Encapsulation : Planet Object

Alexandre Mary, Météo-France

Code training days, 2019 Sept. 12th - Toulouse

Encapsulation

before

```
MODULE YOMZZZ
REAL :: RA
...
```

in subroutine calc.F90 :

```
USE YOMZZZ, ONLY: RA
F = RA * RG...
```

after

```
MODULE YOMZZZ
TYPE EARTH
  REAL :: RA
ENDTYPE EARTH
...
```

in subroutine calc.F90 :

```
SUBROUTINE CALC(YDEARTH, ...)
USE YOMZZZ, ONLY: EARTH
TYPE(EARTH), INTENT(IN) :: YDEARTH
F = YDEARTH%RA * RG...
```

What if we needed to run 2 models on 2 different planets in the same run ?

- YOMCST : definitions of
  - Geoid
  - Solar constant (solar irradiance atop of the atmosphere)
  - Astronomical constants

⇒ module/object/setup YOMPLANET/YRPLANET/SUPLANET

⇒ as a component of GEOMETRY object ?

? is GEOMETRY conceptually available everywhere these constants are used ?

? implications on setup (order of calls)

↪ replace access through the code

↪ compile & checkpack !

Keep in mind

- when ***rephasing*** an old branch on a new cycle, pay attention to ***new encapsulations***
- when introducing new variable(s) :
    - ***In which object should that go ?***
    - ***Should I declare it directly in derived type or create a new sub-object ?***
- passing by arguments : ***compromise*** between passing whole objects and too many small sub-objects/variables
- each object should have its ***dedicated setup routine*** (so-called *"constructor"*)
- (and *"destructor"*, i.e. deallocation of arrays if necessary)