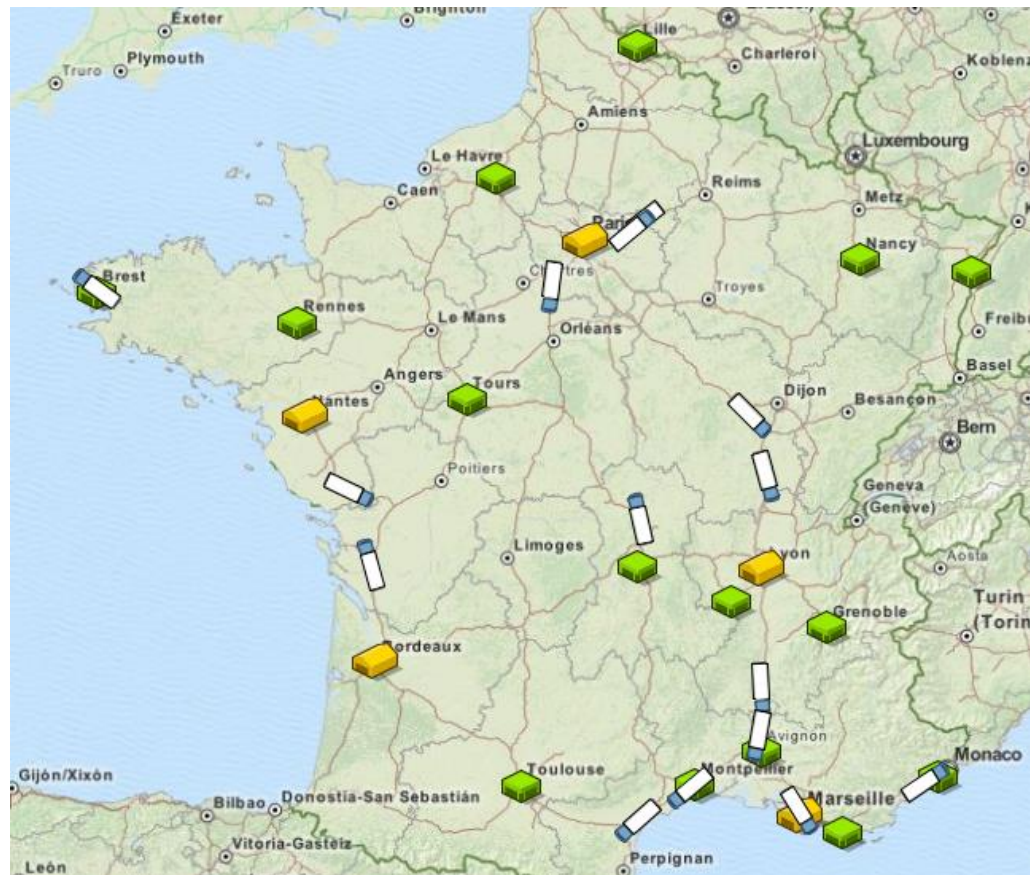


Модель цепочки поставок

Эта презентация является частью
стандартной программы обучения





Модель цепочки поставок

- Постройте имитационную модель простой цепочки поставок.
- Есть несколько дистрибьюторов и ритейлеров, расположенных в различных точках.
- Время от времени ритейлеры заказывают у дистрибьюторов определенное количество товара. Товар доставляется с помощью грузовиков. Доставив требуемый товар ритейлеру, грузовик возвращается к ближайшему дистрибьютору.



Создаем дистрибьюторов и ритейлеров. Фаза 1

- В первой фазе мы добавим в нашу модель агентов – дистрибьюторов и ритейлеров - и поместим их в пространство, заданное картой ГИС.

-
- Мы покажем, как:
 - Создавать популяции агентов, считывая данные из файла MS Excel
 - Расставлять агентов на карте ГИС



Цепочка поставок. Фаза 1. Шаг 1

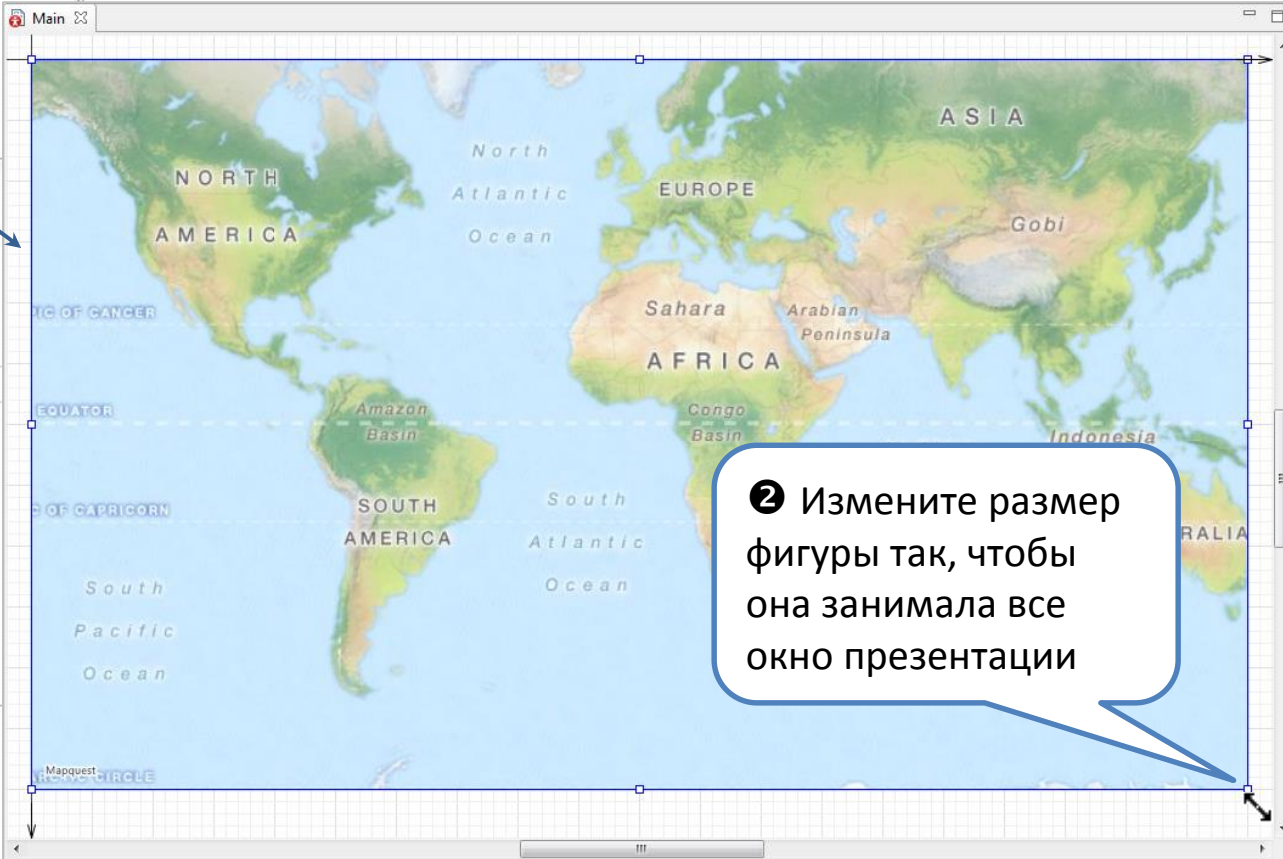
Проекты Палитра

Разметка пространства

- Путь
- Прямоугольный узел
- Многоугольный узел
- Точечный узел
- Аттрактор
- Стеллаж
- ГИС
 - ГИС Карта**
 - ГИС Точка
 - ГИС Маршрут
 - ГИС Регион
- Пешеходы
- Железные дороги

1 Добавьте Карту ГИС

2 Измените размер фигуры так, чтобы она занимала все окно презентации



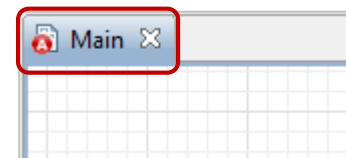
Создайте новую модель. Назовите ее *Supply Chain*.

Выберите **часы** в качестве **Единиц модельного времени**

Мы хотим создать агентную модель с несколькими типами агентов: дистрибьюторами, ритейлерами, грузовиками. Агенты будут жить в геопространственной среде, заданной картой ГИС. Прежде всего мы добавим на презентацию модели карту ГИС.

- 1 Перетащите элемент **ГИС Карта** из секции **ГИС** палитры **Разметка пространства**.
- 2 Измените размер фигуры, которая отображает карту. Пусть карта, на которой мы расположим агентов, занимает все окно презентации модели.

Совет: Пожалуйста, обращайте внимание на имя типа агента, редактируемого в данный момент времени:



ГИС

- Вы можете добавлять в ваши модели карты ГИС (GIS, Geographic Information System). Таким образом вы можете создавать модели с привязкой к географическим данным.
- Карты ГИС обычно используются в агентных моделях. Вы можете помещать агентов на ГИС карту, перемещать агентов с заданной скоростью из одной точки в другую, выполнять действия при достижении цели, рисовать анимацию агентов, устанавливая связи между агентами в соответствии с их расположением в пространстве и т.д.
- Реализация ГИС в AnyLogic основана на технологии OpenMap GIS.



Цепочка поставок. Фаза 1. Шаг 2

❶ Сделайте двойной щелчок мыши по карте, чтобы редактировать ее содержимое

❷ Вы можете перемещаться по карте, удерживая левую кнопку мыши.

Прокрутите колесо мыши от или на себя, чтобы изменить масштаб карты



❶ Двойным щелчком мыши по карте, или вызвав контекстное меню правой кнопкой мыши (пункт **Редактировать карту**), войдите в режим редактирования карты, чтобы выбрать область для модели в нужном масштабе.

