Простая прграмма Операторы языка Операторы языка (2) Размещение операторов

# FORTRAN-77 некоторые операторы

Михайленко К. И.

Башгосуниверситет Институт механики Уфимского научного центра РАН

(2)

```
prog1.f
    program prog1
    read *, a, b, i
    print *, a, b, i
    end
```

```
g g77 -o prog2 prog2.fg ./prog2
```

```
prog2.f

program prog2
implicit none
real a, b
integer i

read *, a, b, i
print *, a, b, i
end
```

```
$ g77 -o prog2 prog2.f
$ ./prog2
```

# Выполнение программы prog2.f

```
prog2.f
    program prog2
    implicit none
    real a, b
    integer i
    read *, a, b, i
    print *, a, b, i
    end
```

```
$ g77 -o prog2 prog2.f
$ ./prog2
-
```

# Выполнение программы prog2.f

```
prog2.f

program prog2
implicit none
real a, b
integer i

read *, a, b, i
print *, a, b, i
end
```

```
$ g77 -o prog2 prog2.f
$ ./prog2
.3 2.5 8_
```

# Выполнение программы prog2.f

```
prog2.f

    program prog2
    implicit none
    real a, b
    integer i

    read *, a, b, i
    print *, a, b, i

end
```

```
$ g77 -o prog2 prog2.f
$ ./prog2
.3 2.5 8
   0.300000012 2.5 8
$ _
```

Onepaтop program
Oneparop end
Onepatopы описания типа
Onepatop read
Onepatop print

# Оператор program

#### Форма записи:

- Оператор поименования программы
- Всегда записывается первым
- Не является обязательным

# Оператор program

#### Форма записи:

- Оператор поименования программы
- Всегда записывается первым
- Не является обязательным

Onepaтop program
Onepaтop end
Onepatopы описания типа
Onepatop read
Onepatop print

# Оператор program

#### Форма записи:

- Оператор поименования программы
- Всегда записывается первым
- Не является обязательным

Onepaтop program
Onepaтop end
Onepatopы описания типа
Onepatop read
Onepatop print

# Оператор program

#### Форма записи:

- Оператор поименования программы
- Всегда записывается первым
- Не является обязательным

Oператор program
Onepaтop end
Oператоры описания типа
Оператор read
Оператор print

# Оператор end

#### Форма записи:

- Оператор завершения программы
- Указывает конец последовательности операторов и комментариев
- Всегда записывается последним
- Оператор завершения программного блока

Оператор program
Oneparop end
Операторы описания типа
Оператор read
Оператор print

# Оператор end

#### Форма записи:

- Оператор завершения программы
- Указывает конец последовательности операторов и комментариев
- Всегда записывается последним
- Оператор завершения программного блока

# Оператор end

#### Форма записи:

- Оператор завершения программы
- Указывает конец последовательности операторов и комментариев
- Всегда записывается последним
- Оператор завершения программного блока

# Оператор end

#### Форма записи:

- Оператор завершения программы
- Указывает конец последовательности операторов и комментариев
- Всегда записывается последним
- Оператор завершения программного блока

## Оператор end

#### Форма записи:

- Оператор завершения программы
- Указывает конец последовательности операторов и комментариев
- Всегда записывается последним
- Оператор завершения программного блока

#### INTEGER list

REAL list

DOUBLE PRECISION list

COMPLEX list

LOGICAL list

CHARACTER list

```
read *, a, b, i
print *, a, b, i
```

Oператор program Oператор end Операторы описания типа Оператор read Оператор print

# Операторы описания типа

#### INTEGER list

REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list
CHARACTER list

### prog2.f

program prog2
implicit none
real a, b
integer i

read \*, a, b, i
print \*, a, b, i

end

# INTEGER list REAL list DOUBLE PRECISION list COMPLEX list LOGICAL list CHARACTER list

```
prog2.f
      program prog2
      implicit none
      real a, b
      integer i
      read *, a, b, i
      print *, a, b, i
      end
```

INTEGER list
REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list
CHARACTER list

```
prog2.f
      program prog2
      implicit none
      real a, b
      integer i
      double precision Alpha, Beta
      read *, a, b, i
      print *, a, b, i
      end
```

INTEGER list
REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list

```
prog2.f
      program prog2
      implicit none
      real a, b
      integer i
      complex Gamma, Delta, Eps
      read *, a, b, i
      print *, a, b, i
      end
```

