

Аннотация

Настоящая книга содержит описание входного языка ЭВМ “МИР” и “МИР-1” – АЛМИР-65. Приводятся краткие сведения о вычислительной машине “МИР-1”.

Описание основано на книге “Программирование для ЭЦВМ “МИР-1” (В.А. Пономарёв; М., Советское радио, 1975).

С автором можно связаться по e-mail: pateralex@mail.ru

1-я редакция
28.03.2003 г.

Александр Савватеев

Оглавление

| | |
|--|-----------|
| 1. ВЫЧИСЛИТЕЛЬНАЯ МАШИНА “МИР-1” | 4 |
| 1.1. НАЗНАЧЕНИЕ И ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ЭВМ “МИР-1” | 4 |
| 1.2. УСТРОЙСТВО ЭВМ | 4 |
| 1.2.1. Устройство обмена информацией (УОИ) | 5 |
| 1.2.2. Устройство микропрограммного управления (УМУ) | 7 |
| 1.2.3. Запоминающее устройство (ЗУ) | 7 |
| 1.2.4. Арифметическое устройство (АУ) | 8 |
| 1.2.5. Устройство электропитания (УЭП) | 9 |
| 2. ЯЗЫК АЛМИР-65 | 10 |
| 2.1. АЛФАВИТ ЯЗЫКА АЛМИР-65 | 10 |
| 2.1.1. Буквы | 10 |
| 2.1.2. Цифры | 10 |
| 2.1.3. Знаки операций | 10 |
| 2.1.4. Знаки отношений | 10 |
| 2.1.5. Спецификаторы | 10 |
| Разделители | 11 |
| Скобки | 11 |
| Специальные символы | 11 |
| Специальные знаки | 11 |
| Резервные символы | 12 |
| 2.2. СЛУЖЕБНЫЕ СЛОВА | 12 |
| 2.3. ИМЕНА СТАНДАРТНЫХ ФУНКЦИЙ | 12 |
| 2.4. СЛОВА | 13 |
| 2.4.1. Числа | 13 |
| 2.4.2. Идентификаторы | 13 |
| 2.4.3. Метки | 14 |
| 2.5. ВЫРАЖЕНИЯ | 14 |
| 2.5.1. Первичные выражения | 14 |
| Числа | 14 |
| Переменные | 14 |
| Функции | 15 |
| Суммы | 15 |
| Произведения | 16 |
| Интегралы | 16 |
| 2.5.2. Арифметические выражения | 17 |
| Простые арифметические выражения | 17 |
| Условные арифметические выражения | 17 |
| 2.6. ОПЕРАТОРЫ ЯЗЫКА АЛМИР-65 | 18 |
| 2.6.1. Простые операторы | 18 |
| Оператор присваивания | 18 |
| Оператор перехода | 18 |
| Оператор "ВЫЧИСЛИТЬ" | 18 |
| Оператор стирания | 19 |
| Оператор останова | 19 |
| Пустой оператор | 19 |
| 2.6.2. Операторы управления выводом | 20 |
| Оператор вывода | 20 |
| Оператор вывода значений | 20 |
| Оператор вывода массива | 20 |
| Оператор вывода заголовка таблицы | 20 |
| Оператор вывода таблицы | 21 |
| Оператор вычисления масштаба | 21 |
| Оператор вывода графика | 21 |
| 2.6.3. Сложные операторы | 22 |
| Составной оператор | 22 |
| Условный оператор | 22 |
| Оператор цикла | 22 |
| 2.7. ОПИСАНИЯ | 23 |
| 2.7.1. Описание простой переменной | 23 |
| 2.7.2. Описание массива | 23 |
| 2.7.3. Описание нестандартной функции | 24 |
| 2.8. ПРОГРАММЫ | 24 |
| 2.8.1. Основные программы | 24 |
| Указатель разрядности | 24 |
| Операторная часть | 25 |
| Описательная часть | 25 |

| | |
|--|----|
| 2.8.2. Вторичные программы..... | 25 |
| Вторичные программы, работающие с исходным текстом..... | 25 |
| Вторичные программы, работающие с оттранслированным текстом..... | 25 |

1. Вычислительная машина “МИР-1”

1.1. Назначение и технические характеристики ЭВМ “МИР-1”

ЭВМ “МИР-1” (Машина для Инженерных Расчётов) была разработана Институтом кибернетики АН УССР под руководством академика В.М. Глушкова. Входной язык машины, названный АЛМИР-65, был разработан коллективом в составе В.М. Глушкова, А.А. Стогния, А.А. Летичевского, В.П. Клименко, А.А. Дородницыной и др.

В более поздних моделях ЭВМ – “МИР-2” и “МИР-3” – язык АЛМИР-65 трансформировался в язык Аналитик, однако сохранилась программная совместимость “снизу вверх”.

Машина “МИР-1” относится к классу малых ЭВМ 2-го поколения. Она предназначена для автоматизации инженерных расчётов. В частности, на ней могут решаться следующие задачи:

- системы линейных алгебраических уравнений до 20-го порядка;
- системы обыкновенных дифференциальных уравнений до 16-го порядка;
- дифференциальные уравнения в частных производных;
- системы нелинейных уравнений до 6-го порядка;
- интегральные уравнения;
- нахождение собственных векторов для матриц до 10-го порядка;
- нахождение максимальных собственных значений для матриц до 18-го порядка.

Вычислительная машина фактически является аппаратно-микропрограммно-программным интерпретатором языка АЛМИР-65. С точки зрения программиста она оперирует десятичными числами любой разрядности в пределах имеющегося объёма памяти (требуемая разрядность указывается в самой программе).

Ввод программы и вывод результатов осуществляется посредством электронной пишущей машинки с широкой кареткой. Скорость печати достигает 10 символов в секунду.

ЭВМ имеет оперативную память объёмом 4096 12-разрядных ячеек. Каждая ячейка может хранить один из символов входного языка, служебное слово или идентификатор стандартной функции.

В качестве внешней памяти используется восьмидорожечная перфолента (шесть дорожек – код символа, седьмая – признак буквы, восьмая – контрольная). Скорость вывода на перфоратор ПЛ-80 – до 80 символов в секунду, скорость ввода с устройства FS-1501 – до 1500 символов в секунду.

Быстродействие машины при выполнении арифметических операций над 5-разрядными числами составляет 200 – 300 оп/с.

Машина потребляет не более 1,5 кВт энергии, питаясь от трёхфазной сети переменного тока (380 В, 50 Гц). Допускается колебание напряжения в пределах от +10 до -15 %.

Для работы необходима относительная влажность воздуха не более 80 % и температура от +10 до +35 °С.

Для нормальной эксплуатации ЭВМ требуется площадь не менее 16 м². Масса машины около 400 кг.

Время перехода в рабочий режим – не более 2 мин.

1.2. Устройство ЭВМ

Конструктивно вычислительная машина “Мир-1” выполнена в виде двух столов. На одном из них смонтирована пишущая машинка, на другом – отладочный пульт. Кроме этих внешних устройств, в состав машины входят:

- устройство обмена информацией;
- устройство микропрограммного управления;
- запоминающее устройство;
- арифметическое устройство;
- устройство электропитания.

1.2.1. Устройство обмена информацией (УОИ)

Основной функцией УОИ является обеспечение связи человека с машиной. Это достигается с помощью соответствующих устройств, позволяющих выполнять следующие действия.

1. Вводить программы и исходную информацию в запоминающее устройство машины в следующих режимах:

- с пишущей машинки;
- с пишущей машинки с одновременной перфорацией;
- с устройства ввода с перфоленты;
- с устройства ввода с перфоленты с одновременной печатью на пишущей машинке.

2. Выводить результаты работы на пишущую машинку.

3. Выводить программу из запоминающего устройства на перфоратор и пишущую машинку по вторичным программам “Проверить”, “Перфорировать”, “Печатать”.

4. Задавать рабочие и отладочные режимы работы машины.

5. Запускать и останавливать работу машины.

6. Вручную заносить коды в основные регистры и память машины.

7. Заносить содержимое любой ячейки памяти в регистр числа.

8. Отображать состояние основных регистров и управляющих схем.

