FORTRAN 95

Podprogramy
czyli
jak pisać żeby było czytelnie

Waldemar Kulig
Zakład Chemii Teoretycznej
UJ

Podprogramy

Podprogram to wydzielony blok instrukcji programu identyfikowany przez nazwę oraz (jeżeli to konieczne) listę argumentów formalnych.

Instrukcje podprogramu powinny tworzyć logiczną całość realizującą jakiś cel.

Podprogramy, których działanie polega na wykonaniu ciągu instrukcji nazywamy **procedurami** (z ang. subroutine), natomiast programy, które wyznaczają (zwracają) pewną waertość nazywamy **funkcjami**.

Podprogramy - deklaracja

1) **PROCEDURY**

```
subroutine nazwa_procedury(lista_argumentow)

......

end subroutine
```

2) **FUNKCJE**

```
typ_zwracanej_zmiennej function nazwa_funkcji(lista_argumentow)

.....

end function
```

Podprogramy - deklaracja

Przykłady:

```
subroutine swap(a, b)
real :: a, b, temp ! Deklaracja zmiennych
temp = a
a = b
b = temp
end subroutine
```

```
real function pierwiastek(x)
real::x  !deklaracja zmiennych
pierwiastek = sqrt(x)
end function
```

Deklaracja podprogramów następuje na końcu!!! programu

Podprogramy - wywołanie

1) procedury

call nazwa_procedury(lista_argumentow)

2) funkcje

zmienna = nazwa_funkcji(lista_argumentow)

Przykład:
call czytaj(a, n)
suma = dodaj(d, n)
call zamien(ala, ola)

Podprogramy

```
program suma_elementow
implicit none

real, dimension(10) :: a
real :: suma, sumuj

call czytaj(a, 10)
suma = sumuj(a, 10)
write(*,*) 'Suma elementow wynosi', suma
call wypisz(a, 10)
end

subroutine czytaj(b, n)
real, dimension(n) :: b
integer :: i, n

do i = 1, n
   write(*,*)'Wpisz', i, 'element'
   read(*,*) b(i)
end do

end subroutine
```

```
subroutine wypisz(d, n)
real, dimension(n) :: d
integer :: i, n

write(*,*) (d(i), i=1,n)

end subroutine

real function sumuj(g, n)
integer::i, n
real, dimension(n) :: g

sumuj = 0.0
do i = 1, n
   sumuj = sumuj + g(i)
end do

end function
```