

Язык программирования FORTRAN-77

базовые понятия

Михайленко К. И.

Башгосуниверситет
Институт механики Уфимского научного центра РАН

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

История

Fortran	1954, November
Fortran I	1956, October
Fortran II	1957
Fortran III	1958 (end)
Fortran IV	1962
Fortran 66 ANS	1966 (Fortran IV)
Fortran 77 ANSI	1978, April (Fortran V)
Fortran 90 ISO	1991
Fortran 95 ISO	1997, December 15
Fortran 2000	(draft) 2002, September 30
Fortran 2003	2004, November 30

Текст программы

Файл hello.f

```
program Hello
print*, 'Hello, World!'
end
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```


Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Компилятор gcc

GCC the GNU Compilers Collection

g77 GNU Fortran-77 compiler

```
$ ls  
hello.f
```

Компиляция (трансляция)

```
$ g77 -c hello.f
```

```
$ ls  
hello.f hello.o
```

Линковка (сборка)

```
$ g77 -o hello hello.o
```

```
$ ls  
hello hello.f hello.o
```

```
$ ls  
hello.f
```

Компиляция и линковка

```
$ g77 -o hello hello.f
```

```
$ ls  
hello hello.f
```

Выполнение программы

```
$ ./hello  
Hello, World!
```

Разрешённые символы

26 букв **ABCDEFGHIJKLMNOPQRSTUVWXYZ**

или (обычно нет различия)

abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки $+ - * / = () . , ' : \$$

(обычно «'» и «"» идентичны)

СИМВОЛ `_`

(обычно трактуется как буква)

СИМВОЛ `□`

(игнорируется компиляторами;

может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв **ABCDEFGHIJKLMNOPQRSTUVWXYZ**
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки **+ - * / = () . , ' : \$**
(обычно «'» и «"» идентичны)

СИМВОЛ **_**
(обычно трактуется как буква)

СИМВОЛ **□**
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки +-*/=() . , ' : \$
(обычно «'» и «"» идентичны)

СИМВОЛ _
(обычно трактуется как буква)

СИМВОЛ □
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки $+ - * / = () . , ' : \$$
(обычно «'» и «"» идентичны)

СИМВОЛ `_`
(обычно трактуется как буква)

СИМВОЛ `□`
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки $+ - * / = () . , ' : \$$
(обычно «'» и «"» идентичны)

СИМВОЛ `_`
(обычно трактуется как буква)

СИМВОЛ `□`
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки +-*/=() . , ' : \$
(обычно «'» и «"» идентичны)

СИМВОЛ -
(обычно трактуется как буква)

СИМВОЛ □
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки +-*/=() . , ' : \$
(обычно «'» и «"» идентичны)

СИМВОЛ _
(обычно трактуется как буква)

СИМВОЛ □
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Разрешённые символы

26 букв ABCDEFGHJKLMNOPQRSTUVWXYZ
или (обычно нет различия)
abcdefghijklmnopqrstuvwxyz

10 цифр 0123456789

знаки +-*/=() . , ' : \$
(обычно «'» и «"» идентичны)

СИМВОЛ _
(обычно трактуется как буква)

СИМВОЛ □
(игнорируется компиляторами;
может быть использован в любых количествах)

Остальные символы могут присутствовать в строковых данных

Структура строки

1	2	3	4	5	6	7 ... 72	73
						Комментарий	
						Текст команды	
						6 Признак строки продолжения (\forall символ) (до 19 строк продолжения; часто – больше)	
1...5					Метка (число)		
1	Признак комментария (символ «C» или «*»)						

Структура строки

1	2	3	4	5	6	7 ... 72	73
							Комментарий
						Текст команды	
					6	Признак строки продолжения (\forall символ) (до 19 строк продолжения; часто – больше)	
1...5						Метка (число)	
1	Признак комментария (символ «C» или «*»)						

Структура строки

1	2	3	4	5	6	7 ... 72	73
							Комментарий
						Текст команды	
					6	Признак строки продолжения (✓ символ) (до 19 строк продолжения; часто – больше)	
1...5						Метка (число)	
1	Признак комментария (символ «C» или «*»)						

Структура строки

1	2	3	4	5	6	7 ... 72	73
						Текст команды	Комментарий
1...5					6	Признак строки продолжения (\forall символ) (до 19 строк продолжения; часто – больше)	
					1	Метка (число)	
1	Признак комментария (символ «C» или «*»)						

Структура строки

1	2	3	4	5	6	7 ... 72	73
						Комментарий	
						Текст команды	
						6 Признак строки продолжения (\forall символ) (до 19 строк продолжения; часто – больше)	
1...5					Метка (число)		
1	Признак комментария (символ «C» или «*»)						

Типы данных

- **integer**
- real
- double precision
- complex
- logical
- character

Типы данных

- integer
- **real**
- double precision
- complex
- logical
- character

Типы данных

- integer
- real
- **double precision**
- complex
- logical
- character

Типы данных

- integer
- real
- double precision