Для выполнения этих функций УОИ имеет следующие устройства и узлы:

- пишущую машинку “SOEMTRON”;
- устройство ввода с перфоленты FS-1501;
- ленточный перфоратор ПЛ-80;
- отладочный пульт;
- схемы управления.

Пишущая машинка “SOEMTRON” предназначена для определения режима работы машины, ручного ввода программы, вывода программы и результатов работы на печать.

Режимы работы определяются с помощью десятиклавишного переключателя. Предусмотрены следующие режимы.

1. ВВОД – ввод информации с пишущей машинки с одновременной перфорацией (при нажатой клавише “ВП”) или без неё либо ввод с устройства ввода с перфоленты без печати или с одновременной печатью вводимой информации (при нажатой клавише “ВТ”).

2. СЧЕТ – работа по заданной программе.

3. С₁ (СТОП 1) – работа с остановкой на каждой микрокоманде.

4. Т₂ (СТОП 2) – работа с остановкой после выполнения оператора внутреннего языка машины.

5. О₃ (СТОП 3) – работа с печатью и остановкой после выполнения каждого арифметического выражения.

6. П₄ (СТОП 4) – работа с остановкой после выполнения каждого оператора входного языка.

7. П.ВВОД (повторный ввод) – ввод вторичных программ с пишущей машинки с перфорацией и без неё либо с устройства ввода с перфоленты без печати или с печатью.

8. ТЕСТ 1 – выполнение тестовой задачи № 1.

9. ТЕСТ 2 – выполнение тестовой задачи № 2.

10. РЕЖИМ – определение кодов начала работы программ одного из выбранных режимов ВВОД, П.ВВОД, СЧЕТ, ТЕСТ 1, ТЕСТ 2. Режимы работы с остановкой (С₁, Т₂, О₃ и П₄) имеют тот же код начала программы, что и режим СЧЕТ. Кроме того, клавиша РЕЖИМ обеспечивает установку в нуль необходимых регистров, счётчиков и триггеров перед пуском машины, т.е. производит начальную установку ЭВМ.

С помощью клавиши “РЕЖИМ” можно остановить машину в процессе работы.

Ручной ввод программы в память машины осуществляется с помощью клавиатуры пишущей машинки, на которой имеются все символы и знаки, составляющие алфавит языка АЛМИР-65. Каждый символ кодируется девятиразрядным двоичным кодом, поступающим в память ЭВМ.

При выводе результатов работы пишущая машинка распечатывает символ, соответствующий коду, находящемуся в регистре числа.

Устройство ввода с перфоленты FS-1501 служит для ввода информации с перфоленты в память ЭВМ. Считывание ведётся фотоэлектрическим способом. Устройство обеспечивает работу с пяти-, шести-, семи- и восьмидорожечными перфолентами. В машине “Мир-1” используется восьмидорожечная перфолента шириной 25,4 мм.

Для непосредственного кодирования символов используются первые шесть дорожек перфоленты. На седьмой дорожке пробивается признак буквы, а на восьмой – признак чётности числа всех пробивок на первых семи дорожках.

Ленточный перфоратор ПЛ-80 обеспечивает вывод на перфоленту информации, находящейся в памяти машины.

Отладочный пульт предназначен для определения отладочных режимов работы машины, занесения кодов в регистры и счётчики и для отображения состояния последних. Он используется также при выполнении профилактических работ и поиске неисправностей.

Основными органами управления и индикации отладочного пульта являются:

– две кнопки “ПИТАНИЕ”, предназначенные для включения и выключения электропитания;

– кнопка “ПУСК”, запускающая работу машины. Её функция совпадает с клавишей “СЧЕТ”. Когда ЭВМ работает, горит сигнальная лампа “ПУСК”;

– кнопка “Н.УСТ”, включенная параллельно клавише “РЕЖИМ” и выполняющая те же функции, т.е. начальную установку ЭВМ;

– сигнальная лампа “ОСТ”, сигнализирующая об останове машины;

– кнопки “ЗАНЕСЕНИЕ”, служащие для ручного занесения кодов, набранных на соответствующих тумблерных регистрах “ЗАНЕСЕНИЕ” в оперативные регистры Р1 – Р6, регистр адреса РА и регистр числа РЧ, а также в счётчики матрицы операторов (СчМОп) и матрицы информационной (СчМИ). Любое занесение информации производится в остановленном состоянии машины;

– тумблеры “ОСТАНОВКА”, предназначенные для остановки машины при совпадении набранного кода с кодом, находящимся в СчМИ, СчМОп или РЧ в зависимости от того, какой из тумблеров включен;

– тумблеры “КОНТРОЛЬ”, управляющие различными режимами работы. Тумблер “КОНТРОЛЬ 4” управляет шириной печатаемой на пишущей машинке информации (65 или 141 символ в строке). Тумблер “КОНТРОЛЬ 5” отключает контроль нечётности информации, считываемой с перфоленты; он может использоваться для чтения семидорожечных перфолент. Тумблер “КОНТРОЛЬ 6” обеспечивает режим ввода информации с перфоленты, а также вывод на перфоленту с одновременной печатью на пишущей машинке. Тумблеры “КОНТРОЛЬ 7” (“запись”), “КОНТРОЛЬ 8” (“чтение”) и “КОНТРОЛЬ 9” (“чтение массива”) позволяют выполнить операции обращения к памяти машины. Обращение происходит при нажатии кнопки “ПУСК”

в режиме “СЧЁТ”. Если включен тумблер “КОНТРОЛЬ 7”, произойдёт запись кода из регистра РЧ в память по адресу в РА. Для тумблера “КОНТРОЛЬ 8” будет выполнено чтение ячейки по адресу РА в регистр РЧ. При включенном тумблере “КОНТРОЛЬ 9” при каждом нажатии на кнопку “ПУСК” также будет производиться чтение ячейки памяти по адресу в РА в регистр РЧ, после чего содержимое РА будет увеличиваться на единицу;

– группа индикаторных ламп, информирующих о состоянии регистров РА, РЧ, Р1 – Р6, СчМИ, СчМОп, регистров арифметического устройства Р11, Р12, Р13, РП, триггеров схемы сравнения и управления печатью. Последние две группы лампочек используются при поиске неисправностей.

1.2.2. Устройство микропрограммного управления (УМУ)

Устройство микропрограммного управления служит для обеспечения выполнения рабочей программы. Оно также организует взаимодействие всех других устройств машины.

В состав УМУ входят следующие узлы:

- блок матрицы информационной (МИ);
- матрица операторов (МОп);
- счётчик матрицы информационной (СчМИ);
- счётчик матрицы операторов (СчМОп);
- схемы управления матрицами.

Информационная матрица содержит алгоритм работы машины, записанный на её внутреннем языке. Алгоритм представляет собой набор следующих программ:

- предварительной обработки входной информации;
- распознавания и выполнения операторов программы на языке АЛМИР-65;
- выполнения арифметических операций;
- вычисления элементарных функций.

Всего МИ содержит 4096 команд.

Матрица операторов хранит микропрограмму, обеспечивающую выполнение операторов информационной матрицы. Всего в МОп размещено 1024 микрокоманды.

Счётчик МИ хранит адрес оператора в информационной матрице, а *счётчик МОп* – адрес микрокоманды в матрице операторов.

Схема управления матрицами обеспечивает выработку необходимых управляющих сигналов.

1.2.3. Запоминающее устройство (ЗУ)

Запоминающее устройство служит для хранения символьной, числовой и адресной информации, вводимой в машину или вырабатываемой в процессе выполнения программы.

Основными узлами ЗУ являются:

- магнитное оперативное запоминающее устройство (МОЗУ);
- регистр адреса (РА);
- блок оперативных регистров;
- схема сравнения и анализа кодов.

Магнитное оперативное запоминающее устройство построено по матричной схеме и включает в себя блок числовых матриц (БЧМ, всего 12 матриц), диодную матрицу и матрицу сопротивлений.

Запоминающими элементами числовых матриц являются кольцевые ферриты с прямоугольной петлёй гистерезиса. Размер ферритов $1,4 \times 1 \times 0,65$ мм. Каждая числовая матрица имеет 4096 ферритов. Для хранения каждого разряда запоминаемого

слова служит один феррит. Индекс разряда определяется номером числовой матрицы в МОЗУ. Все 12 числовых матриц работают параллельно, обеспечивая одновременный доступ к каждому разряду одной и той же 12-разрядной ячейки памяти.