INTEGER list
REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list

```
prog2.f
      program prog2
      implicit none
      real a, b
      integer i
      logical if1, if2
      read *, a, b, i
      print *, a, b, i
      end
```

prog2.f

# Операторы описания типа

INTEGER list
REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list
CHARACTER list

```
program prog2
implicit none
real a, b
integer i
character symbol

read *, a, b, i
print *, a, b, i
end
```

INTEGER list
REAL list
DOUBLE PRECISION list
COMPLEX list
LOGICAL list
CHARACTER list

```
read *, a, b, i
print *, a, b, i
```

#### Форма записи:

- Оператор чтения
- Считывает данные и размещает их в памяти (присваивает значения переменным)
- \* в данном контексте означает чтение под управлением списка из стандартного потока ввода
- Более сложная запись оператора READ позднее

#### Форма записи:

- Оператор чтения
- Считывает данные и размещает их в памяти (присваивает значения переменным)
- \* в данном контексте означает чтение под управлением списка из стандартного потока ввода
- Более сложная запись оператора READ позднее

#### Форма записи:

- Оператор чтения
- Считывает данные и размещает их в памяти (присваивает значения переменным)
- \* в данном контексте означает чтение под управлением списка из стандартного потока ввода
- Более сложная запись оператора READ позднее

#### Форма записи:

- Оператор чтения
- Считывает данные и размещает их в памяти (присваивает значения переменным)
- \* в данном контексте означает чтение под управлением списка из стандартного потока ввода
- Более сложная запись оператора READ позднее

#### Форма записи:

- Оператор чтения
- Считывает данные и размещает их в памяти (присваивает значения переменным)
- \* в данном контексте означает чтение под управлением списка из стандартного потока ввода
- Более сложная запись оператора READ позднее

Оператор program Оператор end Операторы описания типа Оператор read Оператор print

# Оператор read

- Каждый оператор READ начинает чтение с новой записи данных (новой строки), независимо от наличия несчитанных данных в текущей записи
- В случае недостатка значений в записи данных для определения значений всех переменных в списке оператора READ, производится чтение из следующей записи (строки).

- Каждый оператор READ начинает чтение с новой записи данных (новой строки), независимо от наличия несчитанных данных в текущей записи
- В случае недостатка значений в записи данных для определения значений всех переменных в списке оператора READ, производится чтение из следующей записи (строки).

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
-
```

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78_
```

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78
2240_
```

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78
2240
3.1415 -0.000500000024 2240
$ _
```

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78
2240
3.1415 -0.000500000024 2240
$ ./prog3_
```

# Программа prog3.f

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78
2240
   3.1415 -0.000500000024 2240
$ ./prog3
3.1415
-.0005
78_
```

# Программа prog3.f

```
prog3.f
        program prog3
        implicit none
        real a, b
        integer i
        read *, a, b
        read *, i
        print *, a, b, i
        end
```

```
$ g77 -o prog3 prog3.f
$ ./prog3
3.1415 -0.0005 78
2240
3.1415 -0.000500000024 2240
$ ./prog3
3.1415
-.0005
78
3.1415 -0.000500000024 78
$
```

#### Форма записи:

- Оператор записи в стандартный поток вывода
- \* в данном контексте означает запись под управлением списка
- список list допускает наличие переменных, констант и арифметических выражений
- Более сложная запись оператора PRINT позднее

Оператор program Оператор end Операторы описания типа Оператор read Оператор print

## Оператор print

#### Форма записи:

- Оператор записи в стандартный поток вывода
- \* в данном контексте означает запись под управлением списка
- список list допускает наличие переменных, констант и арифметических выражений
- Более сложная запись оператора PRINT позднее

#### Форма записи:

- Оператор записи в стандартный поток вывода
- \* в данном контексте означает запись под управлением списка
- список list допускает наличие переменных, констант и арифметических выражений
- Более сложная запись оператора PRINT позднее

#### Форма записи:

- Оператор записи в стандартный поток вывода
- \* в данном контексте означает запись под управлением списка
- список list допускает наличие переменных, констант и арифметических выражений
- Более сложная запись оператора PRINT позднее

#### Форма записи:

- Оператор записи в стандартный поток вывода
- \* в данном контексте означает запись под управлением списка
- список list допускает наличие переменных, констант и арифметических выражений
- Более сложная запись оператора PRINT позднее

#### Форма записи:

#### variable = const

variable=variable variable=expression

```
read *, i
print *, a, b, j
```

```
Форма записи:

variable=const

variable=variable
```

```
read *, i
print *, a, b, j
```