Графический редактор за пределами карты будет затемнен, когда карта находится в режиме редактирования.

❷ Мы выбрали Францию в качестве места действия. Чтобы перетаскивать отображаемую область, перемещайте указатель, удерживая левую кнопку мыши. Чтобы увеличить масштаб карты, прокрутите колесо мыши от себя. При этом будет приближена та область карты, на которой находится указатель мыши.

Найдите на карте Францию и масштабируйте рабочую область карты для удобства использования.

Снова дважды щелкните по карте или просто щелкните в области графического редактора, чтобы выйти из этого режима.

ГИС Карта

- ГИС карта в AnyLogic по умолчанию отображает слой тайлов, которые загружаются с выбранного сервера. Тайловая карта - это карта, состоящая из незаметно соединенных отдельных изображений, количество которых зависит от масштаба карты.



Цепочка поставок. Фаза 1. Шаг 3

- Наши данные хранятся в книге MS Excel.
- Файл содержит два листа с данными по дистрибьюторам и ритейлерам.
- Давайте импортируем данные во встроенную БД AnyLogic прежде, чем использовать их в модели.

	A	B	C
1	Location	N trucks	
2	Marseille	4	
3	Paris	5	
4	Bordeaux	2	
5	Lyon	3	
6	Nantes	1	
7			
8			

	A		
1	Location		
2	Toulouse		
3	Lille		
4	Nice		
5	Brest		
6	Strasbourg		
7	Grenoble		
8	Rennes		
9	Rouen		
10	Toulon		
11	Montpellier		
12	Avignon		
13	Saint Etienne		
14	Tours		
15	Clermont-Ferrand		
16	Nancy		
17			

◀ ▶

Distributors

Retailers

◀ ▶

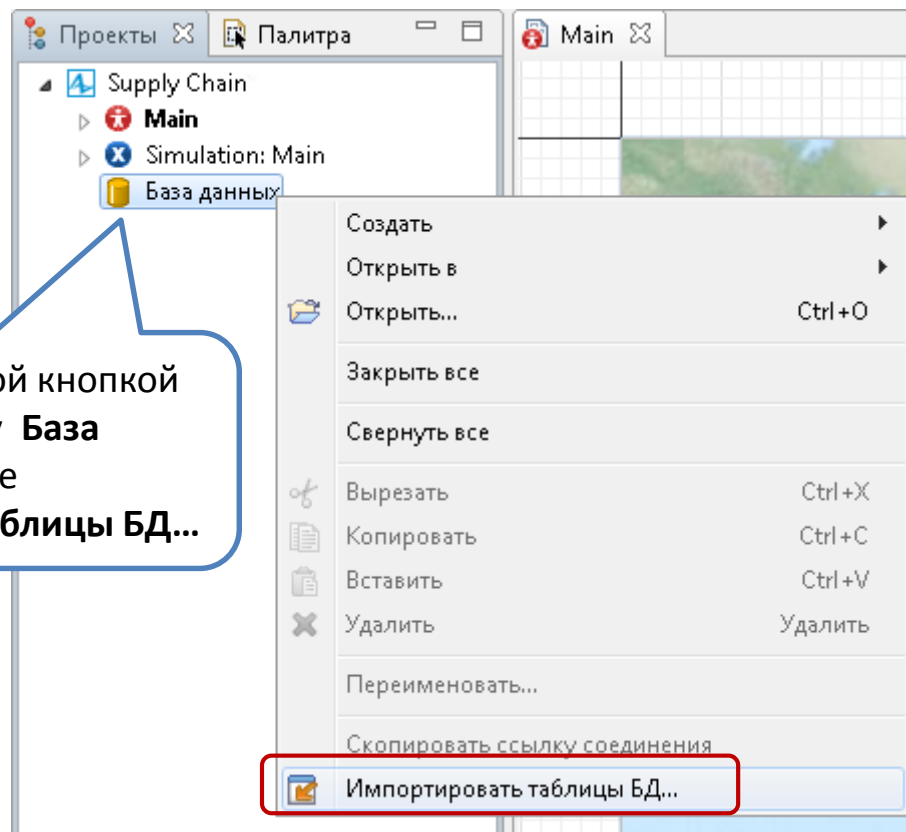
Distributors

Retailers

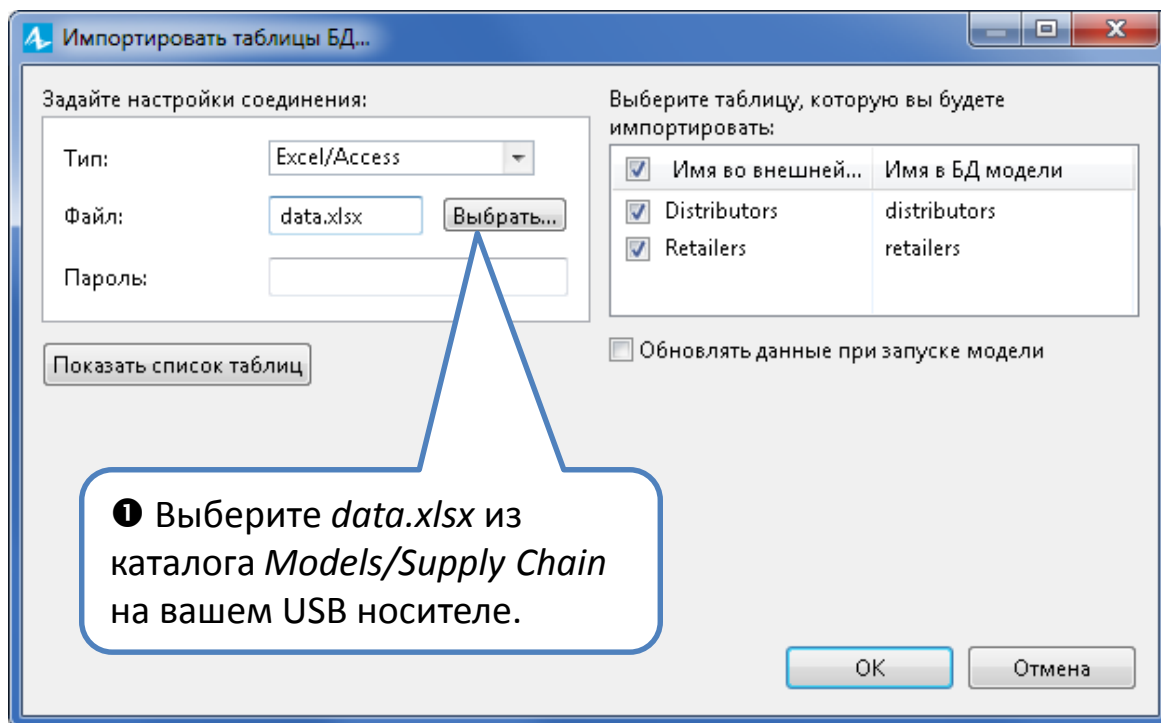


Цепочка поставок. Фаза 1. Шаг 4

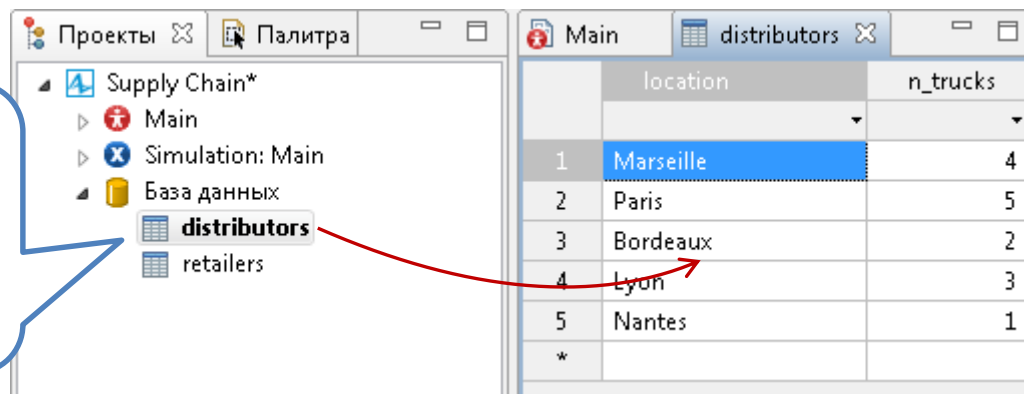
❶ Щелкните правой кнопкой мыши по элементу **База данных** и выберите **Импортировать таблицы БД...**



Цепочка поставок. Фаза 1. Шаг 5



2 Вы увидите, что в **Базе данных** было создано две таблицы: *distributors* и *retailers*. Щелкните два раза по таблице, чтобы просмотреть ее содержимое.



Импортируем данные из внешней БД (файл MS Excel) во встроенную БД AnyLogic.

