كيفية استعمال البرامج الجزئية في لغة فرتران 90
How to use sub-programs in Fortran 90

نسخة اولية

كثيرا ما تصادفنا في عمليات البرمجة إعادة كتابة مجموعة تعليمات او جزء من البرنامج عدة مرات في اماكن مختلفة في نفس البرنامج الاساسي مثل حساب قيمة دالة f(x) عند قيم مختلفة لـ x . لتجنب مثل هذه المسائل وغيرها يوفر نهج استعمال البرامج الجزئية طريقة جيدة وواضحة وفعالة للبرمجة. وبصفة عامة تكمن أهمية البرامج الجزئية في تسهيل تطوير برامج مشاريع ضخمة بوضوح أكثر وبسهولة ويسر. نريد في هذا الدرس ان نشرح كيفية استعمال البرامج الجزئية عن طريق امثلة متنوعة في لغة فورتران 90. لهذا ننصح ان يكون القارئ على دراية اولية بالبرمجة بلغة فرتران 77.
البرامج الجزئية في فرتران على نوعين: subroutines و functions .

• البرامج الجزئي من نوع subroutine:

وهو عبارة عن سلسلة من الأوامر يمكن استدعائها لتنفيذها عدة مرات حسب الحاجة انطلاقا من مواضع معينة في البرنامج الأساسي أو من مواضع معينة من برامج جزئية أخرى بنوعيها. يتميز البرنامج الجزئي subroutine بـ:

- **موضع** وله ثلاث حالات :
  - في **البرنامج الاساسي** بعد كلمة **contains** وقبل كلمة **end program** وفي هذه الحالة يسمى برنامج جزئي **داخلي** interne.
  - في **البرنامج الاساسي** بعد كلمة **end program** او في **ملف مستقل** عن البرنامج الاساسي و يسمى في هذه الحالة برنامج جزئي **خارجي** externe.
- **اسم** يستدعى به عن طريق التعليمة **call** .

```
call nom_subroutine(arguments)
```

- **قد** يملك مجموعة من المداخيل inputs و النواتج output تقع كلها في عمدته arguments وهي مجموعة من المتغيرات يصرح بها داخل البرنامج الجزئي. في هذا التصريح ينصح بتمييز متغيرات العمدة بالتعليمة **intent :**

intent(in) إذا كانت وضيفة احد متغيرات العمدة يستعمل فقط لإدخال قيمة متغير.
intent(out) إذا كانت وضيفة احد متغيرات العمدة يستعمل فقط لإخراج قيمة متغير.
intent(inout) إذا كانت وضيفة احد متغيرات العمدة يستعمل لإدخال وإخراج قيمة متغير بمعنى أن قيمته تتغير بعد الدخول للبرنامج الجزئي.
هذا التمييز مهم لتتبع أخطاء البرنامج ففي حالة استعمال متغير عمدة في خير محله يشير المترجم compiler إلى خطا.

- كل المتغيرات **المستعملة فقط** في البرنامج الجزئي هي متغيرات **محلية** له, بمعنى غير معرفة في البرنامج الأساسي.

- كل متغيرات البرنامج **الأساسي** هي متغيرات عامة globale بالنسبة للبرامج الجزئية **الداخلية** باستثناء متغيرات العمدة لذا ليس من الضروري إعادة تصريحها داخل هذه البرامج الجزئية **الداخلية**.
- لا يصرح بصنف **اسمه**.

• البرامج الجزئي من نوع function (دالة):

وهو برنامج شبيه جدا بالبرنامج الجزئي subroutine ويختلف عنه في:

- من الضروري أن يكون له متغيرات ادخال في عمدته.
- يرجع ناتج الحساب اوالعمل المنوط به في متغير يحمل نفس اسمه لذلك يجب التصريح بصنف اسم **الدالة** في المكان الذي تعرف فيه وفي أي قسم وضيفي يستخدمها.
- لايستدعى بـ call بل يستدعى كما تكتب الدوال في الرياضيات

```
y=nom_fonction( arguments )
```

- صنف نتيجة الحساب يمكن أن يصرح به قبل اسم الدالة (باستثناء حالة أن تكون نتيجة الدالة مصفوفة):

```
real function maxi( t )
. . .
end function maxi
```

أو بالتصريح المباشر لمتغير اسم الدالة

```
function maxi( t )
    . . .
    real   :: maxi
    . . .
end function maxi
```