Ёмкость МОЗУ – 4096 12-разрядных слов. Время выборки числа из ячейки памяти не превышает 2,5 мкс, а полный цикл обращения к МОЗУ занимает четыре такта машины и занимает 16 мкс.

Регистр адреса служит для приёма адреса ячейки и его хранения в процессе выполнения операции обращения к ЗУ. Устройство микропрограммного управления, запрашивая доступ к памяти, задаёт первоначальное содержимое РА и направление его изменения, поскольку разряды чисел хранятся в соседних ячейках памяти и выбираются или записываются последовательно. РА состоит из 12 разрядов.

Регистр числа является временным хранилищем содержимого, читаемого или записываемого в МОЗУ. Он содержит 12 разрядов.

Блок оперативных регистров состоит из оперативных регистров Р1 – Р6 и блока кодовых формирователей ФК1. Регистры Р1 – Р6 являются сверхоперативной памятью ЭВМ и обеспечивают хранение числовой, адресной и символьной информации. Совместно с РА они образуют счётчик, работающий по сигналам устройства микропрограммного управления. Приём информации, поступающей из РЧ и СчМИ, в регистры осуществляется через кодовые формирователи.

Блок ФК1 служит для формирования сигналов обмена информацией между регистрами, СчМИ и МИ.

Схема сравнения и анализа кодов выполняет анализ и сравнение содержимого РЧ с выходами ФК1. Результатом является выработка соответствующего сигнала, передаваемого в устройство микропрограммного управления.

1.2.4. Арифметическое устройство (АУ)

Арифметическое устройство предназначено для преобразования числовой информации. Преобразование чисел осуществляется посредством выполнения операций алгебраического сложения и умножения десятичных цифр, запоминания переноса и прибавления переноса, полученного на предыдущем такте.

Конструктивно АУ является последовательным десятичным арифметическим устройством табличного типа, обеспечивающим обработку чисел произвольной разрядности. Особенностью ЭВМ “Мир-1” является хранение чисел в виде комбинации цифр, каждая из которых имеет собственный знак. В общем виде целое число в такой системе может быть записано так:

$$A = \alpha_n a_n 10^n + \alpha_{n-1} a_{n-1} 10^{n-1} + \dots + \alpha_1 a_1 10^1 + \alpha_0 a_0,$$

Где $a_0 - a_n$ – десятичные цифры, принимающие значения от 0 до 9; $\alpha_0 - \alpha_n$ – знаки каждого разряда десятичного числа, принимающие значения -1 и +1.

В состав АУ входят следующие функциональные узлы:

- матрица арифметическая (МА);
- блок регистров (Р11, Р12, Р13, Р13', РП, РП');
- схема формирования сигналов.

Арифметическая матрица представляет собой постоянное запоминающее устройство, в котором хранятся таблицы сложения и умножения двух десятичных цифр. Кроме того, МА содержит вспомогательную таблицу для приведения всех цифр числа к одному знаку.

Блок регистров служит для непосредственного управления МА, для приёма исходной цифровой информации и результатов выполнения арифметических операций..

Регистр Р11 служит для приёма из РЧ очередной цифры множителя и хранения её до окончания операции умножения.

Регистр P12 служит для приёма первого операнда и управления МА.

Регистр P13 предназначен для приёма второго операнда, а также фиксации результата выполнения арифметической операции.

Регистр P13' обеспечивает приём кода второго операнда из регистра P13 и управляет работой МА.

Регистры РП и РП' предназначены для приёма и хранения возможного переноса при выполнении арифметических операций.

Каждый из перечисленных регистров имеет пять двоичных разрядов: четыре предназначены для хранения собственно цифры, пятый – знака цифры.

Схема формирования сигналов служит для выработки управляющих сигналов, выбирающих нужную таблицу МА.

Управление работой АУ осуществляет устройство микропрограммного управления. Оно выдаёт в арифметическое устройство сигналы арифметических операций, передачи кодов в регистры, а также управляет обменом информацией с запоминающим устройством.

1.2.5. Устройство электропитания (УЭП)

Устройство электропитания преобразует трёхфазное переменное напряжение электрической сети в постоянные напряжения, необходимые для функционирования вычислительной машины: -60, -27, -12,6, -6,3 и +6,3 В. Необходимые уровни напряжений устанавливаются не позднее 5 минут после включения ЭВМ.

2. Язык АЛМИР-65

2.1. Алфавит языка АЛМИР-65

Алфавит языка АЛМИР-65 отличается от алфавитов современных языков программирования. Наряду с традиционными символами он содержит некоторое количество специальных символов, отсутствующих в современных языках. Клавиатура пишущей машинки наряду с общепринятыми содержит несколько дополнительных клавиш, позволяющих вводить уникальные для языка АЛМИР-65 символы.

Знаки алфавита делятся на буквы, цифры, знаки операций, знаки отношений и спецификаторы. Буквы и цифры не имеют самостоятельного значения и служат для образования более сложных конструкций, например, служебных слов, идентификаторов и меток. Знаки операций, знаки отношений и спецификаторы обычно имеют какие-либо определённые функции. Каждый символ алфавита языка АЛМИР-65, не входящий в состав служебного слова или имени стандартной функции, занимает в памяти ЭВМ отдельную ячейку. Каждое служебное слово и имя любой стандартной функции занимают также единственную ячейку памяти. В процессе ввода программы с пишущей машинки происходит преобразование служебных слов и имён стандартных функций во внутренние коды, которые и заносятся в память.

2.1.1. Буквы

В языке АЛМИР-65 используется 45 заглавных букв: 31 буква русского алфавита (без “Ё” и “Ъ”) и 14 букв латинского алфавита, не совпадающих по своему начертанию с буквами русского алфавита (“D”, “F”, “G”, “I”, “J”, “L”, “N”, “Q”, “R”, “S”, “U”, “V”, “W”, “Z”).

Буквы не имеют индивидуального смысла. Они используются для образования идентификаторов, меток и других синтаксических конструкций.

2.1.2. Цифры

Как и современные языки, АЛМИР-65 использует 10 десятичных цифр от “0” до “9”. Они используются для образования чисел, меток и некоторых других конструкций.

2.1.3. Знаки операций

В языке АЛМИР-65 определено пять арифметических операций, задаваемых соответствующими символами: “+” (сложение), “-” (вычитание), “×” (умножение), “/” (деление) и “↑” (возведение в степень). Знаки операций применяются при образовании арифметических выражений, а также при записи чисел.

2.1.4. Знаки отношений

АЛМИР-65 использует пять знаков отношений: “<” (меньше), “≤” (меньше или равно), “=” (равно), “>” (больше) и “≥” (больше или равно). Они предназначены для образования условных выражений и других элементов языка.

2.1.5. Спецификаторы

Спецификаторы делятся на пять групп: разделители, скобки, специальные символы, специальные знаки и резервные символы. Они используются либо для образования некоторых специальных синтаксических единиц языка АЛМИР-65, либо для облегчения восприятия программы. Некоторые из этих знаков имеют чисто техническое назначение и при составлении программ не используются.

Разделители

Язык АЛМИР-65 использует четыре разделителя: “,” (запятая), “;” (точка с запятой), “.” (точка) и “₁₀” (десятичный порядок).

Разделительные знаки отделяют синтаксические единицы программы. Запятая используется для разделения элементов различных списков: переменных, индексов переменных и массивов, операторов вывода, параметров функций и т.п. Точка с запятой разделяет операторы и описания. Точка отделяет целую часть числа от дробной. Подстрочный знак десятичного порядка “₁₀” применяется для информирования о том, что далее следует порядок десятичного числа.

Скобки

К скобкам относятся знаки круглых и квадратных скобок, а также кавычки, всего пять символов (“(”, “)”, “[”, “]” и “””).

Круглые скобки используются для фиксации различных синтаксических единиц, имеющих самостоятельное значение. Например, в круглые скобки заключают параметры стандартных и простых функций, например: EXP(X), ALPHA(Z) и т.п. Кроме того, круглые скобки используются для заключения специальных конструкций под знак суммы, произведения или интеграла, например: $\sum(I=IO, N, A)$.

