

СИСТЕМА ВВОДА/ВЫВОДА А

ЭЛЕМЕНТЫ ОПЕРАТОРОВ ВВОДА/ВЫВОДА

ЭЛЕМЕНТЫ ОПЕРАТОРОВ ВВОДА/ВЫВОДА.



Различные операторы ввода/вывода имеют параметры и аргументы, определяющие источники и цели передачи данных, а так же другие особенности операции ввода/вывода. Мы рассмотрим следующие элементы:

- 1. Спецификация устройства.*
- 2. Спецификация формата.*
- 3. Список ввода/вывода.*

Спецификация устройства.

Спецификация устройства в операторе ввода/вывода может принимать одну из следующих форм:

● *-

спецификация

WRITE (*,*) 'Экран'

- *Первая * относится к клавиатуре или к экрану и определяет устройство.*

СПЕЦИФИКАЦИЯ УСТРОЙСТВА

● целое
выражен
ие

WRITE (10,*) 'Файл 10:'

- *Целое число относится к внешнему файлу, связанному с номером устройства 10 (* означает устройство номер ноль).*
- *Допустимы спецификации устройства в диапазоне от -32767 до 32767.*

● СИМВОЛЬН
ая
переменн
ая

**CHARACTER*10 STRING
WRITE (STRING, '(I10)') IVAL**

- *Символьная переменная STRING относится к внутреннему файлу.*

СПЕЦИФИКАЦИЯ УСТРОЙСТВА

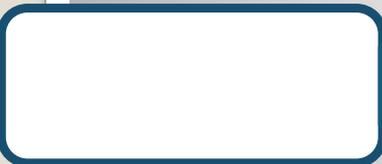
Спецификация формата.

Спецификация формата в операторе ввода/вывода может принимать одну из следующих форм:



● Метка
оператора

```
WRITE (*,990) I, J, K  
990  FORMAT (1X,2I5,I3)
```



● Имя
целой
переменной

```
ASSIGN 990 TO IFMT  
990  FORMAT (1X,2I5,I3)  
WRITE (*,IFMT) I, J, K
```

СПЕЦИФИКАЦИЯ ФОРМАТА

● Символьное выражение

```
WRITE (*,'(1X,2I5,I3)') I,J,K
```

Значением символьного выражения является спецификация формата.

● Символьная переменная

```
CHARACTER * 11 FMTCH
```

```
FMTCH = '(1X,2I5,I3)'
```

```
WRITE (*, FMTCH) I, J, K
```

● * спецификация

```
WRITE (*,*) I, J, K
```

*Выражение * указывает на передачу данных при вводе/выводе списком.*

СПЕЦИФИКАЦИЯ ФОРМАТА

Список ввода/вывода.

- *Список ввода/вывода определяет данные, значения которых передаются операторами READ и WRITE.*
- *Список ввода/вывода может быть и пустым, но обычно состоит из входных/выходных данных и включает в себя списки неявного DO, разделенные запятыми.*
- *Входные данные могут быть определены в списке ввода/вывода оператора READ, а выходные - в списке оператора WRITE.*

