КРЪГОВ АЛГОРИТЪМ ЗА РАЗПРЕДЕЛЕНО ВЗАИМНО ИЗКЛЮЧВАНЕ

Ring-Based Distributed Mutual Exclusion (ME)  
(Formalized Specification)

**Pi::ME**

**{SYNOPSIS}**

**Basic variant of the algorithm for distributed mutual exclusion over logical communication ring.**

**{ASSUMPTIONS}**

**No failures of any kind (neither processes nor channels) are allowed, by condition.**

**Only under this condition the two mandatory requirements for this class of distributed algorithms - ME1 (safety) and ME2 (feasibility, liveness) are guaranteed. The desirable, but not mandatory, condition ME3 (fairness) is not met for this algorithm by definition.**

**{SYSTEM CONSTANTS}**

String MRK\_ME // type of message „ME token*“*

**{MESSAGES}**

<mrk\_me>

**{SET OF STATES}**

<State> := {ME\_INIT, ME\_RELEASED, ME\_WANTED, ME\_HELD}

**{INTERNAL STATE SPACE}**

State meState // current process state

**{E::INTERNAL STATE SPACE}**

Id idCoordinator // “coordinator” process identifier

**{СЪБИТИЯ}**

**OnInit:**

meStatus := ME\_INIT

idCoordinator := ⊥

**OnRingUp:**

idCoordinator := Election()

**OnAfterElection:**

**If** i = idCoordinator

StartMarker()

**EndIf**

meStatus := ME\_RELEASED

**OnEnter:**

**If** meStatus = ME\_RELEASED

meStatus := ME\_WANTED

**EndIf**

**OnReceiptOf <mrk\_me>:**

**If** meStatus = ME\_WANTED

meStatus := ME\_HELD

**Else If** meStatus = ME\_RELEASED

Send <mrk\_me>

**EndIf**

**OnRelease:**

**If** meStatus = ME\_HELD

meStatus := ME\_RELEASED

Send <mrk\_me>

**EndIf**