Другой областью применения круглых скобок являются сложные операторы, в которых они необходимы для заключения внутренних операторов. Их можно использовать в общепринятом математическом смысле, т.е. при группировке отдельных фрагментов сложных арифметических выражений.

Квадратные скобки, называемые в языке АЛМИР-65 индексными, служат для размещения в них индексов, определяющих местоположение элементов векторов и матриц, например: X[I] или A[I, J+2]. Кроме того, индексные скобки используются для заключения последовательностей символов, выводимых на печать, например: "ВЫВОД" "ЗНАЧЕНИЙ" [ТЕМПЕРАТУРА =].

К скобкам отнесены также кавычки, выполняющие роль скобок при написании служебных слов, чтобы отличить их от идентификаторов. Так, в приведённом выше примере заключённые в кавычки слова “ВЫВОД” и “ЗНАЧЕНИЙ” являются служебными словами, а слово “ТЕМПЕРАТУРА”, не заключённое в кавычки – строкой символов, выводимой в данном случае на печать.

Специальные символы

Язык АЛМИР-65 определяет семь специальных символов: “√”, “Σ”, “Π”, “∫”, “ε”, “F”¹ и “∞”. Эти символы служат для обозначения некоторых специальных функций языка: квадратного корня, суммы, произведения, интеграла, целой и дробной части числа. Символ бесконечности используется при программировании некоторых арифметических выражений, например, суммы с бесконечным верхним пределом суммирования: $\sum(I = IO, \infty, EPS, A)$.

Специальные знаки

В группу специальных знаков входят восемь символов, имеющих вспомогательное значение.

Знаки “(BT)” и “(BP)” указывают клавиши, используемые для включения устройства ввода с перфоленты и ленточного перфоратора соответственно.

Знак “→” указывает клавишу перевода каретки пишущей машинки.

Знак “*” служит для исправления программы посредством затирания последнего неправильно набранного символа.

Знак “π” специального назначения не имеет.

¹ Современные шрифты не содержат символа, точно совпадающего с данным символом АЛМИР-65, поэтому используется наиболее похожий на него по начертанию из имеющихся. Символ АЛМИРа фактически является большой латинской рукописной буквой F.

Два знака являются пробельными символами, использующимися для оформления программы. Один из них вводится в ЭВМ посредством нажатия обычной клавиши пробела, он обозначается символом “_”. Второй знак вводится при нажатии специальной “пустой” клавиши на нижнем регистре, он обозначается знаком “Ø”.

Последний из специальных знаков вводится нажатием пустой клавиши на верхнем регистре и обозначается символом “Ø”. В отличие от знака “Ø”, он воспринимается как ошибка, если только не используется внутри последовательностей символов, где применяется в качестве пробела. Например, оператор "ВЫВОД" [ПОТЕРИОСЫРЬЯ] обеспечит печать строки символов “ПОТЕРИ СЫРЬЯ”. Другие пробельные символы (“_” и “Ø”) внутри строки символов будут проигнорированы.

Резервные символы

В языке АЛМИР-65 имеется пять резервных символов: “Щ”, “М”, “Ч”, “Я” и “Д”. Они предназначены для размещения пробелов в символьных строках и эквивалентны символу “Ø”.

2.2. Служебные слова

Язык АЛМИР-65 содержит 31 служебное слово. Они используются в качестве конструктивных элементов операторов и вторичных программ. Все служебные слова всегда заключаются в кавычки, благодаря чему машина отличает их от идентификаторов. Таким образом, запись "СТОП" означает служебное слово “СТОП”, а запись СТОП – идентификатор “СТОП”.

Служебные слова могут записываться в сокращённом виде, для чего используются несколько первоначальных символов этих слов. Следующие за ними символы могут не записываться. Ниже приведён список служебных слов. Обязательная часть указана большими буквами, необязательная – малыми.

| | | | |
|-----------|-----------|---------------|-------------|
| ВМесто | Если | МАССив | РАЗРядность |
| ВЫВод | ЗАГоловок | МАСШтаб | СТЕреть |
| ВЫПолнить | ЗАМенить | Метка | СТОП |
| ВЫЧислить | ЗАПисать | НА | СТРока |
| ГДЕ | Значений | ПЕРфорировать | ТАблица |
| График | Иначе | ПЕЧатать | ТО |
| Для | ИДТИ | ПРОбел | Шаг |
| ДО | КОНец | ПРОверить | |

Следует обратить внимание, что два служебных слова – “ПРОбел” и “ПРОверить” – сокращаются одинаково, до символов “ПР”. Это связано с тем, что слово “ПРОбел” может встречаться только в первичных программах, а слово “ПРОверить” – только во вторичных программах.

Хотя каждое служебное слово состоит из нескольких символов (в том числе двух кавычек, ограничивающих слово), в памяти ЭВМ оно занимает только одну ячейку памяти. Преобразование служебных слов во внутренний код происходит в процессе ввода программы.

2.3. Имена стандартных функций

В языке АЛМИР-65 имеется 20 стандартных функций. Каждая из функций имеет единственный параметр, который должен быть заключён в круглые скобки. Параметром может быть любое арифметическое выражение с учётом ограничений, накладываемых самими функциями (например, функция извлечения квадратного корня требует, чтобы её параметр был неотрицательным числом).

Ниже приведён список стандартных функций языка АЛМИР-65.

SIN синус

| | |
|----------------|--|
| COS | косинус |
| TG | тангенс |
| CTG | котангенс |
| ARCSIN | арксинус |
| ARCCOS | арккосинус |
| ARCTG | арктангенс |
| ARCCTG | арккотангенс |
| SH | гиперболический синус |
| CH | гиперболический косинус |
| TH | гиперболический тангенс |
| CTH | гиперболический котангенс |
| EXP | показательная функция |
| LN | натуральный логарифм |
| LG | десятичный логарифм |
| ABS | абсолютное значение (модуль) |
| SIGN | знак (возвращает +1 для положительного значения параметра, 0 для нулевого и -1 для отрицательного) |
| $\sqrt{\quad}$ | квадратный корень |
| ε | целая часть числа |
| F | дробная часть числа |

Аргументы тригонометрических функций задаются в радианах.

Хотя большая часть имён стандартных функций включает несколько символов, в памяти они занимают одну ячейку, поскольку при вводе программы производится преобразование имён этих функций во внутренний код.

2.4. Слова

К словам АЛМИРа относятся элементарные конструкции, представляющие собой последовательности допустимых знаков. Сюда входят числа, идентификаторы и метки.

2.4.1. Числа

АЛМИР-65 различает целые и дробные (по современной терминологии – вещественные) числа. Целые числа не могут иметь в своём составе десятичной точки и порядка. Если число имеет десятичную точку, оно считается дробным, даже если дробная часть равна нулю. Например, число “7” считается целым, а число “7.0” – дробным. Если целая часть дробного числа равна нулю, она может быть опущена. Например, записи “0.7” и “.7” эквивалентны. Однако, если указывается десятичная точка, обязательно должна присутствовать дробная часть числа, т.е. записи “7” и “7.0” являются допустимыми, а запись “7.” недопустима.

Порядок следует за мантиссой и отделяется от неё знаком “ 10 ”. Этот знак употребляется точно в том же смысле, что буква “E” в записи вещественных чисел современных языков программирования. Например, запись языка АЛМИР-65 “3.2 $_{10}^5$ ” соответствует записи “3.2E5” современных языков и математической записи $3,2 \cdot 10^5$. Порядок обязательно должен быть целым числом.

И мантисса, и порядок числа могут иметь знак “+” или “-”. Если знак не указывается, предполагается знак “плюс”.

2.4.2. Идентификаторы

Идентификаторы языка АЛМИР-65 состоят из букв и цифр и начинаются обязательно с буквы. Они служат для обозначения переменных, массивов, функций и др. Длина идентификаторов формально не ограничена (ограничения связаны с объёмом доступной памяти ЭВМ).

Идентификаторы, обозначающие различные элементы программы, могут совпадать. Например, в операторе

$$A.A = A(A[A])$$

первый идентификатор “А” является именем метки, второй – простой переменной, третий – функции, четвёртый – массива, а пятый – элемента одномерного массива. Интерпретатор определяет конкретный смысл идентификатора в зависимости от его места в операторе и окружающих его символов. Так, если за идентификатором следует открывающая круглая скобка, этот идентификатор является именем функции.

