

7100

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

ДУБНА



7100

Экз. чит. зала

11 - 7100

С.Х. Бычваров

ОПИСАНИЕ КОНЦЕПЦИИ  
СТАНДАРТНОГО УМОЛЧАНИЯ  
В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PL/1

**1973**

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКИ И АВТОМАТИЗАЦИИ

11 - 7100

С.Х. Бычваров

ОПИСАНИЕ КОНЦЕПЦИИ  
СТАНДАРТНОГО УМОЛЧАНИЯ  
В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PL/1

**Научно-техническая  
библиотека  
ОИЯИ**

Концепция стандартного умолчания определяется в языке программирования PL/1 /2/ как концепция умолчания, которая вероятнее всего потребуется программисту. Если некоторому программисту не подходит концепция стандартного умолчания, он может ее изменить при помощи оператора DEFAULT. Таким образом, программист будет использовать язык PL/1 с концепцией умолчания, самой подходящей для его конкретных применений.

Информация о величинах в языке PL/1 накапливается последовательно из следующих источников: явные объявления, контекстуальные объявления, неявные объявления, операторы DEFAULT, правила стандартного умолчания (задающие концепцию стандартного умолчания).

Описание концепции умолчания в /1,2/ имеет ряд недостатков. Поэтому были выделены основные понятия /3/, которые являются удобными для ее описания. В работе /4/ предложен конкретный язык описания концепции умолчания. Средства этого языка являются конкретным представлением выделенных основных понятий. При реализации концепции умолчания в трансляторах удобно программу (определяющую концепцию умолчания в терминах выделенных основных понятий) эквивалентно преобразовывать в таблицу. Синтаксис и семантика таблицы изложены в работе /5/. Найдены критерии /6/, которым должна удовлетворять программа для того, чтобы ее можно было эквивалентно преобразовывать в таблицу. Предложен алгоритм /5,6/ перевода программы в таблицу.

Оператор DEFAULT содержит конструкцию "VALUE (спецификация -значения)", которая в принципе дает больше возможностей при задании концепции умолчания, чем конструкция "описатель . . .". По нашему твердому убеждению, в полном языке PL/1 конструкцию "описатель . . ." нужно расширить до принципиальных возможностей конструкции "VALUE (спецификация-значения)".

В данной работе приведена программа на конкретном языке и таблица, определяющие концепцию стандартного умолчания в языке PL/1.

Любому элементу в этой программе (то есть любому элементу из множества альтернативных элементов) или любому элементу в этой таблице соответствует некоторое понятие в языке PL/1. Следующим элементам соответствуют описатели в языке PL/1 (имя описателя приведено в круглых скобках вслед за именем элемента, если они различны между собой; в случае надобности даются и некоторые пояснения): ALIGNED, AREA, AUTOMATIC, BACKWARDS, BASED, BINARY (описатели U и V для констант и описатель BINARY), BINARY-interpreted (описатель B для констант), BIT, BITSTREAM, BUFFERED, BUILTIN, CHARACTER, COMPLEX (описатель I для констант и описатель COMPLEX), CONNECTED, CONTROLLED, DECIMAL (описатель D для констант и описатель DECIMAL), DEFINED, Dimension, DIRECT, ENTRY, ENVIRONMENT, EVENT, EXCLUSIVE, EXTERNAL, FILE, FIXED (описатель F для констант и описатель FIXED), FLOAT (описатель E для констант и описатель FLOAT), GENERIC, INITIAL, INPUT, INTERNAL, IRREDUCIBLE, KEYED, LABEL, L, Length, Number-of-digits (описатель p для констант и левая часть описателя Precision), OFFSET, OPTIONS, OUTPUT, Parameter, PICTURE-C (описатель PICTURE для данных типа символьных строк), PICTURE-N (описатель PICTURE для числовых данных), POINTER, PRINT, REAL, RECORD, REDUCIBLE, RETURNS, SECONDARY, SEQUENTIAL, Scale-factor (описатель q для констант и правая часть описателя Precision), Size, STATIC, STREAM, TASK, TRANSIENT, UNALIGNED, UNBUFFERED, UPDATE, VARIABLE, VARYING.

Элементы NONCONTROLLED и UNKEYED соответствуют нововведенные описатели NONCONTROLLED и UNKEYED (элементы CONTROLLED и NONCONTROLLED, а также элементы KEYED и UNKEYED являются подмножествами некоторых множеств альтернативных элементов). Это расширение языка PL/1 вызвано естественным применением принятого метода <sup>13/</sup> описания концепции умолчания.

Приведем соответствие между остальными элементами в программе и понятиями в языке PL/1.

