

Лабораторная работа № 1

Агентная модель потребительского рынка в Anylogic¹

Краткая теория

Агентное моделирование - относительно новый метод моделирования. Поначалу оно являлось преимущественно предметом теоретических дискуссий в академических кругах, а начиная с 2000-х годов разработчики имитационных моделей стали использовать его на практике. Переход к агентному моделированию был вызван:

- Желанием глубже изучить системы, которые сложно описать традиционными методами моделирования.
- Развитием технологии агентного моделирования (объектно-ориентированное моделирование, диаграммы состояний).
- Быстрому росту мощности процессоров и объема оперативной памяти компьютеров. Агентные модели более требовательны к ресурсам, чем модели системной динамики или дискретно-событийные модели.

Агентное моделирование предлагает разработчику моделей альтернативный взгляд на поведение системы.

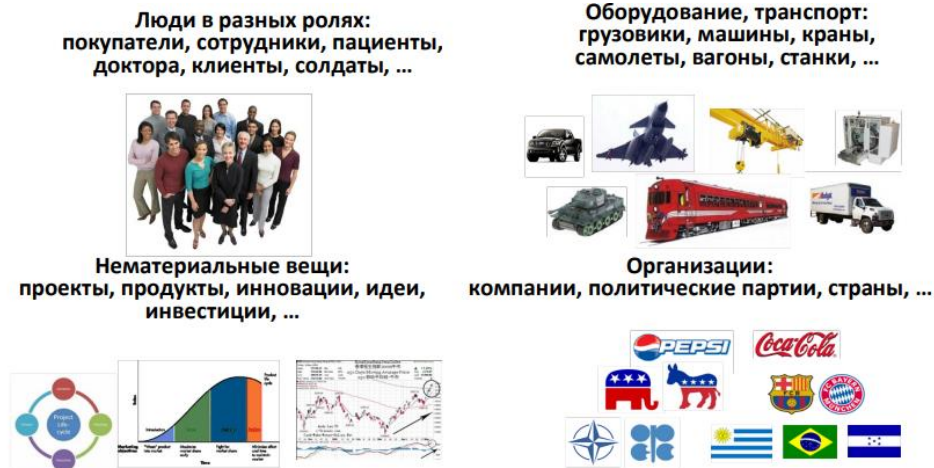
- ✓ **Вы можете не знать ни поведения системы в целом, ни ее главных переменных и зависимостей между ними, или не видеть четкой схемы процессов, но при этом понимать, как ведут себя отдельные элементы системы. В таком случае вы можете начать создание модели с идентификации моделируемых объектов (агентов) и задания их поведения. Иногда вам может понадобиться объединить агентов в сеть и позволить им взаимодействовать друг с другом, либо же поместить агентов в среду, которая имеет свою собственную динамику. Таким образом, глобальное поведение системы формируется из многих десятков (тысяч, миллионов) параллельно протекающих процессов.**

На данный момент не существует стандартного языка агентного моделирования. Структура агентной модели может быть задана как графически, так и с помощью сценариев. Поведение агента может быть задано различными способами. Если у агента есть состояние, от которого зависят его действия и реакции, то его поведение лучше всего задавать с помощью диаграммы состояний. Иногда поведение агента задается действиями, выполняемыми при наступлении определенных событий.

Иногда внутренняя динамика агента лучше всего задается с помощью дискретных событий или системной динамики. Так же и динамика среды, в которой живут агенты, может моделироваться с помощью традиционных методологий. По этой причине многие агентные модели совмещают в себе несколько подходов к моделированию.

Агентами могут быть самые разные объекты: транспортные средства, оборудование, проекты, организации, земельные участки, люди и так далее.

¹ Григорьев И. AnyLogic за 3 дня: практическое пособие по имитационному моделированию. - 2022. - 272 с.



Ученые до сих пор спорят, какими именно свойствами должен обладать объект, чтобы называться агентом: способностью действовать и реагировать на действия других, ориентироваться в пространстве, обучаться, взаимодействовать и общаться, обладать «интеллектом» и т.п. На практике в агентных моделях вам могут встретиться агенты любых типов: одни общаются друг с другом, а другие находятся в полной изоляции; одни живут в пространстве, а другие – нет; одни обучаются и приспосабливаются, а другие никогда не меняют своего поведения.

Приведем несколько полезных фактов об агентах, чтобы многообразие теорий не вводило вас в заблуждение:

- **Агенты не являются клеточными автоматами** и не обязательно обитают в дискретном пространстве (как в игре «Жизнь»). Во многих агентных моделях пространство вообще отсутствует. Когда пространство все же необходимо, оно чаще всего является непрерывным (это может быть карта мира или план здания).
- **Агенты – не обязательно люди.** Агентом может быть все, что угодно: транспортное средство, оборудование, проект, организация или даже идея.
- **Агентом может быть объект, кажущийся абсолютно пассивным.** Например, в модели нефтепровода вы можете представить сегмент трубы как агента, задав для него графики техобслуживания, вероятности происхождения аварий, логику проведения ремонтных работ, затраты и т.д.
- **Агентов в модели может быть как много, так и мало.** При этом агенты могут быть как одного типа, так и разных.
- **Существуют агентные модели, в которых агенты вообще не взаимодействуют друг с другом.** Например, в моделях потребления алкоголя, развития ожирения или хронических заболеваний индивидуальная динамика агента зависит только от его личных параметров и, в некоторых случаях, от среды.

Практическое задание

Построим агентную модель, которая поможет нам изучить процесс вывода нового продукта на рынок.

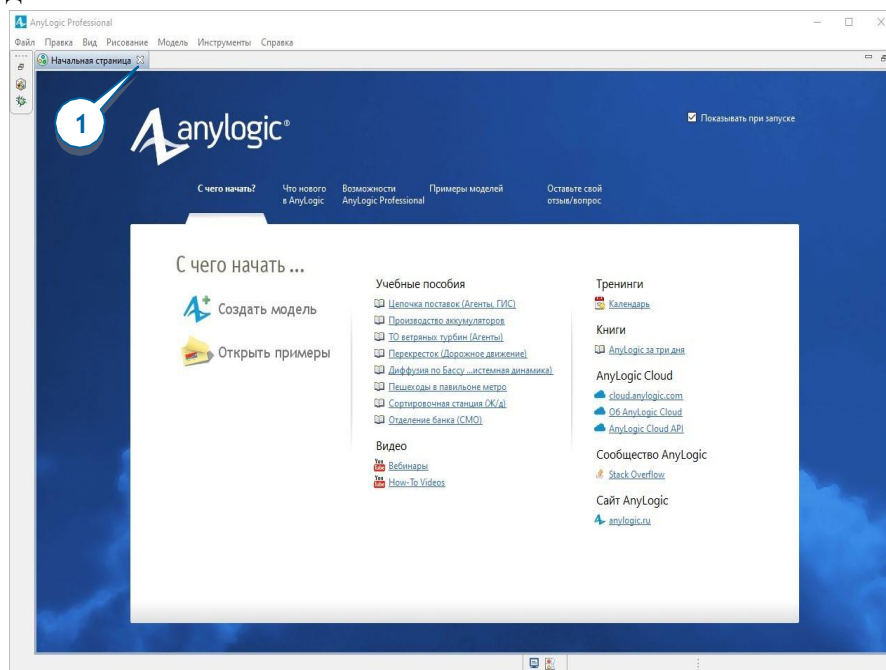
- Мы рассмотрим относительно небольшой потребительский рынок численностью в 5000 человек. С точки зрения реализации модели каждый потребитель будет являться агентом.
- Поскольку мы рассматриваем процесс вывода на рынок нового продукта, то изначально никто этим продуктом не пользуется.

- Люди начнут покупать продукт под влиянием рекламы.
- После этого начального этапа куда более сильное влияние на продажи будет оказывать общение людей друг с другом, рекомендации и положительные отзывы потребителей продукта, побуждающие других на его приобретение.

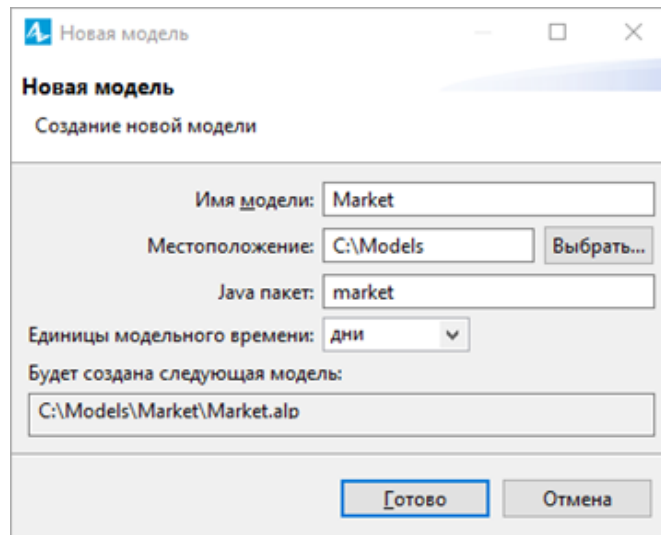
1. Создание популяции агентов

Начнем с создания простой модели, которая продемонстрирует влияние рекламы на начальные продажи продукта. Люди в нашей модели поначалу не будут пользоваться продуктом, но потенциально могут быть в нем заинтересованы. Для начала мы создадим популяцию агентов, а потом зададим то, как люди приобретают товар под влиянием рекламы.

Запустите AnyLogic. Откроется *Начальная страница*. *Начальная страница* предлагает обзор программы AnyLogic и ее функционала, а также позволяет открывать различные примеры моделей.



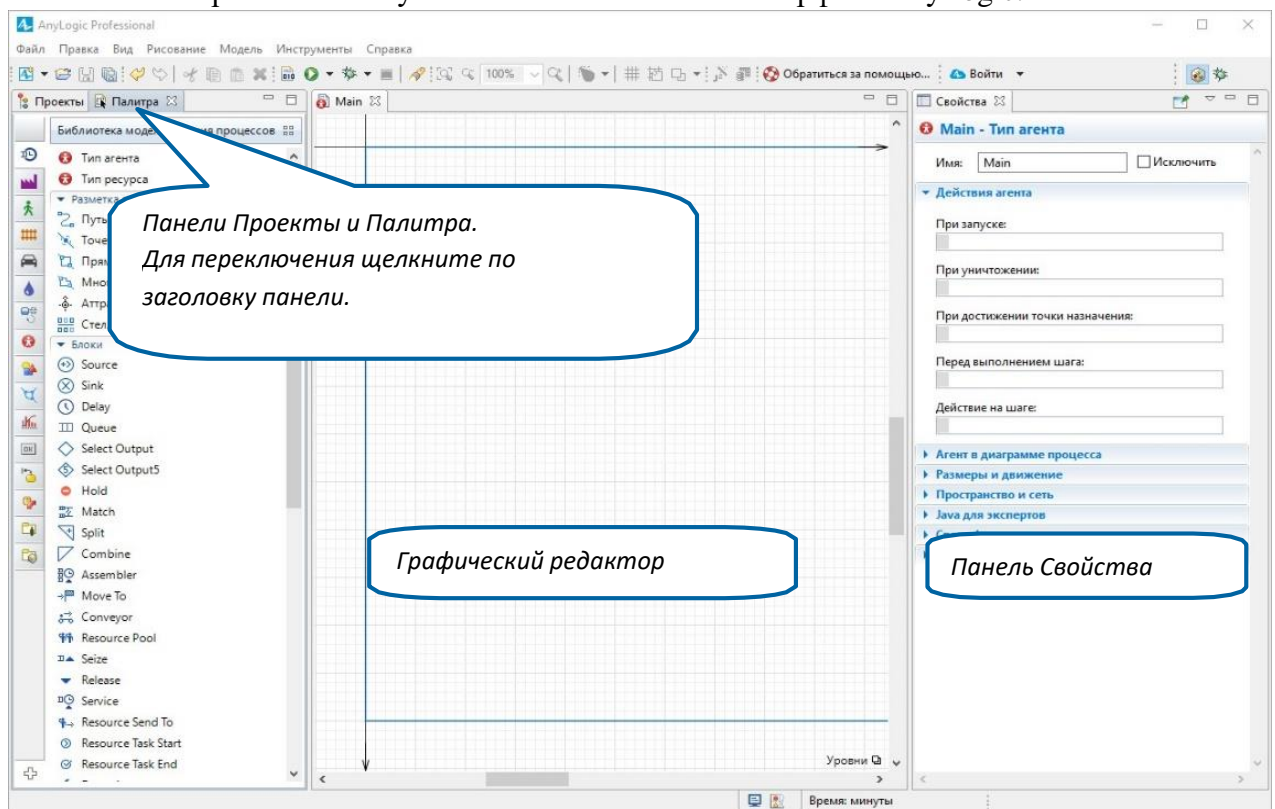
1. Закройте Начальную страницу и создайте новую модель. Для этого выберите **Файл > Создать > Модель** из главного меню AnyLogic. Откроется диалоговое окно **Новая модель**.



2. В поле **Имя модели** введите имя новой модели: Market.
3. В поле **Местоположение** выберите каталог, в котором вы хотите сохранить файлы модели.

4. Щелкните по кнопке **Готово**. Будет создана новая модель.

Самое время бегло изучить пользовательский интерфейс AnyLogic.



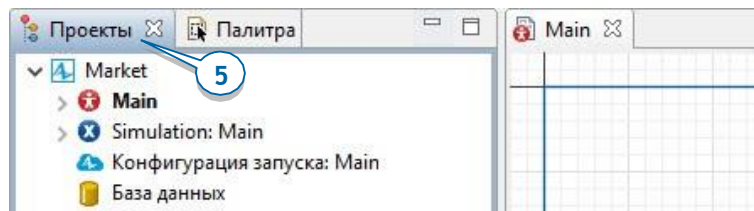
Рабочее пространство AnyLogic

- **Графический редактор** позволяет редактировать диаграмму агента. Вы можете добавлять элементы на диаграмму, перетаскивая их из **Палитры** на холст редактора. Синяя прямоугольная рамка ограничивает ту область холста, которая будет отображаться в окне модели при ее запуске.
- **Панель Проекты** отображает содержимое моделей AnyLogic, открытых в рабочем пространстве в текущий момент. Элементы каждой модели отображаются в виде

иерархического дерева, для облегчения навигации.

- Панель **Палитра** содержит все графические элементы AnyLogic, сгруппированные в отдельные палитры. Чтобы добавить тот или иной элемент в модель, перетащите соответствующий элемент из палитры в графический редактор.
- Панель **Свойства** позволяет вам просматривать и изменять свойства выделенных в текущий момент элементов модели.
- Чтобы открыть или закрыть панель, выберите в меню **Вид** соответствующий пункт с именем панели.
- Чтобы изменить размер панели, перетащите мышью ее границу.
- Вы всегда можете воспользоваться опцией **Восстановить расположение панелей** в меню **Инструменты**, чтобы вернуть расположение панелей по умолчанию.

5. Давайте откроем панель **Проекты**, чтобы изучить структуру новой модели. Панели **Палитра** и **Проекты** находятся в левой части рабочего пространства, и вы можете переключиться с панели **Палитра** на панель **Проекты**, щелкнув по заголовку панели.

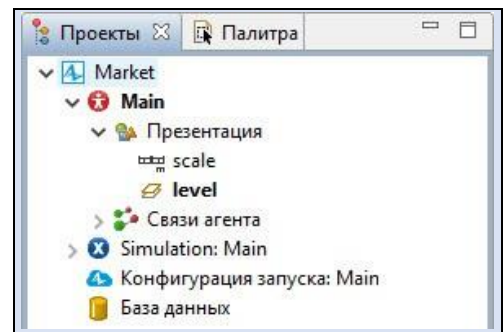


Навигация по элементам модели в панели **Проекты**

- Панель **Проекты** предоставляет простой и быстрый доступ к содержимому моделей, открытых в рабочем пространстве AnyLogic. AnyLogic отображает структуру каждой модели в виде иерархического дерева.
- По умолчанию в каждой модели создается один тип агента – **Main** – и один эксперимент **Simulation**.


хранящий настройки запуска этой модели. **Конфигурация запуска** позволяет настраивать входные и выходные параметры модели перед ее загрузкой в облако AnyLogic Cloud.