Идентификаторами формально являются имена стандартных функций. Однако в силу вышеуказанного правила эти идентификаторы могут быть использованы в качестве имён любых элементов программы, кроме функций.

В идентификаторах не рекомендуется использовать буквы “Г” и “Ш”, поскольку нижние регистры этих клавиш заняты для включения считывателя с перфоленты (ВТ) и перфоратора (ВП).

2.4.3. Метки

Метки используются для присвоения имён определённым операторам программы для выполнения условных или безусловных переходов на них.

Метка является либо идентификатором, за которым следует символ “.” (точка), либо целым числом без знака, также заканчивающимся точкой.

Метка перед оператором не влияет на выполнение этого оператора, она лишь позволяет передать ему управление в нарушение естественного (последовательного) порядка следования операторов.

Язык АЛМИР-65 позволяет использовать метки в любом месте программы, в том числе и в конце составного оператора (в последнем случае метка помечает пустой оператор). Синтаксически несколько операторов могут быть помечены одинаковыми метками, однако в реальности будет использоваться лишь первая из них, а остальные метки будут проигнорированы. В современных языках программирования дублирование меток привело бы к возникновению синтаксической ошибки.

2.5. Выражения

В языке АЛМИР-65 различают простые и условные арифметические выражения. И те, и другие состоят из одного или нескольких первичных выражений.

2.5.1. Первичные выражения

Первичное выражение может представлять собой:

- число без знака (целое или дробное, с показателем или без такового);
- переменную;
- функцию;
- сумму;
- произведение;
- интеграл.

Первичное выражение сводится к определённому числовому значению.

Числа

Формат чисел рассматривался выше и в дополнительных пояснениях не нуждается. Значением числа является оно само. При необходимости интерпретатором осуществляется преобразование целого числа в дробное, а также изменение точности чисел.

Переменные

Язык АЛМИР-65 поддерживает работу с простыми переменными и переменными с индексами (массивами).

Единственным типом данных, хранимым в переменных, является число. Целые числа хранятся в том виде, в каком введены, и не переводятся автоматически в экспоненциальную форму. Вещественные числа хранятся в экспоненциальной форме с заданной точностью.

Простые переменные могут быть определены в описательной части программы (начинающейся с зарезервированного слова "ГДЕ") или в операторной части путём использования в левой части операции присваивания или в качестве управляющей переменной цикла. К моменту использования значения переменной оно должно быть определено. Доступ к переменной осуществляется по её имени (идентификатору).

Переменные с индексами определяются в описательной части программы. АЛМИР-65 поддерживает одно- и двумерные массивы, соответствующие в математике векторам и матрицам. Доступ к элементу массива осуществляется с помощью имени массива и индекса. Индекс заключается в квадратные скобки, следующие непосредственно за именем массива, и представляет собой арифметическое выражение. Если происходит обращение к элементу двумерного массива, используются два индекса, находящиеся внутри общей пары квадратных скобок и разделяемые запятой. Примеры: MAS1[I]; MAS2[I-3,J×5]. Если при вычислении индексного выражения получено дробное число, в качестве индекса используется его целая часть.

Функции

АЛМИР-65 имеет набор стандартных функций, описанных выше, в подразделе 2.3 "Имена стандартных функций". Кроме того, он позволяет программисту определять собственные функции, называемые нестандартными.

В отличие от современных языков программирования, функции в языке АЛМИР-65 понимаются узко математически и предназначены исключительно для вычисления значения выражения.

Нестандартная функция определяется в описательной части программы. Её определение имеет следующий вид:

$$\text{имя}(\text{список}) = \text{выражение}$$

Здесь *имя* – имя нестандартной функции; *список* – список формальных параметров функции, разделённых запятыми; *выражение* – арифметическое выражение, вычисляющее значение функции при её вызове.

Вызов функции осуществляется тем же способом, что и в современных языках программирования: указывается имя функции, за которым в круглых скобках следует список фактических параметров, разделённых запятыми.

Использование выгодно в том случае, когда вычисления по одной и той же формуле необходимо производить в разных местах программы. В этом случае в каждом таком месте достаточно обратиться к функции, реализующей требуемые вычисления, а не писать каждый раз выражение, осуществляющее эти вычисления.

Суммы

Для упрощения реализации часто встречающихся в математике операций суммирования вида $\sum_{i=a}^b c$ и $\sum_{i=a}^{\infty} c$ в языке АЛМИР-65 предусмотрен специальная функция суммирования. Её запись выглядит следующим образом:

$$\Sigma (\text{переменная} = \text{нач, кон, выражение})$$

или

$$\Sigma (\text{переменная} = \text{нач, } \infty, \text{точность, выражение})$$

Вычисление суммы производится следующим образом. Сначала вычисляются начальная и конечная границы суммирования, т.е. параметры *нач* и *кон* (во втором случае – только значение *нач*). Они могут быть заданы произвольными арифметическими выражениями, однако используются только целые части вычисленных значений.

Вычисленное значение начальной границы присваивается указанной *переменной*. Затем производится вычисление значения *выражения*, полученный результат прибавляется к некоторой внутренней переменной, в которой накапливается вычисленная сумма; первоначально значение этой переменной равно нулю. После вычисления *вы-*

ражения значение *переменной* увеличивается на единицу и, если оно не превосходит верхнюю границу (значение *кон*), повторяется вычисление *выражения* для нового значения *переменной* и добавление его к накапливаемой сумме, увеличение *переменной*, сравнение с верхней границей и т.д. Когда значение *переменной* станет больше верхней границы, вычисление суммы будет закончено; результатом будет накопленное значение суммы из внутренней *переменной*. Например, результатом вычисления выражения $\sum(I=0,2,I)$ будет значение 3.

Действия при выполнении вычислений при записи второго вида (с бесконечным верхним пределом) в целом аналогичны вышеописанным. Разница заключается в том, что вычисление суммы прекращается, когда разность между только что вычисленным и предыдущим значениями *выражения* не превосходит заданной *точности*.

Операция суммирования может быть запрограммирована вручную с помощью оператора цикла и нескольких операций присваивания. Однако “ручное” программирование занимает больше памяти, а полученная программа выполняется существенно медленнее (АЛМИР-65 – фактически интерпретатор, а не компилятор).

Произведения

Как и в случае сумм, для вычисления произведений вида $\prod_{i=a}^b c$ и $\prod_{i=a}^{\infty} c$ язык АЛМИР-65 предусматривает специальные конструкции следующего вида:

\prod (*переменная* = *нач*, *кон*, *выражение*)

или

\prod (*переменная* = *нач*, ∞ , *точность*, *выражение*)

Алгоритм их вычисления полностью совпадает с алгоритмом выполнения операции суммирования (см. предыдущий подраздел). Единственная разница заключается в том, что во внутренней *переменной* накапливается не сумма, а произведение значений *выражения* при разных значениях *переменной*; первоначальное значение внутренней *переменной* равно единице.

Примером может служить вычисление факториала: $FAC = \prod(I=1, N, I)$. Для $N = 0$ будет получено правильное значение $FAC = 1$, поскольку проверка на выход *переменной* I за верхний предел N производится после итерации, то есть после вычисления *выражения* (в данном случае – просто I).

Интегралы

Для приблизительного вычисления значения определённого интеграла² вида $\int_a^b f(x)dx$ язык АЛМИР-65 предоставляет специальную конструкцию, записываемую следующим образом:

∫(*переменная* = *нач*, *кон*, *число*, *выражение*)

Здесь *переменная* является простой *переменной*, значение которой изменяется от нижней до верхней границы (*нач* и *кон* соответственно), а *выражение* соответствует подынтегральной функции $f(x)$. В процессе вычисления значения интеграла интервал

² Точное численное вычисление значений интегралов возможно только после того, как оно будет вычислено аналитически, т.е. после того, как запись в виде интеграла будет сведена к обычному арифметическому выражению. Автоматическое решение такой задачи в 1960-х годах была совершенно неразрешима из-за крайне ограниченных с нынешней точки зрения вычислительных возможностей ЭВМ. Сейчас имеются пакеты символьных вычислений, например, Maple и Mathematica, которые во многих случаях сами способны определять способ точного вычисления значений тех или иных интегралов. Однако в некоторых случаях они оказываются бессильны или, ещё хуже, дают неверный ответ, поскольку задача автоматизированного аналитического решения любых интегралов полностью, вероятно, неразрешима.