Элемент	Понятие в языке PL/1
Arithmetic	арифметическая константа
Base-element	основной элемент структуры
Condition	ситуация
Constant	константа
DECIMAL-interpreted	десятичная интерпретация арифметической константы
Descriptor	список описателей параметра
I-N	идентификатор, начинающийся с букв от I по N
Major-structure	старшая структура
Minor-structure	младшая структура
N-I-N	идентификатор, который не начинается с букв от I по N, либо нет идентификатора.
RETURNS-attr-list	список описателей возвращаемого значения
String	строковая константа

Чтобы упростить программу и таблицу (которые определяют концепцию стандартного умолчания в языке программирования PL/1), расширена совокупность допустимых множеств элементов <sup>14/</sup>. Например, множества {RETURNS-attr-list, OPTIONS}, {Descriptor, INITIAL}, {STATIC, INITIAL, LABEL}, {STATIC, INITIAL, POINTER} и {STATIC, INITIAL, OFFSET} являются допустимыми <sup>12/</sup>.

В программе, приведенной в параграфе I, только следующие правила стандартного умолчания

```
{DECIMAL | BINARY | REAL | COMPLEX} ADD FLOAT ;
{FLOAT | FIXED | REAL | COMPLEX} ADD DECIMAL ;
```

не удовлетворяют критериям возможности перевода программы в таблицу. Следовательно, выполнение программы эквивалентно последовательному выполнению этих правил стандартного умолчания и действий, которые определяются таблицей.

Реализация концепции умолчания при помощи таблицы упрощается, если во всех определенных ею множествах порядок элементов один и тот же. Приведенная в параграфе 2 таблица обладает этим свойством.

I. Программа на конкретном языке, которая определяет концепцию стандартного умолчания в языке PL/1.

```

PROCEDURE Standard-default-rules;
Condition-constant {{ I-N | N-I-N }, Condition } ;
BUILTIN-GENERIC-LABEL-constant {{ I-N | N-I-N },
    { BUILTIN | GENERIC | Constant, LABEL } } ;
String-constant { Constant, String };
Arithmetic-constant { Constant, Arithmetic };
RETURNS-attr-list {{ I-N | N-I-N }, RETURNS-attr-list };
O2 I-N-ALIGNED;
    O3 TEFLOPA { TASK | EVENT | FILE | LABEL | OFFSET |
        POINTER | AREA };
    O3 ENTRY-VARIABLE { ENTRY | IRREDUCIBLE | REDUCIBLE |
        OPTIONS | RETURNS };
O2 String { CHARACTER | BIT | VARYING | Length | PICTURE - C };
O2 Arithmetic ;
Base-element {{ I-N | N-I-N }, Base-element };
O2 I-N-ALIGNED ;
    O3 TEFLOPA { TASK | EVENT | FILE | LABEL | OFFSET |
        POINTER | AREA };
    O3 ENTRY-VARIABLE { ENTRY | IRREDUCIBLE |
        REDUCIBLE | OPTIONS | RETURNS };
O2 String { CHARACTER | BIT | VARYING | Length | PICTURE-C };
O2 Arithmetic ;
Minor-structure {{ I-N | N-I-N }, Minor-structure };
Major-structure {{ I-N | N-I-N }, Major-structure };
O2 Major-structure-parameter-descriptor { Parameter |
    Descriptor };

```

```

O2 Major-structure-level-one-variable ;
VARIABLE- dimension { I-N | N-I-N };
O2 I-N-ALIGNED ;
    O3 TEFLOPA { TASK | EVENT | FILE, VARIABLE |
        LABEL | OFFSET | POINTER | AREA };
    O4 Scope-storage-class-parameter-descriptor
        { Parameter | Descriptor };
    O4 Scope-storage-class ;
O3 ENTRY-VARIABLE {{ ENTRY | IRREDUCIBLE |
    REDUCIBLE | OPTIONS | RETURNS },
    { AUTOMATIC | BASED | CONNECTED |
    CONTROLLED | DEFINED | Descriptor |
    Dimension | INITIAL | NONCONTROLLED |
    Parameter | SECONDARY | STATIC |
    UNALIGNED | VARIABLE }} ;
    O4 Scope-storage-class-parameter-descriptor
        { Parameter | Descriptor };
    O4 Scope-storage-class ;
ENTRY-constant {{ I-N | N-I-N }, { ENTRY | REDUCIBLE |
    IRREDUCIBLE | OPTIONS | RETURNS }} ;
FILE-constant {{ I-N | N-I-N }, FILE } ;
VARIABLE-dimension { I-N | N-I-N };
O2 String { CHARACTER | BIT | Length | VARYING | PICTURE-C };
    O3 Scope-storage-class-parameter-descriptor { Parameter |
        Descriptor } ;
    O3 Scope-storage-class ;
O2 Arithmetic ;