СПИСОК ВВОДА/ВЫВОДА

● **Входные
данные**

```
DIMENSION J(10), K(20)  
READ (*,*) I, J[2], K
```

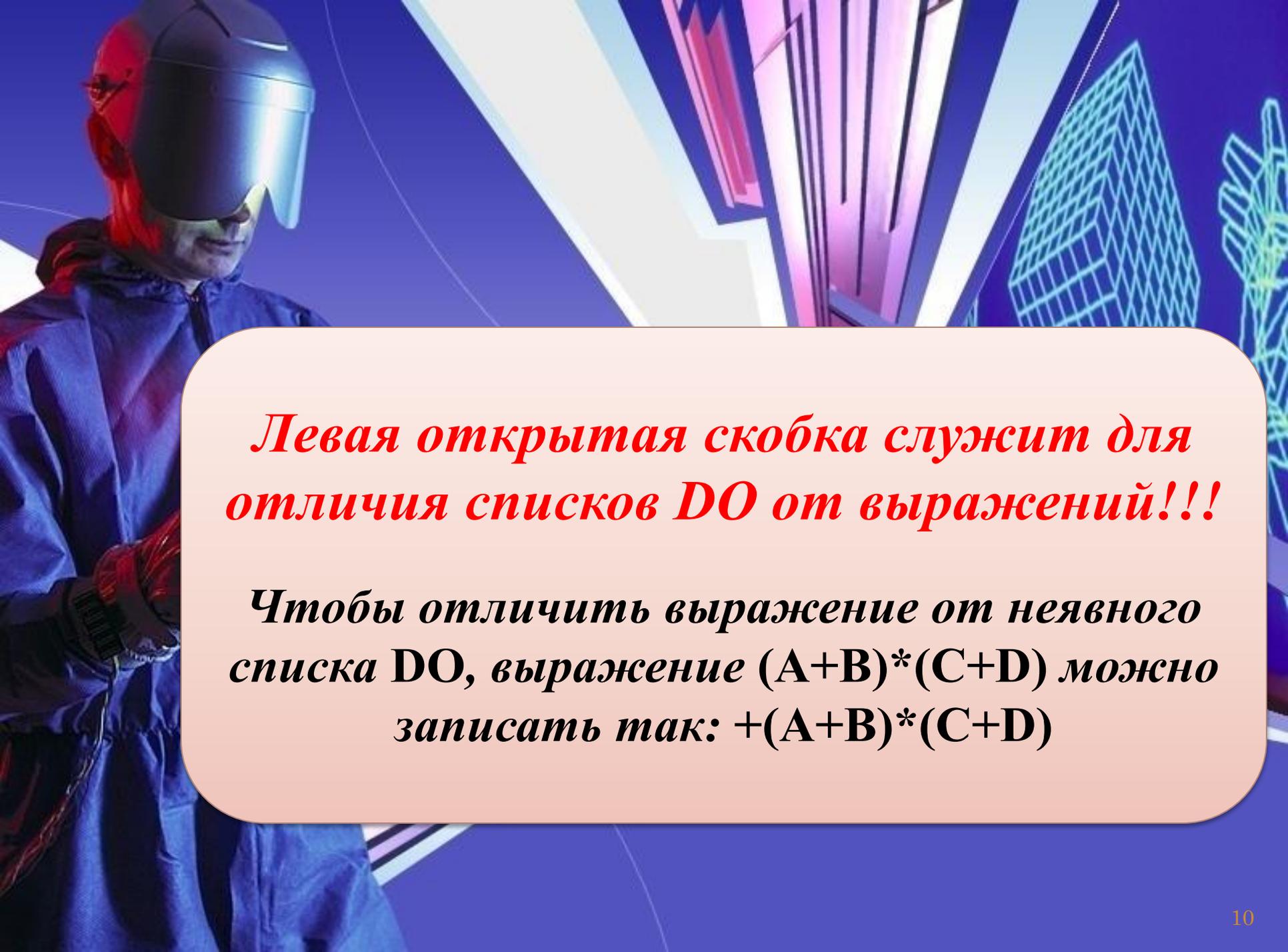
- *Входные данные представляют собой имя переменной, имя элемента массива или имя массива.*
- *Имя массива определяет все элементы массива, последовательно расположенные в памяти.*

● **Выходны
е данные**

```
DIMENSION J, K  
WRITE (*,*) I, J[2], K  
WRITE (*,*) +(A+B)*(C+D)
```

- *Выходные данные могут иметь такой же вид, как перечисленные входные данные, но могут принимать вид выражений, не начинающихся знаком открытой скобки "(".*

СПИСОК ВВОДА/ВЫВОДА



Левая открытая скобка служит для отличия списков DO от выражений!!!

Чтобы отличить выражение от неявного списка DO, выражение $(A+B)*(C+D)$ можно записать так: $+(A+B)*(C+D)$

● Неявные списки DO

```
DIMENSION K(10)
READ (*,*) N, (K(I), I=3,N)
WRITE (*,*) (K(J), J=1,N), J
```

- Неявные списки DO можно определить, как данные в списке ввода/вывода операторов READ и WRITE.
- Они имеют следующий формат:

(список-в/в, пер=выр1,выр2[,выр3])

СПИСОК-В/В определен так же, как и элементы оператора ввода/вывода (включая списки внутреннего неявного DO).

пер, выр1, выр2 и выр3 определены так же, как и для оператора DO.