интегрирования делится на заданное *число* отрезков интегрирования, оно обязательно должно быть чётным.

Вычисление интегралов производится по формуле Симпсона, согласно которой значение интеграла заменяется конечной суммой параболических трапеций. Формула Симпсона является точной для многочленов до третьей степени включительно.

ЭВМ "Мир-1" позволяет вычислять кратные, т.е. вложенные интегралы, однако кратность не должна превышать пяти.

Пример. Для вычисления интеграла $I = \int_0^2 \int_{0,5}^{1,5} \frac{e^{-a(x-y)}}{1+b(x+y)} dx \cdot dy$ при числе отрезков интегрирования 10, $a=0.8$ и $b=0.4$ в языке АЛМИР-65 используется запись следующего вида:

$$I = \int(X=0, 2, 10, \int(Y=.5, 1.5, 10, EXP(-A \times (X \times Y)) / (1 + B \times (X + Y))))$$

При вычислениях по этой формуле с точностью пять знаков будет получено значение $I=2,4576_{10}-1$.

2.5.2. Арифметические выражения

Арифметические выражения языка АЛМИР-65 делятся на простые и условные. Результатом вычисления выражения является целое или вещественное число. Целый результат получается только в том случае, если применяемые операции и исходные типы операндов гарантируют получение целого результата (например, при сложении значений двух целочисленных переменных), в противном случае всегда получается вещественный результат, даже если с математической точки зрения он может быть без потери точности представлен в виде целого.

Простые арифметические выражения

Простые арифметические выражения записываются в целом так же, как и арифметические выражения современных языков программирования, и выполняют те же самые функции. Они состоят из первичных выражений (чисел, переменных, функций, сумм, произведений, интегралов – см. выше), соединённых друг с другом с помощью знаков арифметических операций и при необходимости сгруппированных с помощью круглых скобок.

Арифметические выражения языка АЛМИР-65 отличаются от выражений современных языков программирования следующими особенностями:

- в качестве операции умножения используется знак "×", а не "*" (звёздочка);
- имеется операция возведения в степень "↑", отсутствующая в большинстве языков программирования.
- помимо числовых величин, переменных и функций, в языке АЛМИР-65 возможно использование операций вычисления сумм, произведений и интегралов.

Приоритеты операций стандартные: самой приоритетной является операция возведения в степень, затем следуют операции умножения и деления; самый низкий приоритет – у сложения и вычитания. С помощью круглых скобок порядок вычислений может быть изменён по обычным математическим правилам.

Условные арифметические выражения

Условные арифметические выражения обеспечивают вычисление по одной или другой формуле в зависимости от выполнения того или иного условия. Синтаксически они записываются следующим образом:

"Если" *условие* "ТО" (*выражение 1*) "Иначе" (*выражение 2*)

Условие является комбинацией двух простых арифметических выражений, соединённых одной из операций отношения. Если *условие* соблюдается, то результатом условного арифметического выражения будет значение простого *выражения 1*, если не соблюдается – простого *выражения 2*.

Выражение 2, в отличие от *выражения 1*, также может быть условным. Таким образом, вложенность условных выражений язык АЛМИР-65 не обеспечивает, но поддерживает их соединение “в цепочку”.

Пример. Результатом вычисления условного выражения

"Если" $A > B$ "ТО" $(A + B)$ "Иначе" $(A - B)$

при $A = 2$, $B = 3$ будет значение -1 , а при $A = 3$, $B = 2$ – значение 5 .

2.6. Операторы языка АЛМИР-65

Операторы делятся на три группы – простые операторы, операторы управления выводом и сложные операторы. При записи операторы отделяются точками с запятой (подобно современному языку Паскаль).

Каждый оператор может иметь метку, размещаемую перед ним и отделяемую символом “.” (точка; см. подраздел 2.4.3 “Метки”), например:

МЕТКА . $A = B - C$;

2.6.1. Простые операторы

К простым относятся операторы, не содержащие внутри себя других операторов. Сюда входят:

- оператор присваивания;
- оператор перехода;
- оператор "ВЫчислить";
- оператор стирания;
- оператор останова;
- пустой оператор.

Оператор присваивания

Оператор присваивания в языке АЛМИР-65 совершенно традиционен и имеет следующий вид:

переменная = арифметическое выражение

В левой части указывается имя простой переменной или имя массива вместе со значениями его индексов. В правой части задаётся арифметическое выражение (простое или условное).

Значения всех переменных, входящих в состав выражения, должны быть определены до выполнения оператора присваивания.

Оператор перехода

Оператор перехода имеет следующий формат:

"НА" *метка*

Функционально оператор перехода эквивалентен оператору *goto*, имеющемуся в большинстве современных языков программирования. Как уже указывалось, любой оператор программы на языке АЛМИР-65 может иметь метку, которая записывается перед оператором и отделяется от него точкой. Оператор "НА" осуществляет переход на оператор, помеченный заданной *меткой*. Если этой меткой помечены несколько операторов, переход произойдёт на первый из операторов программы, имеющий данную метку (АЛМИР-65 допускает такую возможность; современные языки расценят это как ошибку).

Оператор "ВЫчислить"

Этот оператор имеет следующий формат:

"ВЫчислить" *список простых переменных*

Имена простых переменных в списке разделяются запятыми.

Оператор "ВЫЧИСЛИТЬ" не имеет прямых аналогов в современных языках программирования. Смысл его заключается в следующем.

В описательной части программы (начинающейся со служебного слова "ГДЕ", будет описана ниже) описываются простые переменные. Описания выглядят как операторы присваивания, т.е. указывается имя простой переменной, знак "=" и арифметическое выражение. Когда выполняется оператор "ВЫЧИСЛИТЬ", в котором указывается имя простой переменной, описанной подобным образом, производится вычисление выражения, заданного в описании, и полученное значение присваивается данной переменной. Поскольку в выражении могут использоваться различные переменные и функции, а оператор "ВЫЧИСЛИТЬ" с именем одной и той же переменной может встречаться в программе неоднократно, каждый раз при его выполнении переменная может принимать различные значения.

Пример. В процессе выполнения следующей программы

```
A = 5;  
B = 6;  
"ВЫЧ" X, K;  
A = 7;  
"ВЫЧ" X, K;  
"ГДЕ" X = A + B; K = A - B;
```

после выполнения первого оператора "ВЫЧИСЛИТЬ" переменная X примет значение 11, а переменная K – значение –1. После второго выполнения оператора "ВЫЧИСЛИТЬ" переменная X будет содержать значение 12, а переменная K – значение 1.

Как видно из приведённого примера, оператор "ВЫЧИСЛИТЬ" предназначен для сокращения объёма программы путём вынесения часто повторяющихся операторов присваивания в описательную часть программы. В современных языках для той же цели набор операторов присваивания может быть вынесен в отдельную процедуру или функцию, которая и будет вызываться по мере надобности.

Оператор стирания

Оператор стирания имеет следующий вид:

"СТЕреть" список простых переменных

Оператор стирания уничтожает в памяти указанные переменные, обеспечивая тем самым её более экономное использование. Стирание массивов не обеспечивается.

Оператор останова

Формат оператора:

"СТОП"

"СТОП" идентификатор

Оператор "СТОП" приостанавливает выполнение программы. Если указан идентификатор, при останове он распечатывается.

Выполнение программы может быть продолжено нажатием клавиши "ПУСК".

Пустой оператор

Пустой оператор не содержит каких-либо символов, кроме, возможно, пробельных. В программе он выглядит как точка с запятой, следующая с интервалом или без такового после другой точки с запятой (как указывалось выше, точка с запятой заканчивает каждый оператор).

Пустой оператор можно использовать совместно с меткой для выполнения операций перехода.

2.6.2. Операторы управления выводом

В данную группу входят оператор вывода, оператор вывода значений, оператор вывода массива, оператор вывода заголовка таблицы, оператор вывода таблицы, оператор вычисления масштаба, оператор вывода графика.