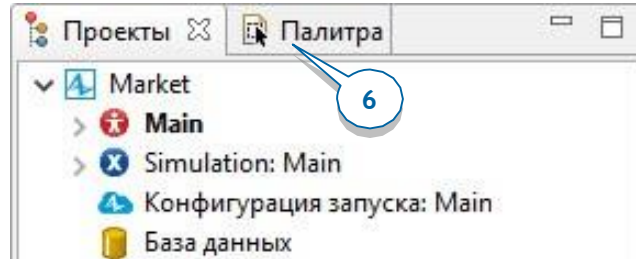
- Двойной щелчок по типу агента или эксперименту открывает его диаграмму в графическом редакторе.
- Также у каждой модели есть своя встроенная **База данных**. База данных изначально пуста, но при необходимости вы можете импортировать в нее данные из внешней базы данных (например, MS Excel), а также собрать информацию о выполнении модели в специальные журналы (для этого нужно выбрать в свойствах базы данных опцию **Записывать в журнал информацию о выполнении модели**).
- Щелчок по элементу модели в дереве выделяет этот элемент и показывает его в центре графического редактора. Если вы не можете найти какой-либо элемент на графической диаграмме, воспользуйтесь таким способом нахождения элемента.




Чтобы добавить в нашу модель потребителей, нам нужно создать новый тип агента (потребитель) и затем создать популяцию агентов, которая будет состоять из заданного количества агентов этого типа. Вы можете выполнить оба этих действия с помощью

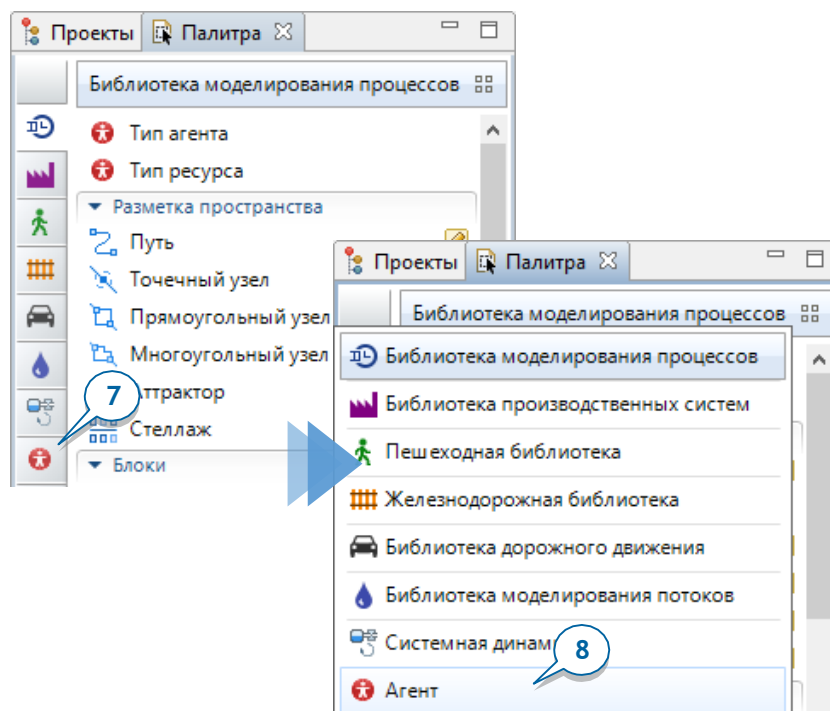
удобного мастера создания агентов.

6. Мы собираемся добавить новый элемент в нашу модель, поэтому давайте перейдем в панель Палитра , щелкнув по заголовку этой панели.




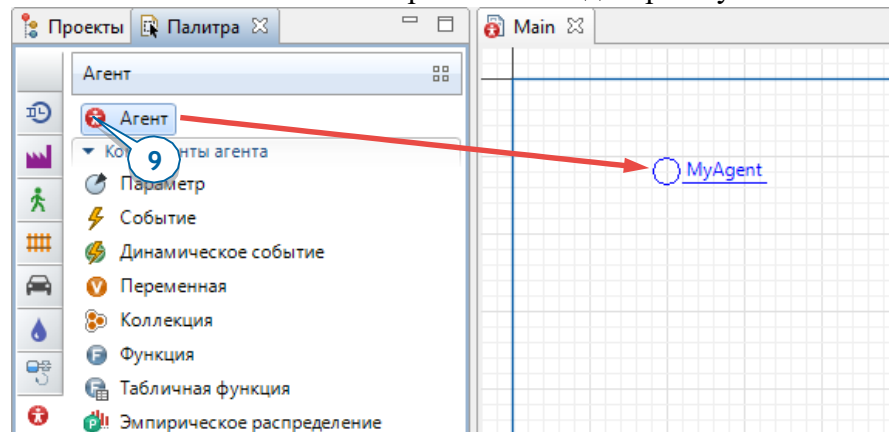
7. Откройте палитру Агент. Чтобы открыть другую палитру, перейдите в панель Палитра и наведите курсор мыши на вертикальную панель навигации.

8. Откроется список всех палитр, и вы сможете выбрать нужную вам палитру. Щелкните в списке по палитре Агент .

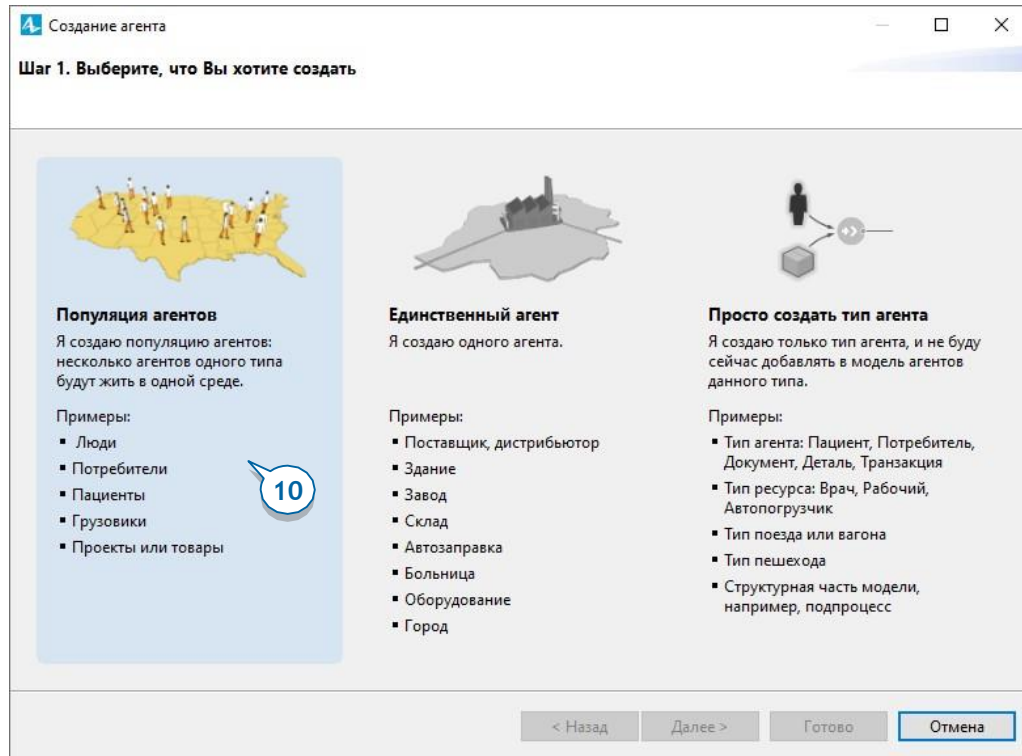


Запомнив значки палитр, вы сможете открывать палитры простым щелчком мыши по нужному значку. В графическом редакторе нашей модели сейчас отображается пустая диаграмма агента *Main*.

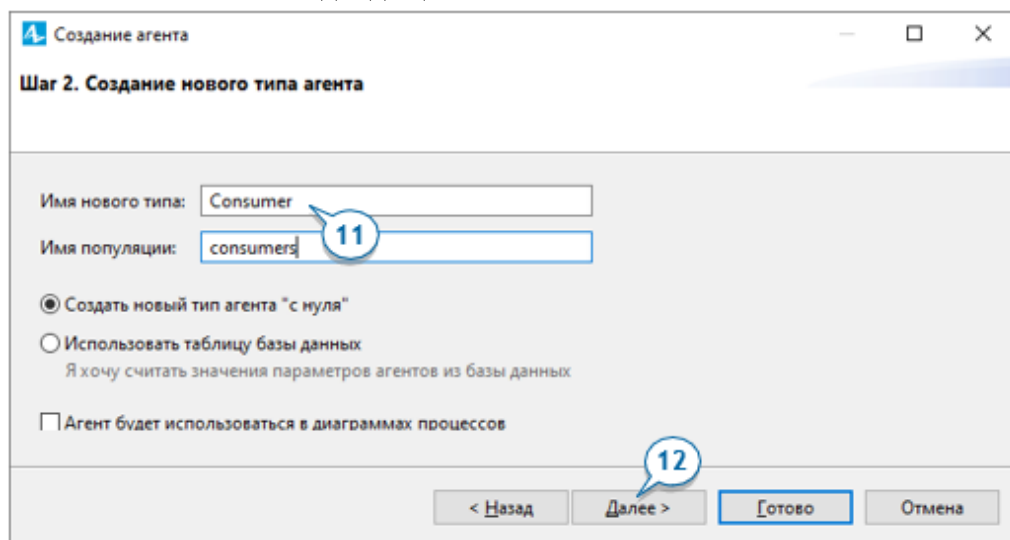
9. Перетащите элемент Агент  из палитры Агент на диаграмму *Main*.



10. Откроется мастер создания агентов **Новый агент**. Мы хотим создать большое количество агентов одного типа, поэтому на первой странице мастера выберите опцию **Популяция агентов**.

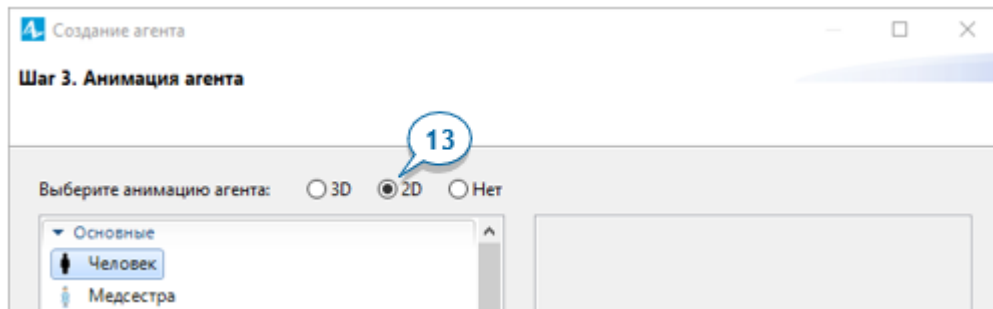


11. На странице мастера **Шаг 2. Создание нового типа агента**, в поле **Имя нового типа**, введите *Consumer*, то есть потребитель. Содержание поля **Имя популяции** автоматически изменится на подходящее *consumers*.



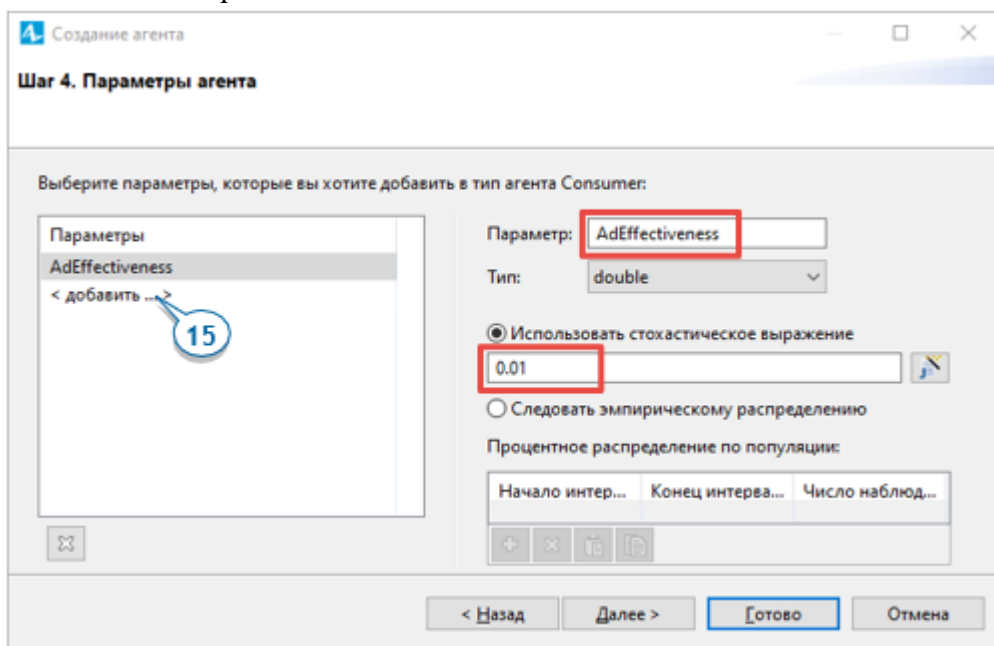
12. Щелкните по кнопке **Далее**.

13. На странице мастера **Шаг 3. Анимация агента** выберите фигуру анимации агента. Поскольку мы создаем простую модель с двумерной анимацией, выберите опцию **2D** и затем выберите первую фигуру (**Человек**) из расположенного ниже списка. Щелкните по кнопке **Далее**.



14. На следующей странице мастера можно задать параметры агента (обычно представляющие собой его статические характеристики).

Мы добавим параметр *AdEffectiveness* (эффективность рекламы), чтобы задать процентную долю потенциальных потребителей, которые захотят купить продукт в течение дня вследствие воздействия рекламы.

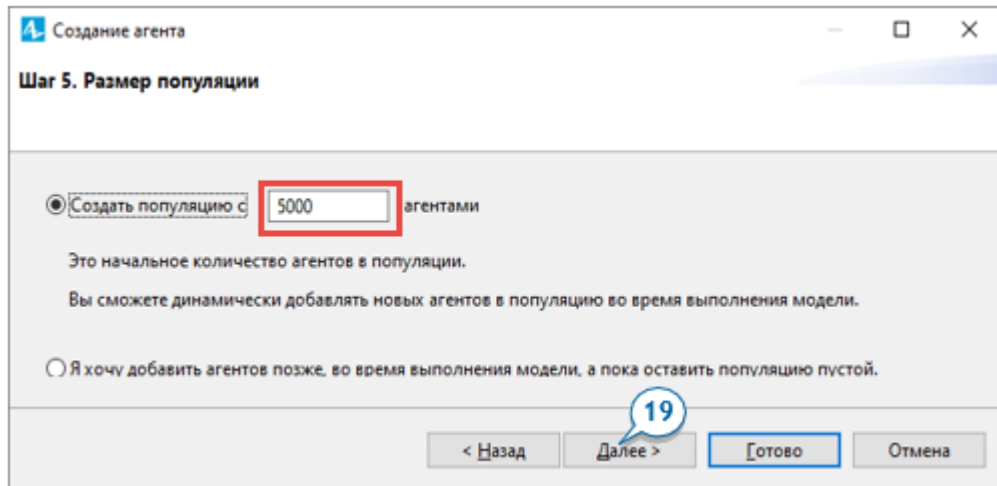


15. В секции слева вы увидите таблицу **Параметры**. Щелкните по строке **<добавить...>**, чтобы создать новый параметр.

16. Справа, в поле **Параметр**, измените заданное по умолчанию имя параметра на *AdEffectiveness*. Выберите в поле **Тип** опцию **double** (параметр будет принимать вещественные значения). Этот параметр задает эффективность рекламы. Мы предполагаем, что за день к решению о приобретении продукта приходит в среднем 1% потенциальных потребителей, поэтому мы задаем 0.01 в качестве значения данного параметра.

17. Щелкните по кнопке **Далее**.

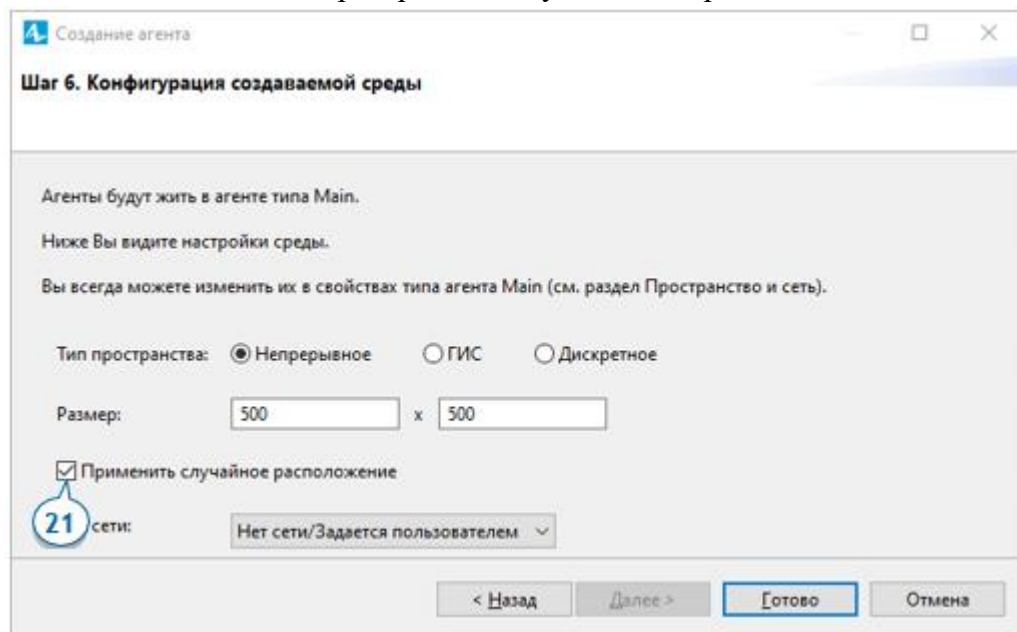
18. На следующей странице мастера, **Размер популяции**, в поле **Создать популяцию с ... агентами** введите значение 5000, чтобы создать 5000 агентов типа *Consumer*. Каждый агент, живущий в создаваемой нами популяции, будет моделировать отдельного агента-потребителя.