СПИСОК ВВОДА/ВЫВОДА

- *но её можно считать тем же оператором READ перед списком неявного DO.*

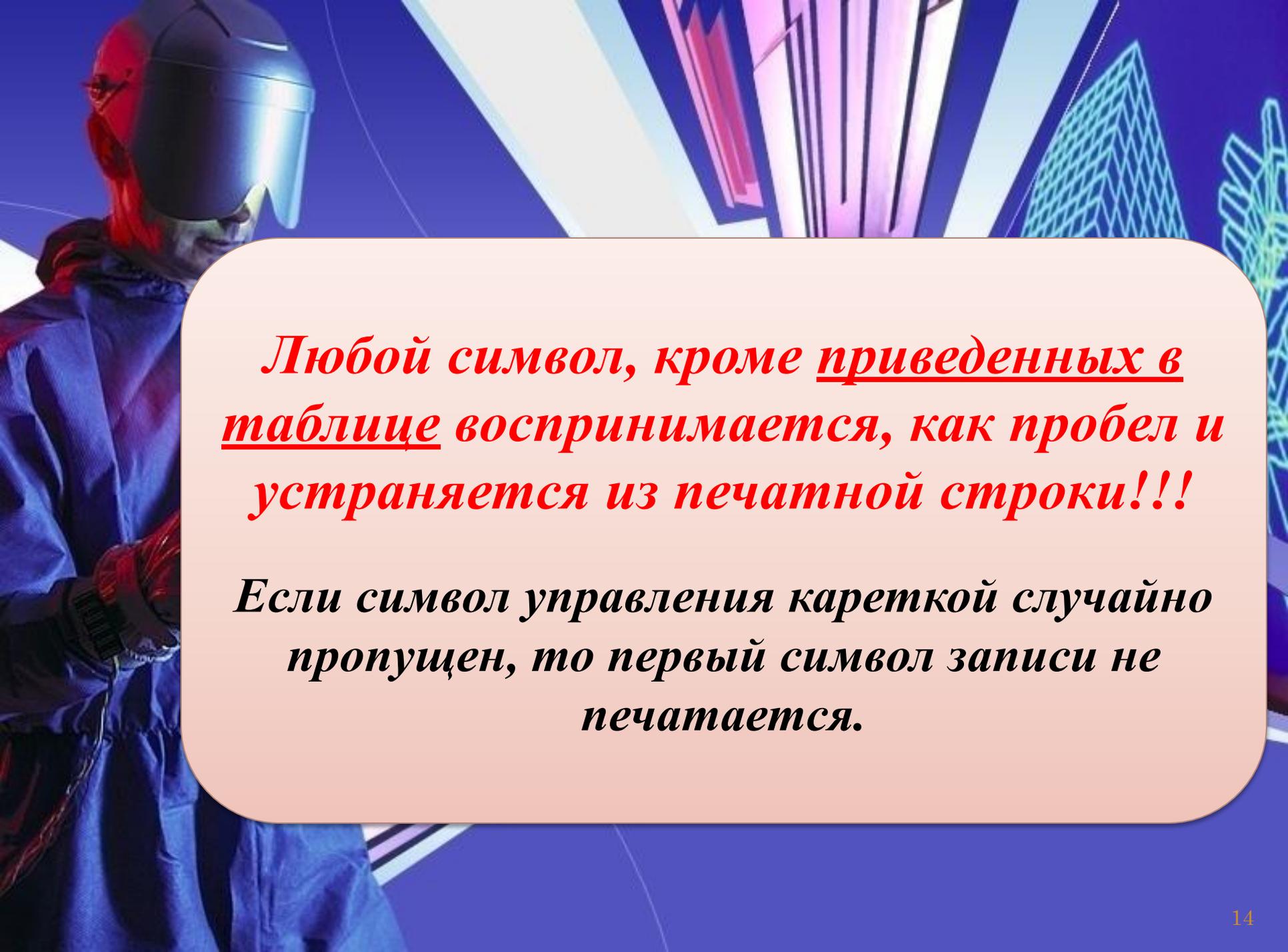
НЕЯВНЫЕ СПИСКИ DO

● Управление кареткой

- *Первый символ любой записи, передаваемый на печать или другое терминальное устройство, включая консоль, не печатается.*
- *Он воспринимается, как символ управления кареткой.*
- *Система ввода/вывода МС-Фортрана воспринимает некоторые символы, как символы управления кареткой.*
- *Эти символы и их действия при печати показаны в таблице:*

СИМВОЛ	ДЕЙСТВИЯ
пробел	Пропуск одной строки
0	Пропуск двух строк
1	Переход в начало следующей строки (игнорируется консолью)
+ (плюс)	Не переходит на следующую строку (можно печатать по этому же месту)

СПИСОК ВВОДА/ВЫВОДА



Любой символ, кроме приведенных в таблице воспринимается, как пробел и устраняется из печатной строки!!!

Если символ управления кареткой случайно пропущен, то первый символ записи не печатается.

