

«به نام خدا»



گروه دینامیک سیالات محاسباتی

دانشگاه صنعتی اصفهان



فرترن

بخش اول (دستورهای کلیدی)

امیررضا هاشمی

کلیه حقوق این کتاب الکترونیک متعلق به گروه دینامیک سیالات محاسباتی دانشگاه صنعتی اصفهان می‌باشد

هر گونه چاپ، تکثیر و استفاده از آن با ذکر منبع بلا مانع است.

<http://CFD.iut.ac.ir>

مقدمه

زبان برنامه‌نویسی فرترن در سال ۱۹۴۸ در دانشگاه منچستر نوشته شده است. کلمه *fortran* مخفف عبارت *formula_transformation* (ترجمه یا تبدیل فرمول) و این امکان را در اختیار کاربران قرار می‌داد تا فرمول‌ها را در برنامه‌ها به صورت کد کامپیوتری درآورند. فرترن از ابتدا به عنوان زبان علمی و مهندسی شناخته شده بود نام فرترن معرف همین موضوع می‌باشد.

فرترن‌ها معمولاً براساس سال تولید آن نام‌گذاری می‌گردند مثلاً فرترن ۶۶ مربوط به سال ۱۹۶۶ و فرترن ۷۷ مربوط به سال ۱۹۷۷ و فرترن ۹۰ مربوط به دهه ۸۰ است که در ابتدا با عنوان فرترن 8x و نسخه تکمیلی آن در حوالی سال ۱۹۹۰ در اختیار برنامه‌نویسان قرار گرفت. بعد از آن نیز فرترن ۹۵ در سال ۱۹۹۵ به بازار آمد که در میان این نسخ فرترن ۷۷ به دلیل پیشرفتهای نسبت به نسخه قبل (فرترن ۶۶) مورد توجه بسیاری قرار گرفت اما با وجود پیشرفتهای صورت گرفته نقص‌هایی از جمله عدم امکان تخصیص "ذخیره دینامیک" و همچنین وجود برخی دستوره‌های نسخ‌شده در برنامه‌نویسی باعث شد تا سازندگان با ارائه نسخه ۹۰ علاوه بر رفع این مشکلات ارائه شد سیستم‌های ذخیره اطلاعات در آن بهبود یافت.

نام‌های مجاز:

نام متغیرها و پارامترها در فرترن از کاراکترهای حرفی، عددی و همچنین *underline* "_" می‌توان استفاده کرد. ملاحظات:

- طول نام‌ها در فرترن تا ۳۱ کاراکتر و اولین کاراکتر آن باید کاراکتر حرفی باشد.
- در فرترن حروف بزرگ و کوچک تفاوتی ندارد. برای مثال: *SeLEct = sEleCT*
- برخی از کلمات به عنوان دستوره‌های کلیدی است و نباید به عنوان نام متغیرها و پارامترها استفاده شود.

برای مثال *abs, sqrt, if, do, ...*

مثال‌هایی از نام‌های غیرمجاز

iut.university

iut university

123_Y

V / U

مثال‌هایی از نام‌های مجاز

iut_university

input

Y_1234

SOLver

متغیرها:

تعریف در فرترن ۹۰	تعریف در فرترن ۷۷	نوع متغیر
<i>INTEGER :: A, I, BETA</i>	<i>INTEGER A, I, BETA</i>	اعداد صحیح ساده
<i>REAL :: FLOAT, SUM</i>	<i>REAL FLOAT, SUM</i>	اعداد اعشاری با دامنه کم
<i>REAL (8) :: FLOAT, SUM</i>	<i>REAL (8) FLOAT, SUM</i>	اعداد اعشاری با دامنه بلند (<i>Double precision</i>)

<code>REAL(KIND = 5) :: FLOAT</code>	-	تعریف اعداد اعشاری با دامنه موردنظر
<code>CHARACTER :: CHAR * 5</code>	<code>CHARACTER CHAR * 5</code>	تعریف کاراکتر (که طول کاراکتر با ضرب اندازه طول در نام کاراکتر تعیین می شود).
<code>LOGICAL :: RIGHT</code>	<code>LOGICAL RIGHT</code>	تعریف عبارات منطقی

دستور `IMPLICIT NONE`:

این دستور در ابتدای هر برنامه یا زیر شاخه نوشته می شود و تعیین نوع ضمنی را توسط کامپایلر غیرفعال می کند. بدین ترتیب باید نام همه متغیرهای یک برنامه را تعریف نمود. این کار از به وجود آمدن برخی اشتباهات در حین کدنویسی مانند غلط املائی و... جلوگیری می کند.

