивания и описания; в этой программе доступны все простые переменные из описываемого алгоритма и из моделей, перечисленных в аннотации алгоритма, вместо идентификатора  $x$  переменной модели  $M$  в этом случае пишется  $x * M$ , то есть указывается модель, в которой переменная описана;

в) использование стандартных программ БСП "Минск-22" с указанием их входных и выходных переменных;

г) использование экстракодов для работы с массивами ввода и вывода.

И2.3.2. Разветвления расходятся только на две ветви по результату проверки условия вида  $a = b$ ,  $a \neq b$ ,  $a \leq b$ ,  $a \geq b$ ,  $a < b$  или  $a > b$ , где  $a, b$  - простые переменные.

И2.4. Исходные данные и результаты

Исходные данные и результаты алгоритма являются переменными. При включении алгоритма в качестве отношения в некоторую модель они становятся внешними переменными отношения и включаются в множество переменных модели. Исходные данные и результаты алгоритма должны быть описаны точно так же, как переменные модели (см. И2.7).

И2.5. Описание модели

В описание модели входят:

- схема модели;
- описания переменных;
- описания отношений.

## И2.6. Схема модели

Схема модели состоит из точек, обозначающих переменные, кружочков, обозначающих отношения, и линий, соединяющих переменные с теми отношениями, которые их связывают.

На схеме указываются идентификаторы обозначенных там переменных и отношений. Можно указывать стрелками, какие переменные являются для отношения входными и какие выходными.

## И2.7. Описание переменных

Для каждой переменной, указанной на схеме модели, задается идентификатор,  
длина,  
тип.

В некоторых случаях задается дополнительно область изменения переменной, информация для ввода-вывода и пример.

И2.7.1. Идентификатором может быть любое сочетание не более 5 букв и цифр, начинающееся с буквы. В пределах одной модели идентификаторы всех переменных должны различаться. При этом идентификаторы переменных в других моделях (даже в тех, которые используются как описания отношений в данной модели) роли не играют.

И2.7.2. Длиной переменной является количество ячеек, занимаемых одним значением переменной, или количество символов в значении переменной.

И2.7.3. Переменные могут иметь один из следующих стандартных типов:

- целый;
- рациональный,
- логический,
- код (последовательность из 37 двоичных символов).

Переменные со стандартным типом занимают по одной ячейке памяти. Для их описания не требуется дополнительных сведений (примеров, описания области изменения и т.п.).

Кроме стандартных типов, допускается использование переменных типа

файл,  
запись,  
текст и т.д.

Нестандартный тип переменной может описываться средствами одного из алгоритмических языков ПЛ/1, ФОРТРАН, АЛГОЛ-68.

Нестандартный тип переменной может быть описан также определением области изменения переменной вместе с указанием машинного представления всех значений переменной. В этом случае необходимы примеры значений.

## И2.8. Описание отношений

Описание отношений начинается с заголовка, который состоит из идентификатора отношения и перечня всех идентификаторов переменных, связанных данным отношением.

Если переменная является всегда исходной величиной для данного отношения, то в заголовке перед ее идентификатором ставится знак "+", если результатом, то "-". Возможны случаи без знака "+", "-" и "±".

За заголовком следует собственно описание отношения в одном из следующих видов:

а) описание алгоритма, оформленное согласно настоящим правилам, без аннотации и описания исходных данных и результатов, с указанием используемых моделей;

б) структурная таблица;

в) формат вместе с примером;

г) отлаженная программа с описанием контрольной задачи и результатами решения последней:

в АКМ,

в "ВЕЛГОЛ-3",

в ССК,

в машинном коде;

д) уравнение;

е) отношение, описанное ранее в той же самой или в какой-либо другой модели;

- ж) модель, описанная ранее;
- з) подпрограмма на языке ФОРТРАН ИФВЭ.

В описаниях по пп. "е" и "ж" указывается, какая переменная из описанной ранее модели соответствует каждой внешней переменной описываемого отношения, а также задается название ранее описанной модели.

## ИЗ. Программирование

Общее

- ИЗ.1. Модули
- ИЗ.2. Переменные
- ИЗ.3. Программы
- ИЗ.4. Уравнения
- ИЗ.5. Таблицы
- ИЗ.6. Форматы
- ИЗ.7. Модели
- ИЗ.8. Вычисления на моделях
- ИЗ.9. Экстракоды
- ИЗ.10. Ограничения на модули

Этой инструкцией определяются правила записи текстов модулей СМП, передаваемых далее на перфорацию. Исходным материалом для записи модулей являются только описания алгоритмов, оформленные согласно И2.

При описании некоторых синтаксических структур в данной инструкции используются следующие обозначения, заимствованные из описания языка PL/I:

1. Название синтаксической единицы берется в угловые скобки "<" и ">", например, <идентификатор>

2. Фигурные скобки "{" и "}", содержащие несколько расположенных столбиком конструкций языка, означают, что надо сделать выбор одной из них.

Пример:

$$\left\{ \begin{array}{l} \text{СИМВ} \\ \text{ДЕС} \end{array} \right\}$$

3. Квадратные скобки "[" и "]" означают необязательное вхождение (возможное отсутствие) взятой в скобки конструкции.

4. Три точки "..." означают возможное повторение непосредственно предшествующей синтаксической единицы или содержимого фигурных или квадратных скобок.

Пример:

$$\left\{ \begin{array}{l} \langle \text{цифра} \rangle \\ \langle \text{буква} \rangle \end{array} \right\} \dots$$

### ИЗ.1. Модули

Машинными описаниями отношений и управляющих программ (см. И2) являются модули. Программирование в СМП является процессом описания модулей на входных языках системы. Каждый модуль программируется отдельно и в большой мере независимо от других.

Модули зависят друг от друга только: 1) по согласованию типов внешних переменных (ИЗ.2.2) и 2) по названиям модельных переменных (ИЗ.2.3).

### ИЗ.1.1. Типы модулей:

программа,  
уравнение,  
формат,  
таблица,  
модель.

### ИЗ.2. Переменные

Способ представления значений переменных и занимаемая переменной память ограничиваются в СМП только возможностями машины "Минск-22", то есть в СМП нет фиксированного набора типов переменных. Но в СМП существует несколько точно определенных способов адресации и способов распределения памяти для переменных.

По области существования переменные делятся на 3 класса:

внутренние,  
внешние,  
модельные.

ИЗ.2.1. Внутренние переменные локализованы в модуле и используются согласно правилам языка, на котором модуль описывается.

ИЗ.2.2. Через внешние переменные осуществляется информационная связь модуля с другими модулями. Непосредственно перед вычислением по модулю внешним переменным присваиваются значения и после выполнения модуля эти значения передаются в модель, где данный модуль используется.

Во время вычисления по модулю его внешние переменные расположены следующим образом:

1) простые переменные (ИЗ.2.4) - в рабочем поле (ячейки с 0100 по 0175<sub>(8)</sub>);

2) простые массивы (ИЗ.2.4) - в поле переменных вне модуля, в I-й блоке МОЗУ, причем адрес простого массива расположен в рабочем поле;

3) файлы (ИЗ.2.4) - на МЛ, причем паспорт файла расположен в начале файла, а адрес адреса паспорта задан в рабочем поле в виде:

$$100 + i) 0000 0000 \text{ а,}$$

где  $a) + \frac{n-1}{12} \frac{pq1}{24}$  дв. разряда

$n$  - длина паспорта;

$p$  - номер механизма МЛ;

$q1$  - адрес начала паспорта на МЛ.

Внешние переменные или их адреса расположены в рабочем поле плотно друг за другом в таком порядке, в каком они перечислены при описании модуля.

ИЗ.2.3. Модельные переменные описываются в модуле типа модель (ИЗ.7) и используются при описании вычислений на модели (ИЗ.8).

При описании модели устанавливается соответствие между модельными переменными и внешними переменными модулей (см. ИЗ.7), а при описании вычислений на модели - между модельными переменными и внутренними переменными модуля (см. ИЗ.8).

Типы внешних или внутренних переменных должны согласоваться с типами соответствующих им модельных переменных. При этом определяющим является тип модельной переменной, задаваемый в описании алгоритма (И2).

ИЗ.2.4. Внешние и модельные переменные делятся по способу адресации и распределению памяти на 3 вида:

1) простые переменные - занимают не более 3 ячеек памяти, имеют прямую адресацию;

2) простые массивы - занимают от 4 до 511 ячеек памяти, имеют косвенную адресацию и располагаются в I-м блоке МОЗУ.

Адрес простого массива, который занимает  $n$  ячеек и расположен с адреса  $a$ , задается в ячейке рабочего поля в виде:

$$\underbrace{+ n - 1}_{13} \quad \underbrace{0000}_{12} \quad \underbrace{a}_{12} \quad \text{дв. разряда}$$

3) файлы (ленточные массивы) - занимают более 511 ячеек памяти, располагаются на МЛ. При работе с файлом в МОЗУ выделяется для файла страница памяти. Сведения о файле хранятся в паспорте файла. Паспорт файла имеет косвенную адресацию, причем адрес паспорта задается так же, как адрес простого массива.