19. Щелкните по кнопке **Далее**.

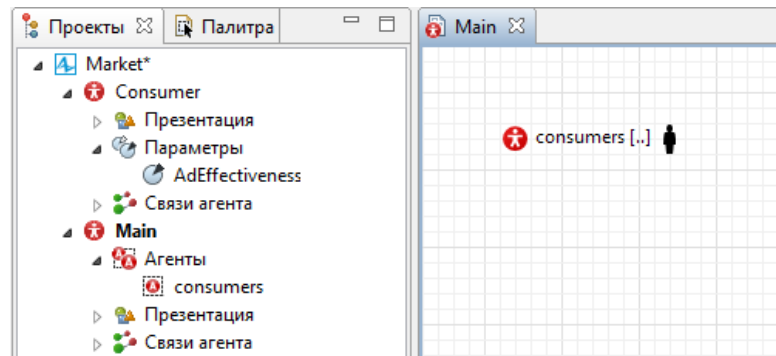
20. На странице мастера **Конфигурация создаваемой среды** оставьте выбранный по умолчанию тип пространства среды (**Непрерывное**) и значения его размерностей **Ширина** и **Высота** (500). Тогда при запуске модели AnyLogic отобразит агентов внутри прямоугольного пространства размером 500x500 пикселей.

21. Выберите опцию **Применить случайное расположение**, чтобы расположить агентов в заданном нами выше пространстве случайным образом.



22. Щелкните по кнопке **Готово**.

23. Давайте откроем панель **Проекты**  и посмотрим, какие именно новые элементы были созданы мастером. Разверните ветви дерева нашей модели, чтобы посмотреть на их содержимое.



В нашей модели теперь два типа агентов: *Main* и *Consumer*.

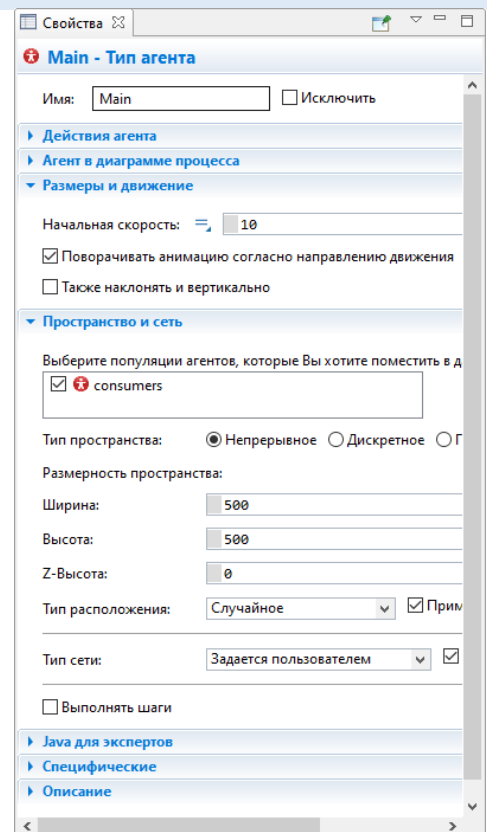
- Тип агента *Consumer* содержит фигуру анимации агента (*person* в ветке **Презентация**) и параметр *AdEffectiveness*.
- Тип агента *Main* содержит популяцию агентов, которая называется *consumers* (набор из 5000 агентов типа *Consumer*).

Среда обитания агентов

Типичная архитектура агентной модели в AnyLogic - агент *Main*, на диаграмме которого заданы популяции агентов других типов. В этом случае агент *Main* играет роль *среды обитания* для других агентов. Среда задает тип пространства, в котором живут агенты, расположение агентов в пространстве и сеть контактов агентов, которая может использоваться при их общении друг с другом.


24. В панели **Проекты**, щелкните по элементу *Main*. Тем самым, вы откроете свойства этого элемента в панели **Свойства** (эта панель находится в правой части окна AnyLogic).

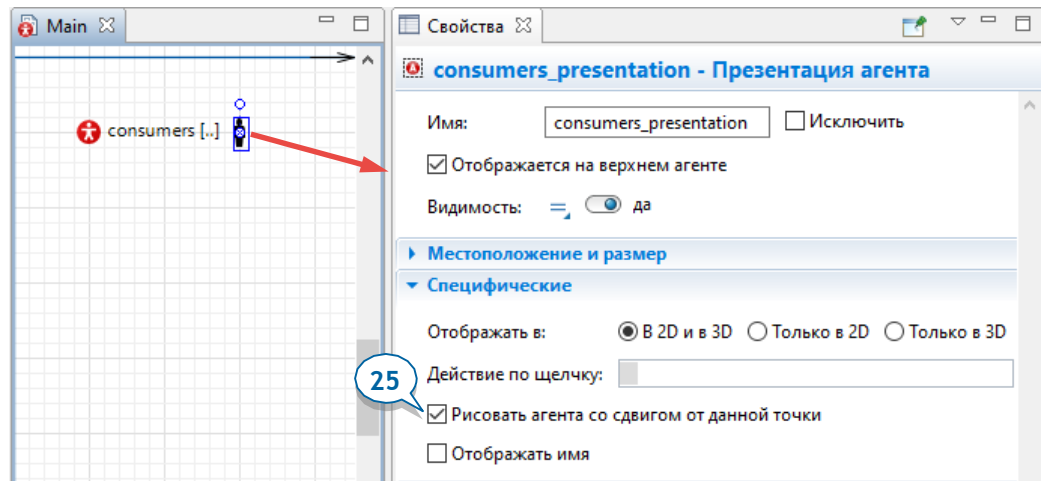
Вы можете изменить настройки среды обитания популяции агентов *consumers* в секции свойств агента *Main* **Пространство и сеть**.



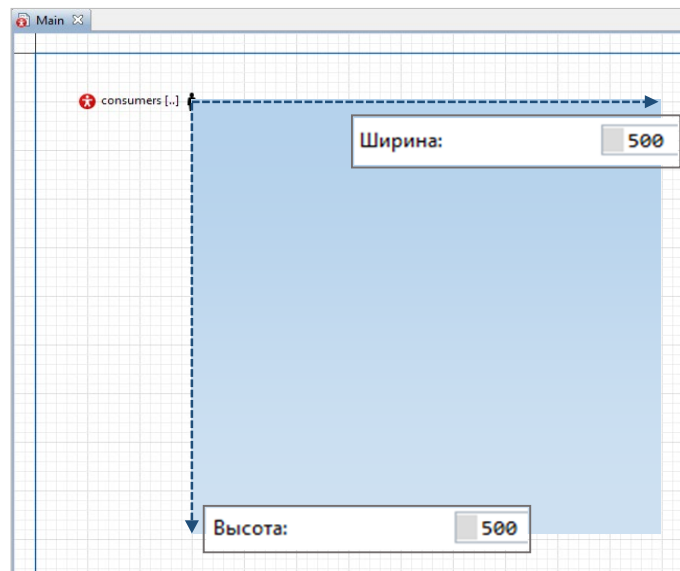
Панель Свойства

- Панель **Свойства** является контекстно-зависимой и отображает свойства выделенного в данный момент элемента.
- Чтобы изменить какое-либо свойство элемента, сначала выделите элемент щелчком в графическом редакторе или в панели **Проекты** и затем перейдите в панель **Свойства**.
- Панель **Свойства** может содержать секции. Чтобы раскрыть или свернуть секцию, щелкните по ее заголовку.
- Имя и тип выделенного элемента отображаются в шапке панели.


25. Выделите фигуру анимации популяции агентов , расположенную на диаграмме *Main*, откройте секцию ее свойств **Специфические** и выберите опцию **Рисовать агента со сдвигом от данной точки**.




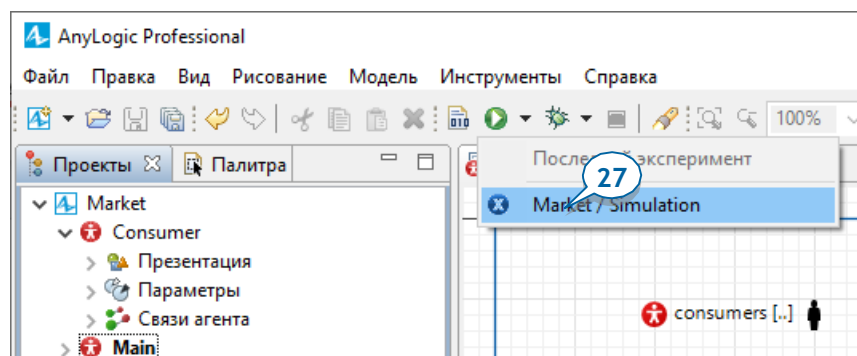
Следующий рисунок показывает, как фигура анимации популяции задает верхний левый угол прямоугольника (в нашем случае - размером 500x500 пикселей), внутри которого будут располагаться фигуры анимации агентов данной популяции во время выполнения модели.



Мы закончили создание простейшей агентной модели, и теперь мы можем запустить ее и понаблюдать за поведением агентов.

26. Щелкните по кнопке панели инструментов **Построить модель** , чтобы скомпилировать нашу модель и проверить ее на наличие ошибок компиляции.

27. Щелкните по маленькому треугольнику справа от кнопки панели инструментов **Запустить** . Выберите из раскрывшегося списка тот эксперимент, который вы хотите запустить. В нашем случае это **Market / Simulation**.



Так как в рабочем пространстве может быть открыто сразу несколько моделей, и у каждой модели может быть несколько экспериментов, а в этом списке находятся все эксперименты всех открытых моделей, то вам следует выбирать из списка именно нужный вам эксперимент.

После того, как вы запустите модель, появится окно модели. Вы увидите презентацию модели - 5000 фигур анимации агентов популяции *consumers*. Так как мы пока не задавали правила поведения агентов, на анимации больше ничего не происходит.

Управлять выполнением модели можно с помощью панели управления, расположенной в нижней части окна запущенной модели.

Управление выполнением модели

▶ Запустить

[Кнопка видна, если в данный момент модель не выполняется] Начинает выполнение. Если же моделирование было приостановлено, то продолжает его с текущего состояния.

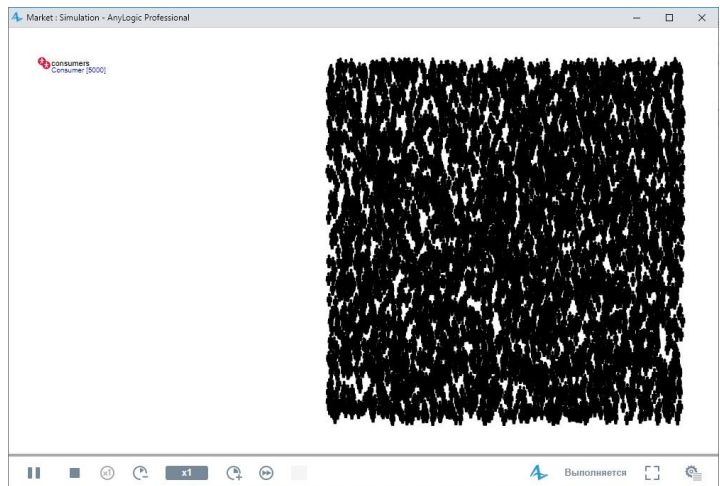
▢ Пауза

[Кнопка видна, если в данный момент модель выполняется] Приостанавливает выполнение

модели. Вы можете продолжить ее выполнение в любой момент времени.

■ Прекратить выполнение эксперимента

Прекращает выполнение модели.



Чтобы убедиться, что модель запущена, проверьте статус текущего прогона модели (**Выполняется**, **Пауза**, **Готов** или **Завершен**), отображаемый в панели разработчика окна модели.

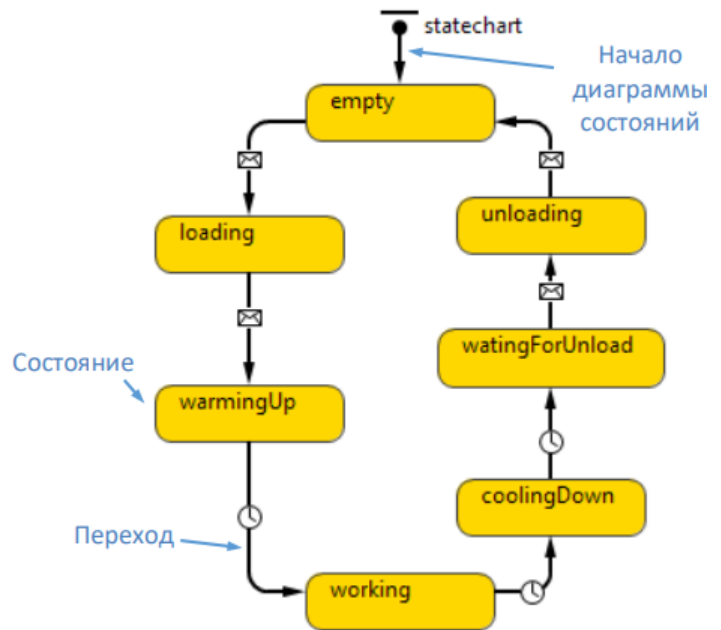
28. Мы готовы продолжить разработку модели. Закройте окно модели, щелкнув по кнопке, расположенной в правом верхнем углу окна.

2. Задание поведения потребителей

Теперь давайте зададим поведение потребителей. Лучше всего задавать поведение агента с помощью диаграммы состояний.

Диаграммы состояний

- *Диаграммы состояний* (карты состояний, стейтчарты) являются самым удобным средством задания поведения агента. Диаграммы состояний содержат *состояния* и *переходы*. Состояния диаграммы являются взаимоисключающими, то есть в каждый момент времени агент может находиться только в одном состоянии. Срабатывание перехода приводит к смене состояния и активации новых переходов. Допускается создание иерархических состояний, которые содержат внутри себя другие состояния и переходы.



- У одного агента может быть сразу несколько диаграмм состояний, каждая из которых описывает независимые аспекты поведения агента.

Мы зададим поведение агента-потребителя как два последовательных состояния:

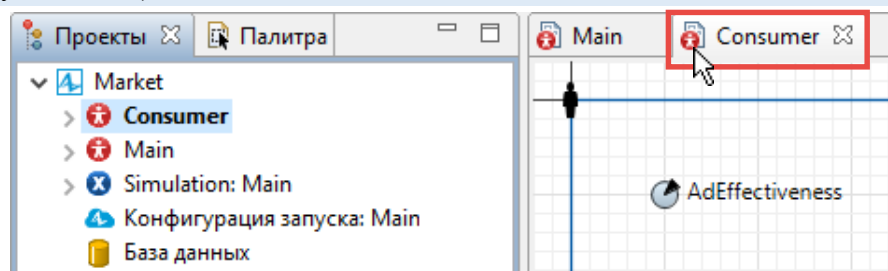
- *PotentialUser* - находящийся в данном состоянии агент является потенциальным покупателем и может быть заинтересован в покупке.
- *User* - потребитель, находящийся в этом состоянии, уже купил продукт.

1. Откройте диаграмму агента-потребителя Consumer, дважды щелкнув по этому типу агента в панели **Проекты**. Вы увидите графическую диаграмму этого агента с фигурой анимации, расположенной на пересечении осей координат, и одним параметром.

Как узнать, какой тип агента вы редактируете?