БД AnyLogic

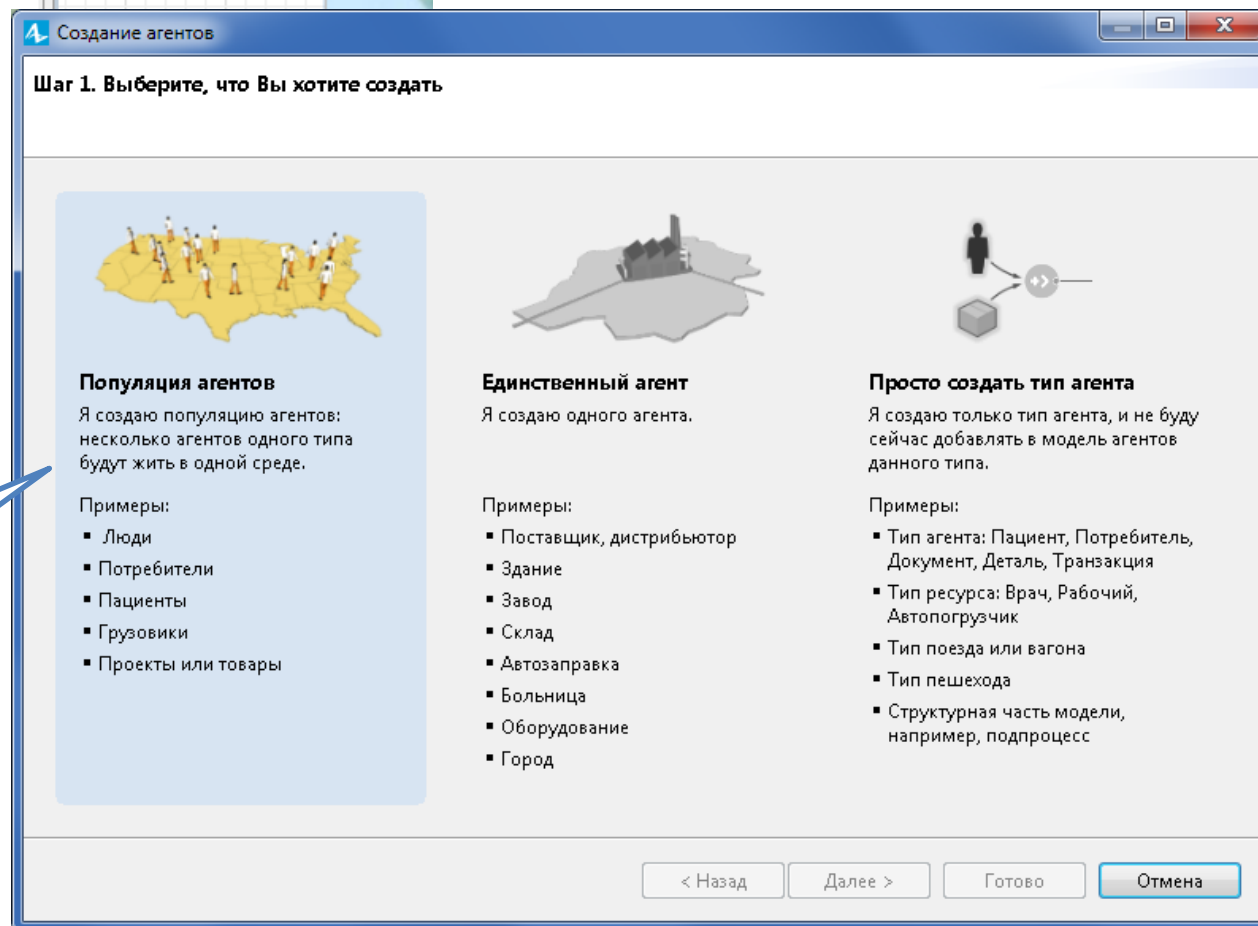
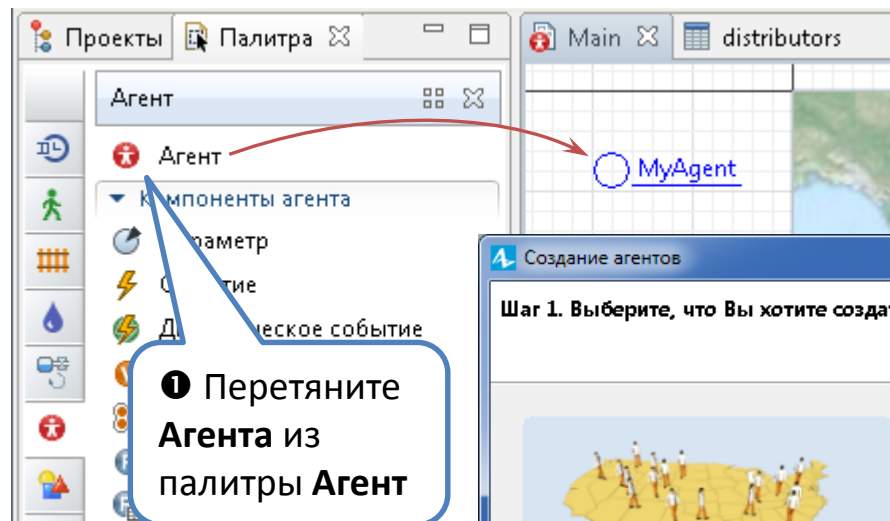
У каждой модели AnyLogic теперь есть встроенная, полностью интегрированная БД, которой можно пользоваться для считывания входных данных и записи данных моделирования. Экспорт модели с БД сейчас настолько же портативен и кросс-платформен, как и сама модель. С новой БД вы можете:

- Считывать значения параметров и настраивать модели
- Создавать популяции агентов с заданными параметрами
- Создавать прибытие заявок в моделировании процессов
- Импортировать данные из других БД или книг Excel и хранить их в готовом и доступном виде
- Вести запись данных диаграмм процессов, переходов диаграмм состояний, передачу сообщений и биографии агентов.
- Видеть использование ресурсов, время в пути, ожидания, обработки
- Хранить и экспортировать статистику, наборы данных, другие журналы
- Экспортировать данные в книги MS Excel.

Вы можете легко импортировать БД в ваш проект AnyLogic из БД или книги MS Excel. Вы также можете создать пустую БД и заполнить таблицы вручную.



Цепочка поставок. Фаза 1. Шаг 6



2 Выберите опцию Популяция агентов



Цепочка поставок. Фаза 1. Шаг 7

Шаг 2. Создание нового типа агента

Имя нового типа:

Имя популяции:

☐ Создать новый тип агента "с нуля"

☒ Использовать таблицу базы данных

☐ Агент будет использоваться в диаграммах процессов

1 Задайте Имя нового типа: *Distributor*

2 Выберите Использовать таблицу базы данных

< Назад **Далее >**

Шаг 3. Выберите таблицу, хранящую данные по агентам данного...

☒ Использовать существующую таблицу:

☐ Импортировать таблицу из внешнего источника данных

Задайте настройки соединения:

Тип:

Файл:

Пароль:

Выберите таблицу, которую вы будете импортировать:

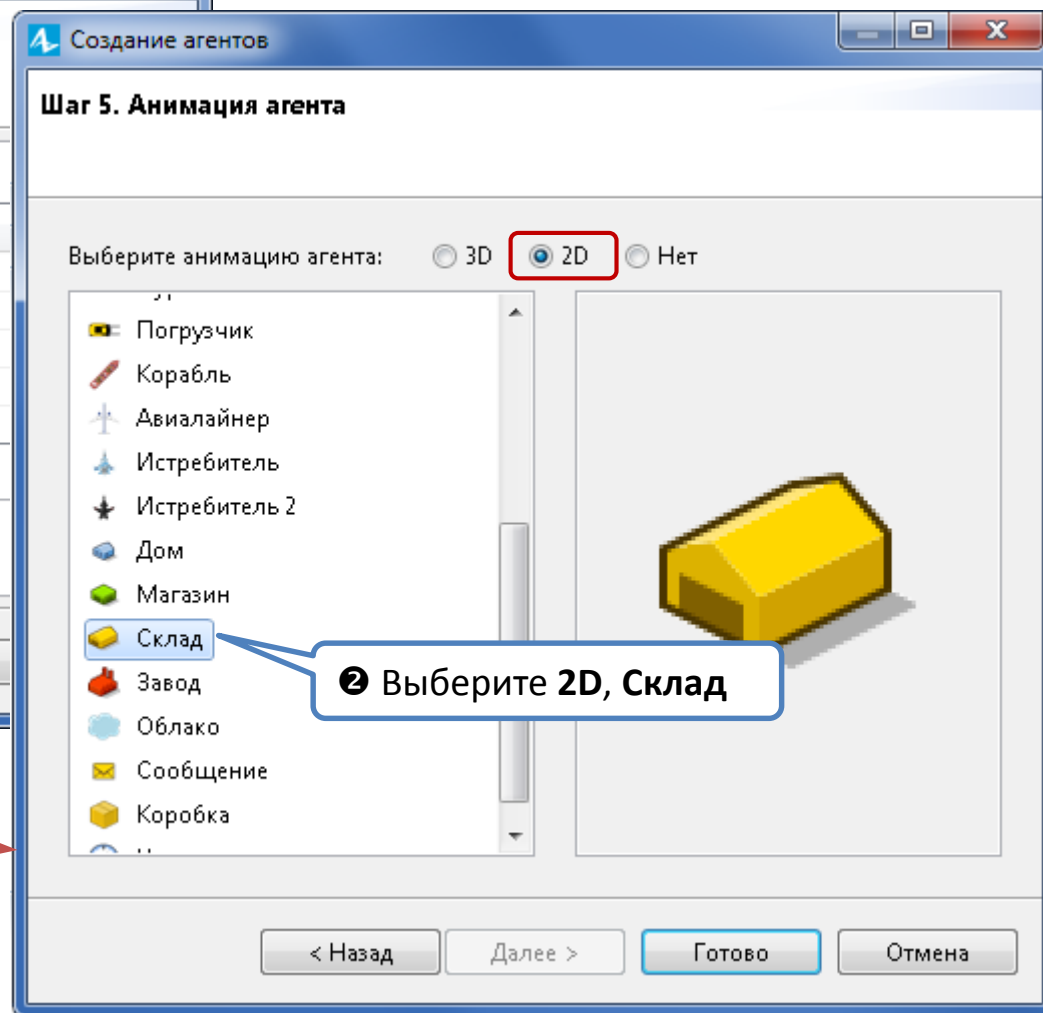
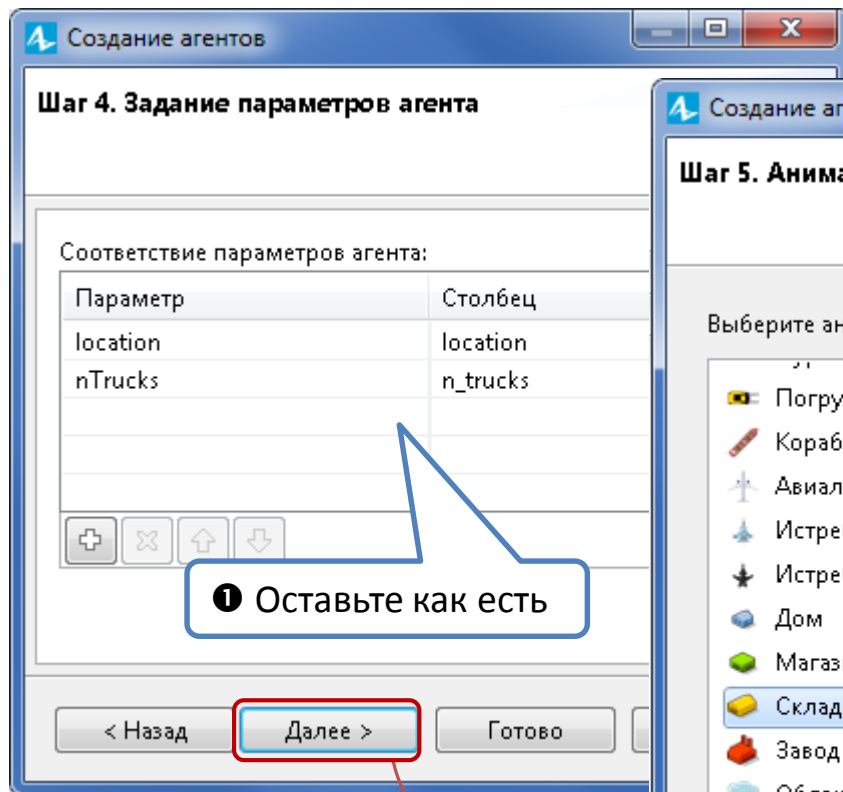
Имя во внешней БД	Имя в БД модели