ФОРМАТНЫЙ ВВОД/ВЫВОД

ФОРМАТНЫЙ ВВОД/ВЫВОД.



Если в операторе READ или WRITE определён формат, то оператор ввода/вывода считается форматным.



Спецификация формата должна начинаться с открывающей скобки и заканчиваться закрывающей скобкой.



Перед левой скобкой могут стоять пробелы.



Символы после правой скобки игнорируются.

Такой формат может быть определен одним из пяти способов.

Следующие пять примеров представляют собой одинаково приемлемые методы определения формата и демонстрируют спецификации формата, описанные ранее.

```
WRITE (*,990) I,J,K  
990  FORMAT (1X,2I5,I3)
```

```
ASSIGN 990 TO IFMT  
990  FORMAT (1X,2I5,I3)  
WRITE (*,IFMT),I,J,K
```

```
WRITE (*,'(1X,2I5,I3)')I,J,K
```

```
CHARACTER*11 FMTCH  
FMTCH = '(1X,2I5,I3)'  
WRITE (*,FMTCH)I,J,K
```

```
WRITE (*,*) I,J,K
```

ФОРМАТНЫЙ ВВОД/ВЫВОД

Взаимодействие формата и списка ввода / вывода

- Если список ввода/вывода содержит хотя бы одно значение, то в спецификации формата должен присутствовать по крайней мере один повторяемый редакционный дескриптор.
- Пустую спецификацию () можно использовать только в случае, если в списке ввода/вывода не определяются никакие данные (в этом случае **WRITE** делает запись нулевой длины, а **READ** переходит к следующей записи).

При выполнении оператора ввода/вывода

наоборот

- *остальные данные управления форматом взаимодействуют непосредственно с записью и не связаны с данными в списке ввода/вывода*

Повторяемый релакционный дескриптор

Γ раз

- *если Γ пропущен, то коэффициент повторения равен единице*

Когда встречается повторяемый редакционный дескриптор, возникает одна из следующих ситуаций:



редакционный дескриптор



никаких данных, в этом случае "контроль"

Так, для следующих операторов:

```
I=5  
WRITE (*,10) I  
10 FORMAT (1X,'I= ', I5, ', J= ',I5)
```

вывод будет выглядеть следующим образом:

```
I= 5, J=
```

- *Однако, если в списке ввода/вывода ещё есть данные, файл помещается в начало следующей записи и контролер формата продолжает обработку формата, начиная в начале спецификации формата и заканчивая последней правой скобкой.*

- *В просмотренной части формата должен быть хотя бы один повторяемый редакционный дескриптор.*

- *Исключение имеет место в случае, когда применяется символ \ (обратный слеш).*

Редакционные дескрипторы

- Редакционные дескрипторы в Фортране определяют форму записи и управляют связью между знаками записи и внутренним форматом данных.
- Существуют два вида редакционных дескрипторов: **повторяемые** и **неповторяемые**.

Неповторяемые редакционные дескрипторы

- Холлеритовское редактирование (H)
 - Редакционный дескриптор **nH** передает последующие **n** символов, включая пробелы, в выходной блок.
 - *Холлеритовское редактирование нельзя применять для ввода.*

● Апостроф ('xxxx')

- *Редакционный дескриптор апостроф имеет вид знаковой константы и порождает эту знаковую константу, передаваемую на выходное устройство.*
- *Внутренние пробелы учитываются.*
- *Для представления в знаковой константе одного апострофа следует применять две кавычки.*
- *Апостроф в качестве спецификации нельзя применять для ввода.*

Примеры редактирования с апострофом и холлеритовского редактирования:

Каждый **WRITE** выводит знаки между слешами: **/ABC'DEF/**

C Спецификация апостроф

```
WRITE (*,970)
```

```
970 FORMAT ('ABC'DEF')
```

```
WRITE (*,('ABC'DEF'))
```

C Использование спецификации H

```
WRITE (*,('8H ABC'DEF'))
```

```
WRITE (*,960)
```

```
960 FORMAT (8H ABC'DEF)
```

Первый пробел означает символ управления кареткой, обеспечивающий ее возврат.