Так как в нашей модели теперь два типа агента, то и наши инструкции будут относиться теперь то к диаграмме одного, то к диаграмме другого агента. Поэтому давайте объясним, как понять, диаграмма какого именно агента открыта в вашем графическом редакторе в текущий момент,

- AnyLogic выделяет активную вкладку графического редактора синим цветом, а также выделяет в дереве модели в панели **Проекты** тот тип агента, диаграмма которого активна в текущий момент.
- Вы можете переключаться между диаграммами разных типов агентов щелчком по заголовку их вкладки в графическом редакторе (например, по заголовку вкладки *Consumer*, как на рисунке ниже):

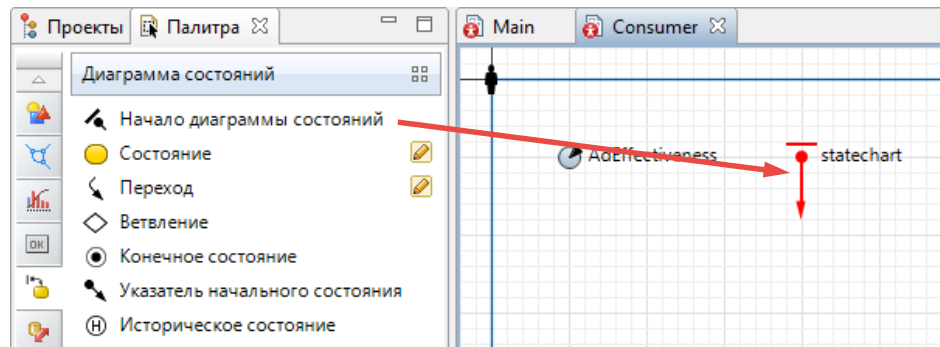





2. Начнем рисовать диаграмму состояний потребителя с добавления двух состояний.

Откройте палитру **Диаграмма состояний** .

3. Перетащите элемент **Начало диаграммы состояний**  из палитры **Диаграмма состояний** на диаграмму *Consumer*. Рисование диаграммы состояний всегда начинается с добавления *начала диаграммы состояний*.

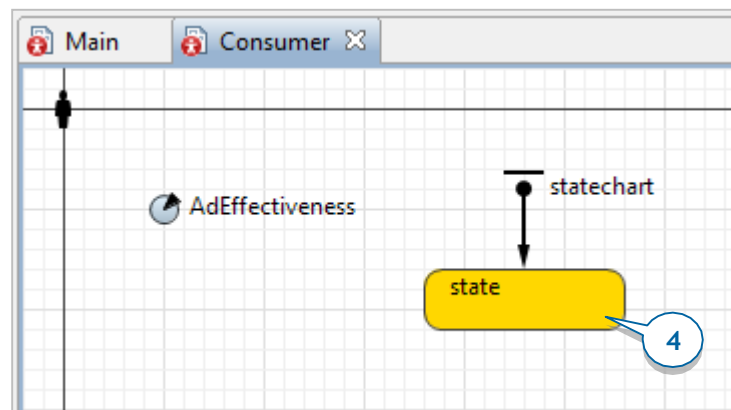
Этот элемент определяет точку, инициирующую управление диаграммой, а также задает имя этой диаграммы состояний.



✓ **Пожалуйста, будьте внимательны – Начало диаграммы состояний  легко спутать с Указателем начального состояния  или Переходом , поскольку их значки похожи.**

Вы увидите, что AnyLogic отображает начало диаграммы состояний красным цветом. Это индикация того, что данный элемент не соединен ни с одним состоянием, и поэтому текущая диаграмма состояний считается незаконченной и неверно заданной. Давайте добавим первое состояние в диаграмму состояний потребителя.

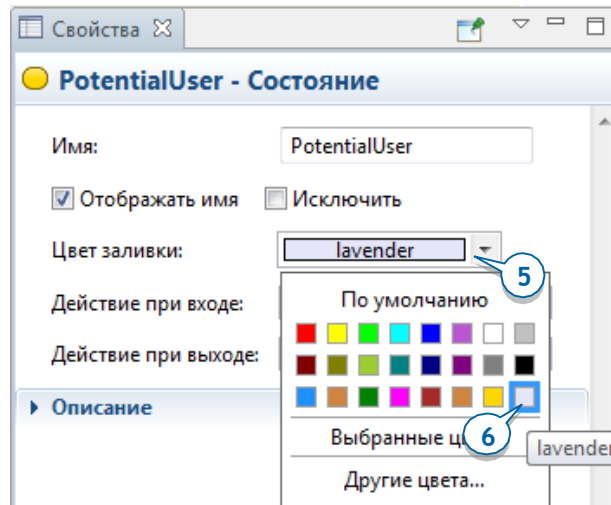
4. Перетащите элемент **Состояние**  из палитры **Диаграмма состояний** в графический редактор и соедините его с началом диаграммы.



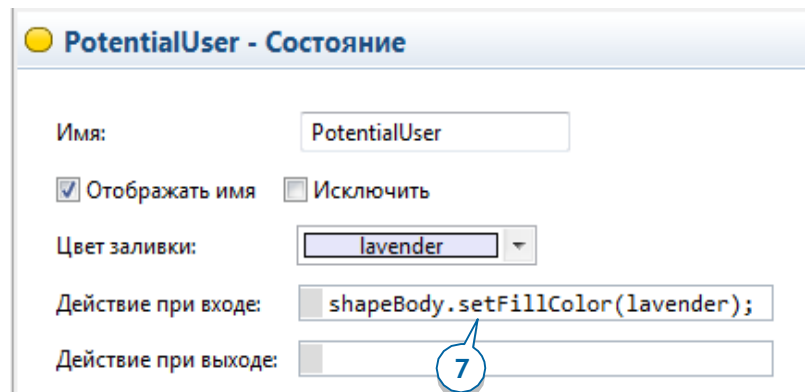
✓ **Убедитесь, что вы рисуете диаграмму состояний на диаграмме агента *Consumer*, а не на диаграмме *Main*.**

5. Выделите состояние в графическом редакторе и измените его свойства. Назовите состояние *PotentialUser*.

6. Щелкните в поле элемента управления **Цвет заливки** и измените цвет заливки состояния на *lavender*.

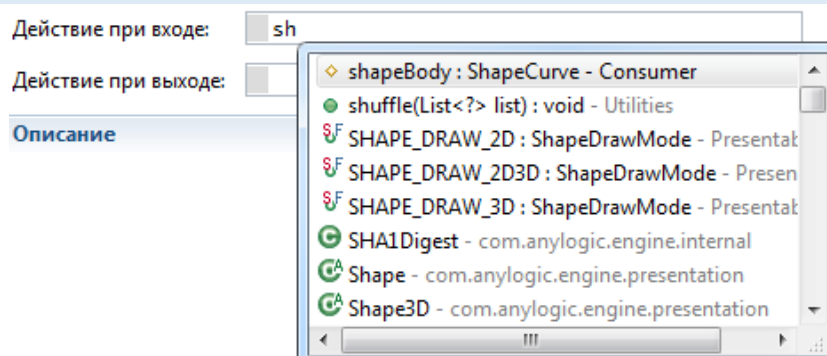


7. Введите следующий Java код в поле состояния **Действие при входе**: `shapeBody.setFillColor(lavender);`



Мастер подстановки кода

- Чтобы не печатать полностью имена элементов и функций, используйте помощник подстановки кода. Чтобы открыть помощник, щелкните в поле редактирования и нажмите на клавиатуре Ctrl+пробел (Alt+пробел на Mac OS). Во всплывающем окне будут перечислены все элементы модели, доступные в текущем контексте, такие как переменные, параметры и функции.
- Прокрутите до имени нужного элемента или введите первые буквы его имени, пока он не будет выделен в списке, затем нажмите Enter, чтобы вставить имя элемента в поле редактирования.



Код, заданный в поле **Действие при входе**, будет выполнен в момент перехода потребителя

в это состояние стейтчарта. Вызываемая здесь функция поможет нам понять, что произошла смена состояния потребителя, изменив цвет его фигуры анимации.

Используемое здесь *shapeBody* является именем фигуры анимации потребителя, которую мы выбрали в мастере создания агентов. (Если вы раскроете ветку **Презентация** типа агента *Consumer* в панели **Проекты**, то вы увидите фигуру *shapeBody* внутри группы *person*).

Далее мы вызываем функцию фигуры *shapeBody*. Чтобы получить доступ к функции элемента, введите имя элемента (*shapeBody*), поставьте точку, затем вызовите мастер подстановки кода, чтобы просмотреть все функции элемента и выбрать функцию из списка. Функция *setFillColor()* является стандартной функцией фигуры, которая позволяет динамически изменять цвет заливки фигуры. У этой функции только один аргумент – новый цвет. Мы передаем в качестве значения имя специальной константы AnyLogic, задающей лавандовый цвет (*lavender*). Примеры других цветовых констант - *red, yellow, green, purple, silver, black* и т.д.

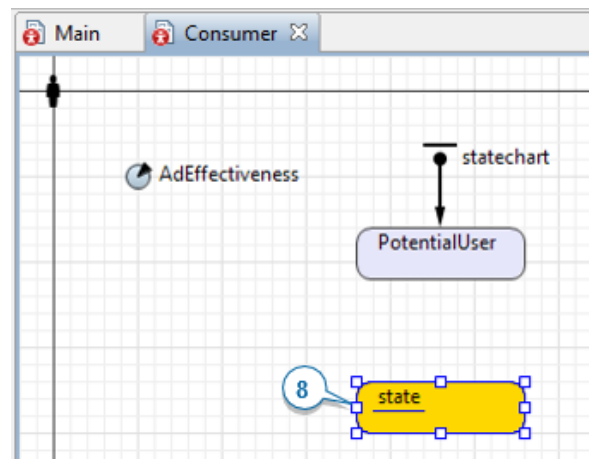
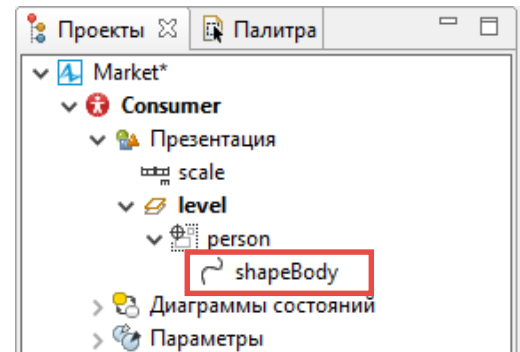
8. Добавьте еще одно состояние в диаграмму состояний потребителя:

9. Измените свойства этого состояния:

Имя: *User*

Цвет заливки: *yellowGreen*

Действие при входе: *shapeBody.setFillColor(yellowGreen);*



User - Состояние



Имя: ☒ Отображать имя

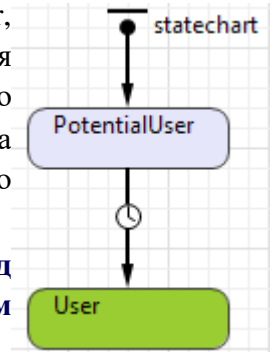
☐ Исклучить

Цвет заливки:

Действие при входе:

Действие при выходе:

10. Нарисуйте переход из состояния *PotentialUser* в состояние *User*, чтобы промоделировать, как человек приобретает продукт (и становится его потребителем). Для этого сделайте двойной щелчок мышью по элементу  **Переход** в палитре **Диаграмма состояний** (значок элемента при этом должен поменяться на ) , затем щелкните сначала по состоянию *PotentialUser*, а потом по состоянию *User*.

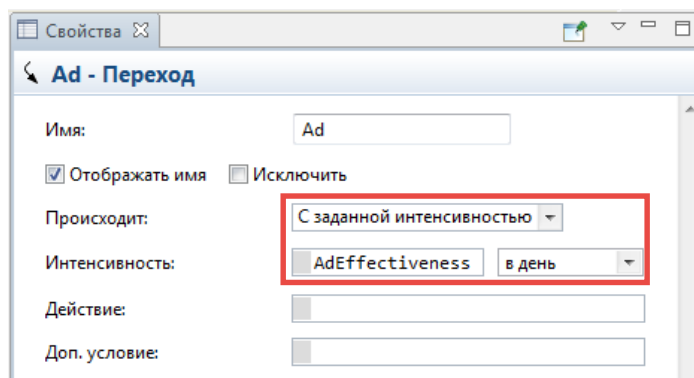




✓ **Убедитесь, что переход соединяет состояния. Если переход некорректно соединен, AnyLogic выделит его красным цветом.**

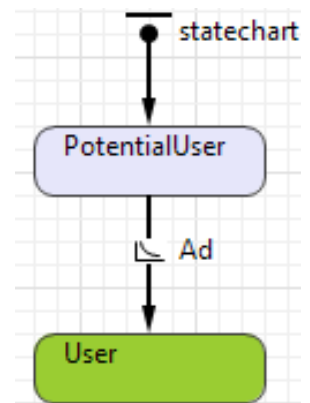
11. Назовите переход *Ad*, потому что он представляет действие рекламы, от английского слова «advertising».

12. Поставьте галочку в свойстве **Отображать имя**, чтобы имя перехода показывалось в графическом редакторе.

13. Переход, ведущий из состояния *PotentialUser* в состояние *User*, будет моделировать процесс покупки продукта под воздействием рекламы. В свойстве перехода **Происходит** выберите опцию **С заданной интенсивностью**. В появившемся поле **Интенсивность** введите имя переменной *AdEffectiveness*, а справа выберите единицы интенсивности срабатывания перехода - **в день**.








Заметьте, значок перехода изменится с  на . Этот значок отображает *тип срабатывания* перехода. Если вы захотите переместить имя перехода или его значок, то вначале выделите этот переход в редакторе, а затем перетащите имя/значок мышью в новое местоположение.



Тип срабатывания перехода

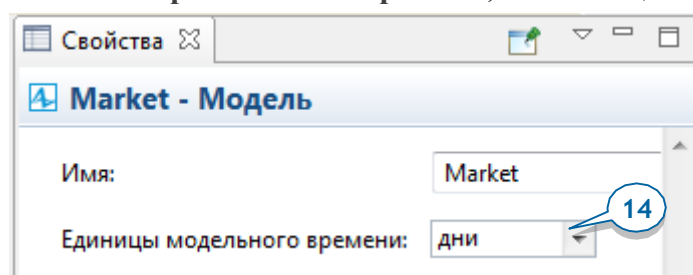
Переход из одного состояния в другое может быть вызван событиями различных типов. Приведенная ниже таблица перечисляет все возможные типы таких событий. Для каждого типа приведен специальный значок, по которому вы легко можете распознать тип перехода в диаграмме состояний.


Переход происходит:	Описание
По таймауту 	Срабатывает после того, как состояние диаграммы пробудет активным в течение заданного промежутка времени (таймаута) с момента перехода управления в это состояние. Таймаут может быть задан как определенным числом, так и стохастическим (случайным)

	выражением.
С заданной интенсивностью 	<p>Моделирует смену состояния через случайный промежуток времени, когда известно среднее время пребывания в состоянии. Аналогичен переходу по таймауту, у которого таймаут считается согласно экспоненциальному распределению, параметром которого является значение интенсивности. Если интенсивность = 0.2, то время пребывания в состоянии будет в среднем равно $1/0.2 = 5$ единиц модельного времени.</p>
При выполнении условия 	<p>Переход отслеживает выполнение заданного логического (булевского) условия и срабатывает, когда это условие будет выполнено. Условие может зависеть как от текущего состояния данного агента, так и от состояний других объектов моделируемой системы.</p> <p>Условие проверяется только в моменты происхождения событий в модели. Чтобы быть уверенными в том, что модель не пропустит момент срабатывания перехода, рекомендуется добавить на диаграмму этого агента циклически срабатывающее событие, и задать для этого события достаточно малый таймаут срабатывания.</p>
При получении сообщения 	<p>Реагирует на получение сообщений от других агентов. Сообщения чаще всего моделируют общение/взаимодействие агентов друг с другом. Вы можете задать в свойствах перехода шаблон сообщения, в этом случае переход будет срабатывать только при получении сообщений, удовлетворяющих заданному шаблону.</p>
По прибытию агента 	<p>Срабатывает по прибытии агента в заданное место назначения.</p> <p>Обратите внимание, что переход данного типа будет срабатывать только в том случае, если движение было начато вследствие вызова функции агента <code>moveTo()</code>.</p>

Наш переход происходит с заданной интенсивностью. В нашем случае, когда управление диаграммы состояний переходит в состояние *PotentialUser*, происходит вычисление времени срабатывания перехода согласно экспоненциальному распределению. Время до покупки продукта для каждого отдельного потребителя будет отличаться, но в среднем продукт будет приобретать 1% потенциальных потребителей в день.