- **complex**
- logical
- character

Типы данных

- integer
- real
- double precision
- complex
- **logical**
- character

Типы данных

- integer
- real
- double precision
- complex
- logical
- **character**

Типы данных

- integer
- real
- double precision
- complex
- logical
- character

Написание целых констант

`integer`
1 55 -167 +7629 0

4 byte (32 bit)

-2147483648 ... 2147483647

Размер данных целого типа

1 55 -167 +7629 0

integer

4 byte (32 bit)

-2147483648 ... 2147483647

Значения данных целого типа

`integer`
1 55 -167 +7629 0

4 byte (32 bit)

`-2147483648 ··· 2147483647`

Значения данных целого типа

`integer`
1 55 -167 +7629 0

4 byte (32 bit)

-2147483648 ··· 2147483647

Написание вещественных констант

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7–8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Написание вещественных констант

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7–8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Написание вещественных констант

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7–8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Размер данных вещественного типа

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7-8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Значения данных вещественного типа

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7-8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Значения данных вещественного типа

real

- С фиксированной запятой:

1.5 55.0 +55. -0.00016 -.16

- С плавающей запятой:

1.5E0 5.5E1 +5.5E+1 -1.6E-4 -.16e+0

4 byte (32 bit)

24 bit мантисса (a), 8 bit показатель степени (b); $a \times 10^b$

7–8 чисел в мантиссе, $-128 \dots 127$ показатель степени

$\pm a.aaaaaaE\pm bbb$

Написание вещественных констант двойной точности

double precision

1.5D0 5.5D1 +5.5D+1 -1.6D-4 -.16d+0

8 byte (64 bit)

48 bit мантисса (a), 16 bit показатель степени (b); $a \times 10^b$

15–16 чисел в мантиссе, $-32768 \dots 32767$ показатель степени
 $\pm a.aaaaaaaaaaaaaaaaE\pm bbbbb$

Размер данных вещественного типа двойной точности

double precision

1.5D0 5.5D1 +5.5D+1 -1.6D-4 -.16d+0

8 byte (64 bit)

48 bit мантисса (a), 16 bit показатель степени (b); $a \times 10^b$

15–16 чисел в мантиссе, $-32768 \dots 32767$ показатель степени
 $\pm a.aaaaaaaaaaaaaaaaE\pm bbbbb$

Значения данных вещественного типа двойной точности

double precision

1.5D0 5.5D1 +5.5D+1 -1.6D-4 -.16d+0

8 byte (64 bit)

48 bit мантисса (a), 16 bit показатель степени (b); $a \times 10^b$

15–16 чисел в мантиссе, $-32768 \dots 32767$ показатель степени

$\pm a.aaaaaaaaaaaaaaaaE\pm bbbbb$

Значения данных вещественного типа двойной точности

double precision

1.5D0 5.5D1 +5.5D+1 -1.6D-4 -.16d+0

8 byte (64 bit)

48 bit мантисса (a), 16 bit показатель степени (b); $a \times 10^b$

15–16 чисел в мантиссе, $-32768 \dots 32767$ показатель степени
 $\pm a.aaaaaaaaaaaaaaaaE\pm bbbbb$

Написание комплексных констант

(Re, Im) **complex** (5.5, 1.2) (+5., -1.) (-1.6E-4, +.1)

2 × 4 byte (64 bit)

пара вещественных значений

Написание комплексных констант

(Re, Im) **complex**
(5.5, 1.2) (+5., -1.) (-1.6E-4, +.1)

2 × 4 byte (64 bit)

пара вещественных значений

Написание комплексных констант

(Re, Im) complex
(5.5, 1.2) (+5., -1.) (-1.6E-4, +.1)

2 × 4 byte (64 bit)

пара вещественных значений

Написание логических констант

logical

`.TRUE.`

`.FALSE.`

`.true.`

`.fAlSe.`

1 byte (реально используется 1 bit)
значения: 0 (`.FALSE.`) и 1 (`.TRUE.`)

Размер логических данных

logical

`.TRUE.`

`.FALSE.`

`.true.`

`.fAlSe.`

1 byte (реально используется 1 bit)
значения: 0 (`.FALSE.`) и 1 (`.TRUE.`)

Размер логических данных

logical

`.TRUE.`

`.FALSE.`

`.true.`

`.fAlSe.`

1 byte (реально используется 1 bit)
значения: 0 (`.FALSE.`) и 1 (`.TRUE.`)

Написание символьных констант

`'a'` `'2+2=4'` `'Группа №38'` `'.fAlSe.'`

n byte;

n — КОЛИЧЕСТВО СИМВОЛОВ

Размер символьных данных

character

'a'

'2+2=4'

'Группа №38'

'.fAlSe.'

n byte;

n — КОЛИЧЕСТВО СИМВОЛОВ

Размер символьных данных

character

'a'

'2+2=4'

'Группа №38'

'.fAlSe.'

n byte;

n — КОЛИЧЕСТВО СИМВОЛОВ

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34

Имена переменных, программных модулей. . .

Стандарт: 6 символов, первый символ — буква

Обычно: > 31 символа

Примеры

Правильно: A BILL Summa i251 X4x9z LImit

Неправильно: A *A 5A K-9 e.g. John's

Формально неправильно: average

Правильное имя: ab_cd_34