```
module HELIOS_data

!  use constants_and_parameters
  use linear_regression_results
  use polynomial_fit
  use HELIOS_files
  implicit none

  type, public                                :: HELIOS

    ! amplitudes describing the Helios surface
    type ( surface_fit )                      :: poly_fit
    type ( HELIOS_output )                    :: HELIOS_tables

    contains

      ! functions

      ! subroutines
      procedure, public                       :: put                    =>

  end type                                       HELIOS

  private                                     :: put_sub

  contains

!   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

    subroutine put_sub ( self, amplitudes, errors, sf_quality, pts_temp
                         array_scaled, array_scaled_characteristics, a

      use constants_and_parameters
      use quality_parameters
      use polynomial_fit
      implicit none

      class ( HELIOS ), target                                :: self
      real ( wp ), pointer                                    :: surface_amplit
      real ( wp ), pointer                                    :: surface_error
      real ( wp ), pointer                                    :: ptr_pts_temp
      real ( wp ), pointer                                    :: ptr_pts_densit

      real ( wp ), optional                                   :: amplitudes
      real ( wp ), optional                                   :: errors
      real ( wp ), optional                                   :: array_scaled
      real ( wp ), optional                                   :: array_normali
      real ( wp ), optional                                   :: pts_temp
      real ( wp ), optional                                   :: pts_density

      type ( surface_fit_quality ),       optional    :: sf_quality
      type ( lr_results ),                optional    :: pts_temp_lr, p
      type ( array_characteristics ),     optional    :: array_scaled_

      ! HELIOS output
      ! maps
      if ( present ( pts_temp ) ) then
        ptr_pts_temp => self % HELIOS_tables % pts_temp
        ptr_pts_temp =  pts_temp
        ptr_pts_temp => null ( )
      end if

      if ( present ( pts_density ) ) then
        ptr_pts_density => self % HELIOS_tables % pts_density
        ptr_pts_density =  pts_density
        ptr_pts_density => null ( )
      end if

      if ( present ( pts_temp_lr ) ) then
        self % HELIOS_tables % pts_temp_lr =  pts_temp_lr
      end if

      if ( present ( pts_density_lr ) ) then
        self % HELIOS_tables % pts_density_lr = pts_density_lr
      end if

      !  array tables
      if ( present ( array_scaled ) ) then
        self % HELIOS_tables % array_scaled = array_scaled
      end if

      if ( present ( array_scaled_characteristics ) ) then
        self % HELIOS_tables % array_scaled_characteristics = array_sc
      end if

      if ( present ( array_normalized ) ) then
        self % HELIOS_tables % array_normalized = array_normalized
      end if

      if ( present ( array_normalized_characteristics ) ) then
        self % HELIOS_tables % array_normalized_characteristics = array
      end if

      ! surface fit
      if ( present ( amplitudes ) ) then
        surface_amplitudes => self % poly_fit % amplitudes
        surface_amplitudes =  amplitudes
        surface_amplitudes => null ( )
      end if

      if ( present ( errors ) ) then
        surface_errors => self % poly_fit % errors
        surface_errors =  errors
        surface_errors => null ( )
      end if

      if ( present ( sf_quality ) ) then
        self % poly_fit % sf_quality = sf_quality
      end if

    end subroutine put_sub
```

```

end module HELIOS_data
```