



Ордена Ленина и ордена Октябрьской Революции

Институт атомной энергии

им. И. В. Курчатова

В. Ю. Веретенков,

ИАЭ-3555/16

В. А. Петров

**БЕСФОРМАТНЫЙ ВВОД ДАННЫХ ПО ИМЕНАМ
В МОНИТОРНОЙ СИСТЕМЕ "ДУБНА"**

Москва 1982

УДК 681.3.06

Ключевые слова: прикладные программы, языки программирования, ввод данных.

Описана система бесформатного ввода данных по именам, предоставляющая пользователю более удобный способ задания данных, чем традиционный форматный ввод или бесформатный ввод данных, определенные в большинстве алгоритмических языков. Подробно изложена методика использования системы, иллюстрированная примерами. Работа может быть использована в качестве инструкции по предлагаемой системе.

ВВЕДЕНИЕ

Традиционный форматный ввод данных не свободен от недостатков:

- данные в соответствии с форматом занимают строго определенные позиции на перфокартах;
- на картах они никак не идентифицируются и их порядок должен строго соответствовать списку ввода;
- количество данных должно совпадать с количеством элементов в списке ввода.

Таким образом, если программа обрабатывает несколько вариантов наборов данных (отличающихся, возможно, одним-двумя числами), то для определения каждого варианта необходимо полностью в строгом формате повторить задание всех данных.

Бесформатный ввод данных по именам, предлагаемый в настоящей работе, свободен от перечисленных недостатков.

Возможны два способа реализации бесформатного ввода данных: "языковый" и "процедурный". Примером первого способа можно считать введение в ФОРТРАН-IV конструкции *NAMELIST* [1], недостатком которого является обязательное условие — задание массива полностью. Второй способ (реализация в виде набора процедур), по мнению автора, более предпочтительный по следующим причинам:

- набор процедур можно использовать из разных языков входящих в систему программирования [2];
 - реализацию набора процедур можно избежать от недостатков как форматного ввода данных, так и ввода данных по *NAMELIST*.
- Модификацию ввода данных по *NAMELIST* рассматривать нецелесообразно, так как это явилось бы нарушением существующих стандартов.

Предлагаемый способ задания данных можно проиллюстрировать следующим примером:

Пример 1:

* EXECUTE

$X = 3.12, Y = 2.42$

$Z(2:7) = 1., 2(2., 3.), -4$

* EOR

По приведенной записи нетрудно сообразить, что после ввода этих карт переменная X примет значение 3.12,

переменная Y — 2.42,

в массив Z со 2 по 7 элементы занесутся 1., 2., 3., 2., 3., — 4.

Карта * EOR — признак конца очередной порции вводимых данных (*End Of Record*).

В рассматриваемом примере X, Y, Z — имена данных на перфокартах соответствующие каким-то переменным (объектам) в программе. Программа, использующая предлагаемую систему бесформатного ввода, перед вводом данных должна с помощью процедур системы установить соответствие символических имен данных (на перфокартах) именам объектов в программе.

1. ОПИСАНИЕ ИМЕН ДАННЫХ

а) Описание символических имен для простых переменных осуществляется с помощью подпрограммы *DECVAR*. Обращение к этой подпрограмме на ФОРТРАНе:

CALL DECVAR(<симв.имя> , <тип> , <адрес>), где

<симв.имя> — идентификатор длиной до шести символов, означает имя, которое задается на перфокартах и указывается в виде текстовой константы либо переменной, значение которой — текст;

<тип> — тип значения объекта (целая константа или переменная):

0 — *INTEGER*

1 — *REAL*

2 — *LOGICAL*

3 — *DOUBLE PRECISION*

4 — *COMPLEX*

Для восьмеричных и текстовых констант <тип> = 0;

<адрес> — простая переменная или переменная с индексом, в которую будет осуществляться ввод, если в вводимой информации встретится указанное <симв.имя>.

```

Пример 2:  CALL DECVAR('X', 1, X)
            CALL DECVAR('Y', 1, F(5))
            .
            .
            .
            * EXECUTE
            X = 1, Y = 4E-2
            * EOR

```

После обращения за вводом данных при заданном описании объектов простая переменная X примет значение 1., а в пятый элемент массива F засылается число 0.04.

Заметим, что производится преобразование целого числа 1 в вещественное, так как тип переменной X — вещественный, и разрешается задание числа в форме

$\langle \text{целое} \rangle E \langle \text{целое} \rangle -$ (без точки)

б) Описание символических имен для массивов осуществляется с помощью подпрограммы *DECOSJ*. Обращение к этой подпрограмме на ФОРТРАНе:

```
CALL DECOSJ ( <симв.имя> , <тип> , <адрес> , <L> , N )
```

Число параметров подпрограммы — переменное и зависит от количества размерностей массива. Значение первых двух было рассмотрено в п. "а".