Паспорт файла имеет вид:

a+0)	+03					название файла
a+1)	}					описание состояния файла для работы с экстракодом
a+2)						
a+3)						
a+4)						
a+5)	+00	$m_0$	$p_0$	$q_0$	$l_0$	
a+6)	+00	$m_1$	$p_1$	$q_1$	$l_1$	
.						
.						
.						
a+5+k)	+00	$m_k$	$p_k$	$q_k$	$l_k$	
a+6+k)	+0000	0000	$m$			

где  $p_i$  - номер ЛПМ;  
 $q_i l_i$  - адрес, с которого расположен файл на ЛПМ с номером  $p_i$ ;  
 $m_i$  - количество зон, занимаемых файлом на ЛПМ с номером  $p_i$ ;  
 $m$  - количество ячеек, занимаемых одной записью (документом) в файле;  
 $k < 7$ .

### И.3.3. Программы

Модуль типа программа описывается на языке символического кодирования (ЯСК) согласно инструкции системы символического кодирования "Минск-22".

ИЗ.3.1. Структура программы на бланках ССК следующая:

010/ + <заглавие>	}	начальные операторы
020//НАЧ/7000 В		
030/ ВХØД/ П/ 177В, ТПП	}	описание внешних переменных
040/ ВНИ/ ЗНАЧ/ <адрес> .....		
.../ ВНК/ ЗНАЧ/ <адрес> .....	}	переадресуемая часть программы - операторы ССК
.../ ТПП/ ØСТ/		
.....	}	ячейка возврата
...//КТРМ/ <название модуля>		
...// КØН / ВХØД	}	непереадресуемая часть программы
	}	конечные операторы

ИЗ.3.2. В описании внешних переменных <адрес> первой внешней переменной равен 100В. Следующим внешним переменным присваиваются адреса так, чтобы они (или их адреса вида  $n-1\ 0000\ a$ ) располагались в рабочем поле плотно друг за другом.

ИЗ.3.3. В переадресуемой части программы перерабатываются при загрузке все адреса  $a$ , если  $7000_{(8)} \leq a \leq 7777_{(8)}$ . При этом адрес  $a$  заменяется на  $a - 7000_{(8)} + a_n$ , где  $a_n$  - адрес, начиная с которого загружается модуль. Всем переработанным адресам присваивается признак 2-го блока МОЗУ. Из-за этого, в частности, нельзя в переадресуемую часть программы записывать операторы ЫЛТЛ, ЫБТЛ, ЫШПГИ.

ИЗ.3.4. Непереадресуемая часть программы содержит константы, не зависящие от расположения модуля, и память внутренних переменных модуля.

ИЗ.3.5. Внутренние переменные модуля могут располагаться в буферной памяти (ячейки 3000<sub>(8)</sub> - 3777<sub>(8)</sub>). Но при этом необходимо учитывать то, что при выполнении системных команд (ИЗ.8) содержимое буфера не сохраняется.

ИЗ.3.6. При рекурсивном использовании модуля в пределах одного экземпляра модели сохраняется только содержимое ячеек 0001, 0002, 0003 и 0177<sub>(8)</sub>.

#### ИЗ.4. Уравнения

Модуль типа уравнение выражает отношение  $f(x_1, x_2, \dots, x_k)=0$ ;  $k \leq 17$ ;  $x_1, x_2, \dots, x_k$  - внешние переменные модуля. Внешние переменные модуля типа уравнение должны быть простыми переменными, значение каждой из них занимает одну ячейку и представлено в форме числа с плавающей запятой.

Модуль типа уравнение записывается на ЯСК в виде программы, вычисляющей значение  $f(x_1, x_2, \dots, x_k)$  с плавающей запятой и присваивающей вычисленное значение переменной-Ф.

Структура модуля следующая:

010/ + <заглавие>	}	начальные операторы
020// НАЧ/ 7000 В		
030/ ВХ0Д/ П/ 177В, ТПП		
040/ Х1/ ЗНАЧ/ 100 В	}	описание внешних переменных и переменной Ф
050/ Х2/ ЗНАЧ/ 101 В		
.....	}	переадресуемая часть модуля - операторы ССК
.../ ХК/ ЗНАЧ/ 100В + к - 1		
.../ Ф/ ЗНАЧ/ 100В + к	}	ячейка возврата
.....		
// П / Ф, 0	}	непереадресуемая часть программы
// ТПП/ 0СТ		
.....	}	конечные операторы
...// КТРМ/ <название модуля>		
...// К0Н / ВХ0Д		

Замечания, сделанные в И3.3 относительно описания внешних переменных (И3.3.2), переадресуемой части (И3.3.3) и непереадресуемой части (И3.3.4), остаются в силе и для уравнений.

### И3.5. Таблицы

Модуль типа таблица описывает структурную таблицу, которая при описании алгоритма записывается на специальном бланке (см. И2.4 и [27]).

Текст модуля типа таблица записывается в 5 частях, которые соответствуют следующим частям структурной таблицы:

- I - массив значений, выбираемых из таблицы,
- II - условия выборки строк,
- III - правила выборки строк,
- IV - условия выборки столбцов,
- V - правила выборки столбцов.

И3.5.1. I часть таблицы (I подтаблица) записывается в следующем виде:

(ТАБЛИЦА: < N<sub>M</sub> >/СЖАТИЕ: < ШК >/  
 ТИП: < q >/СТРОК: < m >/ СТОЛБЦОВ: < n > /  
 ПРЕОБРАЗОВАНИЕ: < α<sub>10</sub> >/< α<sub>8</sub> >/  
 (< z<sub>11</sub> >/< z<sub>12</sub> >/.../< z<sub>1n</sub> >/)  
 .....  
 (< z<sub>m1</sub> >/< z<sub>m2</sub> >/.../< z<sub>mn</sub> >/))

где N<sub>M</sub> - название модуля;

< ШК > - указывает наличие шкалы (см. [2] I-4.7, I-4.8) и может иметь следующие значения:

< ШК > =  $\begin{cases} = 0, & \text{если значения основной таблицы запоминаются без шкалы,} \\ = 1, & \text{если значения основной таблицы запоминаются со шкалой;} \end{cases}$

< q > - тип, который указывает, сколько ячеек занимает одно значение в основной таблице (z).

$q$  может иметь значения 1; 2; 3; 4; 5; 6; 0,3 и 0,5.  
( $\langle q \rangle = 0,3$ , если значение  $z$  занимает 1/3 ячейки памяти, т.е. в одну ячейку записываются три значения  $z$ ).

$\langle m \rangle$  - число строк основного массива таблицы (в десятичной системе)  $m \leq 50$ ;

$\langle n \rangle$  - число столбцов основного массива таблицы (в десятичной системе)  $n \leq 36$ ;

$\langle \alpha_{10} \rangle$  - восьмеричный код (до 12 символов), который показывает по столбцам необходимость преобразования значений  $z$  из десятичной системы с запятой в число с плавающей запятой.  $\alpha_{10}$  определяет машинное слово длиной 36 двоичных разрядов, у которого в  $j$ -м (слева) разряде 1, если необходимо преобразовать значения  $j$ -го столбца  $z_{ij}$  ( $i = 1, 2, 3, \dots, m$ );

$\langle \alpha_8 \rangle$  - восьмеричный код (до 12 символов), который аналогично  $\alpha_{10}$  показывает необходимость преобразования значений  $z$  из кода M2 в восьмеричную (двоичную) систему с фиксированной запятой;

$\langle z_{mm} \rangle$  - значения основной таблицы.

Примечания: 1) Все числа (значения  $z$ ), находящиеся в одном столбце основной таблицы, должны быть записаны в одной системе исчисления, но разные столбцы таблицы могут быть в разных системах (в десятичной или восьмеричной системах, или в коде M2).

2) Если в основной таблице отсутствует какое-нибудь значение  $z_{ij}$ , т.е. какая-нибудь графа таблицы пустая, то имеется 2 возможности:

а) на месте пустой графы пишут "-0",

б) пустую графу (пустые графы) пропускают, но в таком случае перед следующим значением в данной строке надо записать номер этого столбца  $j$  и апостроф (').

Пример:

(ТАБЛИЦА: ПРИМ1/СЖАТИЕ: 0/ ТИП: 1/

СТРОК: 4/СТОЛБЦОВ:3/ПРЕОБРАЗОВАНИЕ: 1/2/

(40 000/ 30 500/ 109 066/)

(12 500/ - 0/ 28 400/)

(10 000/ 7600/ 36 000/)

(ТЫСЯЧА/ 3' 7500/ )

И.3.5.2. П часть (П подтаблица) - переменные и условия выборки строк записываются следующим образом:

(СТРОКИ/

(< x > / < условие > / < признак > / < значение > / )

.....  
.....

(< x<sub>ℓ</sub> > / < условие > / < признак > / < значение > / ))

где: < x > - название переменной (до 6 знаков),

ℓ - общее число условий выборки строки (до 36),

< условие > - может быть

РА - равно " = ",

МР - меньше или равно " ≤ ",

НР - не равно " ≠ ",

ПМ - принадлежит к массиву " ∈ ",

НМ - не принадлежит к массиву " ∉ ",

< признак > - признак типа значения, может быть Т, Д или В,

где: Т - значение переменной в телеграфном коде М2,

Д - значение переменной является числом десятичной системы,

В - значение переменной является числом восьмеричной системы.

Соответственно признаку < значение > будет

для Д - до 9 десятичных символов, включая запятую со знаком или без знака;

для В - до 12 восьмеричных символов со знаком или без знака;

для Т - до 6 символов кода М2.

