**Система имитационного моделирования AnyLogic**

**AnyLogic**– уникальный инструмент имит. моделир-я, поддерживающ. на единой платформе абсолютно все существующ. подходы дискретно-событийного и непрерывного моделир-я. AnyLogicимеет развитый базовый язык дискретного и смешананный дискретно-непрерывного моделир-я. Библиотека AnyLogicEnterprise Library предоставляет высокоуровнев. интерфейс для быстрого создания дискретно-событийных моделей с пом. блок-схем. Графич. представление систем с пом. блок-схем широко исп-ся во многих важных сферах деятельности: производстве, логистике, системах обслуживания, бизнес-процессах, моделир-ии компьют. и телекоммуникацион. сетей.

Класс Entity яв-ся базовым классом для всех сообщений, кот. посылаются между активными объектами библиотеки Enterprise Library.

Под **заявкой** в библиотеке Enterprise Library может пониматься:

• заявка в ее обычном понимании (продукт, потребитель, пакет данных, документ),

• ресурс (оператор, машина, критическая секция),

• транспортер (поезд, автобус, корабль, автопогрузчик).

Заявки в их традиц. понимании генерир-ся объектами Source, проходят через смоделированную систему, где они обрабатыв-ся, обслужив-ся, транспортир-ся, конкурируют за право обладания ресурсами и эту систему покидают.

Ресурсы, созданные объектами Resource, могут быть заняты заявками для выполнения каких-то задач, после чего они освобожд-ся и возвращ-ся в объект Resource.

Транспортеры, также как и обычные заявки, создаются объектами Source, затем передаются в объекты Node и исп-ся для транспортировки других заявок между узлами и вдоль сегментов сети.

Объект класса Entity может исп-ся в любой из этих ролей, и при необходимости даже менять свою роль во время работы модели.

**Source** - Источник заявок. Обычно исп-ся в качестве начальной точки потока заявок, или как генератор ресурсов, транспортеров.

**Sink -** Уничтожает поступивш. заявки. Обычно исп-ся в качестве конечной точки потока заявок.

**Enter -** Пересылает заявки, переданные этому объекту либо “явно” через входной порт inputExternal, либо с помощью функции объекта take(), дальше по блок-схеме.

**SelectOutput -** Принимает заявку, и затем, в зависимости от заданного условия, передает ее на один из двух выходных портов.

**Queue -**  моделирует очередь, он хранит поступающ. заявки в опред. порядке: FIFO (заявки помещ-ся в очередь в порядке поступления), LIFO (заявки помещ-ся в порядке, обратном поступлению), RANDOM (заявки помещ-ся в произвольные места очереди) или PRIORITY (заявки помещ-ся в очередь в соответствии со значением своих полей priority).