$\langle \text{адрес} \rangle$ — имя массива по программе, в которой будет производиться ввод;

$\langle L \rangle$ — список длин массива по размерностям для одномерного массива — это одно число, для двумерного — два, для трехмерного — три;

N — количество размерностей массива.

```
Пример 3: DIMENSION Z(4,2), A(100)
```

```

            .
            .
            .
            CALL DECOSJ('Z', 1, Z, 4, 2, 2)
            CALL DECOSJ('X50', 1, A(50), 51, 1)
            .
            .
            .
            * EXECUTE
            Z(2:4) = 1., 2., 7., Z(1,2) = 10.,
            X50(1:51) = 1., 50(0.)
            * EOR

```

После обращения за вводом данных элементы массива $Z(2,1)$, $Z(3,1)$, $Z(4,1)$, $Z(1,2)$ примут значения 1., 2., 7., 10. В элемент $A(50)$ зашлется 1., а элементы массива A с индексом от 51 до 100 — обнулятся. Заметим, что у двумерного массива Z задан всего только один индекс. По фортранному правилу умолчания недостающие индексы полагаются равными 1.

2. ВВОД ДАННЫХ

После того как описаны все зводимые объекты с помощью подпрограмм *DECVAR* и *DECOBJ*, можно обратиться за вводом данных к подпрограмме *INFREE*. Это обращение на ФОРТРАНе записывается как:

CALL INFREE (MPRI, MER)

где *MPRI* — управляет печатью вводимых данных. *MPRI=1* — данные распечатываются в виде, в каком они набиты на картах; *MPRI=0* — печати нет.

MER — признак ошибки. После возврата из *INFREE* значение *MER=0* означает, что ввод прошел нормально, а *MER≠0* — были ошибки при вводе. Если перед обращением к *INFREE* выполнить оператор *MER=-1*, то произойдет блокировка печати диагностики. Но и в этом случае значение *MER* после возврата содержит признак ошибки.

Правила записи данных на бланках. Данные записываются на бланках с 1 до 72 позицию. Пробелы вне текстовых констант игнорируются. Допускается перенос информации на следующую строку в любом месте. Разделителем данных служит запятая. Обобщенным аналогом оператора присваивания по терминологии описываемой системы является элемент ввода, представляющий список левых частей (простых переменных и неявных циклов), куда производится засылка. Типы значений левых частей могут различаться. Список значений рассылаемых данных находится в правой части, он может содержать числа (целые, вещественные, двойной точности), логические значения, текстовые строки либо набор значений (заключенный в скобки) с коэффициентом повторения. Признаком конца списка элементов ввода является строка **EOR* (записанная с первой позиции). Приведем нестрогое описание записи данных в "квазиэкзусовской" форме: *СЗВ: <список элементов ввода> := <пусто> | <элемент ввода>, <СЗВ>*

ЭВ: <элемент звзда> ::= <список левых частей> | <правая часть>
 СЛЧ: <список левых частей> ::= <левая часть> | <СЛЧ> <левая часть>
 ЛЧ: <левая часть> ::= <простая переменная> = | <невный цикл> =
 ПП: <простая переменная> ::= <идентификатор>
 НЦ: <невный цикл> ::= <идентификатор> < (список граничных пар) >
 СГП: <список граничных пар> ::= <граничная пара> | <СГП> <граничная пара>
 ГП: <граничная пара> ::= <целое без знака> <целое без знака> : <целое без знака>
 ПЧ: <правая часть> ::= <элемент ПЧ> | <ПЧ> , <элемент ПЧ>
 ЭПЧ: <элемент ПЧ> ::= <объект> | <коэффициент повторения> < (список объектов) >
 КП: <коэффициент повт.> ::= <целое без знака>
 СОБ: <список объектов> ::= <объект> | <СОБ> , <объект>
 ОБ: <объект> ::= <число> | <логическое значение> | <текст>
 ЛЗ: <логическое значения> ::= .T | .F

Число — восьмиричное, целое, вещественное, с двойной точностью, записанное по правилам ФОРТРАНА. При несопадении типов значений левой и правой частей данные правой части приводятся (если это возможно) к типу значений левой части. Так при I — целом запись $I = 1.2$ допустима, а $I = 1.2$ запрещена. При задании в левой части объекта типа *COMPLEX* интерпретируются два последовательных числа как одно комплексное. Допустимо также задание комплексного числа по правилам языка ФОРТРАН.

Пример 4:

```

PROGRAM TEST
  REAL MAT(4,4), BEK(4)
  LOGICAL END
  CALL DECVAR ('KON', 2, END)
  CALL DECOBJ ('M', 1, MAT, 4, 4, 2)
  CALL DECOBJ ('B', 1, BEK, 4, 1)
  END = .FALSE.
1 MER = 0
  CALL INFREE (1, MER)
  IF (MER.NE.0) STOP
  IF (END) GO TO <КОНЕЦ>
  <ОБРАБОТКА>
  GO TO 1
  
```

```

END
*EXECUTE
M(1:4, 1:4) = 16(0),
M(1,1) = M(2,2) = M(3,3) = M(4,4) = 1,
B(1:4) = 4(1E2)
*EOR
M(4,1) = M(1,4) = 2.
*EOR
KOH = .T
*EOR
*END FILE

```

В этом примере обрабатываются два набора данных:

— единичная матрица MAT (4,4) и вектор BEK (4), значения компонент которого равны 100.,

— подпревращенная матрица MAT с прежним вектором BEK.