Форма записи:

variable=const

variable=variable

variable=expression

```
read *, i
print *, a, b, j
```

Форма записи:

variable=const

variable=variable

variable=expression

```
prog4.f
        program prog4
        implicit none
        real a, b
        integer i, j
        read *, i
        a = 3.1415
        b = -.0005
        j = i
        print *, a, b, j
        end
```

```
PARAMETER (name=const)
```

```
PARAMETER (name=expression)
PARAMETER (name1=value1, name2=value2, ...)
```

- Определение символьного имени константы
- Недопустипо переопределение значения, закреплённого за символьным именем, в процессе выполнения программы

```
PARAMETER (name=const)

PARAMETER (name=expression)

PARAMETER (name1=value1 name2=value2
```

- Определение символьного имени константы
- Недопустипо переопределение значения, закреплённого за символьным именем, в процессе выполнения программы

```
PARAMETER (name=const)

PARAMETER (name=expression)

PARAMETER (name1=value1, name2=value2, ...)
```

- Определение символьного имени константы
- Недопустипо переопределение значения, закреплённого за символьным именем, в процессе выполнения программы

```
PARAMETER (name=const)

PARAMETER (name=expression)

PARAMETER (name1=value1, name2=value2, ...)
```

- Определение символьного имени константы
- Недопустипо переопределение значения, закреплённого за символьным именем, в процессе выполнения программы

```
PARAMETER (name=const)

PARAMETER (name=expression)

PARAMETER (name1=value1, name2=value2, ...)
```

- Определение символьного имени константы
- Недопустипо переопределение значения, закреплённого за символьным именем, в процессе выполнения программы

```
prog5-1.f
     program prog5
     implicit none
     integer i
     parameter (i=2)

i=3
     print *, i
     end
```

```
prog5-1.f
     program prog5
     implicit none
     integer i
```

parameter (i=2) i=3

print \*, i
end

#### Форма записи:

#### DATA varlist/constlist/

DATA varlist1/constlist1/, varlist2/constlist2/, ...

- Инициализация переменных на этапе компиляции
- Инициализированные переменные могут изменять значения в процессе выполнения программы

```
DATA varlist/constlist/
DATA varlist1/constlist1/, varlist2/constlist2/, ...
```

- Инициализация переменных на этапе компиляции
- Инициализированные переменные могут изменять значения в процессе выполнения программы

```
DATA varlist/constlist/
DATA varlist1/constlist1/, varlist2/constlist2/, ...
```

- Инициализация переменных на этапе компиляции
- Инициализированные переменные могут изменять значения в процессе выполнения программы

```
DATA varlist/constlist/
DATA varlist1/constlist1/, varlist2/constlist2/, ...
```

- Инициализация переменных на этапе компиляции
- Инициализированные переменные могут изменять значения в процессе выполнения программы

#### Без оператора data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
a = pi
b = -.0005
i1 = 1
i2 = 1
i3 = 1
print *, a, b
print *, i1, i2, i3
end
```

```
print *, a, b
print *, i1, i2, i3
```

#### Без оператора data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
a = pi

b = -.0005

i1 = 1
```

i2 = 1i3 = 1

print \*, a, b
print \*, i1, i2, i3
end

#### C оператором data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
data a, b / pi, -.0005 /, & i1, i2, i3 / 3*1 /
```

```
print *, a, b
print *, i1, i2, i3
end
```

#### Без оператора data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
a = pi
b = -.0005
i1 = 1
i2 = 1
```

i3 = 1

print \*, a, b
print \*, i1, i2, i3
end

#### C оператором data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
data a, b / pi, -.0005 /, & i1, i2, i3 / 3*1 /
```

```
print *, a, b
print *, i1, i2, i3
end
```

#### Без оператора data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
a = pi

b = -.0005

i1 = 1
```

i2 = 1i3 = 1

print \*, a, b
print \*, i1, i2, i3
end

#### C оператором data

```
program prog6
implicit none
real a, b, pi
integer i1, i2, i3
parameter (pi=3.1415)
```

```
data a, b / pi, -.0005 /, & i1, i2, i3 / 3*1 /
```

```
print *, a, b
print *, i1, i2, i3
end
```

#### Взаимное расположение операторов

	PROGRAM,	FUNCTION,	SUBROUTINE, BLOCK DATA
			IMPLICIT
			INTEGER, REAL,
		PARAMETER	DOUBLE PRECISION,
Операторы	FORMAT,		COMPLEX, LOGICAL,
комментария	ENTRY		CHARACTER, DIMENSION
			Операторы-функции
		DATA	Исполняемые
			операторы
END			