Оператор вывода

Оператор вывода имеет следующий вид:

"ВЫВод" *список элементов вывода*

Элементы вывода в списке разделяются запятыми. Имеются шесть видов элементов вывода:

простое арифметическое выражение

[*последовательность символов*]

"ПРОбел"

"ПРОбел" *целое без знака*

"СТРока"

"СТРока" *целое без знака*

Простое арифметическое выражение вычисляется, после чего распечатывается оно само и его значение. В принципе возможно использование и условных арифметических выражений, но они должны заключаться в круглые скобки, превращаясь формально в простые выражения.

Последовательность символов в квадратных скобках распечатывается так, как есть (скобки не печатаются; в современных языках программирования для вывода строки символов она обычно заключается в кавычки или апострофы).

Элементы вывода со служебными словами "ПРОбел" и "СТРока" предназначены соответственно для вывода одного или нескольких пробелов или для пропуска одной или нескольких строк. Количество пробелов или строк указывается *целым без знака*; если оно не задано, выводится один пробел или пропускается одна строка.

Оператор вывода значений

Синтаксис этого оператора таков:

"ВЫВод" "ЗНачений" *список элементов вывода*

Оператор вывода значений аналогичен обычному оператору вывода. Разница заключается в том, что при выводе на печать арифметических выражений печатаются только их значения, но не сами выражения.

Оператор вывода массива

Формат этого оператора следующий:

"ВЫВод" "МАССива" *идентификатор массива*

Оператор вывода массива распечатывает идентификатор и размерность массива, а начиная со следующей строки – его содержимое. Одномерный массив распечатывается в строку, двухмерный – в виде матрицы.

На каждый элемент массива отводится $R+7$ позиций на пишущей машинке, где R – установленная точность вычислений. Поскольку целые числа хранятся в машине в том виде, в каком введены, и не переводятся автоматически в экспоненциальную форму, возможна ситуация, когда для вывода числа потребуется большее число позиций. В этом случае произойдет аварийный останов.

Оператор вывода заголовка таблицы

Оператор вывода заголовка таблицы имеет следующий вид:

"ВЫВод" "ЗАГоловка" "ТАБлицы" *номер таблицы, список названий столбцов*

При выполнении этого оператора печатается строка вида “ТАБЛИЦА n ”, где n – номер таблицы (одна цифра). Затем на следующей строке печатаются названия столбцов. Каждое название в списке представляет собой любую последовательность символов, названия разделяются запятыми. Запятая может встречаться и в названиях, однако только внутри круглых скобок, при этом печатаются и скобки, и содержащиеся в них символы, в том числе запяты. Пробельные символы учитываются при определении длины заголовка и выводятся на печать, пустые символы игнорируются.

Вычислительная машина запоминает номер таблицы и начальные позиции столбцов, подсчитываемые в процессе вывода заголовка. В дальнейшем при печати таблицы каждый столбец будет начинаться с соответствующей позиции.

Оператор вывода таблицы

Формат оператора вывода таблицы:

"Вывод" "Таблицы" номер таблицы, список элементов таблицы

Если в программе оператору вывода таблицы предшествовал оператор вывода заголовка таблицы (это определяется путём анализа номера таблицы), то оператор вывода таблицы распечатывает строку, состоящую из элементов таблицы. Каждый элемент является простым арифметическим выражением (условное выражение должно заключаться в скобки, превращаясь при этом в простое выражение); его вывод производится с позиции, соответствующей его заголовку.

Если оператор вывода таблицы используется без оператора вывода заголовка таблицы, сначала печатается строка “ТАБЛИЦА n ”, где n – номер таблицы (одна цифра). Затем распечатывается строка, содержащая выражения, являющиеся элементами таблицы, при этом определяются начальные позиции каждого столбца. После этого распечатываются значения элементов таблицы. Таким образом, в этом случае оператор вывода таблицы одновременно является оператором вывода заголовка таблицы, в котором роль заголовков играют арифметические выражения, являющиеся элементами таблицы.

Оператор вычисления масштаба

Формат оператора вычисления масштаба следующий:

"МАСШтаб" нижняя граница, верхняя граница

Этот оператор задаёт масштаб, используемый при выводе графиков с помощью оператора вывода графика. Эти два оператора могут использоваться только совместно.

Нижняя и верхняя границы указываются в виде простых арифметических выражений. Они задают пределы изменения значения функции, график которой будет построен. Абсолютная величина разности между верхней и нижней границами делится на число позиций в строке пишущей машинки (110 для широкого формата, 55 для узкого); полученное значение принимается в качестве цены одного деления шкалы графика. Оно хранится в памяти до выполнения следующего оператора вычисления масштаба. Запоминается также значение, соответствующее самой левой позиции графика. Таким образом, один оператор вычисления масштаба может предшествовать любому числу операторов вывода графика.

Оператор вывода графика

Оператор вывода графика имеет следующий вид:

*"График" функция графика
"График" функция графика, аргумент графика*

И функция, и аргумент задаются как простые арифметические выражения.

Вычисляется значение функции. Затем на основании информации, запомненной при выполнении оператора вычисления масштаба, определяется, какой позиции строки пишущей машинки соответствует вычисленное значение, и в этой строке печатается символ “*” (звёздочка). Если использован второй формат оператора вывода графика,

производится вычисление аргумента, и полученное значение печатается в конце этой же строки. На этом выполнение оператора вывода графика заканчивается.

Для вывода нескольких значений, т.е. для построения полноценного графика (насколько это возможно при использовании в качестве устройства вывода пишущей машинки) используется несколько операторов вывода графика. В действительности обычно используется один оператор, выполняющийся в цикле.

Если при выполнении оператора вывода графика обнаруживается, что значение функции слишком велико или мало и не может быть выведено на печать, происходит аварийный останов.

2.6.3. Сложные операторы

К сложным относятся составной оператор, условный оператор и оператор цикла.

Составной оператор

Составной оператор состоит из одного или нескольких операторов, заключённых в круглые скобки. Внутри скобок операторы разделяются точками с запятыми, последний оператор может не иметь после себя точки с запятой.

Все современные языки программирования также допускают запись составных операторов, однако для определения их границ круглые скобки не используются. В языке Паскаль для этих целей служат ключевые слова "begin" и "end", в языке Си – фигурные скобки.

Составные операторы используются в тех случаях, когда синтаксис допускает использование одного оператора, а требуется использовать несколько. В этом случае требующиеся операторы оформляются в виде одного составного оператора.

Внутри составного оператора могут находиться другие составные операторы, глубина вложенности языком не ограничивается. Возможно выполнение переходов на любой оператор как внутри составного оператора, так и из него³.

Условный оператор

Формат условного оператора следующий:

условие (список операторов)
условие (список операторов) "Иначе" (список операторов)

Первый формат называют коротким условным оператором, второй – полным условным оператором.

Условие записывается в виде двух простых арифметических выражений, разделённых операцией отношения. Им предшествует служебное слово "Если".

Если условие соблюдается, выполняются операторы, записанные в круглых скобках, следующих непосредственно за условием. Если условие не соблюдается, то короткий условный оператор никаких действий не выполняет, а в полном условном операторе выполняются операторы, записанные в круглых скобках после служебного слова "Иначе". Таким образом, условный оператор является эквивалентом оператора "if" современных языков программирования.

Оператор цикла

Оператор цикла записываются в одном из двух форматов:

"Для" параметр = выражение 1 "Шаг" выражение 2 "Выполнить" оператор

"Для" параметр = выражение 1 "Шаг" выражение 2 "До" выражение 3 "Выполнить" оператор

³ Синтаксис языка АЛМИР-65, таким образом, отнюдь не способствует структурному программированию, а как "программирование без goto" оно вообще оказывается невозможным, поскольку в АЛМИРе отсутствуют эквиваленты циклов while и repeat... until языка Паскаль (while и do... while языка Си), что приводит в общем случае к невозможности обойтись без операторов перехода.

Параметром цикла является простая переменная. Когда оператор цикла выполняется первый раз, этой переменной присваивается значение *выражения 1*. В первом формате после этого сразу выполняется *оператор*, во втором сначала проверяется условие завершения цикла (см. ниже), и только в том случае, если цикл ещё не должен быть завершён, выполняется *оператор*. Обычно в цикле выполняется составной оператор.