Пример:

(СТРОКИ/

(ШКОЛА/ РА/ Т/ СРЕДН/)

(ШКОЛА/ РА/ Т/ ТЕХНИК/)

(ШКОЛА/ РА/ Т/ ВЫСШ/)

(ШКОЛА/ РА/ Т/ ТРУДОВ/)

(РАЙОН/ РА/ Т/ ЦЕНТР/)

(РАЙОН/ РА/ Т/ МОРСК/))

ИЗ.5.3. Ш часть (Ш подтаблица) - восьмеричные коды условий выборки строк - записываются в следующем виде:

(( $\langle \alpha_1 \rangle /$ )  
( $\langle \alpha_2 \rangle /$ )  
( $\langle \alpha_3 \rangle /$ )  
.....  
( $\langle \alpha_m \rangle /$ )),

причем коды для выборки строк  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_m$  получаются следующим образом: любой квадрат поля в Ш части таблицы можно рассматривать как двоичную цифру - пустой квадрат "0" и квадрат с крестиком - "1". Эти двоичные цифры группируют в каждой строке по три. Каждая полученная строка и представляет собой один восьмеричный код.

Пример:

(( 42/) (22/) (12/) (05/)).

Примечание. "0" в начале кода условий выборки надо обязательно писать. Коды 05 и 5 имеют различное содержание.

ИЗ.5.4. IU часть (IU подтаблица) - переменные и условия выборки столбцов - записываются так же, как подтаблица П, только вместо слова "СТРОКИ" записывается слово "СТОЛБЦЫ".

Пример:

(СТОЛБЦЫ/  
(ПРИЕМ/ МР/ Д/ 1970/  
(ПРИЕМ/ МР/ Д/ 1974/  
(ТИП/ РА/ Т/ СТАЦ/  
(ТИП/ РА/ Т/ ЗАОЧН/))

ИЗ.5.5. У часть (У подтаблица) - восьмеричные коды для выборки столбцов - записываются аналогично подтаблице Ш.

Пример:

(( 5/) (44/) (34/))

Примечание. "0" в конце кода условий выборки не надо записывать (значит, коды 5 и 50 имеют одинаковое содержание).

ИЗ.5.6. Если условия выборки строки (столбца) отличаются друг от друга только по значению переменной, то П (IУ) часть таблицы можно записывать и следующим образом:

```
((<x>/<условие>/<признак>/<значение> )  
(4' <значение >/)  
.....  
.....)
```

Пример:

```
П подтаблица:  
(СТРОКИ/  
(ШКОЛА/ РА/ Т/ СРЕДН/)  
(4' ТЕХНИК/)  
(4' ВЫСШ/)  
(4' ТРУДОВ/)  
(РАЙОН/ РА/ Т/ ЦЕНТР/)  
(4' МОРСК /))
```

```
IУ подтаблица:  
(СТОЛБЦЫ/  
(ПРИЕМ/ МР/ Д/ 1970/)  
(4' 1974/)  
(ТИП/ РА/ Т/ СТАЦ/)  
(4' ЗАОЧН/))
```

Полная запись модуля типа таблица:

(ТАБЛИЦА:ПРИМ1/СЖ:0/ТИП:1/СТР:4/СТ:3/1/2/  
(40 000/ 30 500/ 109 066/) (12 500/ -0/  
28 400/) (10 000/ 7600/ 36 000/) (ТЫСЯЧА/3, 7500/))  
(СТРОКИ/ (ШКОЛА/РА/Т/СРЕДН/)(4, ТЕХНИК/  
(4, ВЫСШ/) (4, ТРУДОВ/) (РАЙОН/ РА/ Т/  
ЦЕНТР/) (4, МОРСК/))  
(( 42/) (22/) (12/) (05/)) и т.д.

Если условия выборки строк (столбцов) содержат только одну переменную и условие "=", "≤" или "ε", и если коды выборки строки (столбца) в двоичной системе (на поле Ш (У) подтаблицы) образуют диагональ матрицы (в том случае общее число условий выборки строки (столбца) равняется числу строк (столбцов)), то Ш (У) подтаблицу можно не записывать вообще.

Если выборка строк (столбцов) происходит без условий, т.е. номер строки (столбца) задается при выборке, то П и Ш (1У и У) подтаблицы отсутствуют.

### ИЗ.6. Форматы

Модуль типа формат записывается в виде:

(ФОРМАТ  $\lfloor$  <название формата>  $\lfloor$  <название документа>  $\lfloor$   
РЕКВИЗИТЫ: (<назв. реквизита>  $\lfloor$  <тип реквизита>  $\lfloor$  <длина>  $\lfloor$  )  
.....  
(<назв. реквизита>  $\lfloor$  <тип реквизита>  $\lfloor$  <длина>  $\lfloor$  )

<Название документа> и <название реквизита> - идентификаторы, обозначающие внешние переменные модуля (формата).

<тип реквизита> может быть

- ДЕС - десятичный,
- ВОСЬМ - восьмеричный,
- СИМВ - символьный,
- ДВ - двоичный.

< длина реквизита > - количество символов или цифр указанного типа в значении реквизита. Реквизит с длиной  $\ell$  занимает память

- 4  $\ell$  битов при типе реквизита ДЕС,
- 3  $\ell$  битов при типе реквизита ВОСЬМ,
- 6  $\ell$  битов при типе реквизита СИМВ,
- $\ell$  битов при типе реквизита ДВ.

Числа с плавающей запятой или с фиксированной запятой, занимающие точно одну ячейку, должны описываться как ДВ длиной  $\ell = 36$ .

### ИЗ.7. Модели

В модуле типа модель описываются модельные переменные и отношения между ними. Для каждого отношения указывается модуль, который используется при вычислениях по данному отношению.

Модель - единственный модуль, который может описываться и транслироваться по частям. Будем говорить, что модель задается одним или несколькими описаниями.

ИЗ.7.1. Первое описание модели имеет вид:

$$\left. \begin{array}{l} \text{НОВЫЙ} \\ \text{НОВАЯ} \\ \text{НОВОЕ} \end{array} \right\} \text{ } \_ \text{ } \langle \text{название модели} \rangle \_ \_ \langle \text{шифр МЛ} \rangle \_ \_ \text{ } \\ \text{[ } \langle \text{продолжение описания} \rangle \_ \_ \text{ ]}$$

Следующие описания имеют вид:

$$\begin{array}{l} \text{(ПРОДОЛЖЕНИЕ } \_ \_ \langle \text{название модели} \rangle \_ \_ \langle \text{шифр МЛ} \rangle \_ \_ \\ \text{ } \langle \text{продолжение описания} \rangle \_ \_ \text{ )} \\ \text{ } \langle \text{название модели} \rangle \text{ } \text{ и } \langle \text{шифр МЛ} \rangle \text{ - идентификаторы.} \end{array}$$

Название присваивается модели в ее первом описании, шифр присваивается МЛ при ее подготовке для архива СМП (см. И5).

Пример:

$$\text{(НОВАЯ } \_ \_ \text{СХЕМА } \_ \_ \text{Ш11 } \_ \_ \text{)}$$

< продолжение описания > содержит описания модельных переменных и отношений, а также изменения, требующие удаления из модели ра-

нее описанных переменных или отношений. Описания переменных, описания отношений и описания удалений могут чередоваться, т.е. <продолжение описания> имеет вид:

$$\left\{ \begin{array}{l} \langle \text{описание переменных} \rangle \\ \langle \text{описание отношения} \rangle \\ \langle \text{описание удаления} \rangle \end{array} \right\} \dots$$

ИЗ.7.2. Каждое описание переменных записывается в виде

$$\left( \begin{array}{l} \text{ПЕР} \\ \text{ПЕРЕМЕННАЯ} \\ \text{ПЕРЕМЕННЫЕ} \end{array} \right) \_ \left[ \langle \text{идентификатор} \rangle \_ \langle \text{длина} \_ \rangle \dots \right)$$

В одном описании можно задать до 25 модельных переменных. Каждая переменная задается идентификатором, за которым следует <длина> - число ячеек, занимаемых значением переменной.

Пример:

(ПЕРЕМЕННЫЕ X Y IO N 2)

ИЗ.7.3. Отношение описывается в виде:

$$\left( \text{ОТНОШЕНИЕ} \_ \langle \text{идентификатор} \rangle \_ \left\{ \begin{array}{l} \text{МОДЕЛЬ} \\ \text{ТАБЛИЦА} \\ \text{ФОРМАТ} \\ \text{ССК} \\ \text{УРАВНЕНИЕ} \end{array} \right\} \_ \right.$$

$\left. \langle \text{название модуля} \rangle \_ \langle \text{степень функциональности} \rangle \_ \right)$

$$\left[ \langle \text{название переменной} \rangle \_ \left\{ \begin{array}{l} \text{ВХ} \\ \text{ВЫХ} \\ \text{СИЛ} \\ \text{СЛ} \end{array} \right\} \_ \right.$$

$$\left\{ \begin{array}{l} \langle : \rangle \\ \langle \text{название переменной} \rangle \end{array} \right\} \_ \dots )$$

<идентификатор> является названием описываемого отношения.



дели. При удалении отношений и переменных сперва следует удалить отношения, а затем - переменные.