- **Позиционное редактирование**
- **(Tc, TLc и TRc)**
 - *Редакционные дескрипторы T, TL и TR определяют позицию в записи, в которую, или с которой будет передаваться следующий символ.*

- *Это позволяет обрабатывать запись более одного раза на входе.*

ПОЗИЦИОННОЕ РЕДАКТИРОВАНИЕ

переда
ча
следую
щего

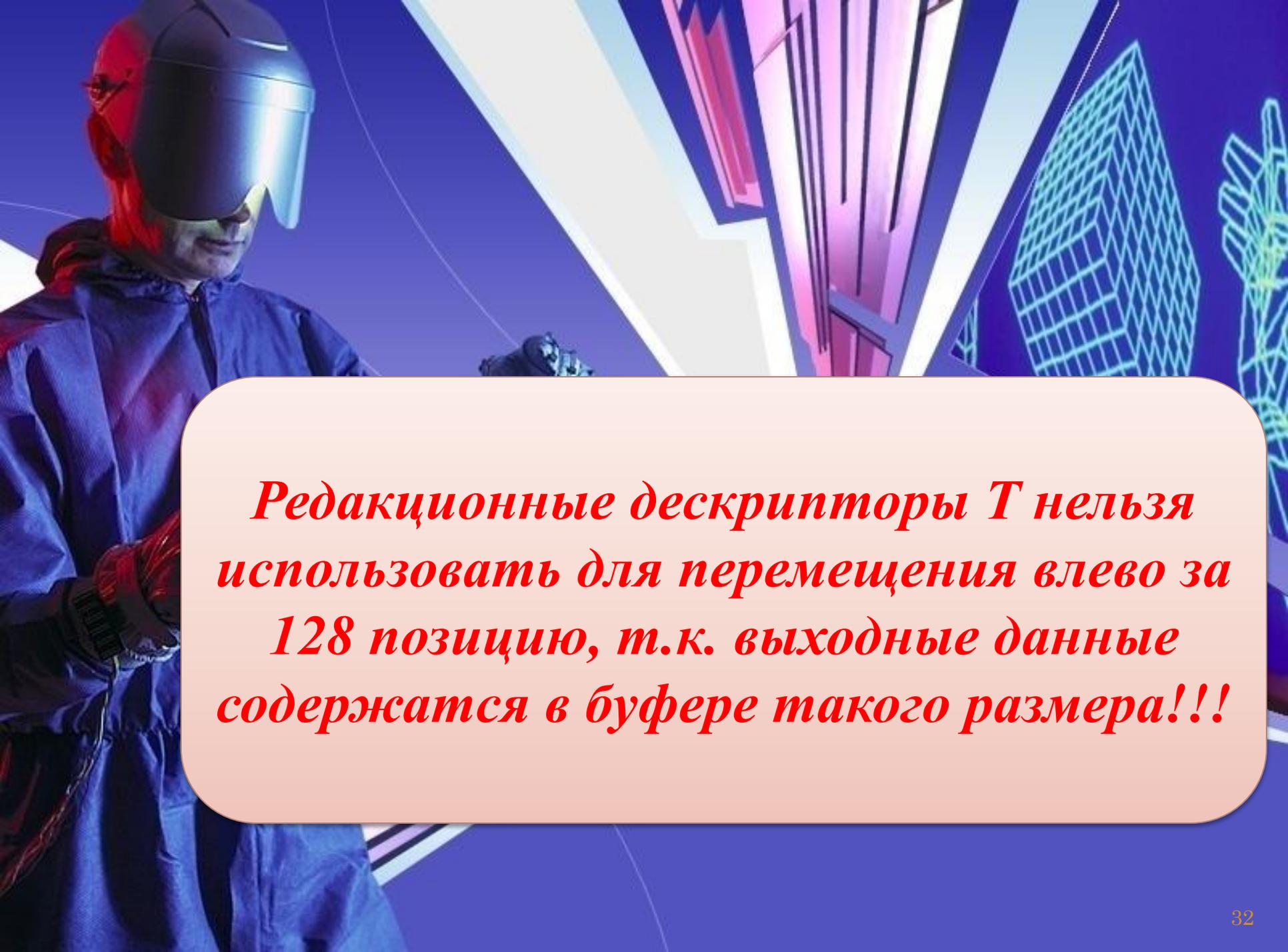
- Знака

следую
щего
знака
должен

проех
одить
на с
знаков
назад
от
текуще
го
полож
ения

- Если
текуще
е
положе
ние
меньше
е или
равно с,
то

ЭЛЕКТРОННОЕ РЕДАКТИРОВАНИЕ



Редакционные дескрипторы T нельзя использовать для перемещения влево за 128 позицию, т.к. выходные данные содержатся в буфере такого размера!!!