☐ Обновлять данные при запуске модели

< Назад **Далее >** Готово Отмена



Цепочка поставок. Фаза 1. Шаг 8



Мы начнем с создания популяции агентов, которая будет моделировать дистрибьюторов.

Создание популяции агентов

- AnyLogic предоставляет простой способ создания популяций агентов простым перетаскиванием элемента **Агент** из палитры **Агент** на диаграмму того типа агента, в котором вы создаете популяцию.
- Мастер создает:
 - Тип агента (*Distributor*). Здесь вы задаете внутреннюю структуру агента, его поведение, рисуете для него анимацию и т.д.
 - Популяцию агентов (*distributors*). Этот объект вложен в тип, в который Вы перетащили элемент **Агент** (*Main*), и состоит из нескольких (5) экземпляров типа агента *Distributor*, каждый из которых моделирует определенного агента-дистрибьютора. Рядом с объектом вы можете увидеть его анимацию.



Цепочка поставок. Фаза 1. Шаг 9

The screenshot shows the AnyLogic software interface. On the left, a 'Main' window contains a red star icon labeled 'distributors [...]'. A red arrow points from this icon to a configuration window titled 'Начальное местоположение' (Initial location). The configuration window has the following options:

- Расположить агентов: (Place agents:)
 - ☐ в месте расположения анимации агента (at the location of the agent animation)
 - ☐ в точке (широта, долгота) (at a point (latitude, longitude))
 - ☐ в узле (at a node)
 - ☒ в 1-м результате ГИС поиска (at the 1st result of the GIS search)
- Название места: (Location name): `self.location`
- Маршруты: (Routes): По умолчанию (By default)

Two callout boxes provide instructions:

- 1 Расположите популяцию агентов за рамкой окна презентации (Place the population of agents outside the presentation window)
- 2 Настройте популяцию, чтобы считать расположения агентов из базы данных (Configure the population to read agent locations from the database)



Расположение агентов в ГИС пространстве

В секции свойств популяции агентов **Начальное местоположение** вы можете выбрать расположение агентов из этой популяции:

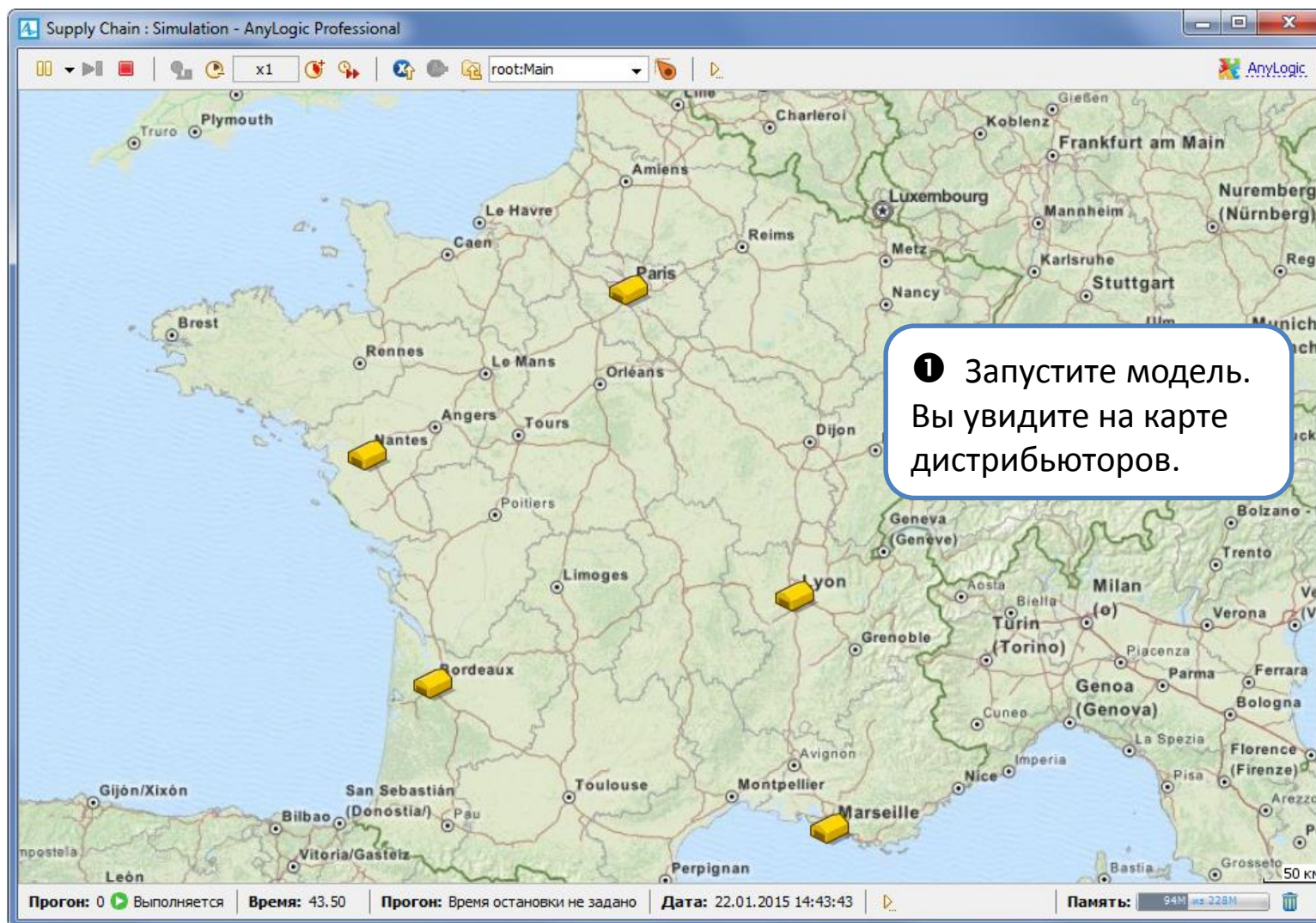
- **В месте расположения анимации объекта** – найдите желтую фигуру склада на вашей карте. Это вложенная презентация агента. Если выбрать этот пункт, агент будет расположен именно там, куда вы его перетяните на ГИС карте.

Если вы не видите вложенную анимацию агента при создании анимации объекта после того, как создадите его популяцию на агенте верхнего уровня Main, нажмите на кнопку **Показать презентацию** в разделе **Специфические** панели **Свойства** популяции агента (к примеру distributors).

- **В точке (широта, долгота)** – вы указываете координаты агента ниже в полях широта, долгота. Если у вас популяция агента, значения координат которого указаны в свойствах lat и lon, пропишите self.lat и self.lon в полях, чтобы расположить агента в пространстве ГИС в соответствии с этими координатами.
- **В узле** – вы создаете места на карте с помощью ГИС точек и указываете имя ГИС точки, в которой вы хотите расположить агента. Этим пунктом можно пользоваться, когда у вас один агент, а не популяция агентов, состоящая из множества агентов.
- **В первом результате ГИС поиска** – Используйте этот вариант, когда места указаны в виде текста (наш случай): "Paris", "Marseille", и т.д. Эти имена используются ГИС картой, которая ищет места с этими именами и выдает первый результат поиска. Если у вашего агента есть параметр, содержащий имя места (как у нас, параметр location), пропишите self.location в поле ниже, чтобы расположить агентов в нужных точках ГИС пространства.