После третьего обращения к подпрограмме *INFREE* логическая переменная *END* примет значение *.TRUE.* и программа перейдет на завершающую часть.

3. ДИАГНОСТИКА ОШИБОК

Диагностику ошибок, возникающую при работе подпрограмм системы, можно разбить на три группы.

а) Диагностика на стадии описания объектов:

— Много объектов — их число превышает 100.

— Число размерностей — при обращении к *DECOSJ* последний параметр не равен 1,2,3.

— Тип объекта — он не равен 0,1,2,3,4,

— Большая длина — одна из размерностей превышает 16384.

— Имя задано повторно — в одном или в разных обращениях к подпрограммам *DECVAR* и *DECOSJ* имя данных указано два раза.

б) Диагностика на стадии синтаксического разбора (первая часть работы *INFREE*). При возникновении синтаксической ошибки не зависимо от заказа на печать вводимых данных (параметр *MPRI*) распечатывается введенная карта с указанием места ошибки. Печать производится в формате:


```

*** ** * * * *
*           I = 1.85
ERR           ↑↑↑
*** ** * * * *

```

Стрелками указывается место ошибки. Эта печать, естественно, блокируется по признаку $MER = -1$.

в) Диагностика на стадии рассылок введенных данных (вторая часть работы *INFREE*):

– Нет заявок на ввод – обращение к *INFREE* без предварительного обращения к *DECVAR* или *DECOBJ*.

– Много вводимых карт – переполнение внутреннего буфера в *INFREE* объемом в 1200 символов, что соответствует 16 полным (1 – 72) картам.

– Нет левой части. Например, ** EXECUTE*
5., 1E8
** EOR*

– Нет правой части.

– Не описан объект – во вводимой информации задано символическое имя, ранее не определенное.

– Несоответствие размерностей – число граничных пар больше, чем описанное количество размерностей.

– Большой индекс – делается попытка записать данные за декларированные размеры массива, переменной.

– Мал список констант <имя массива> – в граничных парах указанного массива задано данных больше, чем их содержит правая часть.

– Велик список констант <имя массива>.

Например: $A(3:4) = B(1:3) = 1, 2, 3.$

– Рассогласование типов – невозможность преобразовать правую часть к типу значения левой.

Диагностику п. 3 можно устраним, подложив в пакет перед картой ** EXECUTE* следующую подпрограмму:

```

*FORTRAN

```

```

BLOCK DATA
COMMON /CODECØ/ LEN, IND, IXX(4)
* , BUF (... )

```

```
DATA LEN /.../ , IND /.../ , IXX /.../
END
```

Вместо многоточия необходимо задать требуемую длину. Стандартная длина равна 200.

4. О РЕАЛИЗАЦИИ СИСТЕМЫ БЕСФОРМАТНОГО ВВОДА

Система бесформатного ввода реализована на ЭВМ БЭСМ-6 и используется как самостоятельно, так и в качестве подсистемы в системе автоматизации диалоговых разработок [3].

Большинство подпрограмм системы написано на ФОРТРАНе, остальные на автокоде *MADLEN*. Системные подпрограммы форматного обмена не используются, что дает значительную экономию памяти. Несколько более медленная работа предлагаемой системы, чем подпрограмм форматного обмена, служит ценой за предоставляемый сервис и удобство.

Список литературы

1. Язык ФОРТРАН/Под ред. В.П. Ширикова. Дубна, 1979.
2. Мазный Г.Л. Программирование на БЭСМ-6 в системе "Дубна". М.: Наука, 1978.
3. Петров В.А. Реализация диалога в проблемно-ориентированной системе математического обеспечения. — В кн.: Материалы 2-го Всесоюз. совещ. "Диалоговые вычислительные комплексы". Серпухов, 1979.

Редактор Л.И. Кирюхина
Технический редактор Н.И. Мазасва

Т-29858. 27.11.81. Формат 60x90/16. Уч.-изл. л. 0.5
Тираж 122. Цена 10 коп. Индекс 3624. Заказ 601

Отпечатано в ИАЗ

РУБРИКАТОР ПРЕПРИНТОВ ИАЭ

- 1. Общая теоретическая и математическая физика**
- 2. Ядерная физика**
- 3. Общие проблемы ядерной энергетики**
- 4. Физика и техника ядерных реакторов**
- 5. Методы и программы расчета ядерных реакторов**
- 6. Теоретическая физика плазмы**
- 7. Экспериментальная физика плазмы и управляемый термоядерный синтез**
- 8. Проблемы термоядерного реактора**
- 9. Физика конденсированного состояния вещества**
- 10. Физика низких температур и техническая сверхпроводимость**
- 11. Радиационная физика твердого тела и радиационное материаловедение**
- 12. Атомная и молекулярная физика**
- 13. Химия и химическая технология**
- 14. Приборы и техника эксперимента**
- 15. Автоматизация и методы обработки экспериментальных данных**
- 16. Вычислительная математика и техника**

Индекс рубрики дается через дробь после основного номера ИАЭ.

Препринт ИАЭ-3555/16. М., 1982