

Pierwsza wersja klasy Simulation

to po prostu przełożenie sygnatury specyfikacji na szkielet klasy Simulation.

Sygnatura specyfikacji	Klasa Simulation
<p>Typy: simprocess</p> <p>plan_of_simulation PQ</p> <p>time</p> <p>eventnotice</p> <p>Operacje: current - the process which is currently executing its commands, current : PQ → SP</p> <p>time - the value of currently simulated time, time : PQ → T</p> <p>schedule - enables planning of events, and hence the processes, schedule : (SP × T) × PQ → PQ</p> <p>hold - permits to suspend a current process for a while, hold : T × PQ → PQ</p> <p>run - causes an immediate execution of the indicated process, run : SP × PQ → PQ</p> <p>passivate - suspends the current process and removes it from the axe of time, passivate : PQ → PQ</p> <p>cancel - removes a named process from the set of events, cancel : SP × PQ → PQ</p> <p>idle? idle : SP → Boolean</p> <p>terminated? terminated : SP → Boolean.</p>	<p>unit Simulation: PriorityQueues class;</p> <p>unit Simprocess: elemFIFO coroutine;</p> <p>unit isIdle: function: Boolean; ...</p> <p>unit isTerminated: function: Boolean; ...</p> <p>end Simprocess;</p> <p>unit EventNotice: elemPQ class; ... end EventNotice;</p> <p>unit PlanSymulacji: QueueHead class;</p> <p>(* operacje w których PQ jest argumentem i wynikiem zadeklarowaliśmy jako metody w PlanieSymulacji czyli PQ *)</p> <p>unit schedule : procedure(p: SimProcess, t: time): ...</p> <p>unit hold: procedure(dt: time); ...</p> <p>unit run: procedure(p: SimProcess); ...</p> <p>unit passivate: procedure; ...</p> <p>unit cancel: procedure; ...</p> <p>unit chooseProcess: procedure; ...</p> <p>unit currentProcess: function: SimProcess; ...</p> <p>unit currentTime: function: time; ...</p> <p>end PlanSymulacji;</p> <p>unit time: class ... end time;</p> <p>var SQS: PlanSymulacji;</p> <p>(* własność S1 jest zagwarantowana: SQS jest kolejką priorytetową!*)</p> <p>end Simulation;</p>