14. Давайте теперь зададим единицы модельного времени. Чтобы изменить настройки модели, переключитесь из **Палитры** в панель **Проекты**, и затем щелкните по элементу



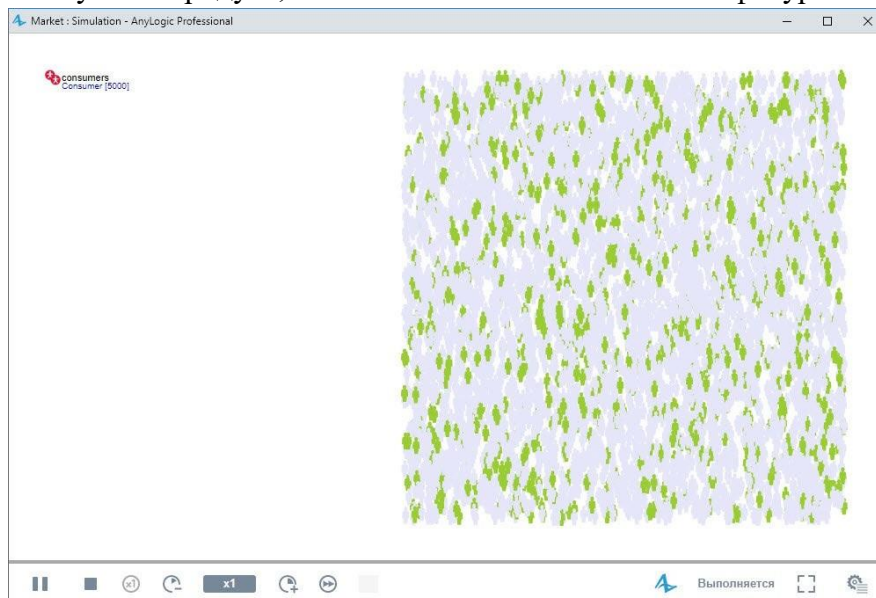
модели в дереве (самый верхний уровень дерева, элемент  Market). Перейдите в панель Свойства и выберите дни в качестве Единиц модельного времени.

Модельное время. Единицы модельного времени

- *Модельное время* – это виртуальное (моделируемое) время ("внутреннее" время "движка" AnyLogic). Модельное время никак не соотносится с реальным временем и часами на компьютере, хотя вы и можете выполнять модель с привязкой модельного времени к реальному.
- Чтобы определить соотношение модельного времени ко времени реального мира, в котором существует моделируемая система, вам нужно задать *единицы модельного времени*. Выберите наиболее подходящие единицы модельного времени в соответствии с длительностью типичных операций в вашей модели.
- К примеру, в пешеходных моделях, в качестве единиц модельного времени, как правило, используются секунды, в моделях производства и системах обслуживания – минуты, а в глобальных экономических, экологических или социальных моделях – месяцы или даже годы.

15. Запустите модель. Популяция агентов постепенно окрашивается в зеленый цвет (изменение, к которому приводит эффект рекламы), пока каждый потенциальный потребитель не купит продукт.

Когда в результате просмотра рекламы агент принимает решение о покупке продукта, его состояние *User* становится активным и выполняется **Действие при входе** этого состояния, меняющее цвет фигуры анимации этого агента на светло-зеленый (*yellowGreen*). Чем больше людей покупают продукт, тем больше появляется зеленых фигур анимации агентов.



Режимы выполнения модели

Вы можете выполнять модель AnyLogic в режиме *реального* или *виртуального времени*.

- В режиме *реального времени* вы устанавливаете соотношение между модельным и реальным временем, выбирая, сколько единиц модельного времени проходит за одну секунду астрономического времени. Как правило, режим реального времени используется тогда, когда вы хотите следить за анимацией моделируемых процессов.
- В режиме *виртуального времени* модель выполняется на максимальной скорости. Режим виртуального времени пригодится вам для того, чтобы моделировать длительный период

жизни модели. При этом соотношение между единицами модельного времени и секундами реального (астрономического) времени не задается, и анимация модели может отображаться "рывками".

Панель управления **Модельное время** позволяет вам настраивать скорость выполнения модели в режиме реального времени:



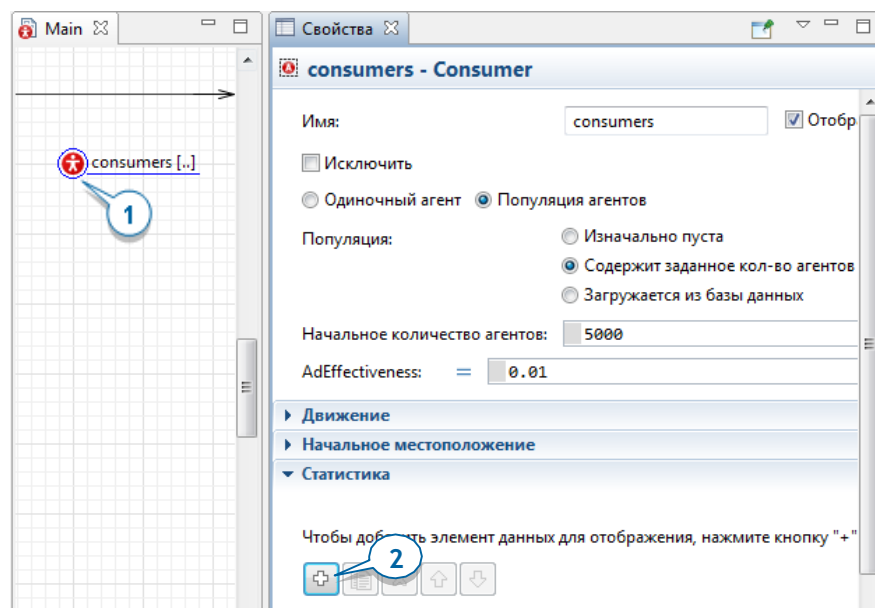
16. Чтобы изменить скорость выполнения модели, щелкните по кнопке **Замедлить** или **Ускорить** на панели управления окна запущенной модели. Увеличьте скорость выполнения модели в десять раз (выбрав значение $x10$), и вы увидите, что популяция будет быстрее окрашиваться в зеленый цвет.

3. Добавление графика для визуализации результатов моделирования

Мы хотим знать, сколько людей приобрело наш продукт в определенный момент времени. Для этого мы зададим функции, которые будут считать количество потребителей и потенциальных потребителей продукта соответственно, а затем добавим график, чтобы наблюдать за динамикой изменения рынка.

1. Сначала зададим функцию, которая будет считать количество потенциальных потребителей. Чтобы добавить новую функцию подсчета статистики по популяции агентов, откройте диаграмму агента *Main*, выделите популяцию агентов *consumers* и перейдите в раздел свойств **Статистика**.


2. Щелкните по кнопке **Добавить**.



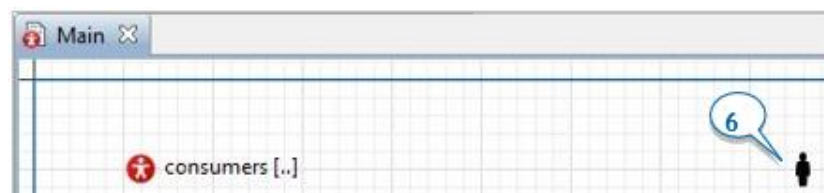
Мы хотим узнать, сколько агентов находятся в состоянии *PotentialUser*.

3. Задайте функцию типа **Кол-во**, в поле **Имя** введите *NPotential*. Функция статистики типа **количество** проходит по всем агентам популяции и подсчитывает тех агентов, для которых выполняется заданное условие.

4. Введите *item.inState(Consumer.PotentialUser)* в качестве **Условия** функции.
- *item* - локальная **переменная**, предоставляющая доступ к агенту, проверяемому в данный момент в процессе итерирования по популяции.
 - Функция *inState()* проверяет, является ли для этого агента активным указанное состояние диаграммы состояний.
 - *PotentialUser* – имя состояния. Поскольку оно имеет смысл для агента определенного типа, мы добавляем к имени префикс соответствующего типа агента - *Consumer*.

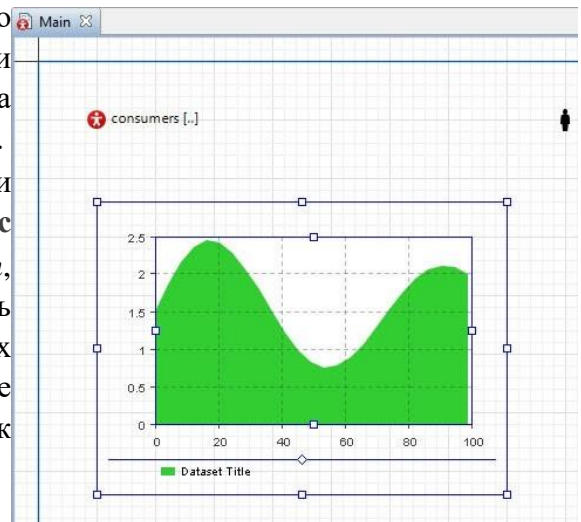
5. Задайте вторую функцию статистики для подсчета потребителей продукта. Назовите ее *NUser*. Пусть она считает количество агентов, для которых выполняется **Условие** *item.inState(Consumer.User)*. Вы можете создать копию ранее созданной функции сбора статистики, щелкнув по кнопке  **Дублировать** и изменив **Имя** и **Условие** созданной функции.

6. На диаграмме *Main*, переместите вправо фигуру презентации популяции агентов *consumers*.



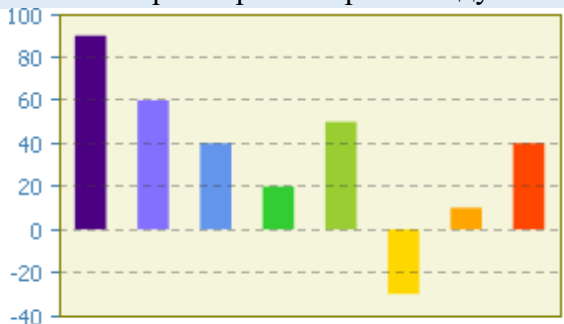
Теперь давайте добавим график для визуального отображения статистики, собираемой заданными только что функциями, и понаблюдаем за динамикой внедрения нового продукта на рынок.

7. Откройте палитру **Статистика** и перетащите элемент **Временная диаграмма с накоплением** из палитры на диаграмму *Main*, чтобы создать график, который будет отображать динамику изменений числа потенциальных потребителей и владельцев продукта. Увеличьте размер временной диаграммы с накоплением, как показано на рисунке сбоку:



Диаграммы

С помощью диаграмм вы можете визуализировать данные, полученные в результате прогона модели. Все диаграммы находятся в палитре **Статистика**, раздел **Диаграммы**. Ниже мы кратко рассмотрим каждую из них.



■ CA: 90 ■ TX: 60 ■ MA: 40 ■ CO: 20
 ■ NY: 50 ■ OK: -30 ■ NC: 10 ■ WA: 40

■ CA: 90 (30.0%)
 ■ TX: 60 (20.0%)
 ■ MA: 40 (13.3%)
 ■ CO: 20 (6.7%)
 ■ NY: 50 (16.7%)
 ■ NC: 40 (13.3%)

Столбиковая диаграмма

Отображает несколько элементов данных в виде столбцов, «растущих» в заданном направлении. Размеры столбцов пропорциональны значениям соответствующих элементов данных.

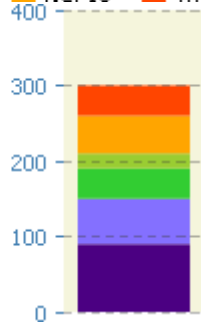


Диаграмма с накоплением

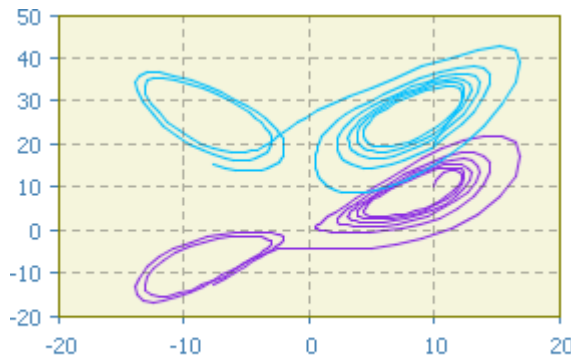
Показывает вклад нескольких элементов данных в суммирующий результат в виде столбцов, расположенных друг над другом. Высота каждого столбца пропорциональна значению соответствующего элемента данных.



Круговая диаграмма

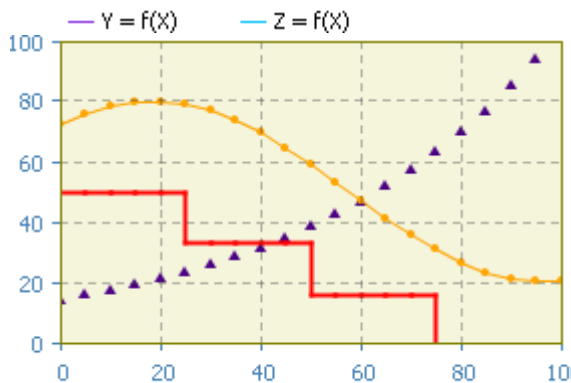
Отображает вклад нескольких элементов данных в общую составляющую в виде секторов круга. Дуги секторов пропорциональны значениям соответствующих элементов данных.

■ CA: 90 (29.0%)
 ■ TX: 60 (19.4%)
 ■ MA: 40 (12.9%)
 ■ CO: 20 (6.5%)
 ■ NY: 50 (16.1%)
 ■ NC: 10 (3.2%)
 ■ WA: 40 (12.9%)



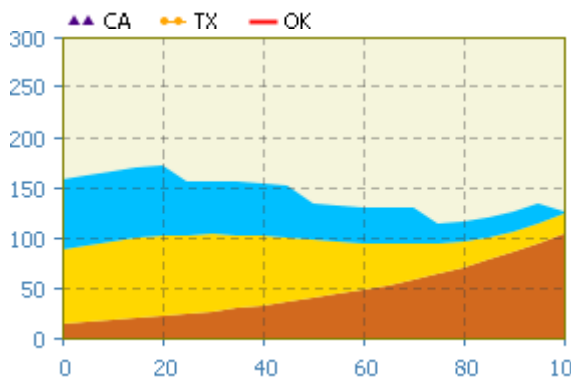
График

Обычно выполняет роль фазовой диаграммы. Отображает зависимость Y-компонент значений набора данных от соответствующих им X-компонент. Каждому измерению набора данных на графике соответствует точка с координатами $\langle x, y \rangle$.



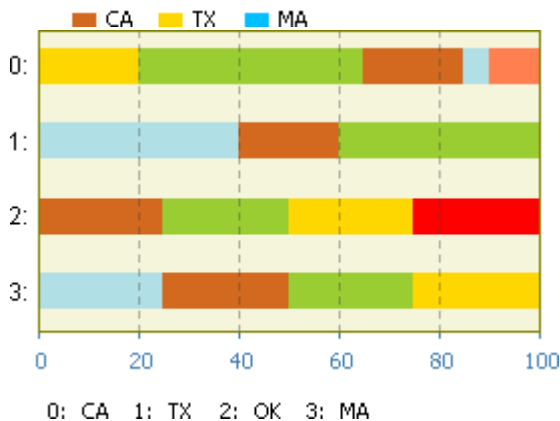
Временной график

Отображает динамику изменения (временной тренд) одного или нескольких наборов данных в течение последних N единиц модельного времени (заданного временного диапазона).



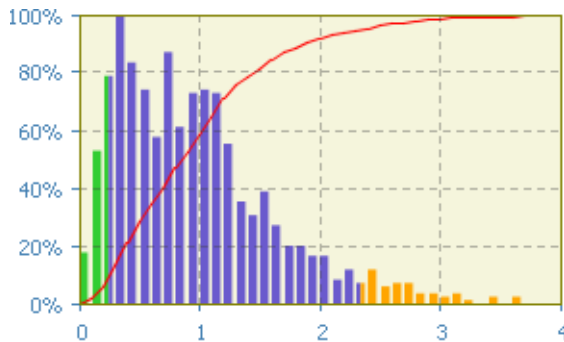
Временная диаграмма с накоплением

Отображает в виде располагающихся друг над другом областей историю вклада нескольких наборов данных в общую составляющую в течение заданного временного диапазона.



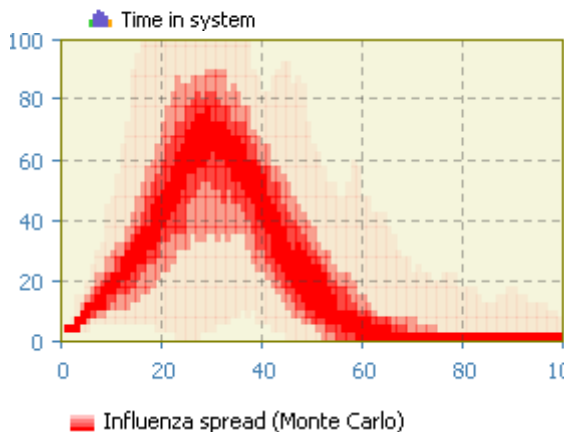
Временная цветовая диаграмма

Отображает динамику изменения данных в виде столбцов, каждый из которых составляется из полосок различного цвета (цвет зависит от того, какое из заданных условий для отображаемого данным столбцом набора данных верно в текущий момент).