Примеры описаний модели:

(НОВЫЙ\_МОТОР\_МЕХ\_

(ПЕР\_ЗАКАЗ\_К\_КУДА\_СРОК\_ВЕС\_

МОЩН\_ПАСП\_25\_МАРШ\_ЦЕХ\_)

(ОТНОШЕНИЕ\_СПЕЦ\_ФОРМАТ\_Ф\_

ПАСП\_СЛ\_:

ВЕС\_СЛ\_:

МОЩН\_СЛ\_:)

(ОТНОШЕНИЕ\_ТЕХНО\_МОДЕЛЬ\_М\_

ЦЕХ\_ВХ\_ЦЕХ\_

ПАСП\_ВХ\_Х\_

МРШР\_ВЫХ\_У\_))

### ИЗ.8. Вычисления на моделях

Программы в СМП могут содержать операторы обращения к модели, в которых описываются вычисления на моделях. Модели, содержащиеся в архиве СМП, образуют среду, в которой работает каждая программа-модуль. До сих пор рассматривалась связь модулей со средой, осуществляемая через внешние переменные модуля. Собственно, внешние переменные модуля можно рассматривать как исходные данные и результаты вычислений по модулю.

Модуль, содержащий операторы обращения к модели, связан со средой еще через эти операторы. В последних можно ссылаться на любые модели, а также на отношения и модельные переменные любых моделей, имеющихся в архиве СМП. Выполнение оператора обращения к модели инициирует вычисления на других модулях, описывающих отношения в разных моделях. Возможно и рекурсивное обращение к тому же самому модулю, в котором записан оператор.

В операторах обращения к модели описываются:

- 1) вычислительные операторы, выполняемые на моделях;
- 2) передача данных из программы в модель и обратно;

3) управление средой - выбор модели, на которой вычисления выполняются.

ИЗ.8.1. Вычислительные операторы на моделях описываются макрокомандами, которые имеют вид:

$$(\langle \text{код операции} \rangle \_ \left[ \left\{ \frac{+}{-} \right\} \_ \langle \text{название переменной} \rangle \_ \right] \dots)$$

$\langle \text{Код операции} \rangle$  - либо название отношения, либо экстракод. Экстракоды обозначают встроенные в систему операторы, выполняемые над модельными переменными (ИЗ.9).

Символ "+" означает, что переменная является исходной величиной для оператора, и "-" означает, что переменная является выходной величиной (результатом) оператора. "С" означает, что переменная является как исходной величиной, так и результатом, а "Л" отмечает переменную, которая при выполнении оператора не используется.

Если кодом операции является название отношения, то в макрокоманде записываются отношениями связанные переменные в таком порядке, в каком они записаны при описании отношения.

Пример:

$$(AI \_ + \_ X \_ - \_ Y \_ - \_ \text{НАПР} \_ + \_ \text{ТОК} \_)$$

Здесь предполагается, что в описании отношения AI переменные перечислены в следующей последовательности: X, Y, НАПР, ТОК.

Последовательность макрокоманд образует макропрограмму - описание алгоритма вычислений на модели. В конце макропрограммы записываются символы "(\\_)"

Пример макропрограммы:

$$(AI \_ + \_ X \_ - \_ Y \_)$$
$$(A3 \_ + \_ Y \_ - \_ V \_)$$
$$(AB \_ + \_ X \_ + \_ Y \_ + \_ V \_ - \_ C \_)$$
$$(\_)$$

Если у отношений, по которым производятся вычисления, нет слабосвязанных переменных, то допускается вместо макропрограммы

писать последовательность названий отношений, называемую вычислительной схемой.

Пример:

(A1 A3 AB)

является вычислительной схемой, описывающей тот же самый алгоритм, который описан в предыдущем примере макропрограммой.

ИЗ.8.2. Передача данных из программы в модель и обратно задается списком переменных и списком этикеток, являющихся частями оператора обращения к модели.

Список переменных имеет форму

$\left\{ \begin{array}{l} \text{ВХ} \\ \text{ВЫХ} \end{array} \right\} \left[ \langle \text{название модельной переменной} \rangle \right] \dots$

В начале выполнения оператора модельными переменными, перед которым стоит ВХ, присваиваются значения некоторых внутренних переменных программы. В конце выполнения оператора значения модельных переменных, отмеченных словом ВЫХ, передаются в программу.

Список этикеток задается в виде псевдоконстант ССК. В каждой псевдоконстанте по три этикетки. Этикетки обозначают переменные программы, значения которых передаются в модель или считываются из модели. Между модельными переменными, записанными в список переменных оператора обращения к модели, и внутренними переменными программы устанавливается соответствие по порядку их записи.

ИЗ.8.3. Управление средой производится словами ПРВ, ПРД, ЗМК, которые определяют, какая модель вызывается в оперативную память для выполнения оператора обращения к модели и какая остается там после его завершения. Модель, находящаяся в оперативной памяти, непосредственно доступна для выполнения вычислений, на ее отношения и переменные можно ссылаться в макрокомандах и вычислительной схеме. Такая модель называется активной.

ПРВ означает, что в оперативную память вызывается новый экземпляр модели, названной в операторе обращения к модели. После выполнения оператора остается активной эта же модель.

ПРД не меняет распределения памяти, т.е. активной остается прежний экземпляр в последний раз в сочетании с ПРВ названной модели.

ЗМК делает на время выполнения оператора обращения к модели активным новый экземпляр модели, названной в этом операторе. После выполнения оператора активной становится прежняя модель.

ИЗ.8.4. Оператор обращения к модели состоит из 2 частей: тела оператора и текста алгоритма, которые могут записываться раздельно, в разных местах символической программы. В некоторых случаях текст алгоритма отсутствует.

Тело оператора имеет вид:

(КОМ ПРД <название модели> <макрокоманда> )

или

(	}	ВЫП	}	{	ПРВ	}	< название модели > _
		ПР			ПРД		
		ОПС			ЗМК		
		ЗАП					
		ВЫТР					
		ОПТР					

< список переменных > [ < список этикеток > ]

[ < этикетка алгоритма > ]

Та часть тела оператора, которая взята в скобки "(" и ")", записывается на бланке ССК в виде текстовых констант (КТРМ). Остальная часть тела оператора – этикетки, которые записываются на бланке ССК в виде псевдоконстант.

Первое слово в теле оператора определяет тип оператора. Название модели указывает, на какой модели оператор выполняется. Остальные части тела оператора рассмотрены в ИЗ.8.1 и ИЗ.8.3.

Текст алгоритма является либо макропрограммой, либо вычислительной схемой (ИЗ.8.1). Текст алгоритма снабжается этикеткой

и записывается на бланке ССК в виде текстовых констант (КТРМ). Этикета текста алгоритма указывается также в теле оператора.

Рассмотрим теперь назначение разных типов операторов обращения к модели.

КОМ означает, что выполняется макрокоманда, указанная в теле оператора. Текст алгоритма отсутствует.

ВЫП означает, что на указанной модели решается задача, заключающаяся в вычислении значений выходных переменных по значениям входных переменных. (Выходные и входные переменные отмечены в списке переменных словами ВЫХ и ВХ). Текст алгоритма отсутствует. Сведения, необходимые для решения задачи, должны содержаться в модели.

ПР - выполняется макропрограмма, являющаяся текстом алгоритма.

ОПС - составляется и выполняется макропрограмма, соответствующая вычислительной схеме в тексте алгоритма.

ЗАП - происходит только передача данных из программы в модель и из модели в программу. Текст алгоритма отсутствует.

ВЫТР и ОПТР выполняются аналогично ВЫП и ОПС. Разница только в том, что для ВЫТР и ОПТР макропрограмма составляется во время трансляции, а для ВЫП и ОПС - во время выполнения программы.

ИЗ.8.5. Примеры записи тела оператора обращения к модели. X1, X2, ... - переменные модели; A1, A2, ... - названия отношений; P1, P2, ... - внутренние переменные программы; M - название модели; МПР - этикетка описания алгоритма.

```
010// КТРМ / (ВЫП П
020/// РВ М В
030/// X X1 В
040/// ВХ X2
050///)
060// ПСК / P1, P2, 0
010// КТРМ / (ПР ПР
020/// Д М )
030// ПСК / МПР, 0, 0
```

010// КТРМ/ (ПР\_ЭМ  
 020/// К\_М\_ВХ  
 030/// \_ХI\_ВЫ  
 040/// Х\_Х2\_В  
 050/// Х\_Х3\_)  
 060// ПСК / P1, P2, P3  
 070/// МПР, 0, 0

### ИЗ.9. Экстракоды

Экстракоды используются для работы с файлами, редактирования и печати текстов и для присвоения значений одним модельных переменных другим.

ИЗ.9.1. Работа с файлом должна начинаться с открытия файла и кончатся закрытием файла. Это выполняется экстракодами

(ОТКР\_[С\_<название файла>\_]...)

и

(ЗАКР\_[С\_<название файла>\_]...)

Примеры:

(ОТКР\_С\_МI\_С\_КАРТЫ\_С\_ТЕМА\_)

(ЗАКР\_С\_МI\_С\_КАРТЫ\_)

Все файлы - последовательного типа с записями фиксированной длины. После открытия файла становится доступной первая запись. Доступная запись может считываться из файла и присваиваться документу в качестве значения экстракодом

(ВЗЯТЬ\_+[+\_<название файла>\_]\_<название документа>] ...)