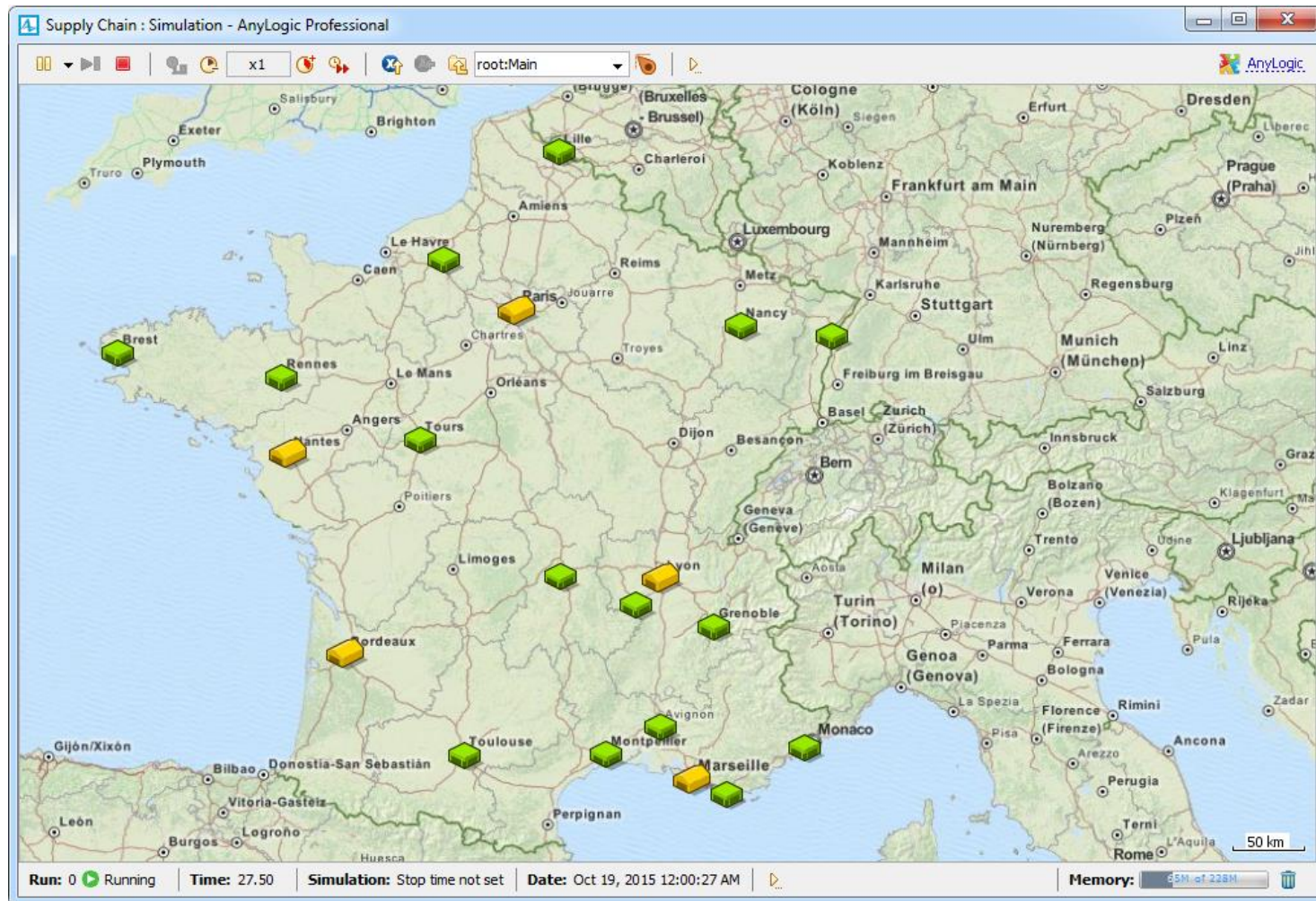


Цепочка поставок. Фаза 1. Шаг 10



Цепочка поставок. Фаза 1. Шаг 11

- Создайте ритейлеров так же, как создали дистрибьюторов. Назовите тип агента Retailer и считайте данные о местоположении из БД AnyLogic.

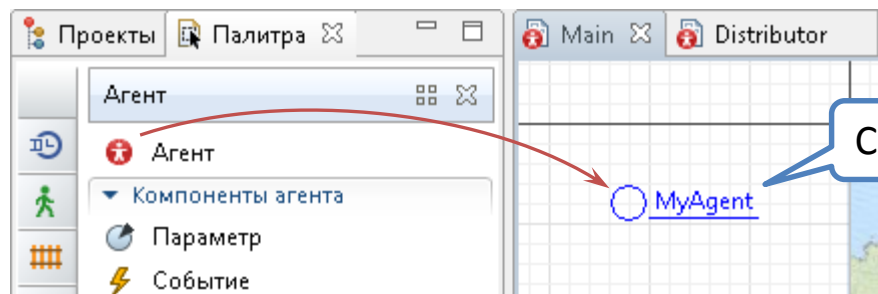


Цепочка поставок. Фаза 2

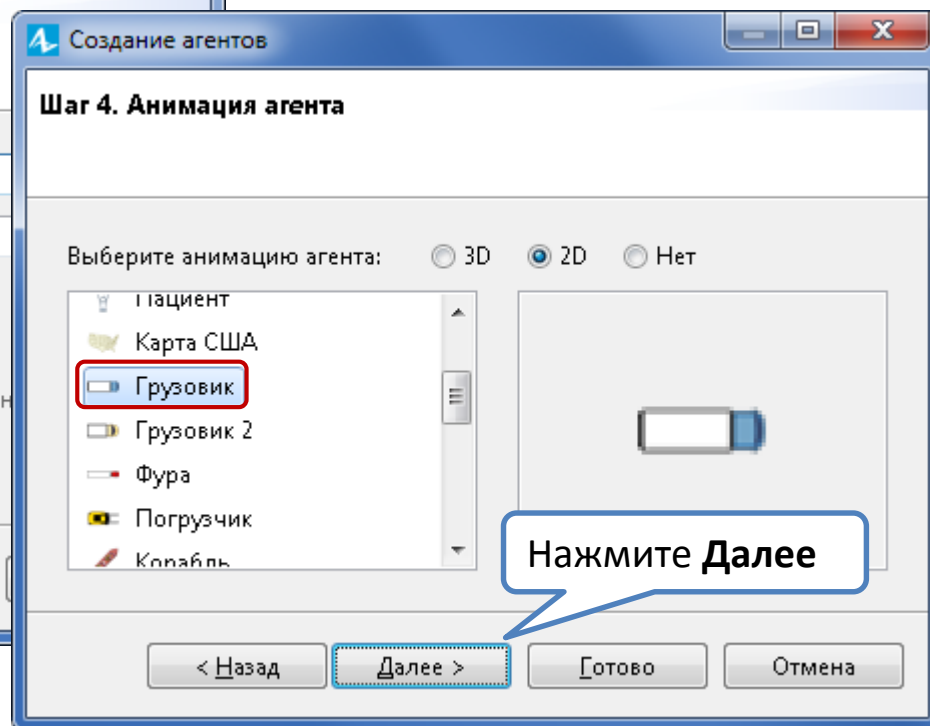
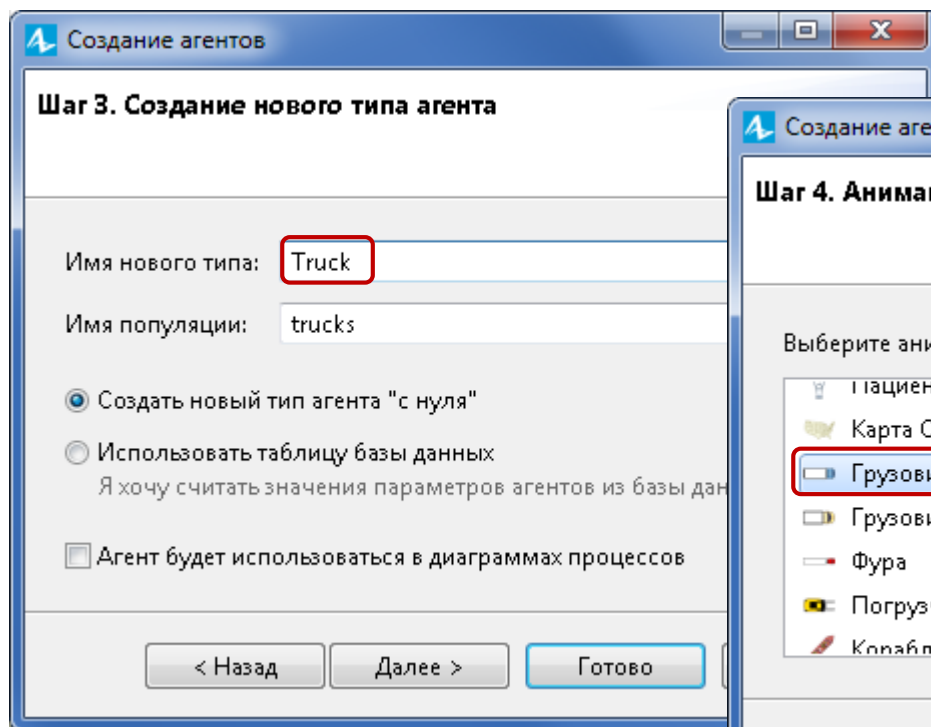
- Сейчас мы создадим еще одну популяцию агентов - грузовики.
- Мы будем считывать данные (количество грузовиков и их расположение) из базы данных.



