

Wait



Этот блок похож на блок Queue с одним исключением: он поддерживает изъятие в ручном режиме (методы free(), или freeAll()). Этот блок не имеет определённого порядка (кроме случаев, когда включено вытеснение).

Параметры

Тип агента

Тип агентов, проходящих через блок.

Далее: T

Вместимость

[Параметр виден, если не выбрана опция *Максимальная вместимость*]

Вместимость очереди.

Тип: int capacity

Значение по умолчанию: 100

Максимальная вместимость

Если опция выбрана (true), то вместимость очереди будет максимально возможной (ограничена константой Integer.MAX_VALUE).

Синтаксис: boolean maximumCapacity

Значение по умолчанию: false

Место агентов

Фигура разметки ([узел](#) или [путь](#)), где располагаются агенты, пока они находятся в этом блоке.

Синтаксис: AnimationStaticLocationProvider entityLocation

Специфические

Разрешить уход по таймауту

Если опция выбрана (true), то агенты могут покидать очередь по таймауту. После проведения в очереди максимально допустимого времени агенты будут покидать объект через специальный порт *outTimeout*.

Синтаксис: boolean enableTimeout

Значение по умолчанию: false

Таймаут [динамический]

[Параметр виден, если выбрана опция *Разрешить уход по таймауту*]
Выражение, вычисляющее значение таймаута (максимально допустимое время, которое агент может провести в очереди) для агента.

Тип значения: double

Локальная переменная: T agent - агент.

Разрешить вытеснение

Если опция выбрана (true), агенты располагаются в очереди соответственно их приоритетам и могут быть вытеснены агентами с более высоким приоритетом.

Синтаксис: boolean enablePreemption

Значение по умолчанию: false

Вытеснять

[Параметр виден, если выбрана опция *Разрешить вытеснение*] Задает, как приходящие агенты вытесняют агентов, уже находящихся в очереди.

Вытеснять можно агента: *Самого последнего агента, Агента с самым низким приоритетом, По сравнению агентов, Самого старшего агента*

Имя: queuing

Допустимые значения: Wait.QUEUING_FIFO - *Самого последнего*

агента

Wait.QUEUEING_PRIORITY - *Агента с самым низким приоритетом*

Wait.QUEUEING_COMPARISON - *По сравнению агентов*

Wait.QUEUEING_LIFO - *Самого старого агента*

Приоритет агента [динамический]

[Параметр виден, если **Вытеснить**: *Агента с самым низким приоритетом*] Приоритет поступающего агента (чем больше значение, тем выше приоритет).

Тип значения: double

Локальная переменная: T agent - агент.

Значение по умолчанию: 0

"agent1 может вытеснить agent2" [динамический]

[Параметр виден, если **Вытеснить**: *По сравнению агентов*] Код сравнения, который должен возвращать true, если агент *agent1* имеет более высокий приоритет чем агент *agent2* (иначе, false, включая случаи равных приоритетов).

Тип значения: boolean

Локальная переменная: Tagent1, Tagent2 - агенты.

Значение по умолчанию: false

Выталкивать агентов

Если опция выбрана (true), то агенты, созданные этим блоком, будут немедленно вытолкнуты дальше независимо от состояния следующего блока.

Если опция не выбрана, агенты будут не выталкиваться, а выходить постепенно: только когда следующий блок готов принять еще одного агента, он запрашивает его у этого блока, и тогда агент проходит дальше.

Синтаксис: boolean pushProtocol

Значение по умолчанию: false

Вернуть агента в исходную точку

Если опция выбрана, агенты возвращаются в свое начальное местоположение ([узел](#) или [путь](#), где они находились до того, как попали в этот блок), после того, как покинут фигуру разметки, заданную в параметре **Место агентов**.

Синтаксис: `boolean restoreEntityLocationOnExit`

Включить сбор статистики

По умолчанию, статистика собирается для всех блоков Библиотеки Моделирования Процессов. Тем не менее, вы можете настроить этот параметр и выключить сбор статистики, чтобы улучшить динамику модели. Для этого добавьте блок [PML Settings](#) и отключите опцию ***Включить статистику по умолчанию***. Таким образом, вы отключите сбор статистики для всех блоков диаграммы процесса в этой модели. Но вы можете включить статистику для некоторых конкретных блоков, включив эту опцию ***Включить сбор статистики*** в свойствах самого блока.

Синтаксис: `boolean forceStatisticsCollection`

Значение по умолчанию: `false`

Действия

Во всех действиях текущий агент доступен как локальная переменная agent.

При входе [код]

Код, выполняемый, когда агент поступает в объект (и помещается в очередь).

При выходе [код]

Код, выполняемый, когда агент покидает объект через порт *out* (обычным способом).

При уходе по таймауту [код]

[Параметр виден, если выбрана опция ***Разрешить уход по таймауту***]
Код, выполняемый, когда агент покидает объект по таймауту (прождав в очереди максимально допустимое время) через специальный порт *outTimeout*.

При вытеснении [код]

[Параметр виден, если выбрана опция ***Разрешить вытеснение***] Код, выполняемый, когда агент покидает объект через порт *outPreempted* в результате вытеснения.

При извлечении [код]

Код, выполняемый, когда агент умышленно извлекается из блока посредством вызова функции агента `remove()`. Этот код вызывается автоматически после вызова функции `remove()`.

Локальная переменная: `T agent` - агент.

Переменные

StatisticsContinuous statsSize

Статистика длины очереди. Статистика собирается если выбрана опция объекта ***Включить статистику по умолчанию*** в блоке [PML Settings](#).

Функции

`int size()` - возвращает количество агентов, находящихся в данный момент в очереди.

`T get(int index)` - возвращает агента, находящегося в позиции с номером `index` (ближайшая к выходу из очереди позиция имеет номер 0).

`T remove(Agent agent)` - извлекает агента `agent` из очереди и возвращает его. Если такого агента в очереди обнаружено не будет, метод вернет `null`.

`boolean free(Agent agent)` - сообщает очереди освободить указанного агента и направить его к выходному порту *out*.

`boolean freeAll()` - сообщает очереди освободить всех агентов и направить их к выходному порту *out*.

`resetStats()` - удаляет статистику, собранную объектом к текущему моменту времени.

`Iterator iterator()` - возвращает итератор агентов (в порядке: первый - у выхода, последний - самое большое время, оставшееся до его задержки).

Порты

in

Входной порт.

out

Выходной порт.

outTimeout

Выходной порт для агентов, покидающих объект по таймауту.

outPreempted

Выходной порт для агентов, покидающих объект в результате вытеснения.