Пример:

(ВЗЯТЬ\_+\_КАРТЫ\_--КАРТА\_)

В файле можно изменить доступную в данный момент запись экстракодом (ЗАП\_[+\_<название документа>\_]\_<название файла>\_] ...).

Пример:

(ЗАП+ДЕЛО-ДЕЛА)

записывает значение документа ДЕЛО в файл ДЕЛА.

Следующая запись в файле становится доступной после применения экстракода

(ПОДВ[С<название файла>]...)

Пример:

(ПОДВСДЕЛАСКАРТЫ).

### ИЗ.9.2 Экстракод вида

(ПРИСВ [+<название переменной>]...  
[-<название переменной>]...)

присваивает значения переменных с плюсами переменным с минусами. Присваивание происходит последовательно, ячейка за ячейкой, до тех пор, пока не кончатся значения присваиваемых переменных или не будут присвоены значения всем переменным.

При этом части значения одной переменной из разных ячеек могут быть присвоены разным переменным, а новое значение переменной может состояться из значений разных переменных.

### ИЗ.10. Ограничения на модули

ИЗ.10.1. Машинное описание модуля не должно превышать 1000<sub>(8)</sub> ячеек.

ИЗ.10.2. Количество внешних переменных модуля не должно превышать 17.

ИЗ.10.3. Суммарная длина переменных (не считая файлов), значения которых записываются в модель по одному оператору обращения к модели, не должна превышать 1000<sub>(8)</sub>.

ИЗ.10.4. Суммарная длина переменных (не считая файлов), значения которых передаются из модели в программу по одному оператору обращения к модели, не должна превышать 1000<sub>(8)</sub>.

- ИЗ.10.5. Количество незаконченных операторов обращения к модели (глубина обращения к СМП) не должно превышать 8.
- ИЗ.10.6. Количество переменных в модели не должно превышать 127.
- ИЗ.10.7. Количество отношений в модели не должно превышать 127.
- ИЗ.10.8. Суммарная длина переменных, используемых при вычислениях на одном экземпляре модели (не считая файлов), не должна превышать 2400<sup>(8)</sup>. При этом суммарная длина всех переменных модели не ограничивается, так как память для переменных распределяется динамически.
- ИЗ.10.9. Одновременно не должно быть открыто более 4 файлов.
- ИЗ.10.10. В архиве модулей на одной магнитной ленте может храниться до 255 модулей.

#### И4. ПЕРФОРИРОВАНИЕ

- И4.1. Перффорирование модулей
- И4.2. Перффорирование приказов МОС
- И4.3. Разметка

#### И4.1. Перфорирование модулей

И4.1.1. Модули, подготовленные для перфорации, должны соответствовать требованиям И3.

И4.1.2. Каждый массив телеграфного текста должен заканчиваться 5 пробелами.

И4.1.3. Программы и уравнения перфорируются согласно инструкции по перфорации СИМП.

И4.1.4. К инструкции по перфорации СИМП имеются следующие добавления:

а) в конце оператора перфорируются вместе со знаком "<" (возврат каретки) также знак "≡". (перевод строки); в "заказе" знак "<" должен предшествовать знаку "≡";

б) между графами оператора и в "заказе" перфорируются "/" (вместо знака "≡");

в) в коде операции (графа "КОП") буква "ч" заменена на цифру "4".

И4.1.5. Таблицы перфорируются согласно правилам перфорации структурных таблиц.

И4.1.6. Описание модели перфорируется в один или несколько массивов. В начале каждого массива должно быть слово **НОВЫЙ**, **НОВАЯ**, **НОВОЕ** или **ПРОДОЛЖЕНИЕ**.

И4.1.7. Формат перфорируется в один массив.

И4.1.8. При перфорации форматов и описании моделей для исправления возникающих ошибок можно использовать знак ":".

Знак ":" гасит все предшествующие символы до знака "┘".

И4.1.9. Таблицы, форматы и описания моделей перфорируются отдельными массивами на одну катушку перфоленты. При этом последним массивом на катушке должен быть пустой массив.

## И4.2. Перфори́рование приказов МОС

И4.2.1. Приказы (для малой операционной системы) перфорируются на одну перфоленту. Между приказами оставить 5 см пустой ленты ("рус"). Каждый приказ заканчивается тремя символами "лат. регистр" подряд. Перфорирование комбинации "начало массива" (начальная граница) необязательно.

И4.2.2. При перфорации приказов возникающие ошибки можно исправить знаком "X". Каждый знак "X" гасит один предыдущий символ.

## И4.3. Разметка

И4.3.1. На каждую отперфорированную перфоленту наносится фамилия перфорирующего, дата и шифр перфоленты, указанной заказе.

## И5. ОПЕРАЦИОННАЯ СИСТЕМА

Общее

И5.1. Программы МОС

И5.2. МЛ

И5.3. Действия оператора

И5.4. Приказы МОС

И5.5. Таблицы остановов

И5.6. Сообщения трансляторов

Работы непосредственно на вычислительной машине в СМП выполняются при помощи малой операционной системы МОС. МОС обеспечивает:

- 1) ведение общего архива и архива модулей;
- 2) трансляцию модулей;
- 3) загрузку модулей в МОЗУ;
- 4) выполнение программ во взаимодействии со средой;
- 5) издание текстов модулей и других записей из архивов;
- 6) отладку.

МОС выполняет работы, задаваемые операционной системе п р и к а з а м и. Приказы задаются с телетайпа или стартстопного ввода последовательно по одному. После выполнения каждого приказа машина выдает сообщение об этом, печатая на телетайпе слово ГОТОВО.

#### И5.1. Программы МОС

С точки зрения оператора МОС делится на следующие программы:

1. Программа вызова резидента.
2. Резидент.
3. Монитор.
4. Отладчик.
5. Транслятор символических команд.
6. Транслятор СМП.
7. Прораб СМП.

Ниже приводятся сведения о программах, необходимые для вызова программ МОС и определения состояния МОС в ходе выполнения приказов.

И5.1.1. Программа вызова резидента вводится с ПЛ и состоит из следующих команд:

0030) -4700 1001 0000  
0031) -4520 0400 0000

0032) -3000 0030 0000

0033) -0000 0000 0000

0034) +3357 6346 7775

Программа вызывает резидента с МЛ в МОЗУ.

Пуск с I0000(8).

нв: Перед пуском убедиться, что в механизм IO запровлена МЛ "СМП".

Программа вызова резидента должна перед каждым ее выполнением вводиться с ПЛ.

И5.1.2. Резидент хранится на МЛ "СМП", занимает оперативную память с I0000(8) по I0377(8). При пуске с команды I0000(8) резидент вызывает в МОЗУ монитор, печатает на телетайпе слово ГОТОВО и готовит МОС к выполнению приказа. Резидент сохраняется в МОЗУ при совместной работе с монитором, отладчиком и прорабом.

И5.1.3. Монитор используется при работе с телетайпом или стартовым вводом ПЛ в начале выполнения каждого приказа.

И5.1.4. Отладчик используется при выполнении приказов отладки.

И5.1.5. Транслятор символических команд используется при работе с общим архивом.

И5.1.6. Транслятор СМП используется при работе с архивом модулей.

И5.1.7. Прорабу СМП подчиняются все обслуживающие программы СМП (не МОС!), кроме транслятора СМП. Прораб может присутствовать в МОЗУ при выполнении приказов трансляции модулей, загрузки модулей и выполнения программ.

## И5.2. Магнитные ленты

При работе с МОС МЛ распределяются следующим образом:

№ мех.	Длина МЛ (зон)	Назначение
00	20(8)	Стандартные программы
01	20(8)	Архив модулей
02	20(8)	Общий архив
10	20(8)	Обслуживающие программы МОС и СМП
11	30(8)	Рабочая лента СМП

Механизмы 03, 12, 13 и все механизмы с номерами  $\geq 20$  могут использоваться рабочими программами без ограничений.

## И5.3. Действия оператора

Оператор выполняет при помощи вычислительной машины для каждой задачи последовательность действий. Последовательность действий задается оператору на бланке заказа (см. табл.)

И5.3.1. Заказ СМП состоит из трех частей. В первой части указывается название задачи, автор заказа, дата выдачи заказа, исполнитель перфориования заказа, исполнитель работы на машине и машинное время на выполнение заказа.

Во второй части заказа записываются используемые магнитные ленты с указанием механизмов для их установки (коммутации, если механизмы не используются под своими номерами). В этой же части перечисляются все шифры перфолент, устанавливаемых на ФСУ, или стартстопный механизм ввода.

Третья часть содержит описания действий, выполняемых для решения задачи.

Действия бывают следующие: приказы МОС (см. И5.4); установка и снятие магнитных лент, перфолент и перфокарт; включение, выключе-

чение, блокировка и разблокировка устройств вычислительной машины.

Все действия, кроме приказов ЦОС, описываются и выполняются по правилам операторской работы вычислительного центра НИПТИ.

Таблица

I	Заказ СМП	Задача Программист:	Дата:		
Перфорировал:		Выполнил:			
Магнитные ленты:				Перфоленты:	
МЕХ.		НОМЕР			
МЕХ.		НОМЕР			
II	00	20		ФСУ	Стартстопный
	01	21			
	02	22			
	03	23			
	10	30			
	11	31			
	13	33			
III	№	Действие	Перфорация	Примечание	
	1				
	2				
	3				
	4				
	5				
	6				
	7				

В заказе записанные приказы перфорируются на одну перфоленту друг за другом так, что каждый приказ кончается тремя символами ЛАТ, за которыми записаны пять пробелов. На ПЛ перед каждым приказом записывается его номер (номер действия, которому соответствует приказ). ПЛ с приказами шифруются и их шифры записываются во 2-й части бланка заказа.

