```
module cpu_timer_class                                  ! time quantum is a
!
  use constants_and_parameters
  implicit none
!
  ! derived data type
  type, public            :: cpu_timer          ! name to instantiate
!
    private
    real ( wp )           :: saved_time         ! saved time in second
    real ( wp )           :: cum_time           ! cumulative time in s
!
    contains  ! bound procedures
!
      ! sequence:  grab  pause, resume, ... pause, resume  stop
      ! stop is equivalent to pause
      procedure, public   :: cpu_timer_grab     => cpu_timer_grab_sub
      procedure, public   :: cpu_timer_pause    => cpu_timer_pause_sub
      procedure, public   :: cpu_timer_resume   => cpu_timer_resume_su
!
      procedure, public   :: cpu_timer_stop     => cpu_timer_stop_fcn
      procedure, public   :: cpu_timer_cum_read => cpu_timer_cum_read_
!
  end type cpu_timer
  ! end derived data type
!
  private                 :: cpu_timer_grab_sub, cpu_timer_pause_sub, c
  private                 :: cpu_timer_stop_fcn, cpu_timer_cum_read_fc
!
  ! methods
  contains                                          ! subroutines and func
!
!   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!
    subroutine cpu_timer_grab_sub ( self )      ! start a timer
!
      implicit none
```

```
38
39        class ( cpu_timer ) :: self                ! timer object
40
41        real ( wp )         :: cpu_time_t0         ! saved time in second
42
43        ! system call for current cpu_time
44        call cpu_time ( cpu_time_t0 )
45
46        self % saved_time = cpu_time_t0            ! unique start time fo
47        self % cum_time   = zero
48
49      end subroutine cpu_timer_grab_sub
50
51  !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
52
53      subroutine cpu_timer_pause_sub ( self )     ! update the cumulativ
54
55        implicit none
56
57        class ( cpu_timer ) :: self                ! timer object
58
59        real ( wp )         :: cpu_time_now
60
61        ! system call for current cpu_time
62        call cpu_time ( cpu_time_now )
63
64        ! guard against the erroneous sequence of  pause, pause, ..., pa
65        self % cum_time   = self % cum_time + ( cpu_time_now - self % sa
66        self % saved_time = cpu_time_now           ! unique start time fo
67
68      end subroutine cpu_timer_pause_sub
69
70  !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
71
72      subroutine cpu_timer_resume_sub ( self )    ! restart the time
73
74        implicit none
```

```
75
76        class ( cpu_timer ) :: self                ! timer object
77
78        real ( wp )         :: cpu_time_now        ! saved time in second
79
80        ! system call for current cpu_time
81        call cpu_time ( cpu_time_now )
82
83        self % saved_time = cpu_time_now           ! unique start time fo
84
85      end subroutine cpu_timer_resume_sub
86
87  !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
88
89      function cpu_timer_stop_fcn ( self ) result ( cpu_time_cum )  ! re
90
91        implicit none
92
93        class ( cpu_timer ) :: self                ! timer object
94
95        real ( wp )         :: cpu_time_now        ! saved time in second
96        real ( wp )         :: cpu_time_cum        ! accumulated time
97
98        ! system call for current cpu_time
99        call cpu_time ( cpu_time_now )
100
101       ! guard against the erroneous sequence of  pause, pause, ..., pa
102       cpu_time_cum      = self % cum_time + ( cpu_time_now - self % sa
103       self % cum_time   = cpu_time_cum
104       self % saved_time = cpu_time_now           ! unique start time fo
105
106     end function cpu_timer_stop_fcn
107
108 !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
109
110     function cpu_timer_cum_read_fcn ( self ) result ( cpu_cum_time )
111
```

```
112       implicit none
113
114       class ( cpu_timer ) :: self                ! timer object
115
116       real ( wp )         :: cpu_cum_time        ! cumulative time buff
117
118       cpu_cum_time = self % cum_time             ! read cumulative time
119
120     end function cpu_timer_cum_read_fcn
121
122 end module cpu_timer_class
123
124 !     ###################################################################
125 !     #
126 !     ###################################################################
127
128 module clock_timer_class                         ! time quantum is proc
129
130   use constants_and_parameters
131   implicit none
132
133   ! derived data type
134   type, public              :: clock_timer       ! name to instantiate
135
136     private
137     integer ( lint )        :: saved_time        ! saved time in second
138
139     contains  ! bound procedures
140
141       procedure, public     :: clock_start_timer  => clock_start_timer
142       procedure, public     :: clock_elapsed_time => clock_elapsed_tim
143
144   end type clock_timer
145   ! end derived data type
146
147   private :: clock_start_timer_sub, clock_elapsed_time_fcn
148
```

```
149   ! methods
150   contains                                          ! subroutines and func
151
152 !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
153
154     subroutine clock_start_timer_sub ( self )
155
156       implicit none
157
158       class ( clock_timer ) :: self             ! timer object
159
160       integer ( lint )      :: clock_count_start! saved time in ticks
161
162       ! system call for current clock_time
163       call system_clock ( clock_count_start )
164
165       self % saved_time = clock_count_start     ! unique start time fo
166
167     end subroutine clock_start_timer_sub
168
169 !   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
170
171     real function clock_elapsed_time_fcn ( self )
172
173       implicit none
174
175       class ( clock_timer ) :: self             ! clock timer object
176       integer ( lint )      :: clock_count_stop, clock_count_rate, clo
177
178       ! system call for current clock_time
179       call system_clock ( clock_count_stop, clock_count_rate, clock_co
180
181       ! units are compiler dependent
182       clock_count_delta = clock_count_stop - self % saved_time
183       ! convert to seconds
184       clock_elapsed_time_fcn = dble ( clock_count_delta ) / clock_coun
185
```

```
186     end function clock_elapsed_time_fcn
187
188 end module clock_timer_class
```