Цепочка поставок. Фаза 2. Шаг 1



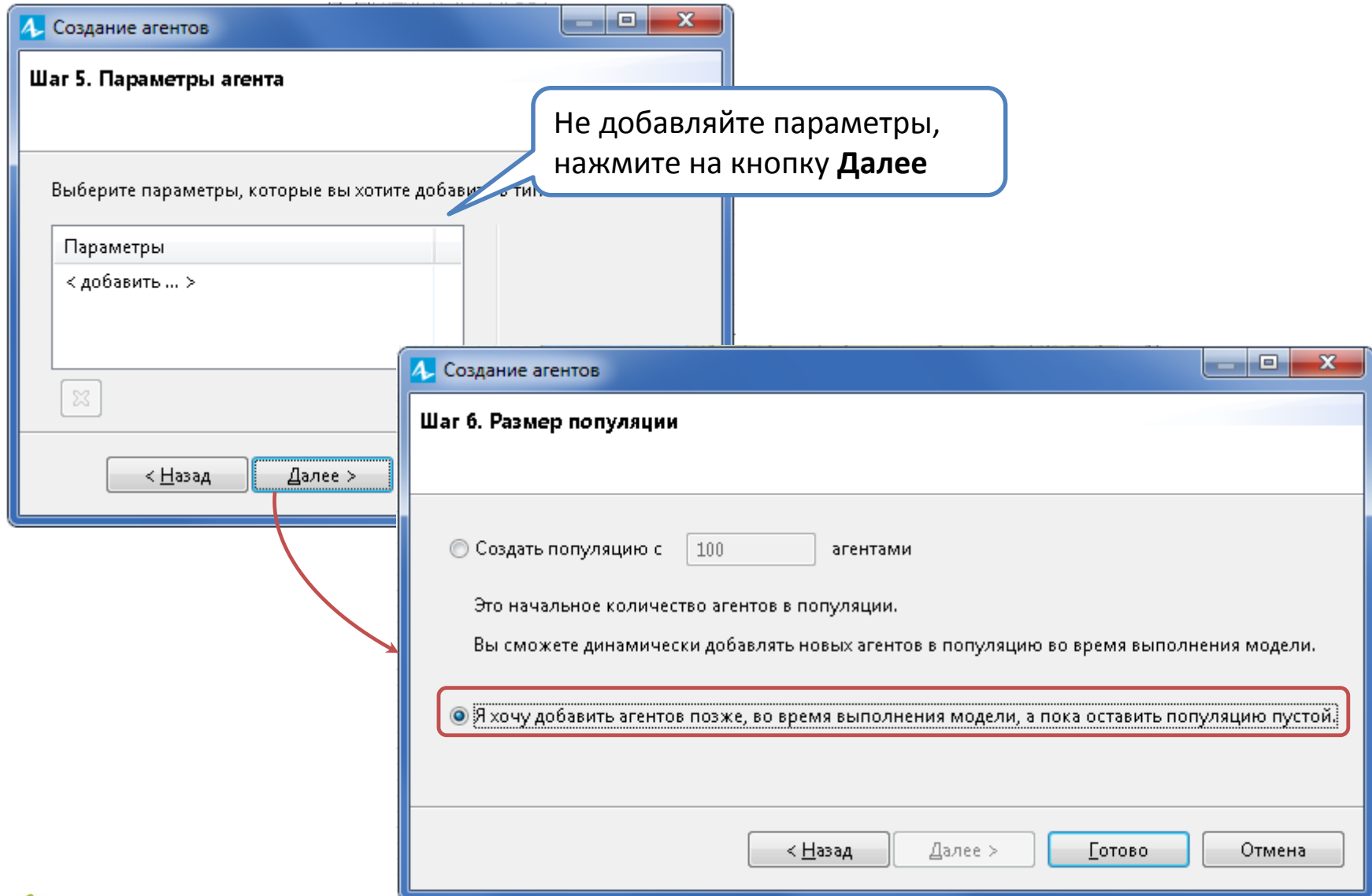
Создайте популяцию грузовиков



Нажмите Далее



Цепочка поставок. Фаза 2. Шаг 2



Цепочка поставок. Фаза 2. Шаг 3

The screenshot displays the AnyLogic software interface. On the left, the 'Distributor' map shows a geographical area with several entities: 'distributors [...]', 'retailers [...]', and 'trucks [...]'. A blue callout bubble points to the 'trucks' entity with the text 'Измените настройки trucks'.

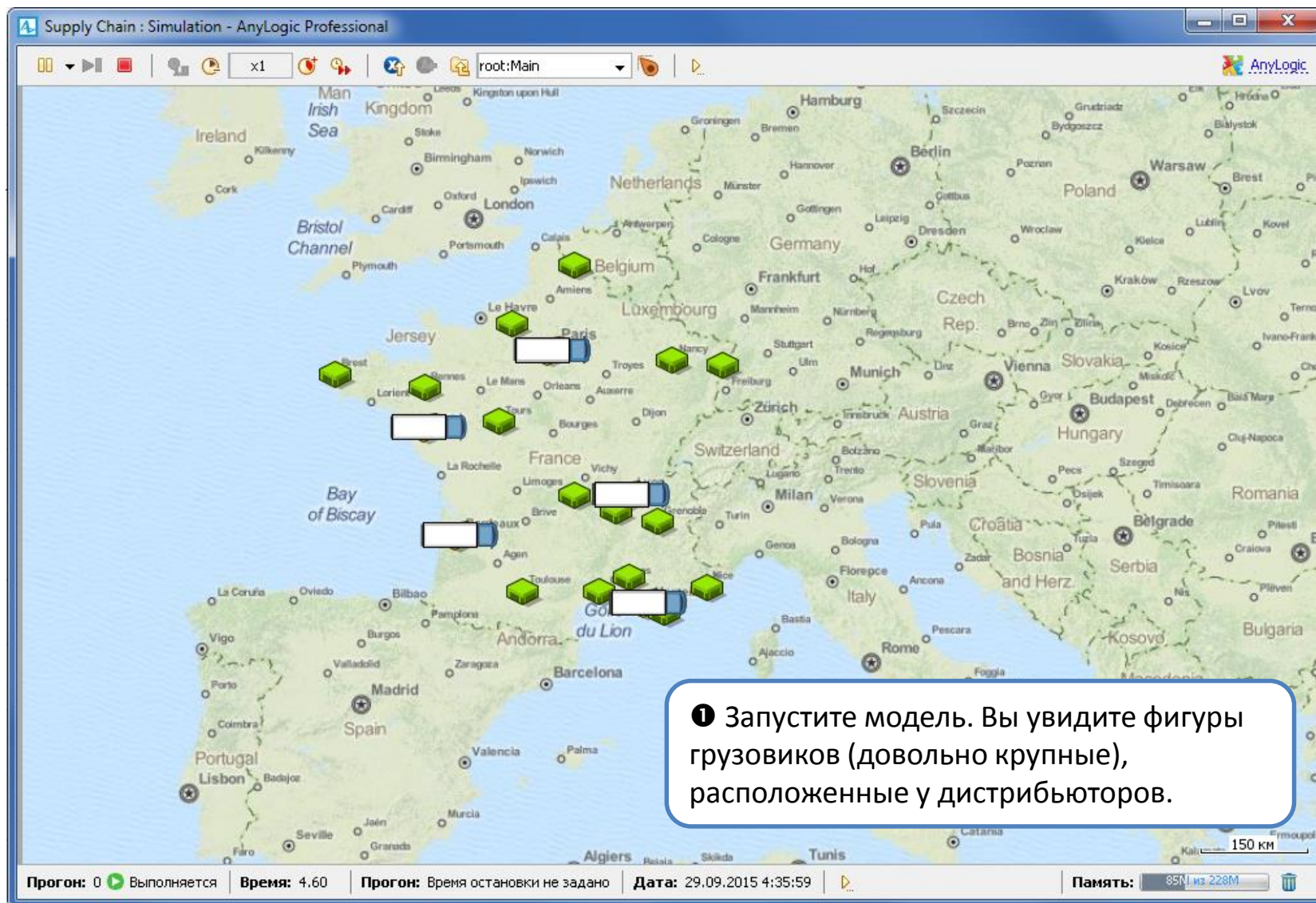
On the right, the 'Свойства' (Properties) window for the 'trucks - Truck' entity is open. The window contains the following settings:

- Имя: trucks
- ☒ Отображать имя ☐ Исключить
- ☐ Одиночный агент ☒ Популяция агентов
- Популяция: ☐ Изначально пуста ☐ Содержит заданное кол-во агентов ☒ Загружается из базы данных
- Таблица: distributors
- Условия выборки: (Buttons: +, -, X, Up, Down)
- Режим: ☐ Один агент для каждой записи в БД ☒ Несколько агентов по одной записи
- Кол-во задано в столбце: n_trucks
- Соответствие параметров агента:

Параметр	Столбец
GIS location name	location



Цепочка поставок. Фаза 2. Шаг 4





Фаза 3. Определяем логику

- Сейчас мы смоделируем запрос товаров ритейлерами и доставку от дистрибьюторов к ритейлерам.
- Для этого нам нужно создать спрос на продукты, присвоить его грузовикам и заставить их передвигаться от их расположения у дистрибьюторов к ритейлерам и назад.



Цепочка поставок. Фаза 3. Шаг 1

1 Перетяните Агента из палитры Агент

2 Выберите Просто создать тип агента

Шаг 1. Выберите, что Вы хотите создать

Популяция агентов
Я создаю популяцию агентов: несколько агентов одного типа будут жить в одной среде.

Примеры:

- Люди
- Потребители
- Пациенты
- Грузовики
- Проекты или товары

Единственный агент
Я создаю одного агента

Примеры:

- Здание
- Завод
- Склад
- Автозаправка
- Больница
- Оборудование
- Город

Просто создать тип агента
Я создаю только тип агента, и не буду сейчас добавлять в модель агентов данного типа.