И5.3.2. Приказы МОС выполняются следующими приемами:

1) Ввод резидента:

а) ввод программы вызова резидента с ПЛ;

б) проверка контрольной суммы ввода

КС = - 7777 7777 7777;

в) пуск с адреса 0030(8).

2) Включение телетайпа и стартстопного механизма.

3) Установка ПЛ с очередными выполняемыми приказами на стартстопный механизм.

4) Пуск резидента с адреса 10000.

После этого МОС печатает на телетайпе слово НАЧАЛО и выполняет последовательно все приказы до первого встретившегося приказа ОТЛАДКА, ВЕРНУТЬ или СТОП. При выполнении каждого приказа на телетайпе печатается текст приказа и после его выполнения слово ГОТОВО, если при выполнении приказа не предусмотрен останов машины.

Каждая последовательность подряд выполняемых приказов должна кончаться приказом ОТЛАДКА, ВЕРНУТЬ или СТОП, так как только по этим приказам происходит останов машины после правильного выполнения работы.

Если при работе машины происходит останов не по приказу ВЕРНУТЬ или СТОП, то следует: 1) действовать по инструкциям, записанным в таблице остановов (И5.5); 2) проверить, не записано ли в заказе дополнительных действий на случай происшедшего останова.

И5.3.3. Выполнение заказа кончается отметками в журнале машины и на бланке заказа и сбором всех входных и выходных носителей информации по данному заказу.

NB! Изменение последовательности выполнения действий или повторное выполнение одного и того же действия приводит к неверному решению задачи и недопустимо.

#### И5.4. Приказы МОС

Приказом МОС является текст, взятый в скобки и начинающийся одним из слов:

ПОДГОТОВИТЬ

СОЗДАТЬ

АРХИВ

ТРАНСЛ\_ССК

ТРАНСЛ\_СМП

ПУСК

ИЗДАТЬ

ОТЛАДКА

НА\_ПРОВЕРКУ

ВЕРНУТЬ

СТОП

Вслед за приказом перфорируется (печатается на телетайпе) три символа "ЛАТ" и не менее пяти пробелов. Это является для монитора сигналом, требующим выполнения приказа. Далее через  $N$ ,  $N_1$ ,  $N_2$ , ... обозначаются целые десятичные числа без знака.

И5.4.1. (ПОДГОТОВИТЬ/ШМЛ/ $N$ /), где ШМЛ - шифр МЛ, запрошенной в механизм ОI, производит подготовку новой (размеченной) МЛ для архива модулей. Архив модулей займет на ленте  $N$  слов,

начиная с адреса 00 0000. В выполнении ПОДГОТОВИТЬ участвуют Резидент, Монитор и Транслятор СМП.

Пример:

(ПОДГОТОВИТЬ/ КАДРЫ/ 24 000/)

И5.4.2. (СОЗДАТЬ/МАС/Н /  $N_1$  / ШМЛ<sub>1</sub>/  $N_2$  /  $N_3$ /), где МАС - идентификатор, ШМЛ<sub>1</sub> - шифр МЛ, запрошенной в механизм с номером  $N_1$ , создает на МЛ с шифром ШМЛ<sub>1</sub> файл. Этот файл получает название МАС и состоит из записей, имеющих длину  $N$  слов, располагается на МЛ, начиная со слова  $N_3$ , и может занимать на ленте до  $N_2$  зон.

Допускается записывать приказ СОЗДАТЬ, в котором указано несколько (до 8) номеров механизмов МЛ и шифров МЛ и для каждой МЛ начальный адрес и количество зон:

(СОЗДАТЬ/МАС/Н /  $N_1$  / ШМЛ<sub>1</sub>/ $N_2$  /  $N_3$  /  
 $N_4$  / ШМЛ<sub>2</sub>/ $N_5$  /  $N_6$  /  
.  
.  
.  
 $N_{3K-2}$  / ШМЛ<sub>K</sub>/ $N_{3K-1}$  /  $N_{3K}$  /)

В этом случае файл создается последовательно на лентах ШМЛ<sub>1</sub>, ШМЛ<sub>2</sub>, ..., ШМЛ<sub>K</sub>.

В выполнении СОЗДАТЬ участвуют Резидент и Монитор.

Пример:

(СОЗДАТЬ/КЛАСС/8/12/Ш11/10/3000/  
13/Ш12/12/1001/)

И5.4.3. (АРХИВ/ МШ//  $N_1$  / М1 ///  $N_2$  / М2 ///...), где МШ - шифр МЛ архива модулей, переводит модули М1, М2, ... из общего архива и записывает в архив модулей.  $N_1 = 1$ , если модуль М1 содержит операторы обращения к модели, в остальных случаях  $N_1 = 0$ , то же относится к  $N_2$  ... .

В выполнении приказа АРХИВ участвуют Резидент, Монитор и Транслятор СМП.

Пример:

(АРХИВ/Ш1// 0 /ТАВ1/// 0 /ТАВ2/// 1/ПРОГ1///)

И5.4.4. (ТРАНСЛ\_ССК/ШМЛ/) выполняет трансляцию символической программы согласно заказу ССК, установленному на ФСУ к началу выполнения этого приказа. Выполняется программами Резидент, Монитор и Транслятор ССК.

И5.4.5. (ТРАНСЛ\_СМП/ШМЛ/), где ШМЛ - шифр МЛ, транслирует все модули с ПЛ, запрошенной на ФСУ, и записывает их в архив модулей.

Выполняется программами Резидент, Монитор и Транслятор СМП.

Пример:

(ТРАНСЛ\_СМП/ СВАРК /)

И5.4.6. (ПУСК / М / N<sub>1</sub> / N<sub>2</sub> /), где М - название модуля, вызывает модуль в МОЗУ и выполняет вычисления по модулю, если модуль не модель.

Значения внешних переменных, являющиеся исходными данными модуля, должны перед выполнением приказа ПУСК находиться в МОЗУ, начиная с ячейки 0200<sub>(8)</sub> в таком порядке, в каком они будут в рабочем поле при выполнении модуля.

Простые массивы должны находиться в МОЗУ, начиная с адреса 276<sub>(8)</sub>.

Если модуль М является моделью, то он записывается в МОЗУ с адреса 11400<sub>(8)</sub>, иначе с адреса 10000<sub>(8)</sub> + N<sub>1</sub>, а если N<sub>1</sub> не задано, то модуль записывается с 17000<sub>(8)</sub>. N<sub>2</sub> указывает общую длину простых массивов, которые являются внешними переменными модуля.

ПУСК выполняется программами Резидент, Монитор и Прораб.

Примеры:

(ПУСК / КАССА /)

(ПУСК / ПОСТ / 6000 / 280 /)

(ПУСК / ПОСТ // 280 /)

И5.4.7. (ИЗДАТЬ / КАТАЛОГ /) печатает на АЦПУ каталог архива модулей.

(ИЗДАТЬ / МОДЕЛЬ /) печатает на АЦПУ таблицу переменных, таблицу отношений и распределение памяти переменных и отношений той модели, которая находится в этот момент в МОЗУ.

(ИЗДАТЬ/М/), где М - название модуля, печатает на АЦПУ и перфорирует на перфораторе текст (машинный код) модуля с названием М. Приказы ИЗДАТЬ выполняются программами Резидент, Монитор и Отладчик.

И5.4.8. (ОТЛАДКА/М/), где М - название модуля, создает среду для отладки модуля и вызывает модуль с названием М в I-й блок МОЗУ с адреса 7000<sub>(8)</sub> (см. [4] - И7).

Выполняется программами Резидент, Монитор и Отладчик.

Пример:

(ОТЛАДКА/МАС/).

Если название модуля отсутствует, то создается только отладочная среда (см. И7).

И5.4.9. (НА\_ПРОВЕРКУ/М/), где М - название модуля, загруженного в МОЗУ, печатает текст этого модуля на АЦПУ.

Выполняется программами Резидент и Монитор.

Пример:

(НА\_ПРОВЕРКУ / ПАПА /)

И5.4.10. (ВЕРНУТЬ/) восстанавливает то содержимое МОЗУ, которое было перед последним пуском резидента с адреса I0000.

И5.4.11. (СТОП/) останавливает работу машины на команде I57I(8), не восстанавливая при этом содержимого МОЗУ.

И5.4.12. В приказах допускаются комментарии, поясняющие назначение параметров приказов.

Комментарием является любой текст, записанный непосредственно после символа "/", кончающийся символом ":" и не содержащий символов "/", ":", "(" и ")".

