

Alexandru Calotoiu

Fortran challenges



Development of structures

```
PROGRAM type_test
  implicit none

  TYPE simple_type
    REAL :: w(5,5,5), z(5)
    INTEGER :: a
    REAL :: name
  END TYPE simple_type

  REAL :: d(5,5)
  CALL type_test_function(d)
end

SUBROUTINE type_test_function(d)
  REAL d(5,5)
  TYPE(simple_type) :: s
  s%w(1,1,1) = 5.5
  d(2,1) = 5.5 + s%w(1,1,1)
END SUBROUTINE type_test_function
```


Development of structures

```

PROGRAM type_test
  implicit none

  type a_t
    real :: w(5,5,5)
    type(b_t), pointer :: b
  end type a_t

  type b_t
    type(a_t)          :: a
    integer             :: x
  end type b_t

  type c_t
    type(d_t), pointer  :: ab
    integer             :: xz
  end type c_t

  type d_t
    type(c_t)          :: ac
    integer             :: xy
  end type d_t

  REAL :: d(5,5)

  CALL circular_type_test_function(d)
end

```

```

SUBROUTINE circular_type_test_function(d)
  REAL d(5,5)
  TYPE(a_t) :: s
  TYPE(b_t) :: b(3)

  s%w(1,1,1)=5.5
  !s%b=>b(1)
  !s%b%a=>s
  b(1)%x=1
  d(2,1)=5.5+s%w(1,1,1)

END SUBROUTINE circular_type_test_function

```

Development of structures

```
PROGRAM type2_test
  implicit none

  TYPE simple_type
    REAL:: w(5,5,5), z(5)
    INTEGER:: a
  END TYPE simple_type

  TYPE comlex_type
    TYPE(simple_type):: s
    REAL:: b
  END TYPE comlex_type

  TYPE meta_type
    TYPE(comlex_type):: cc
    REAL:: omega
  END TYPE meta_type

  REAL :: d(5,5)
  CALL type2_test_function(d)
end
```

```
SUBROUTINE type2_test_function(d)
  REAL d(5,5)
  TYPE(simple_type) :: s(3)
  TYPE(comlex_type) :: c
  TYPE(meta_type) :: m

  !c%b=1.0
  c%s%w(1,1,1)=5.5
  m%cc%s%a=17
  s(1)%w(1,1,1)=5.5+c%b
  d(2,1)=c%s%w(1,1,1)+s(1)%w(1,1,1)

END SUBROUTINE type2_test_function
```