

Klasa Simulation, wersja 3

```
unit Simulation: PriorityQueues class;
  unit Simprocess: elemFIFO coroutine;
    unit isIdle: function: Boolean; ...
    unit isTerminated: function: Boolean; ...
  end Simprocess;

  unit EventNotice: elemPQ class(p: SimProcess, t: time);
    unit less: virtual function(e: EventNotice): Boolean;
    begin
      result:= t < e.t
    end less;
  end EventNotice;
  (* własność S2 jest spełniona, ponadto obiekty tej klasy mogą być wstawiane do kolejki PQ *)

  unit PlanSymulacji: QueueHead class;
    unit schedule : procedure(p: SimProcess, t: time): ...
    unit hold: procedure(dt: time);
    begin
      ...
      call chooseProcess;
    end hold;
    unit run: procedure(p: SimProcess); ...
    unit passivate: procedure; ...
    unit cancel: procedure; ...
    unit chooseProcess: procedure;
      var e: EventNotice;
    begin
      (* wartością SQS.min jest najmniejszy element, typ elemPQ*)
      e:=SQS.min qua EventNotice; (* dzięki qua odzyskujemy poprawny typ *)
      (* zmienne currTime i currProcess są prywatnymi zmiennymi w klasie Simulation *)
      currProcess:= e.p;
      currTime := e.t;
      attach(e.p);
    end chooseProcess;

    unit currentProcess: function: SimProcess;
    begin
      result := currProcess;
    end currentProcess;
    unit currentTime: function: time; ...
  end PlanSymulacji;

  unit time: class ... end time;

var SQS: PlanSymulacji;
(* własność S1 jest zagwarantowana, SQS jest kolejką priorytetową!*)
var currProcess: SimProcess;
var currTime: time;
```

end Simulation;