يمكن كذلك استخراج نتيجة الحساب في متغير آخر يختلف عن اسم **الدالة** باستعمال التعليمة **result** لكن دون ان يصرح بصنف الدالة قبل التعليمة function **ولا** أن يرفق لمتغير النتيجة التخصيص **intent**:

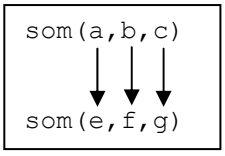
```
function maxi( t ) result(y)
    . . .
    real   :: y
    . . .
end function maxi
```

• امثلة:
Subroutines

مثال1: نعرض هنا برنامجا أساسيا examplesub1.f90 يستدعي برنامج جزئي **خارجي** يقع بعد السطر الحاوي للتعليمة **end program.** يستقبل البرنامج الجزئي متغيرين a وb ويحسب متغيرا ثالثا c .

```
PROGRAM examplesub1
 ! programme principal
 ! la subroutine n'est pas déclarée
 implicit none
 integer::i
 real::a,b,c

 write(*,*)"entrer a et b"
 read(*,*)a,b
 write(*,*)"a = ",a
 write(*,*)"b = ",b

 call som(a,b,c)   ! on utilise call pour appeler une subroutine

 write(*,*)"Les valeurs de a,b,c sont"
 write(*,*) a,b,c

 END PROGRAM examplesub1

! * * * * * * * * * * * * * *

 SUBROUTINE som(e,f,g)
 implicit none
 real,intent(in)::e,f   ! on précise que e et f sont des argument d'entrer
 real,intent(out)::g   ! on précise que g est un argument de sortie
 g=e+f

 END SUBROUTINE som
```

لاحظ أن أسماء المتغيرات في عمدة البرنامج الجزئي ليس من الضروري أن توافق أسماء المتغيرات في البرنامج الأساسي فما يهم هو ترتيبهم فقط بمعنى :



. كذلك جرب أن تغير قيمة المتغير e بإضافة مثلا .e+2 بعد السطر g=e+f فان المترجم سيكشف عن خطىء ناتج عن أن المتغير e لايجب تغيير قيمته لأنه مميز بصفة الادخال فقط **intent(in)** .

يختفي هذا الخطأ بتغيير صفة الإدخال فقط إلى صفة الإدخال والإخراج **intent(inout)** ذلك لان المتغير يصبح قابل للتغيير.

مثال2: نعرض هنا نفس البرنامج الأساسي السابق يستدعي برنامج جزئي **داخلي** يقع في **البرنامج الاساسي** بعد كلمة **contains** وقبل كلمة **end program.** يستقبل البرنامج الجزئي متغيرين a وb ويحسب متغيرا ثالثا c .

```
PROGRAM examplesub1
 ! programme principal
 ! la subroutine n'est pas déclarée
 implicit none
 integer::i
 real::a,b,c

 write(*,*)"entrer a et b"
 read(*,*)a,b
 write(*,*)"a = ",a
 write(*,*)"b = ",b

 call som(a,b,c)   ! on utilise call pour appeler une subroutine

 write(*,*)"Les valeurs de a,b,c sont"
 write(*,*) a,b,c

!************************
 CONTAINS

 SUBROUTINE som(e,f,g)
 implicit none
 real,intent(in)::e,f   ! on précise que e et f sont des argument d'entrer
 real,intent(out)::g   ! on précise que g est un argument de sortie
 g=e+f

 END SUBROUTINE som

 END PROGRAM examplesub1
```

تمرين اكتب برنامجا جزئيا من نوع subroutine اسمه polar يستقبل الإحداثيات الكرتيزية (x,y) ويحسب الإحداثيات القطبية (r,θ) الموافقة لها.

$$
\begin{array}{ll}
r = \sqrt{x^2+y^2} & \\
\theta = \arctan|y/x| & x > 0 \\
\theta = -\arctan|y/x| & x < 0 \\
\theta = \pi/2 & x = 0 \text{ et } y > 0 \\
\theta = -\pi/2 & x = 0 \text{ et } y < 0
\end{array}
$$

مثال3: نعرض هنا نفس البرنامج الأساسي السابق يستدعي برنامج جزئي **خارجي** يقع في **ملف مستقل محفوظ باسم لا على التعيين مثلا** somme.f90 **في نفس المجلد الحاوي على البرنامج**