Гистограмма

Отображает статистику, собранную элементом Данные гистограммы. Высота каждого прямоугольника пропорциональна числу элементов выборки, попавших в соответствующий интервал по оси X.



Гистограмма 2D

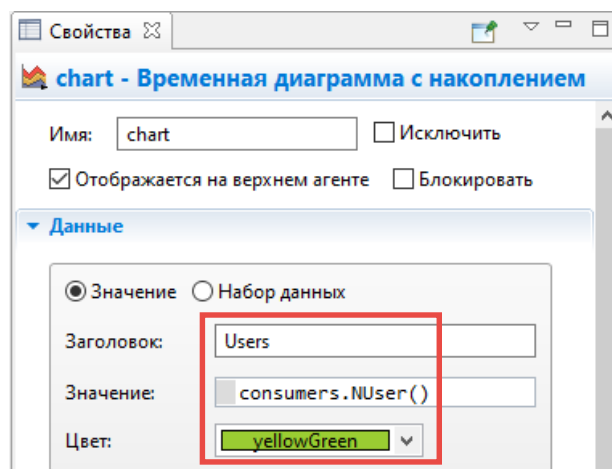
Отображает статистику, собранную элементом Данные двумерной гистограммы. Гистограмма отображается в виде закрашенных разным цветом прямоугольников (ячеек), отражающих значение плотности вероятности в соответствующей точке (X,Y), или вложений ("конвертов").


Укажите, какие данные будет отображать график. Мы воспользуемся теми самыми функциями статистики *NUser* и *NPotential*, которые мы создали ранее для популяции *consumers*.

8. В секции Данные свойств временной диаграммы с накоплением, измените свойства элемента данных следующим образом:

- **Заголовок:** *Users* – заголовок элемента данных.
- **Цвет:** *yellowGreen*
- **Значение:** *consumers.NUser()*

Здесь *consumers* - это имя нашей популяции агентов, а *NUser()* - это функция сбора статистики, которую мы задали ранее.



9. Добавьте еще один элемент данных, щелкнув по кнопке  **Добавить**. Задайте для этого элемента следующие свойства:

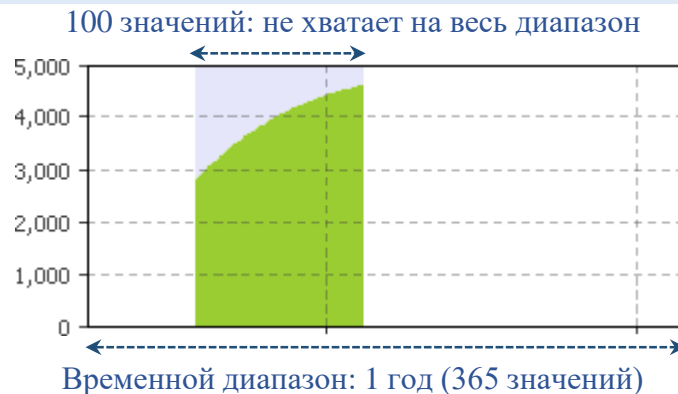
- **Заголовок:** *Potential users*
- **Цвет:** *lavender*

- **Значение:** `consumers.NPotential()`

☒ Значение ☐ Набор данных
 Заголовок:
 Значение:
 Цвет:

Подбор временного масштаба графика

- Диаграммы, используемые для отображения динамики изменения значений во времени (временной график, временная диаграмма с накоплением, временная цветовая диаграмма), позволяют вам настраивать временной диапазон - интервал времени, для которого диаграмма отображает значения.
- Если в процессе выполнения модели вы увидите, что ваш график напоминает приведенный на рисунке ниже, то вам будет нужно увеличить количество значений, отображаемых на графике или же уменьшить временной интервал графика.



Так как мы хотим видеть данные сразу за целый год, нам нужно изменить свойства графика.

10. Перейдите в раздел свойств **Масштаб** и задайте **Временной диапазон**, равный 1 году.

▼ Масштаб

Временной диапазон:

Вертикальная шкала: ☐ Авто ☒ Фиксированный ☐ 100%

От: До:

11. Теперь задайте диапазон значений для оси Y графика. Так как наш график будет показывать статистику для популяции *consumers*, а потребителей в нашей модели 5000, установите **Фиксированный** тип **Вертикальной шкалы** и введите 5000 в поле **До**.

▼ Масштаб

Временной диапазон:

Вертикальная шкала: ☐ Авто ☒ Фиксированный ☐ 100%

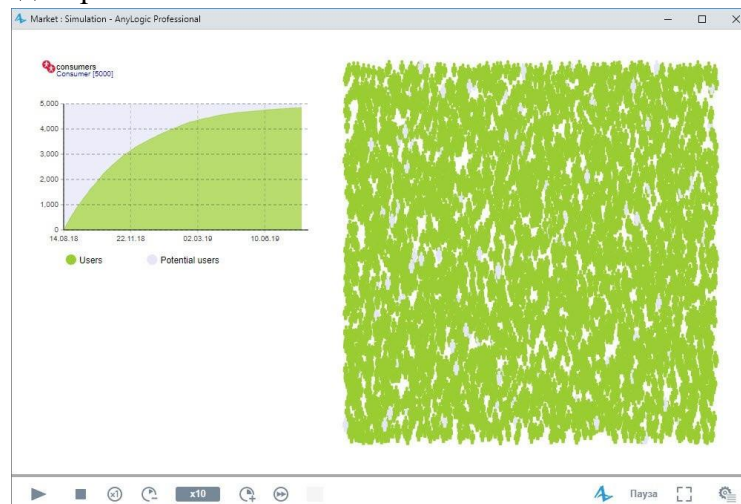
От: До:

12. Теперь, когда мы задали временной диапазон, давайте изменим максимальное

количество значений данных, отображаемых на графике. Перейдите в секцию **Обновление данных** и введите **365** в поле **Отображать до... последних значений**. Так как мы добавляем по одному значению каждый день, то это идеально подходит для временного диапазона длительностью в один год.

13. Перейдите в секцию свойств графика **Внешний вид** и выберите опцию **Модельная дата (только дата)** из списка **Формат временной оси**. Мы меняем формат меток по временной оси графика - теперь они будут отображать только дату, но без времени, тем самым метки станут более компактными, и внешний вид графика улучшится.

14. Запустите модель и наблюдайте за моделируемым процессом с помощью добавленной нами диаграммы.



4. Добавление эффекта рекомендаций

Теперь мы промоделируем эффект, который оказывают на потенциальных потребителей положительные отзывы о продукте его владельцев.

- В нашей модели каждый человек в течение дня будет в среднем общаться с одним своим знакомым.

- Во время общения друг с другом владельцы продукта могут повлиять на потенциальных потребителей. Мы зададим вероятность приобретения продукта потенциальным потребителем под воздействием общения с помощью параметра $AdoptionFraction = 0.01$.

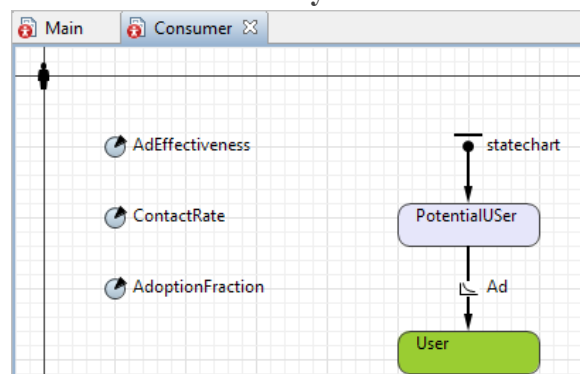
Для начала добавим два новых параметра: *ContactRate* (определяет интенсивность контактов) и *AdoptionFraction* (вероятность приобретения продукта в результате общения с пользователем этого продукта).

1. Откройте диаграмму типа агента *Consumer*, сделав двойной щелчок по элементу *Consumer* в панели **Проекты**.
2. Добавьте параметр, который будет задавать среднее количество контактов потребителя с другими людьми в течение дня. Перетащите элемент **Параметр** из палитры **Агент** на диаграмму агента *Consumer*.
3. Назовите параметр *ContactRate*.
4. В данной модели средняя интенсивность контактов равна одному контакту в день. Перейдите в свойства этого параметра и введите 1 в поле **Значение по умолчанию**.

5. Добавьте еще один параметр, *AdoptionFraction*, который задает вероятность приобретения продукта в результате общения с пользователем этого продукта. В свойствах данного параметра задайте **Значение по умолчанию**: 0.01.

Диаграмма типа агента *Consumer* теперь выглядит так:

Теперь промоделируем общение агентов между собой – те самые разговоры о продукте, побуждающие людей к его приобретению.



Взаимодействие агентов

Взаимодействие агентов в AnyLogic чаще всего реализуется с помощью *передачи сообщений*. Агент может посылать сообщения какому-то определенному агенту или группе агентов. Сообщение может представлять собой строку текста, число, и вообще любой объект Java, со своей структурой данных и множеством параметров.

Для отправки сообщения вызываются специальные функции агента (самые популярные из них приведены ниже):

sendToAll(msg) – агент отправляет сообщение *msg* всем другим агентам своей популяции.

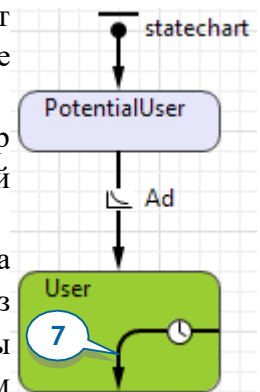
sendToRandom(msg) – агент отправляет сообщение *msg* одному случайно выбранному агенту из своей популяции.

send(msg, agent) – агент отправляет сообщение *msg* указанному агенту *agent* (вы передаете ссылку на агента-получателя с помощью второго аргумента функции)

В нашей модели сообщения будут посылать только те агенты, которые находятся в состоянии *User*. Лучшим способом задать действие, которое агент выполняет, не выходя из текущего состояния, является добавление *внутреннего перехода*.

6. Откройте диаграмму агента *Consumer* и измените размер состояния *User* (см. рисунок ниже), чтобы в него поместился внутренний переход, который мы сейчас добавим.

7. Нарисуйте переход внутри состояния *User*, как показано на рисунке ниже. Для этого перетащите элемент **Переход** из палитры **Диаграмма состояния** внутрь состояния, чтобы начальная точка перехода расположилась на границе состояния. Затем поместите на границу состояния и конечную точку этого перехода. Чтобы добавить изгиб фигуры перехода, сделайте по переходу двойной щелчок мышью.

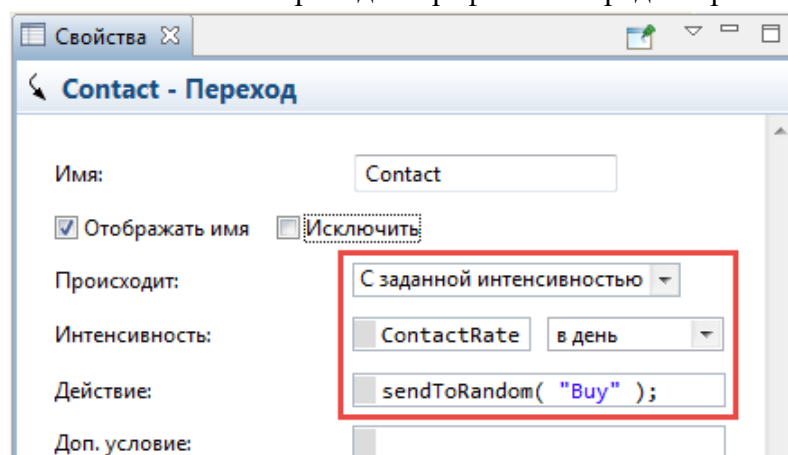


✓ **Внутренние и внешние переходы отличаются по своей логике, поэтому убедитесь, что этот переход лежит полностью внутри состояния.**

Внутренние переходы

- Внутренний переход располагается внутри состояния. Обе крайние точки такого перехода находятся на границе состояния.
- Так как внутренний переход не выходит за границы состояния, то он не выводит диаграмму из этого состояния. При срабатывании такого перехода не выполняются ни действие **При входе**, ни действие **При выходе** этого состояния.

8. Измените свойства перехода. Этот переход будет происходить **С заданной интенсивностью**, равной значению параметра *ContactRate* (чтобы не печатать имя параметра полностью, используйте мастер подстановки кода). Назовите переход *Contact* и включите отображение имени этого перехода в графическом редакторе.



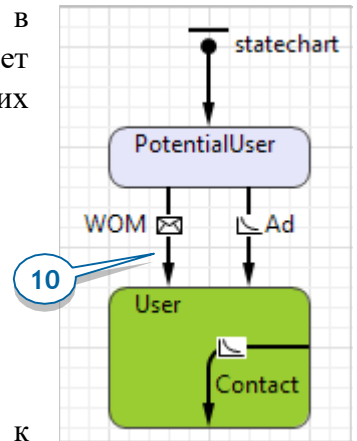
9. Укажите **Действие**, которое должно выполняться при срабатывании перехода: `sendToRandom("Buy");`

Мы создали циклический переход в состоянии *User*. Каждый раз, когда срабатывает этот переход, вызываемая функция `sendToRandom("Buy");` моделирует отправку этим потребителем сообщения "Buy" случайно выбранному агенту. Если агент, который получает сообщение, является потенциальным потребителем (то есть находится в состоянии *PotentialUser*), то текущим состоянием агента-получателя станет состояние *User* (согласно еще одному переходу, который мы нарисует сейчас).

10. Нарисуйте еще один переход из состояния *PotentialUser* в состояние *User* и назовите его *WOM*. Этот переход будет моделировать покупку продукта в результате рекомендаций других людей.

11. Измените свойства перехода:

- В списке **Происходит** выберите **При получении сообщения**.
- В свойстве **Осуществлять переход** выберите **При получении заданного сообщения**.
- В поле **Сообщение** ниже введите сообщение **"Buy"**
- Так как мы понимаем, что не каждый контакт приводит к новым продажам, то мы ограничим долю успешных контактов с помощью параметра *AdoptionFraction*. Задайте следующее **Доп. условие** перехода: `randomTrue(AdoptionFraction)`



Свойства

WOM - Переход

Имя: WOM

☒ Отображать имя ☐ Исключить

Происходит: При получении сообщения

Тип сообщения: Object

Осуществлять переход: ☐ Безусловно ☒ При получении заданного сообщения: ☐ Если выполняется условие

При получении сообщения: "Buy"

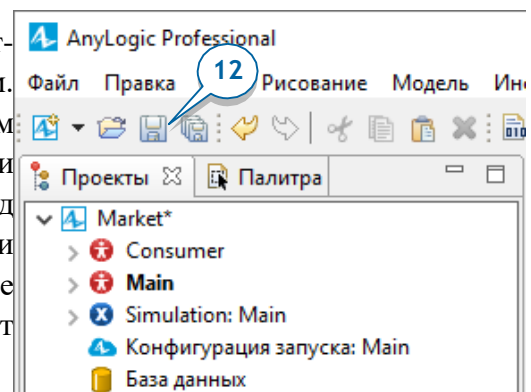
Действие:

Доп. условие: `randomTrue(AdoptionFraction)`

Дополнительные условия переходов

- При переходе в простое состояние инициируются все исходящие переходы этого состояния, и диаграмма состояний начинает ждать, когда один из них произойдет.
- При происхождении события, ведущего к срабатыванию перехода, также оценивается дополнительное условие этого перехода. Если это условие выполняется, то тогда переход срабатывает.

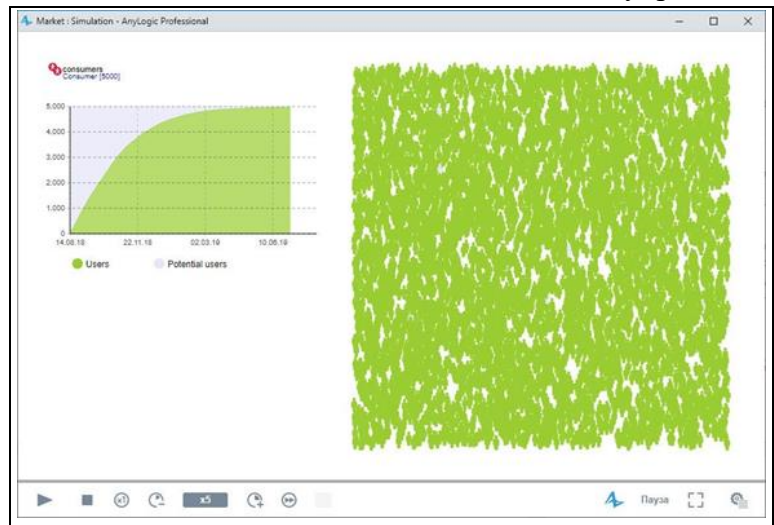
Итак, сейчас в нашей модели агент-потребитель отправляет сообщения другим агентам. Если агент-получатель является потенциальным потребителем (то есть, находится в состоянии *PotentialUser*), то это сообщение вызовет переход управления его диаграммы в состояние *User*. Если диаграмма состояний агента-получателя уже находится в состоянии *User*, то сообщение будет просто проигнорировано.