- **Позиционное редактирование**
- **(X)**
 - *На входе редакционный дескриптор nX пропускает n знаков.*
 - *На выходе редакционный дескриптор nX пишет n пробелов, а дальше запись будет продолжена.*
 - *Если записи нет, то он ничего не делает символ.*

- **Спецификация для необязательного плюса**
- **(SP,SS и S)**
 - *Редакционные дескрипторы SP, SS и S могут применяться для управления дополнительными знаками "плюс" в цифровых выходных полях.*

- **Слеш**

- **(/)**

- *Слеш означает конец передачи данных текущей записи.*
- *При вводе файл помещается в начало следующей записи.*
- *При выводе записывается конец записи и файл располагается так, чтобы записывать в начало следующей записи.*

- **Обратный слеш**

- **(\)**

- *Обычно при окончании "контроллера формата" происходит конец передачи данных текущей записи.*
- *Если последним редакционным дескриптором, воспринятым "контроллером формата" является обратный слеш (\) то автоматический конец записи запрещается, что позволяет последующим операторам ввода/вывода продолжать чтение (или запись) с (или на) этой же записи.*

Обратный слеш (\) широко используется для получения на экране запроса и ответа на одной и той же строке, например:

```
WRITE (*,'(A\)' ) 'Ввод целой величины->'  
READ (*,'(BN, I6)' )I
```

*•ввод с клавиатуры всегда должен заканчиваться нажатием на **ENTER***

ОБРАТНЫЙ СЛЕШ (\)

- Конец управления форматом
- **(:)**
 - *Двоеточие (:) заканчивает управление форматом, если в списке ввода/вывода больше нет данных.*
 - *Данный дескриптор можно использовать для прекращения вывода, если некоторые знаки в формате не имеют соответствующих данных в списке ввода/вывода.*

- Спецификация с масштабным коэффициентом
- **(P)**
 - *Редакционный дескриптор **кР** устанавливает масштабный множитель для последующих редакционных дескрипторов **F** и **E** до следующего редакционного дескриптора **кР**.*
 - *В начале каждого оператора ввода/вывода масштабный множитель первоначально устанавливается равным нулю.*

Масштабный множитель воздействует на редактирование формата следующим образом:

*• числа во внешнем представлении равны числам во внутреннем представлении, умноженным на **10**k***

*• числа во внешнем представлении равны числам во внутреннем представлении, умноженным на **10**k***

•масштабный множитель не оказывает воздействия

*•действительная часть получается умноженной на 10^{*K} , а порядок уменьшается на K (изменяется положение десятичной точки, а не сама выводимая величина).*

- **Интерпретация пробела**
- **(BN и BZ)**
 - *Эти редакционные дескрипторы определяют интерпретацию пробелов в цифровых вводимых полях.*

- *Это приводит к тому, что пробелы, за исключением ведущих, идентичны нулям.*

ИНТЕРПРЕТАЦИЯ ПРОБЕЛА

При игнорировании пробелов все знаки, не являющиеся пробелами, включаются в поле ввода как, если бы они были сдвинуты вправо ведущим числом пробелов, на число игнорированных пробелов.

*Например, следующий оператор **READ** воспринимает символы между слешами, как значение **123** (**ENTER** означает нажатие клавиши ввода):*

```
READ (*,100)I  
100  FORMAT (BN,I6)
```

```
/123 ENTER/  
/123 456 ENTER/  
/123 123ENTER/
```

ИНТЕРПРЕТАЦИЯ ПРОБЕЛА

- *"короткая" подразумевает, что общее число знаков вводимой записи меньше числа знаков, определенных комбинацией описателей формата и элементами списка ввода/вывода.*

Запись дополняется справа пробелами до требуемой длины.

*Так, следующий пример даст величину **123**, а не **12300**:*

READ (*, '(I5)') I

/123 ENTER/

ИНТЕРПРЕТАЦИЯ ПРОБЕЛА



*Редакционный дескриптор VN
совместно с дополнением форматных
записей пробелами делает
интерактивный ввод очень удобным.*

Повторяемые редакционные дескрипторы

*Спецификации **I, F, E, D** и **G** применяются для ввода/вывода цифровых данных.*

Для всех цифровых спецификаций действуют следующие правила:

• При вводе ведущие значения не имеют.

• Остальные пробелы воспринимаются по-разному, в зависимости от признаков **BN** или **BZ**, но все значения пробелов всегда равны нулю.

• Знак плюс присутствует по желанию.