При каждой итерации цикла значение переменной увеличивается на величину *выражения 2* (если это выражение имеет отрицательное значение, то значение управляющей переменной цикла, естественно, будет уменьшаться). В первом формате цикл сам по себе никогда не заканчивается, он может быть прерван только при выполнении *оператора*. Во втором случае выполнение цикла заканчивается, когда значение управляющей переменной будет больше или меньше *выражения 3* (больше, если *выражение 2* больше нуля, и меньше в противном случае). *Оператор* и в этой ситуации может прервать выполнение цикла.

Реализация оператора цикла в ЭВМ “МИР” довольно неэффективна, из-за чего он выполняется медленнее своего программного аналога, состоящего из условного оператора и оператора перехода.

Оператор цикла принципиально соответствует циклу “for” современных языков программирования, хотя и имеет определённые отличия.

2.7. Описания

В языке АЛМИР-65 применяются описания простых переменных, массивов и нестандартных функций.

2.7.1. Описание простой переменной

Синтаксически описание простой переменной совпадает с оператором присваивания:

идентификатор = арифметическое выражение

Встретив описание простой переменной в описательной части программы, вычислительная машина не вычисляет немедленно указанное выражение и не присваивает полученное значение переменной. Это производится в одной из следующих ситуаций:

– если в операторной части программы встречается оператор “ВЫЧИСЛИТЬ”, то значения перечисленных в нём переменных вычисляются в соответствии с их описаниями в описательной части программы (см. выше описание оператора “ВЫЧИСЛИТЬ” в подразделе 2.6.1 “Простые операторы”);

– если при вычислении арифметического выражения встречена переменная, значение которой не было определено с помощью оператора присваивания или оператора “ВЫЧИСЛИТЬ” в операторной части программы, анализируется описательная часть программы, и первое встретившееся описание данной переменной используется для определения её значения и подстановки его в вычисляемое в операторной части выражение.

Таким образом, описание простых переменных в языке АЛМИР-65 принципиально отличается от описания переменных в современных языках. Оно используется не для определения типа переменной и распределения памяти, а для вынесения повторяющихся выражений из основной программы или для присвоения переменным начальных значений.

2.7.2. Описание массива

В языке АЛМИР-65 можно использовать два вида описаний массивов:

идентификатор размерность = список значений
идентификатор размерность

Первый формат используется для так называемых исходных массивов. Встретив такое описание, вычислительная машина выделяет для массива память и заносит в его

ячейки начальные значения. Второй формат описывает рабочий массив и используется только для определения размерности массива и выделения для него памяти.

Размерность – это количество элементов массива, записанное в квадратных скобках. Если описывается одномерный массив, то размерность задаётся единственным числом без знака, если описывается двумерный массив, то задаются два числа без знака, разделяемых запятой.

Список значений представляет собой список чисел, разделённых запятыми. Использование выражений при записи списка значений недопустимо.

Примеры.

A [5] = 2, 6, 4, 5, 10

Б [2, 3] = 1, 2, 3, 4, 5, 6

В [5]

Первые два описания выделяют память под массивы (А из пяти элементов и Б из шести, последний массив состоит из двух строк по три столбца) и присваивает им начальные значения. Третье описание просто определяет одномерный массив В размерностью 5 элементов и выделяет под него память, но не присваивает его элементам начальные значения.

2.7.3. Описание нестандартной функции

В подразделе 2.5.1 “Первичные выражения” уже говорилось об использовании нестандартных функций. Описание нестандартной функции имеет следующий формат:

имя функции (список формальных параметров) = арифметическое выражение

При обращении (вызове) функции в процессе вычисления того или иного арифметического выражения в операторной части программы ЭВМ отыскивает её описание в описательной части, подставляет значения фактических параметров в точке вызова вместо соответствующих формальных параметров, затем вычисляет *арифметическое выражение*, соответствующее данной функции, и полученное значение возвращает в качестве результата функции.

Функция языка АЛМИР-65 является не последовательностью описаний и операторов, как это имеет место в современных языках программирования, она фактически является именованным арифметическим выражением и соответствует понятию функции в математике, а не программировании.

2.8. Программы

В языке АЛМИР-65 существуют понятия основной и вторичной программы.

2.8.1. Основные программы

Основная программа состоит из четырёх частей и имеет следующий вид:

указатель разрядности
операторная часть
описательная часть
"Конец"

Основная программа всегда начинается указателем разрядности, за которым следует операторная часть. Описательная часть может отсутствовать. Программа заканчивается служебным словом "Конец".

Указатель разрядности

Указатель разрядности записывается в одном из двух форматов, не имеющих никаких смысловых отличий:

"РАЗРядность" *целое без знака*
"" *целое без знака*

Указатель разрядности определяет, какой объём памяти выделяется для каждой переменной или элемента массива и с какой точностью выполняются вычисления. Следует помнить, что простые переменные, которым присваиваются целые значения, записанные в обычной, а не экспоненциальной форме, хранят свои значения в том виде, в каком они записаны, т.е. их разрядность может оказаться больше, чем задано, что не приведёт к ошибке (память для простых переменных выделяется при выполнении операторов присваивания или операторов "ВЫЧИСЛИТЬ"). Значения элементов массивов всегда хранятся во внутренней форме в экспоненциальном виде (память под массивы выделяется на основании их описаний).

Операторная часть

В операторной части находятся операторы, выполняемые в процессе работы программы. Операторы отделяются друг от друга точкой запятой. После последнего оператора точка с запятой не требуется.

Описательная часть

Описательная часть начинается служебным словом "ГДЕ", за которым следуют описания простых переменных, массивов и нестандартных функций. Описания друг от друга отделяются точкой с запятой. После последнего описания точка с запятой не ставится, а сразу следует служебное слово "Конец", завершающее программу.

2.8.2. Вторичные программы

В языке АЛМИР-65 имеется семь вторичных программ: программа исправления текста, программа проверки текста, программа перфорации текста, программа замены, программа "МЕТКА" L., программа печати, программа "ГДЕ". Общее назначение вторичных программ – отладка основных программ.

Вторичные программы делятся на две группы: программы, работающие с исходным текстом основной программы (до начала её выполнения), и программы, работающие с уже оттранслированной основной программой, когда выполнен хотя бы один её оператор.

Вторичные программы, работающие с исходным текстом

В первую группу входят программы исправления текста, проверки текста и перфорации текста. Эти программы активизируются следующей последовательностью действий:

- нажимаются клавиши С₁ и П.ВВОД;
- последовательно нажимаются клавиши "РЕЖИМ" и "ПУСК";
- вводится команда, активизирующая вторичную программу;
- последовательно нажимаются клавиши "СЧЁТ", "РЕЖИМ", "ПУСК".

Программа замены текста активизируется посредством ввода команды следующего вида:

"ВМесто" искомая последовательность "ЗАПисать" новая последовательность

Эта команда производит поиск и замену одной последовательности символов на другую.

Программы проверки и перфорации текста вызываются с помощью команд "Проверить" и "Перфорировать", не имеющих параметров. Программа проверки выводит основную программу на пишущую машинку, программа перфорации – на перфоленту.

Вторичные программы, работающие с оттранслированным текстом

Во вторую группу входят программа замены, "МЕТКА" L., программа печати и программа "ГДЕ". Эти программы вызываются с помощью команд "ЗАМенить", "Метка", "ПЕчать" и "ГДЕ" соответственно. Команды вводятся после останова машины, завершившей выполнение основной программы, либо после приостанова машины оператором с помощью нажатия клавиши П₄.

Программа замены позволяет заменить ту или иную часть программы и выполнить один или несколько операторов. В числе операторов может быть оператор перехода на ту или иную метку в основной программе, что позволяет возобновить её выполнение.

Программа “МЕТКА” L. позволяет установить на заданной метке основной программы точку останова. Когда управление дойдёт до этой метки, ЭВМ остановится.

Программа печати выводит на перфоленту содержимое основной памяти машины, т.е. оттранслированную основную программу и связанную с ней информацию.

Программа “ГДЕ” обеспечивает вывод на пишущую машинку той или иной части основной программы, т.е. выполняет функции декомпилятора (это возможно благодаря тому, что АЛМИР-65 является интерпретирующим языком, и программа на нём хранится в памяти в закодированном виде).