12. Перейдите в панель. **Проекты** Если вы видите звездочку у имени модели, значит, в вашей модели есть несохраненные изменения. Щелкните по кнопке панели управления **Сохранить модель**, чтобы сохранить изменения.

13. Запустите модель.

Насыщение рынка теперь будет происходить быстрее, а график покажет известную S-образную кривую выхода нового продукта на рынок.

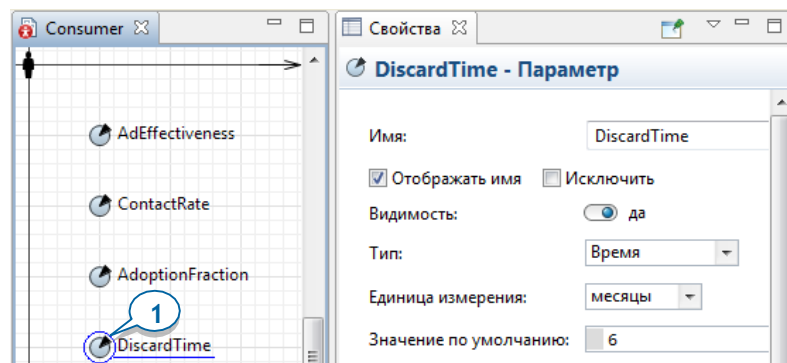


5. Учет повторных продаж продукта


Допустим, что у рассматриваемого нами

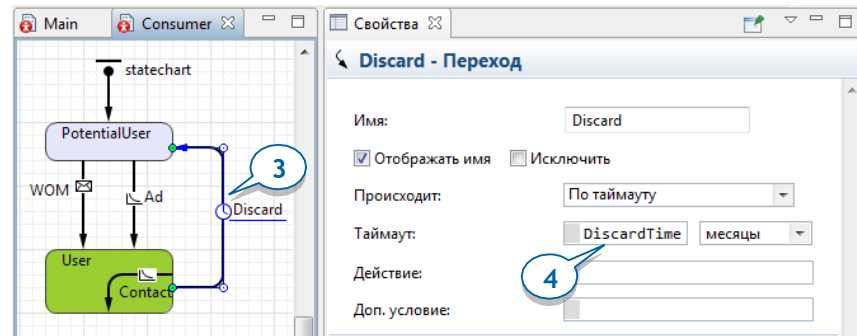
продукта ограниченный срок годности (или срок эксплуатации), равный шести месяцам. Когда потребитель больше не сможет пользоваться продуктом, ему понадобится замена продукта. Мы смоделируем повторные покупки, предположив, что по истечении срока годности товара потребители вновь становятся потенциальными потребителями (то есть, агенты переходят из состояния *User* обратно в состояние *PotentialUser*).

1. Откройте диаграмму агента *Consumer* и добавьте на нее параметр *DiscardTime*.



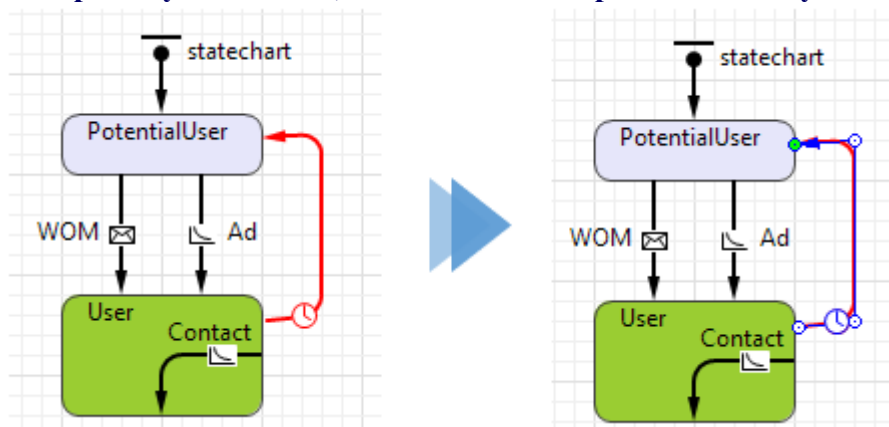
2. Этот параметр задает срок службы нашего продукта. Выберите **Время** в качестве **Типа** параметра, затем выберите **месяцы** в списке **Единица измерения** и введите **6** в поле **Значение по умолчанию**.

3. Нарисуйте переход из состояния *User* в состояние *PotentialUser*, чтобы промоделировать истечение срока службы товара. Мы хотим нарисовать переход сложной формы, как на рисунке ниже. Для этого сделайте двойной щелчок мыши по элементу **Переход** в палитре **Диаграмма состояний** (значок элемента при этом сменится на ). Чтобы начать рисование, щелкните по состоянию *User*, после чего щелкните по холсту диаграммы в тех точках, где вы хотите поместить точки изгиба перехода. Завершите рисование, сделав двойной щелчок мыши по конечному состоянию перехода *PotentialUser*.



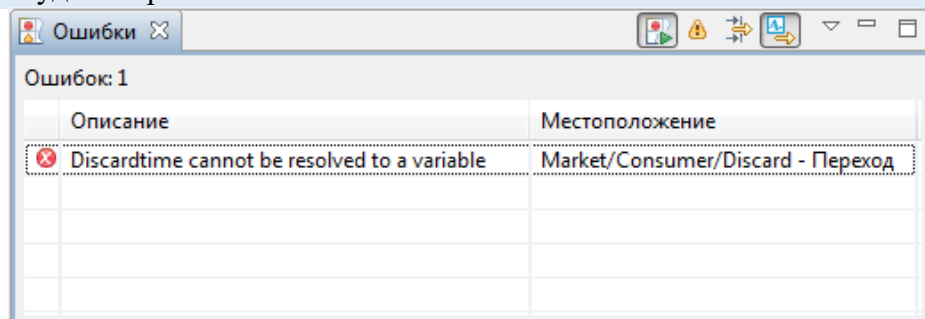
4. Назовите переход *Discard*, пусть он срабатывает по таймауту, равному значению параметра *DiscardTime*. В списке справа от значения, выберите **месяцы**.

- ✓ **AnyLogic выделяет красным цветом переходы, не соединенные с состоянием (как на рисунке слева). Поскольку визуально обе точки у перехода могут казаться соединенными с соответствующими состояниями, воспользуемся дополнительным инструментом диагностики. Чтобы понять, где именно допущена ошибка, выделите такой переход. Точки корректного соединения будут выделены зеленым цветом. Если AnyLogic не подсвечивает зеленым точку соединения перехода и состояния, вам нужно вручную подвинуть эту точку на границу состояния, и тем самым исправить ошибку.**



Исправление опечаток

Иногда при компиляции модели показывается сообщение о том, что была обнаружена ошибка - не удастся распознать то или иное имя:



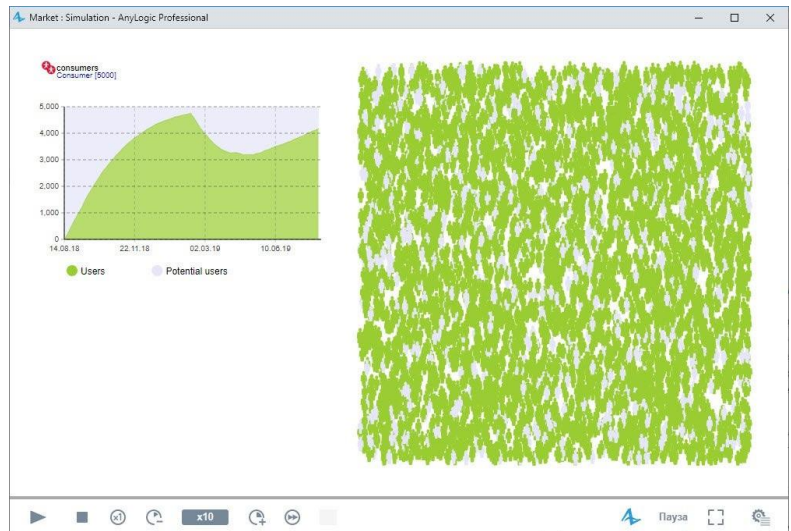
Чаще всего такая ошибка вызвана тем фактом, что AnyLogic учитывает регистр клавиатуры в именах элементов. В данном случае ошибка заключается в том, что буква *t*, с которой начинается слово в имени параметра *DiscardTime*, изначально в имени параметра - заглавная, а пользователь ссылается на это имя, используя прописную букву *t*, и поэтому имена не совпадают, и происходит ошибка компиляции модели.

Чтобы исправить ошибку, дважды щелкните по строке этой ошибки в панели **Ошибки**.

Если ошибка графическая, то AnyLogic подсветит вызвавший ошибку элемент в графическом редакторе. Если ошибка допущена при задании значения какого-то свойства элемента, то AnyLogic откроет свойства этого элемента и выделит предполагаемое место ошибки.

Итак, мы учли в модели ограничение срока службы товара, приводящее к повторным покупкам товара потребителями.

5. Запустите модель. Вы можете заметить изменение динамики продаж продукта: по прошествии определенного времени рост насыщения рынка сменится локальным спадом.

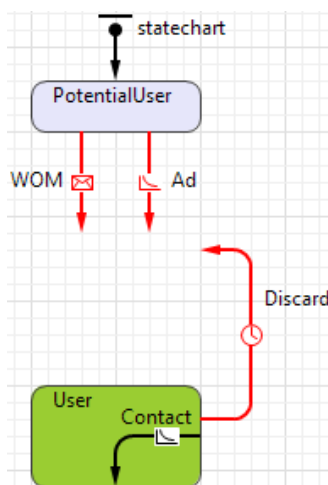


6. Учет времени

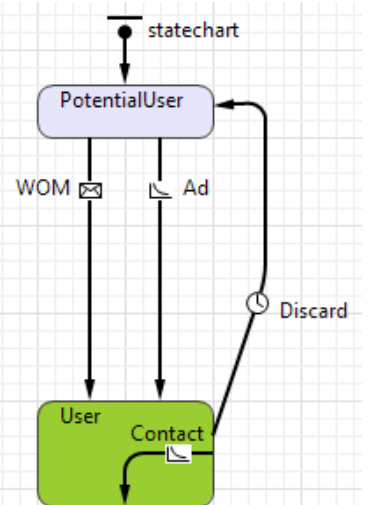
доставки продукта



В нашей текущей модели предполагается, что продукт всегда есть в наличии, и поэтому переход из состояния *PotentialUser* в состояние *User* происходит моментально. Теперь мы усовершенствуем модель, добавив у потребителя еще одно состояние, которое будет соответствовать времени, проходящему с момента принятия решения о покупке продукта до момента появления товара в продаже и доставки его покупателю.

1. Подготовьте место для нового состояния между состояниями *PotentialUser* и *User*, перетащив состояние *User* вниз.



2. Отсоедините состояние *User* от переходов. Выделите переходы *WOM* и *Ad* и переместите их конечные точки выше, затем отсоедините переход *Discard* от состояния *PotentialUser*. Вы заметите, что теперь эти переходы отображаются красным цветом.



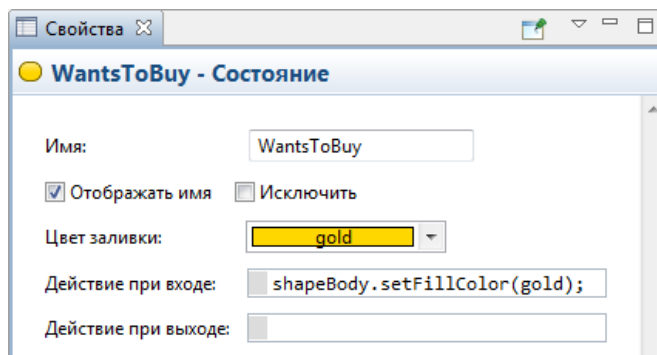
3. Добавьте новое Состояние  из палитры Диаграмма состояний  в середину диаграммы состояний потребителя и назовите его WantsToBuy («хочет купить»). Потребители в этом состоянии решили купить продукт, но продукт пока еще не приобрели.

4. Подсоедините переходы *WOM*, *Ad*, и *Discard* к среднему состоянию *WantsToBuy*.

5. Измените свойства состояния *WantsToBuy*:

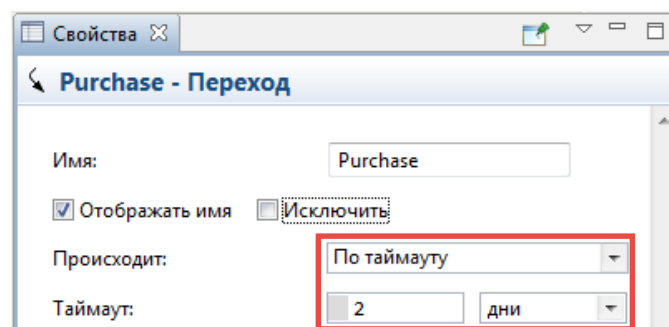
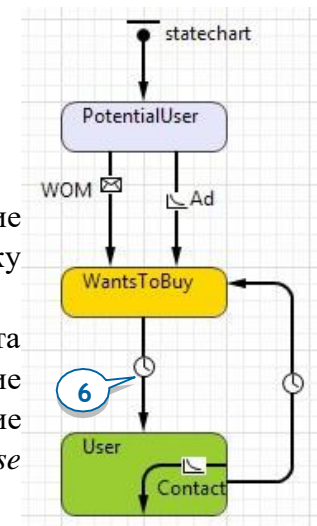
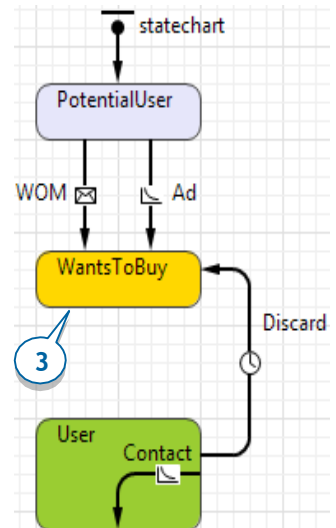
Цвет заливки: *gold*

Действие при входе: *shapeBody.setFillColor(gold);*



6. Добавьте переход из состояния *WantsToBuy* в состояние *User*, чтобы смоделировать доставку и, соответственно, покупку товара. Назовите этот переход *Purchase*.