شروع و پایان یک برنامه:

برنامه ها در فرترن به صورت `Source File` فایل های برآمده از منبع می باشد یعنی برای نوشتن برنامه تنها می توان به فرمت تعیین شده کامپایلر برنامه را نوشت، در فرترن ۷۷ به قبل این شکل برنامه نویسی حتی در نوشتار ثابت (`Fixed format`) بود به طوریکه در نوشتن یک برنامه می بایست ۶ فضای اول هر خط را برای نوشتن شماره دستور توسط کاراکتر هفتم جدا می شد اما در فرترن ۹۰ این قید هم به دلیل محدودیت تایپ حذف شد و شکل برنامه در آن به صورت `Free format` ارائه شده است.

`PROGRAM` نام برنامه (برای مثال `SOLVER`)

دستورات فرترن

`END`

`END PROGRAM`

`END PROGRAM` نام برنامه (برای مثال `SOLVER`)

ملاحظات:

۱. طول هر خط نوشتن برنامه در فرترن ۱۲۰ تا ۸۰ کاراکتر است.
۲. ترجیح بر این است که نوع نوشتن به گونه ای باشد که خطها بسیار بلند نباشد، نهایت طول خط دستورات به گونه ای باشد که در یک صفحه A4 بتوان کد را پرینت نمود.

کاراکترهای خاص:

- !: برای قرار دادن توضیحات در میان برنامه از این کاراکتر استفاده می‌شود و خط متناظر با خود را به رنگ سبز در می‌آید و این خط در حین اجرای برنامه کامپایل نمی‌گردد.
- &: برای دستوراتی که طول دستورات از حد مجاز بیشتر باشد در ابتدای خط دوم برای ایجاد اتصال بین خطوط دستور مورد استفاده قرار می‌گیرد.
- ;: با این کاراکتر می‌توان در یک خط چند دستور را تایپ نمود و با ; از هم جدا نمود معمولاً دستورها در انتهای هر خط پایان می‌یابد و استفاده از این کاراکتر توصیه نمی‌شود.

عملیات جبری:

اولویت بیشتر به توان رساندن (نمایی) **
* ضرب
/ تقسیم
+ جمع
- اولویت کمتر تفریق

مثال: $4**5/2**3*8 = \frac{4^5}{2^3 \times 8} = 16$

اگر نوع عبارت و متغیر یا متغیر و متغیر در جایگزینی‌ها یکسان نباشد برای رفع مشکل از فرمت زیر استفاده می‌شود:

$A = TYPE(B)$

$A = INTEGER$

$B = REAL$

$A = INTEGER(B + A)$

برای مثال:

برخی توابع درونی در فرترن:

$ABS(X)$	قدر مطلق
$INT(X)$	جزء صحیح
$NINT(X)$	نزدیکترین به عدد صحیح
$CEILING(X)$	نزدیکترین به عدد صحیح کوچکتر از X نباشد
$FLOOR(X)$	نزدیکترین به عدد صحیح بزرگتر از X نباشد
$REAL(X)$	تبدیل به اعشاری
$MOD(X,Y), MODULE(X,Y)$	باقیمانده تقسیم
$MAX(X,Y), MIN(X,Y)$	ماکزیمم و مینیمم

عملیات ریاضی:

$\text{sqrt}(x)$	\sqrt{x}	ریشه دوم
$\text{exp}(x)$	$\text{exp}(x)$	نما رساندن
$\log x$	$\ln x$	لگاریتم طبیعی
$\log_{10} x$	$\log_{10} x$	لگاریتم معمولی
$\sin(x)$	$\sin x$	سینوس زاویه
$\cos(x)$	$\cos x$	کسینوس زاویه
$\tan(x)$	$\tan x$	تانژانت زاویه
$\sinh(x)$	$\sinh x$	سینوس هذلولوی
$\cosh(x)$	$\cosh x$	کسینوس هذلولوی
$\tanh(x)$	$\tanh x$	تانژانت هذلولوی
$\sin^{-1}(x)$	$\sin^{-1} x$	سینوس معکوس
$\cos^{-1}(x)$	$\cos^{-1} x$	کسینوس معکوس
$\tan^{-1}(x)$	$\tan^{-1} x$	تانژانت معکوس
$\text{cplx}(x)$	$x + iy$	عدد مختلط بر حسب مولفه‌های اعشاری و موهومی
$\text{conj}(x)$	$x - iy$	مزدوج مختلط