Примеры:

- Тип агента: Пациент, Потребитель, Документ, Деталь, Транзакция
- Тип ресурса: Врач, Рабочий, Автопогрузчик
- Тип поезда или вагона
- Тип пешехода
- Структурная часть модели, например, подпроцесс

< Назад Далее > Готово Отмена



Цепочка поставок. Фаза 3. Шаг 2

Шаг 2. Создание нового типа агента

Имя нового типа:

☒ Создать новый тип агента "с нуля"

☐ Использовать существующий тип агента

☐ Агент будет...

1 Имя нового типа: *Order*

2 Выберите анимацию агента: *Нет*

Выберите анимацию агента: ☐ 3D ☐ 2D ☒ Нет

Шаг 3. Анимация агента

Шаг 4. Параметры агента

Выберите параметры, которые вы хотите добавить в тип агента

Параметры
amount
< добавить ... >

Параметр:

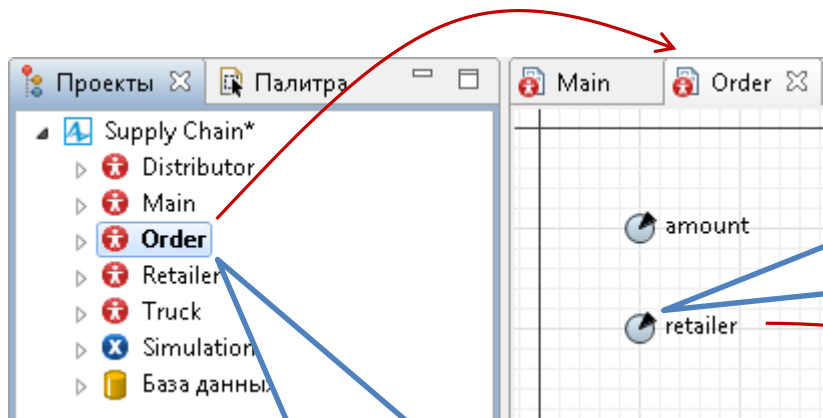
Тип:

Значение по умолчанию:

3 Добавьте параметр - количество товаров, запрошенных этим заказом: *amount* типа *int*

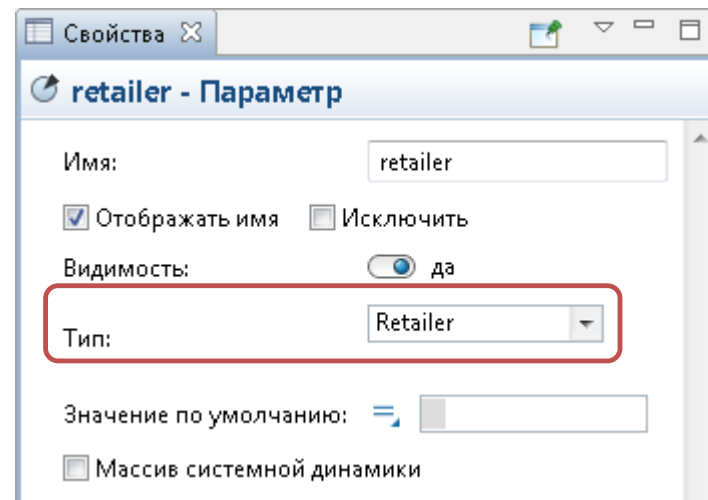


Цепочка поставок. Фаза 3. Шаг 3



❶ Откройте графическую схему *Order* двойным щелчком

❷ Добавьте еще один параметр (Параметр из палитры **Агент**) чтобы запомнить ритейлера - источника этого заказа (*retailer* типа *Retailer*)



Цепочка поставок. Фаза 3. Шаг 4

The screenshot displays the AnyLogic software interface. On the left, the 'Проекты' (Projects) pane shows a tree structure under 'Supply Chain*' with nodes: Distributor, Main, Order, Retailer, **Truck**, Simulation, and База данных. The 'Main' window shows a diagram with a truck icon and a variable icon labeled 'order'. A red arrow points from the 'Переменная' (Variable) icon in the 'Агент' (Agent) pane to the 'order' variable in the diagram. The 'order' variable is highlighted with a red box in the 'Свойства' (Properties) window.

1 Откройте графическую диаграмму *Truck* двойным щелчком

2 Уменьшите размер *Грузовика*

3 Добавьте переменную для хранения текущего заказа

order - Переменная

Имя: order

☒ Отображать имя ☐ Исключить

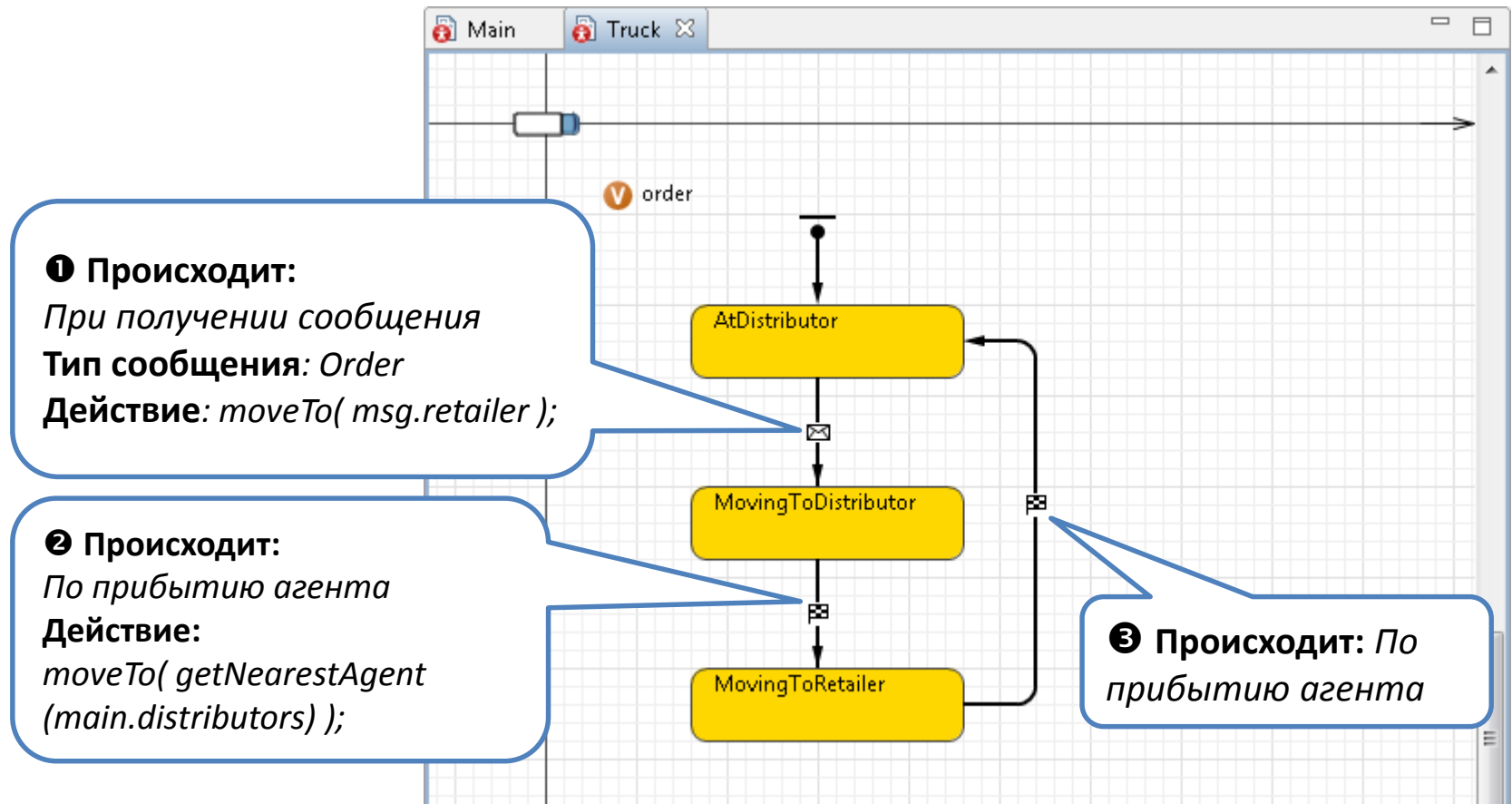
Видимость: ☒ да

Тип: Order

Начальное значение: []



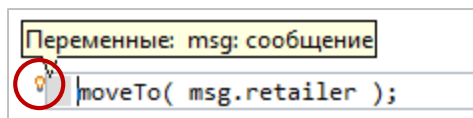
Цепочка поставок. Фаза 3. Шаг 5



Задайте поведение грузовика с помощью диаграммы состояний. В нашем случае, грузовик изначально находится у дистрибьютора (поэтому начальным состоянием является состояние *AtDistributor*). Пока грузовик находится там, он может получить заказ и начать движение к ритейлеру, пославшему этот заказ. Сейчас мы полагаем, что достигнув ритейлера, грузовик немедленно возвращается обратно. Перемещение является длительным действием, и мы можем представить его как состояние. Таким образом, есть еще два состояния: *MovingToRetailer* и *MovingToDistributor*. Чтобы завершить пребывание грузовика в состоянии перемещения, мы используем специальный тип перехода, срабатывающий по приходу агента в место назначения.