```

```

03 Scope-storage-class-parameter-descriptor
  {Parameter | Descriptor} ;
03 Scope-storage-class ;
END Standard-default-rules ;

SUBROUTINE Condition-constant ;
ALT {I-N, N-I-N | Constant | ALIGNED | Condition | EXTERNAL} ;
ADD {Constant | ALIGNED | EXTERNAL} ;
END Condition-constant ;

SUBROUTINE BUILTIN-GENERIC-LABEL-constant ;
ALT {I-N, N-I-N | Constant | ALIGNED | BUILTIN, GENERIC,
  LABEL | INTERNAL} ;
ADD {Constant | ALIGNED | INTERNAL} ;
END BUILTIN-GENERIC-LABEL-constant ;

SUBROUTINE String-constant ;
ALT {Constant | String | UNALIGNED | CHARACTER, BIT | Length} ;
ADD {UNALIGNED | CHARACTER | Length} ;
END String-constant ;

SUBROUTINE Arithmetic-constant ;
ALT {Constant | Arithmetic | ALIGNED | REAL, COMPLEX |
  L, COMPLEX | DECIMAL, BINARY | L, BINARY |
  DECIMAL-interpreted, BINARY-interpreted |
  L, BINARY-interpreted | Number-of-digits |
  FIXED, FLOAT | Scale-factor, FLOAT | L, FLOAT} ;

```

```

ADD {ALIGNED | REAL | DECIMAL | DECIMAL-interpreted |
  Number-of-digits | Scale-factor | FIXED} ;
END Arithmetic-constant ;

SUBROUTINE RETURNS-attr-list ;
ALT RETURNS-attr-list ;
END RETURNS-attr-list ;

SUBROUTINE I-N-ALIGNED ;
ALT {I-N, N-I-N | ALIGNED, UNALIGNED} ;
ADD ALIGNED ;
END I-N-ALIGNED ;

SUBROUTINE TEFLOFA ;
ALT {TASK, EVENT, FILE, LABEL, OFFSET, POINTER,
  {Size | AREA}} ;
ADD Size ;
END TEFLOPA ;

SUBROUTINE ENTRY-VARIABLE ;
ALT {ENTRY | IRREDUCIBLE, REDUCIBLE | OPTIONS | RETURNS} ;
ADD OPTIONS ;
ADD {ENTRY | IRREDUCIBLE | RETURNS} ;
END ENTRY-VARIABLE ;

```

```

SUBROUTINE String ;
ALT {I-N, N-I-N | ALIGNED, UNALIGNED | CHARACTER, BIT,
    PICTURE-C | VARYING PICTURE-C | Length, PICTURE-C } ;
ADT VARYING ;
ADD { UNALIGNED | CHARACTER | Length } ;
END String ;

SUBROUTINE Arithmetic ;
ALT {I-N, N-I-N | ALIGNED, UNALIGNED | REAL,
    COMPLEX | DECIMAL, BINARY, PICTURE-N |
    Number-of-digits, PICTURE-N | FLOAT, Scale-factor,
    PICTURE-N | FLOAT, FIXED, PICTURE-N } ;
{DECIMAL | BINARY | REAL | COMPLEX } ADD FLOAT ;
{FLOAT | FIXED | REAL | COMPLEX } ADD DECIMAL ;
I-N ADD {REAL | FIXED | BINARY} ;
N-I-N ADD {REAL | FLOAT | DECIMAL} ;
PICTURE-N ADD UNALIGNED ;
ADD {ALIGNED | Number-of-digits | Scale-factor | FIXED} ;
END Arithmetic ;

SUBROUTINE Base-element ;
ALT {Base-element | VARIABLE | Dimension | INTERNAL | INITIAL} ;
ADT Dimension, INITIAL ;
ADD {VARIABLE | INTERNAL} ;
END Base-element ;