• Пробелы, присутствующие в системе файла для дополнения записи до требуемой длины, также являются

ПОВТОРЯЕМЫЕ РЕДАКЦИОННЫЕ ДЕСКРИПТОРЫ



оде
пер
при
суде
мве
да
зна
ким
едв
ага
бле
и

чир
лаво
зна
кор
прн
врот
дя
вод
вос

ход
ит
ши
рин
у
пол
я,
то

во
по
еся
за
он

ПОВТОРЯЕМЫЕ РЕДАКЦИОННЫЕ ДЕСКРИПТОРЫ

- При чтении спецификации **I, F, E, D, G** или **L** поле ввода может содержать запятую, окончившую поле.

- *Считывание следующего поля начнется со знака, следующего за запятой.*

- *Пропущенные знаки не существенны.*

Например

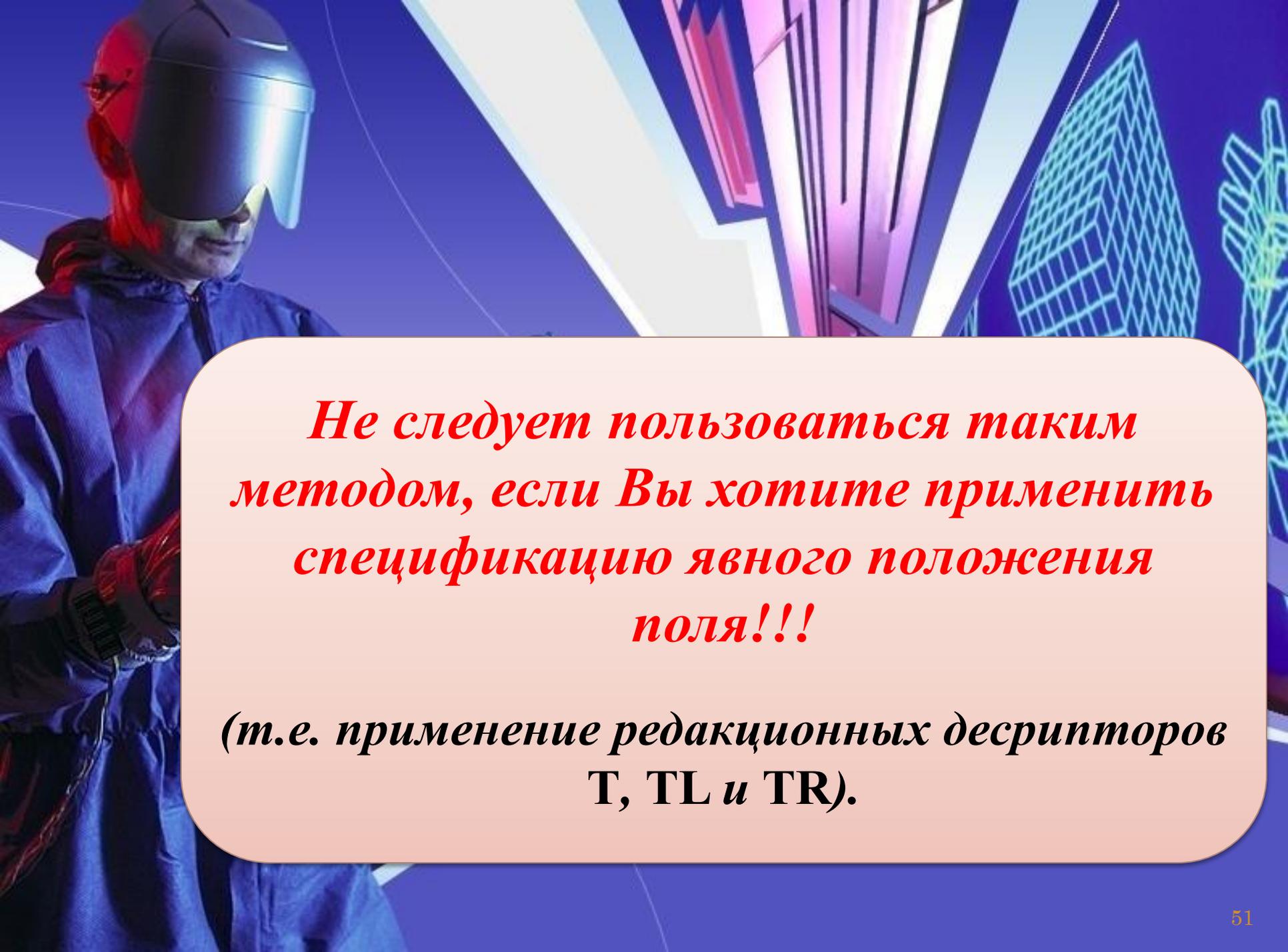
```
READ (*, '(3I5)') I, J, K
```

```
/1,2_,3, ENTER/
```