حلقه‌های تکرار:

دستور DO با شمارنده در فرترن ۹۰:

نام حلقه + توضیحات: $DO \quad I = 0,10,2$

گام حلقه →

پایان حلقه ←

شروع حلقه ←

دستورات فرترن

نام حلقه DO END

$LOOP1: \quad DO \quad I = 0,10,2$

برای مثال:

دستورات فرترن

END DO LOOP1

DO VARIABLE = START, END, PITCH

شکل کلی حلقه

$\left[\frac{(END - START)}{PITCH} \right]$: تعداد دفعاتی که حلقه اجرا می شود:

حلقه های بی نهایت:

DO

دستورات

EXIT

دستورات

END DO

DO I = 0, 100

دستورات

IF (Q <= 0) EXIT

دستورات

END DO

DO

دستورات

CIRCLE

دستورات

END DO

DO I = 0, 100

دستورات

IF (Q <= 0) CIRCLE

دستورات

END DO

دستور حلقه در فرترن ۷۷:

DO LABEL WHILE (logic statement)

دستورات فرترن ۷۷

LABEL CONTINUE

مثال:

DO 100 I = 0, 10

دستورات فرترن ۷۷

100 CONTINUE

عبارات منطقی:

همانطور که گفته شد عبارتهای منطقی به شکل زیر تعریف می گردند که تعریف این عبارتها در حلقه های شرطی برای اعمال قیود موردنیاز است به این ترتیب:

<i>LOGICAL</i> :: <i>RIGHT</i> , <i>WRONG</i>	تعریف عبارت منطقی
<i>TRUE</i> .	درست
<i>FALSE</i> .	غلط
<i>LOGICAL</i> , <i>PARAMETER</i> :: <i>T = TRUE</i> .	برای مثال:

عملگرهای منطقی:

<i>AND</i> .	و
<i>OR</i> .	یا
<i>EQV</i> .	مساوی با
<i>NEQV</i> .	نامساوی با

عملگرهای قیاسی:

فرترن ۷۷	فرترن ۹۰	
<i>lt.</i> <i>less than</i>	<	کوچکتر از
<i>le.</i> <i>less equal</i>	<=	کوچکتر یا مساوی با
<i>gt.</i> <i>greater than</i>	>	بزرگتر از
<i>ge.</i> <i>greater equal</i>	>=	بزرگتر یا مساوی با
<i>eq.</i> <i>equal</i>	==	مساوی با
<i>ne.</i> <i>non equal</i>	~=	مخالف با

دستورات شرطی: (منطقی)
دستور IF:

فرمت کلی:

IF (logic statement) دستورات فرترن

IF (A > B) A = A + B

برای مثال

label : if (logic statement) then

دستور IF بلوکی:

دستورات فرترن

end if label

label : if (10.lt.A.ge.20) then

برای مثال

save = A

end if label

label : if (logic statement) then

دستور IF با دو بلوک:

دستورات فرترن

else

دستورات فرترن

end if label

label : if (A.gt.B) then

برای مثال

C = A / B

else

C = B / A

end if label

label : if (first logic statement) then

دستور IF با بلوک‌های تودرتو:

دستورات فرترن

else if (second logic statement) then

دستورات فرترن

else if (third logic statement) then

دستورات فرترن

else if (nth logic statement) then

دستورات فرترن

else

دستورات فرترن

end if label

label : if (1800 <= year < 1900) then

CENT = 19th

else if (1900 <= year < 2000) then

CENT = 20th

else if (2000 <= year) then

CENT = 21th

end if label

برای مثال

دستور شرطی CASE:

label :select case(logic statement)

case(first logic state)

بلوک اول

دستورات فرترن

case(second logic state)

بلوک دوم

دستورات فرترن

`case(nth logic state)`

بلوک n ام

دستورات فرترن

`case default`

بلوک آخر

دستورات فرترن

`end label case`

`century :select case(year)`

برای مثال:

`case(1800 : 1900)`

`cent = 19th`

`case(1900 : 2000)`

`cent = 20th`

`case(2000 :)`

`cent = 21th`

`end case century`