الأساسي examplesub1.f90**.** يستقبل البرنامج الجزئي متغيرين a وb ويحسب متغيرا ثالثا c . في هذه الحالة يجب أولا ترجمة (compiling) البرنامج الجزئي ثم ترجمة البرنامج الأساسي ثم التنفيذ.

```
PROGRAM examplesub1
 ! programme principal
 ! la subroutine n'est pas déclarée
 implicit none
 integer::i
 real::a,b,c

 write(*,*)"entrer a et b"
 read(*,*)a,b
 write(*,*)"a = ",a
 write(*,*)"b = ",b

 call som(a,b,c)   ! on utilise call pour appeler une subroutine

 write(*,*)"Les valeurs de a,b,c sont"
 write(*,*) a,b,c

 END PROGRAM examplesub1
```

يحوي الملف المستقل somme.f90 البرنامج الجزئي **som**

```
SUBROUTINE som(a,b,c)
 implicit none
 real,intent(in)::a,b   ! on précise que a et b sont des argument d'entrer
 real,intent(out)::c   ! on précise que c est un argument de sortie
 g=e+f

 END SUBROUTINE som
```

مثال4: نعرض هنا برنامجا أساسيا examplesub2.f90 يستدعي برنامج جزئي **داخلي sp** يستقبل عمودا يحوي 100 قيمة عشوائية ويحسب متوسط هذه القيم و اكبر قيمة في العمود ثم يرسل النتيجة في المتغيرين moyenne و maximum.

```
PROGRAM examplesub2
  implicit none
  real ::moyenne maximum
  real, dimension(100) :: tab
  call random_number( tab )        ! subroutine intrinsic
  call sp( tab , moyenne , maximum )
  print* , moyenne, maximum
CONTAINS
 subroutine sp( t , moy , max )

  implicit none
  real , dimension( 100 ) , intent(in) :: t
  real , intent( out ) : : moy , max
  integer :: i
    max = t( 1 ) ; moy = t( 1 )
   do i=2 ,100
     if ( t( i ) > max ) then
      max = t( i )
     end if
      moy = moy + t( i )
   end do
  moy = moy /100
end subroutine sp
END PROGRAM examplesub2
```

Functions

مثال1 : نعرض هنا برنامجا أساسيا examplef1.f90 يستدعي برنامج جزئي **خارجي** (دالة) يقع بعد السطر الحاوي للتعليمة **end program.** تستقبل الدالة 3 متغيرات a وb وx وترجع القيمة a*x+b في اسمها كمتغير.

```
PROGRAM examplef1
 ! programme principal
 implicit none
 integer::i
 real::f,a,b,x,y

 a = 1.d0
 b = 2.d0
 x=0.d0
 Do i=1,100
    x = x + 0.1d0
    write(*,*)x,f(a,b,x)  ! appel de la fonction
 Enddo

 ! autre exemple
 x=0.d0
 Do i=1,100
    x = x + 0.1d0
    y = f(a,b,x)          ! appel de la fonction
    write(*,*)x,y
 Enddo
END PROGRAM examplef1

! * * * * * * * * * * * *

 FUNCTION f(a,b,x)
 implicit none
 real,intent(in)::a,b,x
 real::f     ! dans une fonction son nom est déclaré

 f = a*x + b ! quelque part dans une fonction on doit assigner une valeur à f

 END FUNCTION f
```

مثال2 : نعرض هنا برنامجا أساسيا examplef2.f90 يستدعي برنامج جزئي **داخلي maxi** (دالة) تستقبل الدالة متغيرا واحدا t في شكل عمود يحوي 100 قيمة عشوائية وترجع الدالة اكبر قيمة له في المتغير **maxi**.

```
PROGRAM examplef2
  implicit none
  real :: maximum
  real, dimension(100) :: tab
  call random_number( tab )
  maximum = maxi( tab )
  print* , maximum
CONTAINS
 Function maxi( t )
  implicit none
  real , dimension( 100 ) , intent(in) :: t
  integer :: i
  real :: maxi !- la fonction est déclarée
  maxi = t( 1 )
  do i=2,100
    if ( t( i ) > maxi ) then
     maxi = t( i )
    end if
  end do
end function maxi
END PROGRAM examplef2
```