Локальные переменные

- Обратите внимание, что в коде действия состояния мы используем переменную *msg*. *msg* здесь – это локальная переменная, доступная только в текущем контексте. Чтобы посмотреть на список всех локальных переменных, щелкните в поле и затем наведите курсор мыши на лампочку, которая появится в строке слева от поля:



Перемещение агента в непрерывном и ГИС пространстве

- Чтобы переместить агента в новое место, вызовите метод *moveTo()* этого агента, передав в качестве аргументов метода координаты нового местоположения.
- Обратите внимание на следующие функции (полный список см. в *Справке AnyLogic*):

Функция	Описание
<i>moveTo(double x, double y)</i>	Начинает перемещение агента в точку с заданными координатами.
<i>jumpTo(double x, double y)</i>	Немедленно перемещает агента в заданное место.
<i>moveTo(String geographicPlace)</i>	Начинает перемещение агента в указанный объект на карте.
<i>moveTo(node)</i>	Начинает перемещение агента в указанный объект ГИС разметки на карте: ГИС точку или регион.
<i>stop()</i>	Останавливает агента; агент остается при этом в текущем месте.
<i>double timeToArrival()</i>	Возвращает время, оставшееся до прибытия в точку назначения, в единицах модельного времени.
<i>boolean isMoving()</i>	Проверяет, движется ли агент в данный момент времени.
<i>double getVelocity ()</i>	Возвращает текущую «крейсерскую скорость» агента. Ненулевая скорость не означает, что агент движется в данный момент времени.
<i>setVelocity(double newVelocity)</i>	Задаёт «крейсерскую скорость» агента. Если агент движется, то он продолжит движение с новой скоростью. Если нет, то он не начнет движение, пока вы не вызовете метод <i>moveTo()</i> .
<i>setLatLon(double latitude, double longitude)</i>	Укажите координаты расположения агента.
<i>double getLatitude()</i>	Возвращает значение текущей широты агента.
<i>double getLongitude()</i>	Возвращает значений текущей долготы агента.



Цепочка поставок. Фаза 3. Шаг 6

The screenshot displays the AnyLogic development environment. On the left, the 'Палитра' (Palette) window shows the 'Событие' (Event) icon selected. A red arrow points from this icon to the 'orderProducts' event in the 'Main' diagram. The 'Main' diagram shows a green cube representing the 'Retailer' agent and a 'location' object. A callout box with the text '1 Откройте диаграмму Retailer' points to the 'Retailer' tab. On the right, the 'Свойства' (Properties) window for the 'orderProducts' event is open. It shows the event name 'orderProducts', visibility settings, and a circled 'Действие' (Action) section containing the following Java code:

```
Order order = new Order( uniform_discr(5, 20), this);

Truck truck = getNearestAgent(filter
(main.trucks, t-> t.inState(Truck.AtDistributor)));

if (truck != null) {
    truck.order = order;
    send(order, truck);
}
```



Создайте событие *orderProducts*, которое будет периодически создавать запросы на товары.

Мы создаем новый заказ, вызывая конструктор типа *Order: new Order (amount, this)*. Конструктор принимает два аргумента, инициализирующие поля класса *amount* и *retailer*. Обратите внимание на код Java класса *Order*, чтобы понять порядок следования аргументов. Мы инициализируем поле *retailer* ссылкой на данный объект *Retailer*, доступный из текущего контекста по ключевому слову Java *this*.

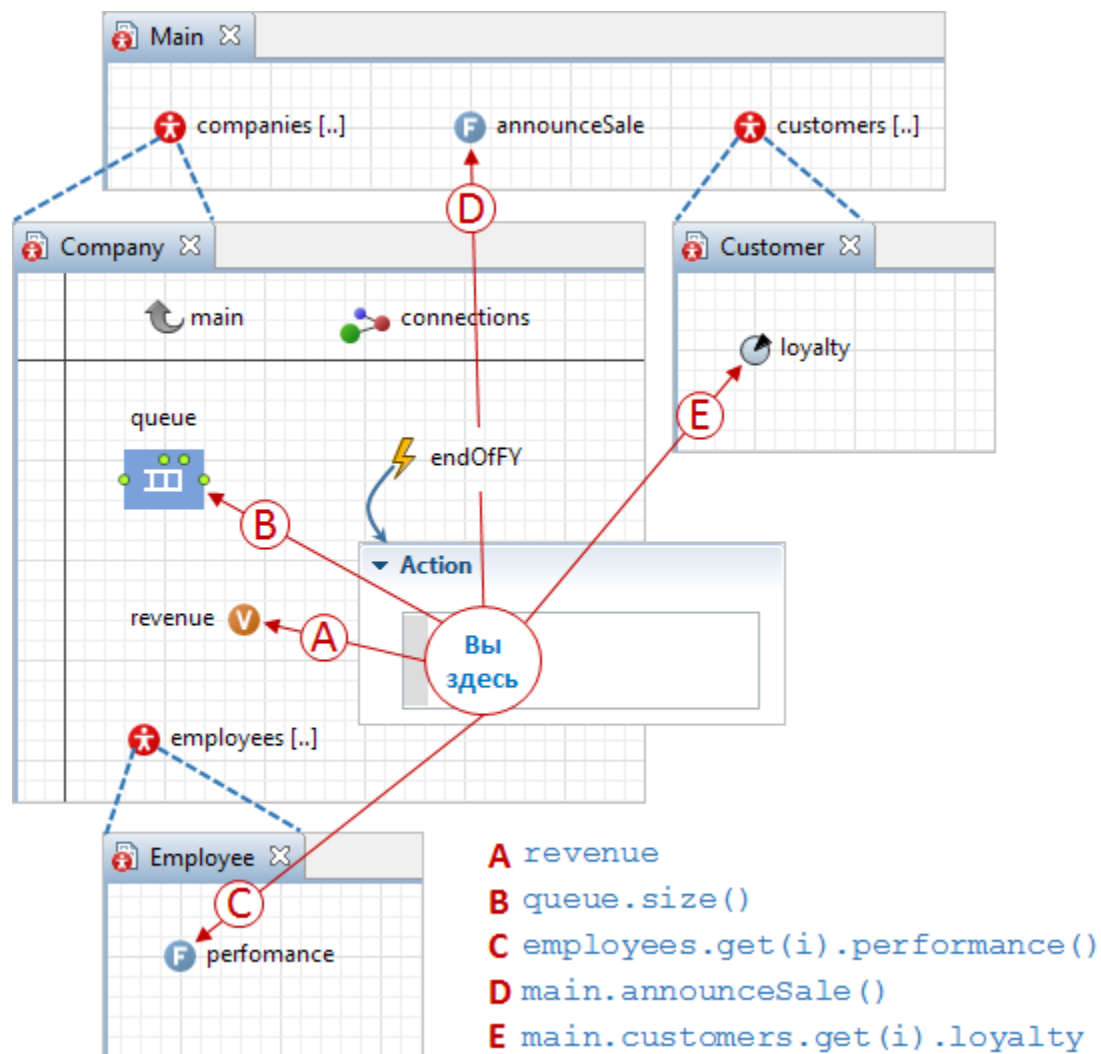
Затем мы ищем ближайший свободный грузовик.

- Функция *getNearestAgent()* возвращает ближайшего агента из указанной популяции. В данном случае мы передадим в качестве аргумента функции не всю популяцию грузовиков *main.trucks*, а отфильтруем эту популяцию с помощью функции *filter()*, выбирая ближайший грузовик только из числа свободных в данный момент.
- Первый аргумент функции *filter()* задает популяцию, по которой будет производиться фильтрация, в этом случае *main.trucks*. Затем задается имя элемента (*t ->*), после чего можно задавать условия фильтрации для элемента, в виде *t.<условие>*. Функция вернет подмножество агентов из популяции, удовлетворяющих это условие.
- Функция *inState(state)* проверяет, находится ли агент в данный момент времени в заданном состоянии. Таким образом, мы отфильтровываем результаты, и рассматриваем только свободные грузовики (которые сейчас находятся у дистрибьютора).

Когда мы находим свободный грузовик, мы присваиваем ему заказ и посылаем сообщение его диаграмме состояний, чтобы инициировать его движение. Если все грузовики окажутся занятыми, то функция вернет значение *null*.



Java код. Как получить доступ к... ?



В AnyLogic Вы не пишете весь код для Java классов от начала до конца. Вместо этого, Вы вводите фрагменты кода и выражения в специальных текстовых полях свойств различных элементов. Поэтому важно всегда понимать, где именно Вы пишете код в данный момент (какому классу он принадлежит), и как получить доступ к другим элементам модели из этого кода.

- Элементы модели, принадлежащие тому же типу агента, доступны просто по именам. Скажем, Вы пишете код в поле свойства **Действие** события *endOfFY* агента *Company*. Чтобы получить доступ к блоку *queue*, просто напишите *queue*. Чтобы увеличить значение переменной *revenue*, напишите *revenue += amount*. Чтобы перезапустить событие *endOfFY* из кода его собственного действия: *endOfFY.restart(year())*
- Чтобы получить доступ к полю объекта, Вы должны поставить точку "." после имени объекта и затем написать имя этого поля. Например, чтобы получить размер очереди *queue*, нужно написать *queue.size()*. В случае популяции агентов, Вам нужно указать, какой именно агент из этой популяции Вам нужен. Чтобы вызвать функцию *performance()* у агента под номером 247 из популяции *employees*, напишите: *employees.get(246).performance()*.
- Если популяция агентов типа *Company* задана на диаграмме агента *Main*, то чтобы получить доступ к агенту *Main* из агента *Company*, Вы должны написать *main*. Поэтому, чтобы вызвать функцию *announceSale()* типа агента *Main*, Вам нужно написать *main.announceSale()*.
- Чтобы получить доступ к агенту другой популяции, живущему в той же среде, нужно вначале перейти на уровень выше, а затем спуститься по иерархии модели вниз, к нужному агенту. Чтобы получить значение переменной *loyalty* определенного агента популяции *customers* из типа агента *Company*, нужно обратиться к типу агента *Main*, а затем - к агенту *Customer* (с нужным индексом): *main.customers.get(i).loyalty*



Цепочка поставок. Фаза 3. Шаг 7