```

```

SUBROUTINE Minor-structure ;
ALT {ALIGNED, UNALIGNED | I-N, N-I-N | Minor-structure |
    VARIABLE | Dimension | INTERNAL} ;
ADT ALIGNED, UNALIGNED, Dimension ;
ADD {VARIABLE | INTERNAL} ;
END Minor-structure ;

SUBROUTINE Major-structure ;
ALT {I-N, N-I-N | ALIGNED, UNALIGNED | Major-structure |
    VARIABLE | Dimension} ;
ADT ALIGNED, UNALIGNED, Dimension ;
ADD VARIABLE ;
END Major-structure ;

SUBROUTINE Major-structure-parameter-descriptor ;
ALT {Parameter, Descriptor | INTERNAL | CONNECTED, CONTROLLED} ;
ADT CONNECTED, CONTROLLED ;
ADD INTERNAL ;
END Major-structure-parameter ;

SUBROUTINE Major-structure-level-one-variable ;
ALT {SECONDARY | INTERNAL, EXTERNAL | AUTOMATIC, BASED,
    CONTROLLED, STATIC, DEFINED | AUTOMATIC, BASED,
    DEFINED, EXTERNAL} ;
ADT SECONDARY ;
ADD {AUTOMATIC | INTERNAL | STATIC} ;
END Major-structure-level-one-variable ;

```

```

SUBROUTINE VARIABLE-dimension ;
ALT {VARIABLE | Dimension} ;
ADT Dimension ;
ADD VARIABLE ;
END VARIABLE-dimension ;

SUBROUTINE Scope-storage-class-parameter-descriptor ;
ALT {Parameter, Descriptor | INTERNAL | {NONCONTROLLED |
CONNECTED} , {CONTROLLED | INITIAL}} ;
ADT CONNECTED, INITIAL ;
ADD {INTERNAL | NONCONTROLLED} ;
END Scope-storage-class-parameter-descriptor ;

SUBROUTINE Scope-storage-class ;
ALT {SECONDARY | INTERNAL, EXTERNAL | AUTOMATIC, BASED,
CONTROLLED, STATIC, DEFINED | AUTOMATIC, BASED,
DEFINED, EXTERNAL | INITIAL, DEFINED} ;
ADT SECONDARY, INITIAL ;
ADD {AUTOMATIC | INTERNAL | STATIC} ;
END Scope-storage-class ;

SUBROUTINE ENTRY-constant ;
ALT {I-N, N-I-N | Constant | ALIGNED | ENTRY |
IRREDUCIBLE, REDUCIBLE | OPTIONS | RETURNS |
EXTERNAL, INTERNAL} ;
ADT OPTIONS ;
ADD {Constant | ALIGNED | ENTRY | IRREDUCIBLE | RETURNS | EXTERNAL} ;
END ENTRY-constant ;

```

```

SUBROUTINE FILE-constant ;
ALT {I-N, N-I-N | Constant | ALIGNED | FILE | EXTERNAL,
INTERNAL | ENVIRONMENT | STREAM, BITSTREAM,
{RECORD | SEQUENTIAL, DIRECT, TRANSIENT |
BUFFERED, UNBUFFERED | TRANSIENT, UPDATE |
KEYED, UNKEYED | BACKWARDS, DIRECT, TRANSIENT |
EXCLUSIVE, SEQUENTIAL, TRANSIENT} | INPUT,
OUTPUT, UPDATE | UNKEYED, DIRECT | BACKWARDS,
OUTPUT, UPDATE | EXCLUSIVE, {INPUT, OUTPUT |
UNKEYED | BACKWARDS, PRINT} | PRINT, {INPUT,
UPDATE | BITSTREAM, RECORD | SEQUENTIAL, DIRECT,
TRANSIENT | BUFFERED, UNBUFFERED | KEYED, UNKEYED} ;
ADT PRINT, BACKWARDS, EXCLUSIVE, ENVIRONMENT ;
ADD {Constant | ALIGNED | EXTERNAL | STREAM | INPUT |
OUTPUT | UPDATE | SEQUENTIAL | DIRECT | UNKEYED |
KEYED | RECORD} ;
DIRECT ADD UNBUFFERED ;
ADD BUFFERED ;
END FILE-constant ;

```

2. Таблица, которая определяет концепцию стандартного умолчания в языке PL/I.

Замечание. В приведенной таблице знак " " имеет тот же смысл, что и знак " U " /5/ .