Примеры приказов с комментариями:

(ПОДГОТОВИТЬ/ЛЕНТУ С ШИФРОМ: ШМЛ/С КОЛИЧЕСТВОМ

СЛОВ: 10 000/)

(СОЗДАТЬ/ФАЙЛ: МАССЫ/С ДЛИНОЙ ЗАПИСЕЙ: 8/ НА

МЕХ.: 20/ЛЕНТА: ЛПИ/ЗОН: 5/

С АДРЕСА: 2000/)

(АРХИВ/НА ЛЕНТЕ: АР1//ТИП ТРАНСЛ.: 1/МОДУЛЬ:ММА///

ТИП ТРАНСЛ.: 0/МОДУЛЬ: КУ ///)

(ТРАНСЛ\_СМП/ НА ЛЕНТУ: ШМЛ/)

(ПУСК/ ЭКРАН/С: 6000/ПАМЯТЬ: 250/)

(НА\_ПРОВЕРКУ/ МФМ/)

(ИЗДАТЬ/ КАТАЛОГ/)

(ИЗДАТЬ/ МОДЕЛЬ/)

(ИЗДАТЬ/МОДУЛЬ: ОПИСЬ/)

(ОТЛАДКА/ МОДУЛЯ: М1/)

(ТРАНСЛ\_ССК/\_ШМЛ/)

(ВЕРНУТЬ/)

(СТОП/)

И5.5. Таблицы остановов

И5.5.1. Остановы программы Монитор

СчАК	Причина	Действие оператора
1	2	3
I535	Незнакомый приказ	Выполнить следующий приказ
I57I	Выполнен приказ СТОП	То же
I627	В механизм заправлена неверная лента (шифр ИЛ в 0041)	Заменить ленту, повторить приказ
I642	Слишком короткая ИЛ в мех. с номером из 0003	Выполнить следующий приказ
I654	Для файла заказано более 8 лент	То же
I674	Название файла уже использовано для другого модуля, который имеется в каталоге	"
I676	Каталог полный	"
I763	Для файла заказан механизм с системными лентами	"
2244	Слишком длинный приказ	"
2663	В системной команде вместо цифр стоят буквы	"
3017	КС считывания системной программе не равна 77...7	Повторить приказ
3075	Нет отношения с заданным названием	Выполнить следующий приказ
3II4	Отношение не введено в МОЗУ	То же
I0027	Выполнен приказ ВЕРНУТЬ	"

## И5.5.2. Остановы программы Отладчик

СчАК	Причина	Действие оператора
1	2	3
10412	При чтении отладчика 064000-067777 с МЛ КΣ неверна	Прекратить работу и выяснить причины изменения информации в отладке (на МЛ № 2II)
10427	При чтении отладчика 060004-067777 с МЛ КΣ неверна	То же
10467	Вызваны отладчик и модуль (если они указывались)	Приступить к отладке
6132	КΣ вызываемого модуля неверна (из архива модулей)	Пуск. Издают модуль с неверной КΣ
6167	Нет такого модуля в архиве модулей	Пуск. Модуля не издают
5035	В памяти нет места для вызываемой СП	Прекратить работу. Отладку необходимо завершить по частям модуля
5072	КΣ при чтении СП неверна	Испорчена МЛ с СП, прекратить работу
6410	В общем архиве нет такого модуля	Выполнить следующий приказ (резидент в МОЗУ)
6424	Запись в архиве не соответствует каталогу	То же
6463	Модуль запрограммирован неверно - адрес машинного слова <7000, или ≥10000 (8)	Выполнить следующий приказ (резидент в МОЗУ)
6563	Испорчен общий архив. КΣ блока МП неверно	То же
6561	Испорчен общий архив (запись не соответствует каталогу)	"

## И5.5.3. Остановы программы Архив

СчАК	Причина	Действие оператора
1	2	3
I7077	Неверная лента с общим архивом	Поставить заказанную МЛ в механизм 02 и ПУСК I7002 или пустить машину с адреса I0000 для выполнения нового приказа
I7101	Испорчен транслятор ССК при СМП	Пуск с I0000 для нового приказа
I7106	В каталоге общего архива идентификатор вводимой СИМП	Выполнить новый приказ (резидент в МОЗУ)
I7110	В общем архиве отсутствует заказанная СИМП	Выполнить новый приказ (резидент в МОЗУ)
I7160	В каталоге общего архива ошибка: начальный адрес больше конечного	То же
I7211	В общем архиве нет места	Выполнить новый приказ (резидента нет в МОЗУ)
I7213	Каталог общего архива заполнен	Выполнить новый приказ (резидент в МОЗУ)
I7315	Неверная работа программы Архив	Выполнить новый приказ (резидента нет в МОЗУ)
I7366	Неверная запись каталога общего архива на МЛ	Пуск с адреса I7354 для повторной записи (резидента нет в МОЗУ)
I7422	Неверная запись в общий архив	Пуск с адреса I7412 для повторной записи (резидента нет в МОЗУ)
I7442	Программа записана более чем в 8 местах	Выполнить новый приказ (резидента нет в МОЗУ)
I7523	В каталоге уже имеется новый идентификатор СИМП	Выполнить новый приказ (резидента нет в МОЗУ)
I7551	Неверный символ в новом идентификаторе СИМП	То же

1	2	3
17553	Новый идентификатор СИМП содержит более 5 символов	Выполнить новый приказ (резидента нет в МОЗУ)
17555	Начальный символ нового идентификатора не буква	То же
17605	В каталоге отсутствует идентификатор при гашении его	"
14010	В каталоге общего архива отсутствует идентификатор (приказ АРХИВ)	Выполнить новый приказ (резидент в МОЗУ)
14024	Испорчен общий архив (запись не соответствует каталогу)	То же
14063	Модуль запрограммирован неверно: адрес машинного слова <7000 (в) или >10000 (в)	"
14071	Длина модуля 0	"
14163	Испорчен общий архив. КΣ блока машинной программы неверная	"
14161	Испорчен общий архив (запись не соответствует каталогу)	"
14240	Испорчен резидент СИМП	Ввести резидент с МЛ и выполнить новый приказ
14264	Шифр архива модулей не соответствует приказу	Выполнить новый приказ (резидент в МОЗУ)
5073	Транслятор системных команд работает неверно	То же
5125	ИТ не поместится при трансляции системных команд	"
5204	Идентификатор содержит более пяти символов (при трансляции системных команд)	"

1	2	3
5256	Отсутствует тип задачи (транслятор системных команд)	Выполнить новый приказ (резидент в МОЗУ)
5277	Тип переменной в ИТ неверный (пер.сист.команд)	То же
5352	Отсутствует операторная схема (транслятор системных команд)	"
5375	Отсутствует псевдопрограмма (транслятор системных команд)	"
5413	Неверный тип переменного в псевдопрограмме (транслятор системных команд)	"
5502	В каталоге моделей отсутствует идентификатор модели (транслятор системных команд)	"
5526	В таблице X отсутствует идентификатор переменной (транслятор системных команд)	"
5576	В таблице A отсутствует идентификатор модуля	"

## И5.5.4. Остановы Транслятора СМП

СчАК	Причина останова	Действие оператора
1	2	3
124	Общая длина модуля больше допустимой (1000 <sub>8</sub> )	Возьми следующий массив и пуск 6406
300	Название изменяемого модуля (в ячейке 0041) отсутствует в массиве КР	То же
1125	Признак типа значения в ячейке 0040 неправильный. Должен быть Т, Д или В в коде М2	Пуск в адреса 6406
1262	Тип таблицы в ячейке 13002 неправильный	То же
1762	Код знака условия выборки строки (столбца) неправильный	"
2054	В ячейке 2210 отсутствует код знака условия выборки строки (столбца)	"
4025	Транслятор ввел пустой массив	"
4032	Вводимый массив не помещается в заданное рабочее поле	"
4437	В программе перевода М2→8 переводимое слово содержит телеграфный знак, отличный от "+", "-", "u", "1", "2", "3", "4", "5", "6", "7", "0". Переводимое слово в ячейках 0061+0063	"
4545	В программе перевода М2→10 переводимое слово в ячейках 0061+0063 содержит недопустимый знак	"

I	2	3
5224	Округляемое слово не содержит кода запятой в двоично-десятичной системе (аргумент в ячейке 0040)	Пуск с адреса 6406
5364	В программе перевода IO,0 → M2 переводимое слово не является десятичным числом с запятой (аргумент в ячейке 0040)	То же
6542	Начало вводимого массива неправильное	"
657I	Длина перерабатываемой части вводимого модуля больше общей длины модуля	"
66I3	Контрольная сумма блока транслятора не совпадает с кодом - 7777 7777 7777	"
666I	Таблица КР заполнена	"

## И5.5.5. Остановы программы Прораб

СчАК	Причина	Действие
I0I30	Испорчена программа Стоп	Пуск для вторичного чтения (резидент в МОБУ)
I0I32	Неверная запись на МЛ. КΣ при контрольном чтении не совпадает	Пуск для повторной записи (резидент в МОБУ)
I6055	Признак причины и причина сокращенно выдается на широкую печать	Выполнить новый приказ (резидент в МОБУ)

## И5.6. Сообщения трансляторов

### И5.6.1. Трансляция системных команд

В машинной программе транслируются все системные команды. Если в системной команде (СК) или в макропрограмме (МК), на которую есть ссылка в СК, встречается ошибка, то печатается информация об ошибке и эта СК (или МК) пропускается. Если при трансляции обнаружены ошибки, то модуль в архив модулей не записывается.

Информация об ошибке печатается на ТБММ в виде трех строк. В первой строке печатается признак ошибки в десятичной системе; во второй строке - адрес программы, где начинается ошибочная СК (в восьмеричной системе). В третьей строке печатается адрес ошибки в восьмеричной системе или порядковый номер слова (символы между двумя "┌") в СК (или в МК) в десятичной системе.