дает:

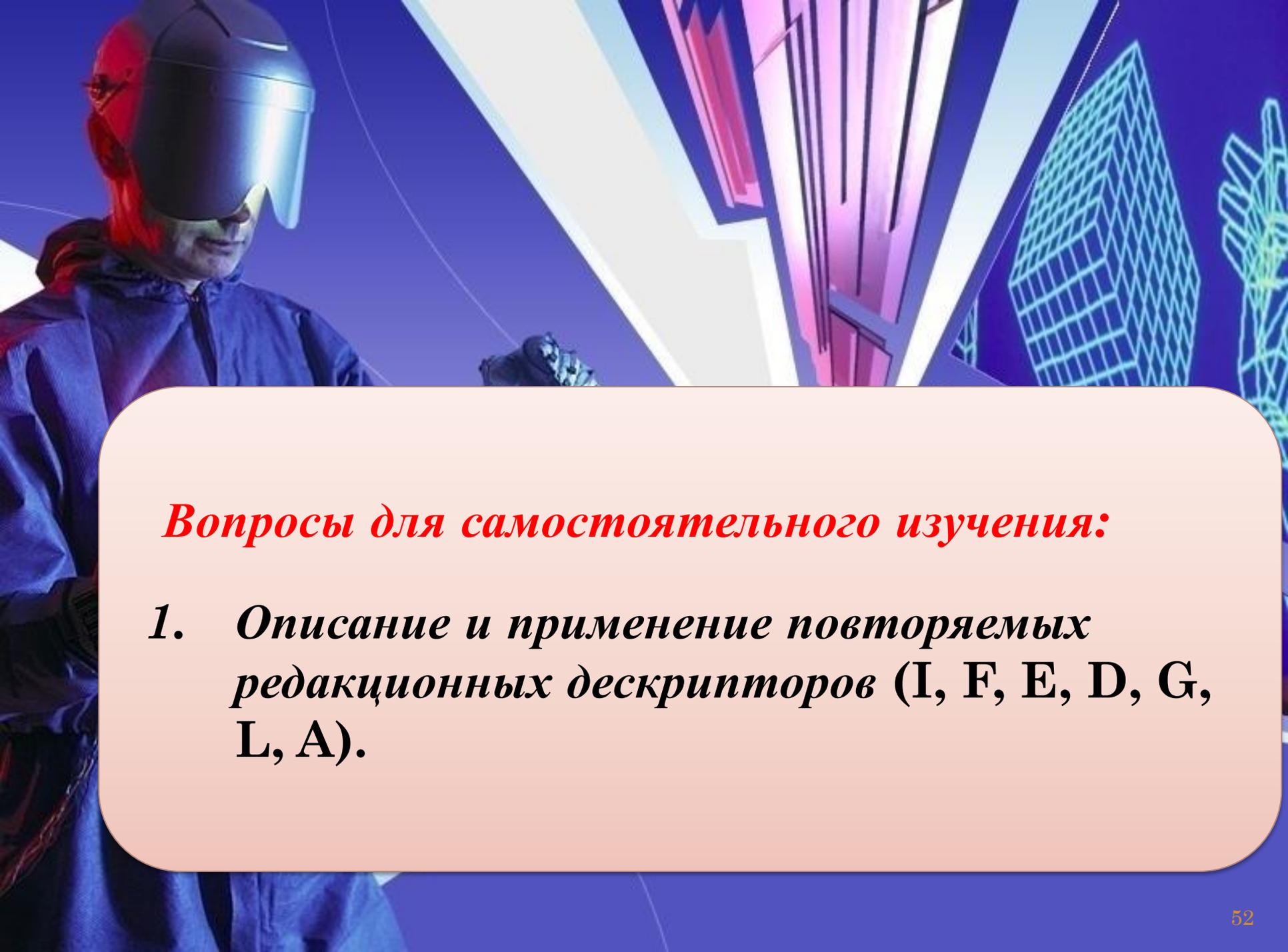
```
I=1, J=20 и K=3
```

ПОВТОРЯЕМЫЕ РЕДАКЦИОННЫЕ ДЕСКРИПТОРЫ



Не следует пользоваться таким методом, если Вы хотите применить спецификацию явного положения поля!!!

(т.е. применение редакционных дескрипторов T, TL и TR).



Вопросы для самостоятельного изучения:

- 1. Описание и применение повторяемых редакционных дескрипторов (I, F, E, D, G, L, A).***