1 = Scale-factor, FIXED  
 2 = N-I-N, {DECIMAL | BINARY}, {FLOAT | 1}  
 3 = I-N, {BINARY | DECIMAL}, {1 | FLOAT}  
 4 = {ALIGNED | UNALIGNED}, Number-of-digits, {2 | 3}  
 5 = {UNALIGNED | ALIGNED}, {I-N | N-I-N}, PICTURE-N  
 6 = {REAL | COMPLEX}, {4 | 5}  
 7 = {CHARACTER | BIT}, [VARYING], Length  
 8 = {UNALIGNED | ALIGNED}, {I-N | N-I-N}, {7 | PICTURE-C}  
 9 = {AUTOMATIC | BASED | CONTROLLED | STATIC}, [INITIAL]  
 10 = [SECONDARY], INTERNAL, {9 | DEFINED}  
 11 = [SECONDARY], EXTERNAL, {STATIC | CONTROLLED}, [INITIAL]  
 12 = NONCONTROLLED, [CONNECTED]  
 13 = CONTROLLED, [INITIAL]  
 14 = {Parameter | Descriptor}, INTERNAL, {12 | 13}  
 15 = VARIABLE, [Dimension], {6 | 8}, {10 | 11 | 14}  
 16 = [PRINT], OUTPUT  
 17 = STREAM, {INPUT | 16}  
 18 = BITSTREAM, {INPUT | OUTPUT}  
 19 = INPUT, {UNKEYED | KEYED}, [BACKWARDS]  
 20 = OUTPUT, {UNKEYED | KEYED}  
 21 = UPDATE, {UNKEYED | KEYED}  
 22 = SEQUENTIAL, {BUFFERED | UNBUFFERED}, {19 | 20 | 21}  
 23 = INPUT, KEYED  
 24 = OUTPUT, KEYED  
 25 = UPDATE, [EXCLUSIVE], KEYED  
 26 = DIRECT, {UNBUFFERED | BUFFERED}, {23 | 24 | 25}  
 27 = INPUT, {UNKEYED | KEYED}

28 = TRANSIENT, {BUFFERED | UNBUFFERED}, {27 | 20}  
 29 = RECORD, {22 | 26 | 28}  
 30 = Constant, ALIGNED, {I-N | N-I-N}, FILE, {EXTERNAL | INTERNAL}, [ENVIRONMENT], {17 | 18 | 29}  
 31 = ENTRY, {IRREDUCIBLE | REDUCIBLE}, [OPTIONS], RETURNS  
 32 = Constant, ALIGNED, {I-N | N-I-N}, 31, {EXTERNAL | INTERNAL}  
 33 = {TASK | EVENT | FILE | LABEL | OFFSET | POINTER}  
 34 = Size, AREA  
 35 = {ALIGNED | UNALIGNED}, {I-N | N-I-N}, {31 | 33 | 34}  
 36 = VARIABLE, [Dimension], 35, {10 | 11 | 14}  
 37 = INTERNAL, {AUTOMATIC | BASED | CONTROLLED | STATIC | DEFINED}  
 38 = EXTERNAL, {STATIC | CONTROLLED}  
 39 = [SECONDARY], {37 | 38}  
 40 = {Parameter | Descriptor}, INTERNAL, {CONNECTED | CONTROLLED}  
 41 = VARIABLE, [Dimension], [ALIGNED | UNALIGNED], {I-N | N-I-N}  
 42 = Major-structure, 41, {39 | 40}  
 43 = Minor-structure, 41, INTERNAL  
 44 = Base-element, VARIABLE, [Dimension], {6 | 8 | 35}, INTERNAL, [INITIAL]  
 45 = RETURNS-attr-list, {6, 8 | 35}  
 46 = {REAL | COMPLEX}, ALIGNED, Number-of-digits, {DECIMAL | BINARY}, {DECIMAL-interpreted | BINARY-interpreted}, {1 | FLOAT}  
 47 = L, REAL, ALIGNED, Number-of-digits, DECIMAL, DECIMAL-interpreted, 1  
 48 = Constant, Arithmetic, {46 | 47}

49 = Constant, String, UNALIGNED, { CHARACTER | BIT } ,  
Length  
50 = Constant, ALIGNED, { I-N | N-I-N }  
51 = 50, { LABEL | GENERIC | BUILTIN } , INTERNAL  
52 = 50, Condition, EXTERNAL  
53 = { 15 | 30 | 31 | 36 | 42 | 43 | 44 | 45 | 48 | 49 | 51 | 52 }

Автор выражает благодарность Н.Н.Говоруно и В.П.Ширикову за помощь и внимание к работе, Е.А.Жоголеву за полезные обсуждения, Г.Л. Мазному - за редакционные замечания.

#### Л И Т Е Р А Т У Р А

1. Универсальный язык программирования PL/1  
(Перевод с английского под редакцией В.М.Курочкина, Москва, 1968).
2. PL/1 Language Specifications  
Order Number GY 33-6003-2 (Major Revision June 1970) IBM 1970 .
3. С.Х.Бычваров, ОИЯИ, II-6737, Дубна, 1972 г.
4. С.Х.Бычваров, ОИЯИ, II-6738, Дубна, 1972 г.
5. С.Х.Бычваров, ОИЯИ, II-6429, Дубна, 1972 г.
6. С.Х.Бычваров, ОИЯИ, II-6875, Дубна, 1973 г.

Рукопись поступила в издательский отдел  
23 апреля 1973 года.