Причины ошибки приведены в таблице:

Признак	Причина
+ 100	В тексте системной команды (СК) или макрокоманде (МК) два символа "("без)" между ними.
+ 101	Неверный конец макропрограммы (в первой ячейке следующего МК нет "(").
+ 102	В СК или МК нет символа ")" (имеется символ "рус").
+ 103	В СК или МК нет символа ")" (при трансляции вышли из пределов программы).
+ 104	Число СК и МК превышает допустимую $(2n_{СК} + n_{МК} > 250)$
+ 105	В СК отсутствует тип команды (ПРВ, ПРД или ЗМК)
+ 107	Текст СК или МК содержит более 102 слов (слово-символы между двумя "┌").

Признак	Причина
+ I20	Массив машинных представлений информационных таблиц (ИТ) при обращении к СМП превышает допустимую величину (510).
+ I21	В рабочем архиве нет модели, указанной в СК.
+ I22	Тип переменной неверный или отсутствует имя переменной (в СК).
+ I24	В модели нет переменной, указанной в СК.
+ I25	Этикетка переменной равняется нулю.
+ I26	Неверная этикетка макропрограммы.
+ I27	См. + I04
+ I30	Массив машинных представлений макропрограмм превышает допустимую величину (510).
+ I31	В модели нет отношения, указанного в МК.
+ I33	В модели нет переменной, указанной в МК.
+ I34	Тип переменной в МК неверный.
+ I40	См. + I04
+ I51	СК вышла из перерабатываемой части программы.
+ I60	Программа после трансляции превышает допустимую величину (512).
+ I61	В программе имеется обращение не в начале МК или СК. При этом печатается адрес команды, сама команда и программа (после трансляции до появления ошибки).

## И5.6.2. Трансляция СМП

Во время работы транслятор выводит на узкую печать сообщения, закодированные в цифровом виде. При установлении ошибки в описании модулей и моделей транслятор, в зависимости от ее характера, либо продолжает трансляцию, принимая некоторое решение, либо производит останов работы, когда дальнейшая трансляция невозможна. Список остановок машины и дальнейшие действия оператора приведены в И5.5.4.

После ввода массива описания модели печатается строка  
+ IIII IIII IIII.

При вводе массива описания формата печатается строка  
+ 2222 2222 2222. После трансляции описания модели или формата печатается строка + 7777 7777 7777.

Сообщения, вызванные ошибками в описаниях, имеют следующий вид:

+ НОЦ		+ НОЦ
+ НФР	или	+ НФР
+ НСЛ		+ НПЕ

где:

- НОЦ - номер ошибки (см. таблицу);
- НФР - номер фразы, где была ошибка;
- НСЛ - номер неправильного слова в фразе;
- НПЕ - номер переменной, которая описана неправильно.

Фраза - это часть описания модели (формата), которая находится в скобках.

Одно сообщение от другого отделяется двумя пропусками строки.

Сообщения транслятора на цифровой печати и реакция транслятора на причину сообщения приведены в таблице:

Номер сообщения	Вид сообщения	Причина сообщения	Реакция транслятора на причину сообщения
1	2	3	4
200	+ НОЩ + НФР + НПЕ	Длина переменной больше допустимой (64.000)	Переменная пропускается
201	+ НОЩ + НФР + НПЕ	В таблице ХР уже было название введенной переменной	Переменная пропускается
202	+ НОЩ + НФР + НПЕ(1)	В таблице ХР нет названия исключенной переменной	Исключение переменной исключается
203	+ НОЩ + НФР + НСЛ(4)	В таблице КР отсутствует название модуля	Отношение пропускается
204	+ НОЩ + НФР + НСЛ(2)	Тип модуля неправильный	Отношение пропускается
205	+ НОЩ + НФР + НСЛ(2)	Исключенная переменная была в массиве	Исключение переменной исключается
206	+ НОЩ + НФР + НПЕ	Тип связи переменной неправильный	Отношение пропускается
207	+ НОЩ + НФР + НПЕ	В таблице ХР отсутствует с отношением связанная переменная	Отношение пропускается
208	+ НОЩ + НФР + 00000000	Таблица STR заполнена	Отношение пропускается

1	2	3	4
209	+ НОЩ + НФР + 000000000	В таблице STR ошибка	Исключение отношения исключается
210	+ НОЩ + НФР + 000000000	Таблица XR заполнена	Переменная пропускается
211	+ НОЩ + НФР + 000000000	Массив AP заполнен	Отношение пропускается
212	+ НОЩ + НФР + НСЛ(2)	В таблице AP отсутствует название исключенного отношения	Исключение отношения пропускается
213	+ НОЩ + НФР(0) + НСЛ(1)	Определитель неправильный	Целый массив пропускается
214	+ НОЩ + НФР(0) + НСЛ(2)	В таблице MM отсутствует название модели	Целый массив пропускается
215	+ НОЩ + НФР(0) + 000000000	Таблица MM заполнена	Целый массив пропускается
216	+ НОЩ + НФР + НСЛ(0)	Число изменений в описании модели больше допустимого	Следующие изменения пропускаются
217	+ НОЩ + НФР + НПЕ	Число связанных переменных больше допустимого	Отношение пропускается
220	+ НОЩ + НФР + НПЕ	Переменная отсутствует в таблице XR второй модели	Отношение пропускается

1	2	3	4
221	+ НОЩ + НФР(0) + НСЛ(3)	Шифр магнитной ленты не совпадает с данными	Целый массив пропускается
222	+ НОЩ + НФР + НСЛ(1)	Название изменения неправильное	Изменение пропускается
223	+ НОЩ + НФР(0) + НСЛ(2)	Название новой модели уже было в массиве КР или ММ	Целый массив пропускается
224	+ НОЩ + НФР + НСЛ	Длина формата больше допустимой	Формат пропускается
225	+ НОЩ + НФР + НСЛ(2)	Тип реквизита неправильный	Формат пропускается
226	+ НОЩ + НФР + НСЛ(2)	Название отношения уже было в массиве АР	Отношение пропускается
227	+ НОЩ + НФР + НПЕ	Название переменной типа массива отсутствует в таблице КР	Переменная пропускается
230	+ НОЩ	Таблица КР заполнена	Останов 666I

Пример печати сообщений на бумаге узкой печати следующий:

+ IIII	IIII	IIII	Начало массива описания модели
+	200		Номер ошибки: 200
+	2		Номер фразы: 2
+	4		Номер переменной, длина которой больше допустимой
+ 7777	7777	7777	Конец трансляции данного массива

## Л и т е р а т у р а

1. Система модульного программирования для ЭВМ "Минск-22". Общее описание, НИПТИ, Таллин, 1970 (ротапринт).
2. Система табличного программирования, научно-технический отчет, ПТНИИ по теме 72-129-2, Таллин, 1969, рукопись.
3. Столяров Г.К., Ковалевич Э.В., Курусь Э.С. Система символического кодирования для ЭВМ "Минск-2(22)", часть I, II, III, Минск, 1967.
4. Система модульного программирования для ЭВМ "Минск-22". Отладка и решение задач, НИПТИ, Таллин, 1970 (ротапринт).

## СО Д Е Р Ж А Н И Е

	Стр.
И1. Организация работ . . . . .	3
Составили Э.Тыгу и К.Тинн	
И2. Описание алгоритмов . . . . .	13
Составил Э.Тыгу	
И3. Программирование . . . . .	19
Составил Э.Тыгу	
И4. Перфорирование . . . . .	43
Составил В.Муст	
И5. Операционная система . . . . .	46
Составил Э.Тыгу	

Экспериментальный комбинат "Бит"  
Таллин, ул. Планеэди, 18  
Подписано в печать 20/УИ 1970 г. Тираж 500 экз.  
Бумага 30x41. Усл.печ.л. 4,32. Уч.-изд.л. 2,15.  
Заказ №272-524.  
Ретсапринт ЭК "Бит". Таллин, ул. Паик, 68

ОПЕЧАТКИ

Стр.	Строка	Напечатано	Следует читать
40	6 снизу	<название документа>]...)	<название документа>_]...)
42	6 снизу	превышать 2400(8)	превышать 2600(8)
53	5 снизу до конца страницы	(АРХИВ/... ... относится к $N_2$ ...	(АРХИВ/МЛ.../ $t_1/t_2$ /МЛ/), где МЛ... - шифр МЛ архива модулей, переводит модуль МЛ из общего архива и записывает в архив модулей $t_1 = \begin{cases} 0 - \text{модуль переписывается впервые} \\ 1 - \text{модуль переписывается не впервые} \end{cases}$ $t_2 = \begin{cases} 0 - \text{перепись без трансляции системных команд} \\ 1 - \text{перепись с трансляцией системных команд} \end{cases}$
54	4 сверху	(АРХИВ/Ш1//0/ТАБ1/// 0/ТАБ2///1/ПРОГ1///)	(АРХИВ/Ш1/0/0/ТАБ1/)
57	7 снизу	на 77...7.	на -77...7
59		Останов СчАК I72II	Останов СчАК I72I3
59		Останов СчАК I72I3	Останов СчАК I72II
60	СчАК 5073- -СчАК 5576		Остановы от СчАК 5073 до СчАК 5576 лишние. Последним остановом программы Архив читать СчАК I4264
65	II снизу	два символа "( "без")" между ними	два символа "( " без ")" между ними

