

Interface-lohko

Funktion välittäminen parametrina

Esimerkiksi yhtälöä ratkaisevalle proseduurille tekisi mieli ilmoittaa parametrina yhtälöä kuvaava funktio, jolloin samassa ohjelmassa voidaan samalla proseduurilla ratkaista useita erilaisia yhtälöitä.

Aloitetaan tarkastelemalla triviaalia esimerkkiä. Olkoon pääohjelma tiedostossa `main.f90`:

```
program main
  real :: x(5) = 1.0, summa
  real, external :: f
  summa=sub(f, x)
  write(*,*) summa
contains
real function sub(f, x)
  real, external :: f
  real x(:)
  sub=f(x)
end function
end program
```

Funktiolle `sub` välitetään funktio `f` parametrina. Parametrina välitettävä aliohjelma on käännettävä erikseen. Tämä ilmoitetaan määreellä `external`.

Oletetaan, että aliohjelma on tiedostossa `sub.f90`:

```
real function f(x)
  real x(:)
  f=sum(x)
end function
```

Käännös ja linkitys tapahtuvat tavalliseen tapaan:

```
f95 -o ohjelma main.f90 sub.f90
```

Erillisenä käännettävän proseduurin kutsun oikeellisuutta ei voi suoraan tarkistaa käännoisaikana. Siksi parametrina välitettävän proseduurin kutsun muoto on hyvä kertoa interface-lohkon avulla:

```
real function integrate(f, a, b)
implicit none

interface
  real function f(x)
  real, intent(in) :: x
  end function
end interface

real, intent(in) :: a, b

sum=f(a)+f(b)
...
integrate=sum
end function
```

Tässä interface sisältää funktion ja sen argumenttien määrittelyn, mutta ei mitään suoritettavaa koodia eikä paikallisten muuttujien määrittelyjä. Interface-lohko kertoo kääntäjälle, miten funktiota kutsutaan, joten kutsun oikea muoto voidaan tarkastaa käännosvaiheessa.