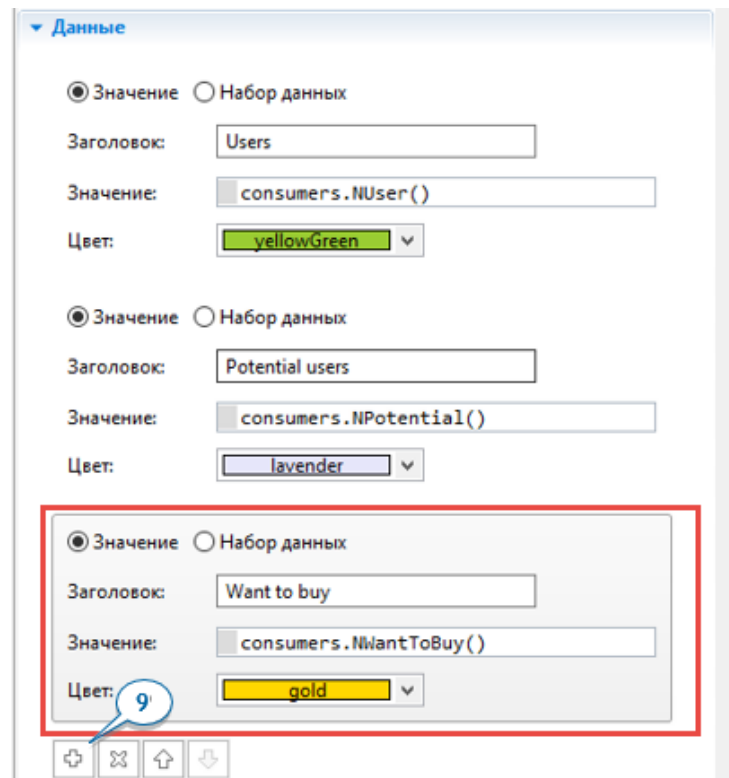
7. Давайте предположим, что в среднем доставка продукта занимает два дня. Это означает, что наш агент перейдет в состояние *User* через два дня после момента перехода в состояние *WantsToBuy*. Исходя из этого, задайте для перехода *Purchase* таймаут длительностью в 2 дня:




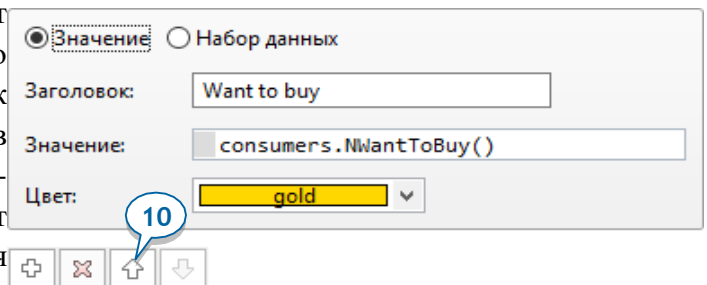
8. Создайте еще одну функцию сбора статистики, чтобы вести учет заявок на приобретение товара. Выделите популяцию *consumers* на диаграмме *Main*, перейдите в секцию свойств *Статистика* и добавьте новую функцию статистики с именем *NWantToBuy*

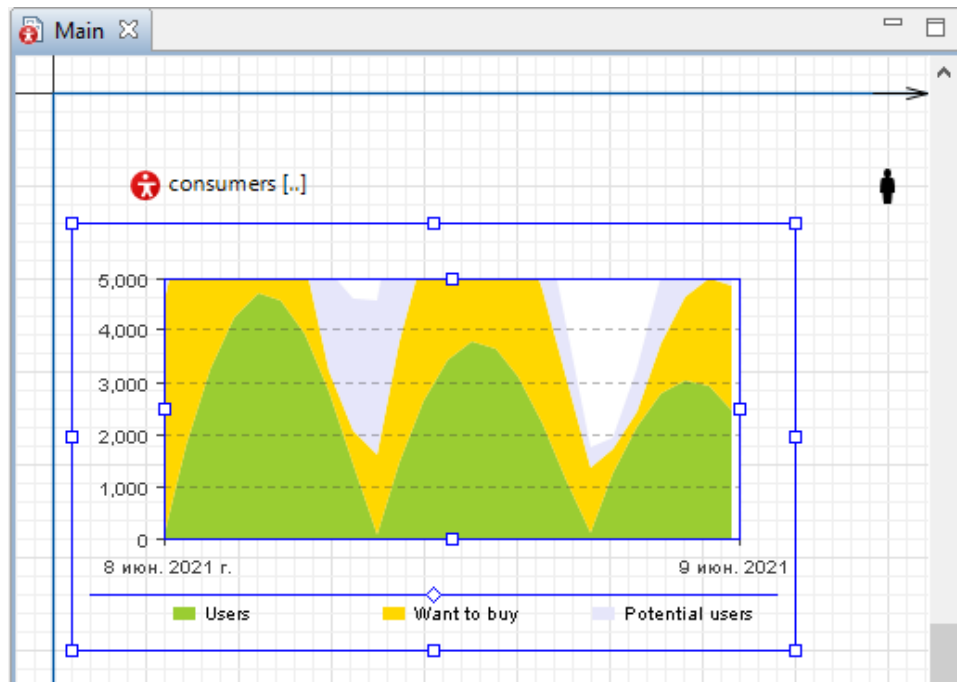
и условием *item.inState(Consumer.WantsToBuy)*

9. Далее, на диаграмме *Main*, выделите наш график и добавьте еще один элемент данных для отображения, со значением *consumers.NWantToBuy()*, заголовком *Want to buy* и цветом *gold*.

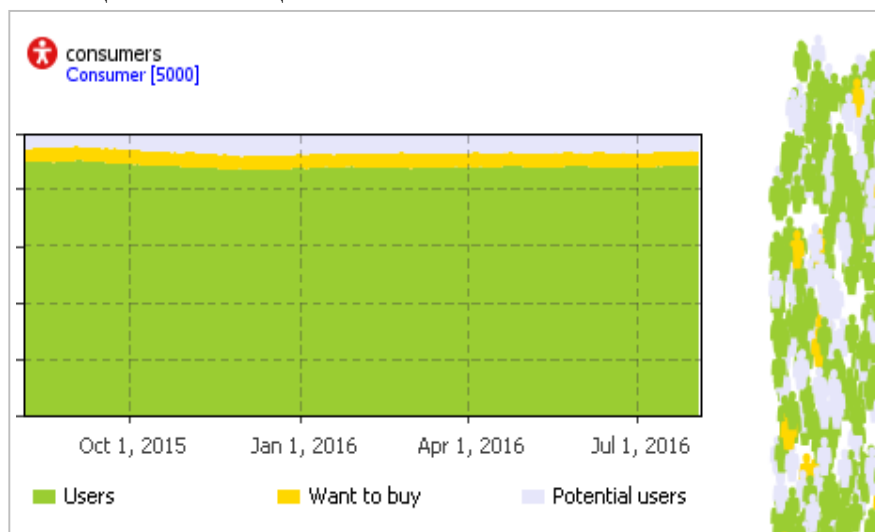


10. Переместите новый элемент данных в середину списка, чтобы во время моделирования график отображал категории на графике в следующем порядке: внизу - пользователи, затем - те, кто хочет приобрести продукт, и верхняя категория - потенциальные потребители продукта. Для этого выделите секцию свойств элемента *Want to buy* и затем щелкните по кнопке **Вверх** .





11. Запустите модель. Люди, ожидающие доставки товара, будут отображаться на графике и на анимации желтым цветом.



7. Моделирование отказов от покупки товара

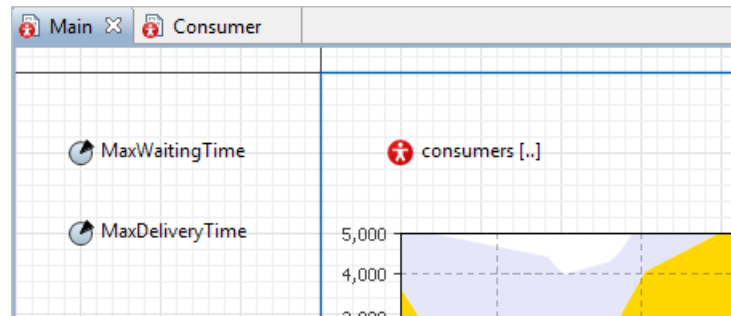
Теперь давайте учтем тот факт, что время, которое потребители согласны потратить на ожидание доставки товара, конечно. Если время доставки превысит предельно допустимое время ожидания, потребитель откажется от покупки.

Давайте начнем с того, что добавим на диаграмму *Main* два параметра, задающих максимальное время доставки товара (25 дней) и максимальное время ожидания доставки (7 дней) соответственно.

1. Откройте диаграмму *Main*.
2. Передвиньте холст графического редактора вправо, чтобы мы могли расположить элементы за пределами видимой области окна модели.

- ✓ Чтобы передвинуть холст графического редактора, нажмите правую кнопку мыши в редакторе и перемещайте мышь, не отпуская кнопку.
- ✓ Синяя прямоугольная рамка на диаграмме *Main* очерчивает границы окна

модели. При запуске модели вы увидите те элементы, которые расположены внутри этой рамки.



3. Создайте два параметра. Параметр *MaxWaitingTime* задает максимальное время, в течение которого потребитель готов ждать доставки продукта (в нашем случае - семь дней).

4. Другой параметр, *MaxDeliveryTime*, задает максимально возможное время доставки товара. Поскольку мы должны учесть специфику работы определенных отечественных служб доставки, зададим значение этого параметра равным 25 дням.

Таким образом, доставка товара может длиться от одного до 25 дней, в среднем же доставка занимает два дня. Давайте изменим значение времени доставки с фиксированного периода, равного двум дням, на стохастическое выражение, которое использует вышеуказанный диапазон значений.

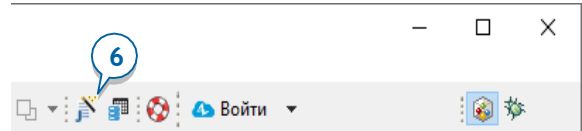
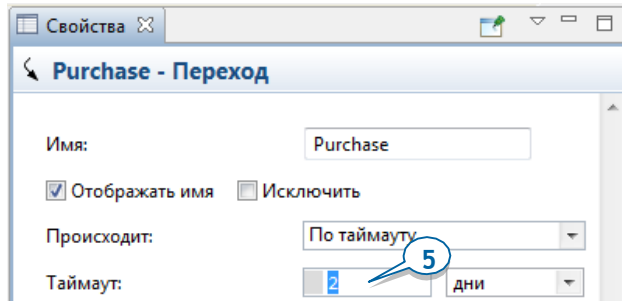
Функции распределения вероятностей

В следующей таблице мы рассмотрим самые часто используемые функции распределения вероятностей. Полный список функций, поддерживаемых AnyLogic, вы можете найти в документации, в разделе Приложение. Java в AnyLogic > Функции AnyLogic.

Функция распределения вероятностей	Описание
<p>Равномерное</p>  <p><code>uniform(min, max)</code></p>	<p>Используется, когда вы знаете минимальное и максимальное значения, но не знаете, являются ли одни значения в этом интервале значений более частотными, чем другие. Поэтому, выбирая данное распределение вероятности, вы просто полагаете, что величина принимает любое значение из заданного интервала с равной вероятностью.</p>
<p>Треугольное</p>  <p><code>triangular(min, mode, max)</code></p>	<p>Используется в том случае, когда у вас недостаточно измерений для построения эмпирического распределения, но вы знаете минимальное, максимальное и наиболее часто встречаемое (модальное) значение. Треугольное распределение обычно используется для задания времени обслуживания или длительности операции.</p>
<p>Экспоненциальное</p>  <p><code>exponential(lambda, min)</code></p>	<p>Описывает время между происхождением событий как Пуассоновский процесс, т.е. цепочку событий, происходящих независимо друг от друга с постоянной (в среднем) интенсивностью. Используется для определения времени между прибытиями посетителей, звонков, заказов, деталей в процессно-ориентированных моделях.</p>
<p>Нормальное</p>  <p><code>normal(sigma, mean)</code></p>	<p>Хорошо описывает данные, группирующиеся вокруг известного среднего значения. Чтобы ограничить значения (например, отсеять отрицательные значения), используйте нормальное распределение с ограничениями или другое распределение (логнормальное, Гамма, Бета или Вейбулла).</p>
<p>Дискретное равномерное</p> 	<p>Используется для моделирования конечного числа результатов, происходящих с равной вероятностью. В интервал возможных значений включаются и минимальное, и максимальное значения, поэтому вызов функции <code>uniform_discr(3, 7)</code> может вернуть 3, 4, 5, 6 или 7.</p>

Ознакомившись с информацией в приведенной выше таблице, можно прийти к выводу, что самым подходящим распределением вероятностей для задания времени ожидания является треугольное.

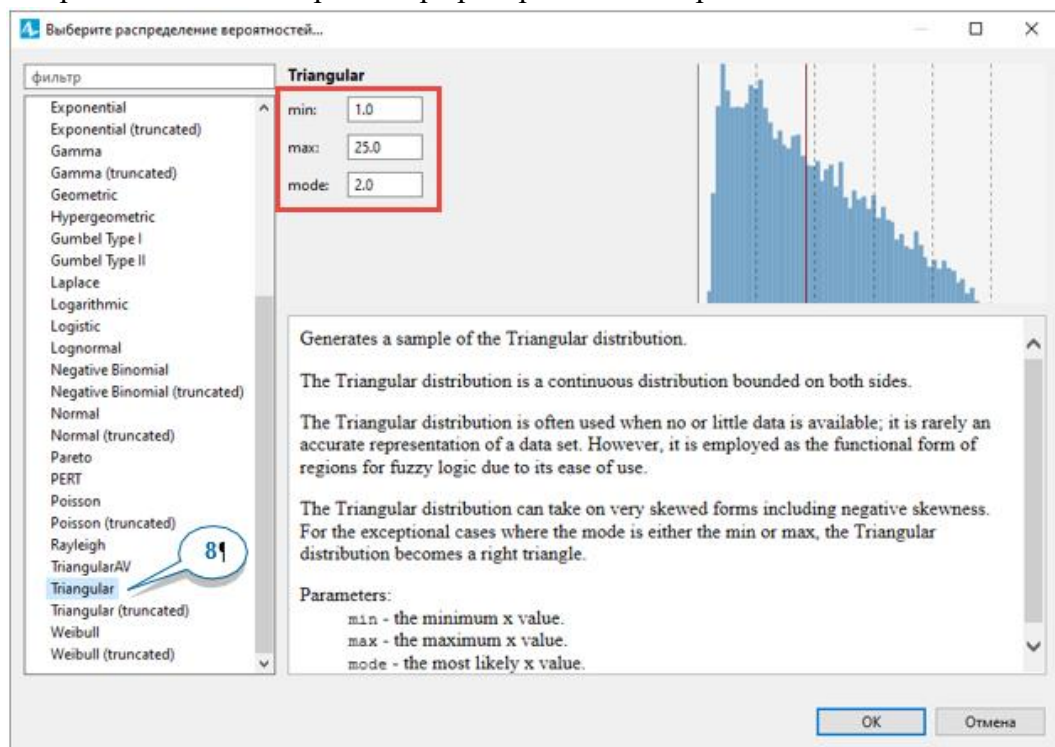
5. Откройте диаграмму агента *Consumer* и выделите переход *Purchase*. Мы хотим изменить значение таймаута, по которому срабатывает переход. Для этого мы воспользуемся мастером выбора функций распределения, который вставит вызов функции в поле свойства перехода. Чтобы заменить текущее значение поля **Таймаут**, выделите его мышью.



6. Щелкните по кнопке панели управления **Выберите распределение вероятностей...**



7. Откроется окно Мастера выбора распределения вероятностей.



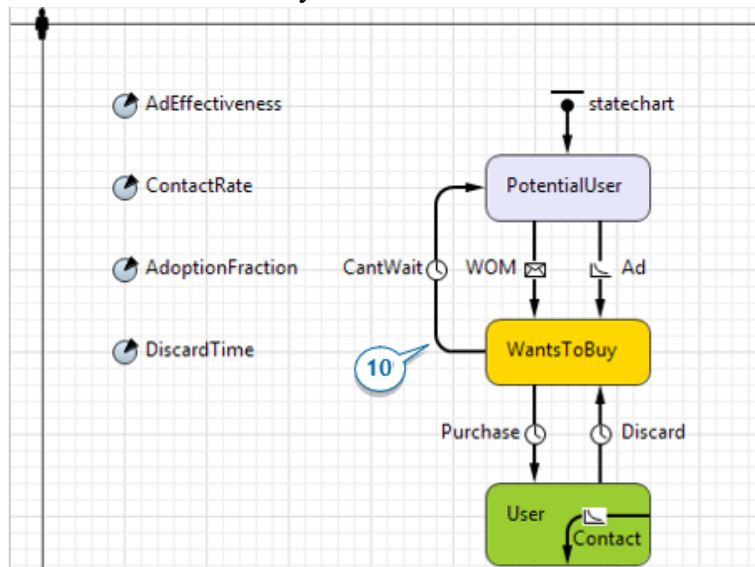
8. Диалоговое окно **Выберите распределение вероятностей** позволяет выбрать одну из поддерживаемых AnyLogic функций распределения вероятностей. Выберите функцию *Triangular* в расположенном слева списке. Введите в поля параметров **min**, **max** и **mode** значения 1, 25 и 2 соответственно. В правом верхнем углу вы увидите автоматически построенную гистограмму значений, сгенерированных функцией с заданными параметрами. Щелкните по кнопке **ОК**, чтобы вставить вызов функции в кодовое поле.

9. В поле задания значения таймаута будет автоматически вставлено выражение *triangular(1, 25, 2)*. Давайте изменим эту строку на выражение *triangular(1, main.MaxDeliveryTime, 2)*

Здесь мы используем префикс *main*, чтобы получить доступ к агенту *Main* из агента *Consumer*.

10. Нарисуйте переход под названием *CantWait*, который выходит из состояния

WantsToBuy и ведет в состояние *PotentialUser*. Этот переход моделирует то, как потребитель отказывается от покупки товара ввиду его долгого отсутствия. В результате диаграмма состояний агента *Consumer* будет выглядеть так:



11. Измените свойства перехода, задав его Таймаут равным `triangularAV(main.MaxWaitingTime, 0.15)` дней.

CantWait - Переход

Имя: CantWait ☐ Отображать имя ☐ Исключить

Происходит: По таймауту

Таймаут: triangularAV(main.MaxWaitingTime, 0.15) дни

Мы задаем максимальное время ожидания с помощью треугольного распределения со средним значением, равным параметру *MaxWaitingTime* (т.е., одной неделе), и отклонением от этого значения, равным 15 процентам.

Мы используем параметр, а не просто указываем соответствующее значение времени для того, чтобы впоследствии иметь возможность варьировать это значение динамически и наблюдать эффект от производимых изменений прямо по ходу моделирования. Одним из способов создания интерактивной модели является добавление элементов управления и связывание их с варьируемыми параметрами.

Элементы управления

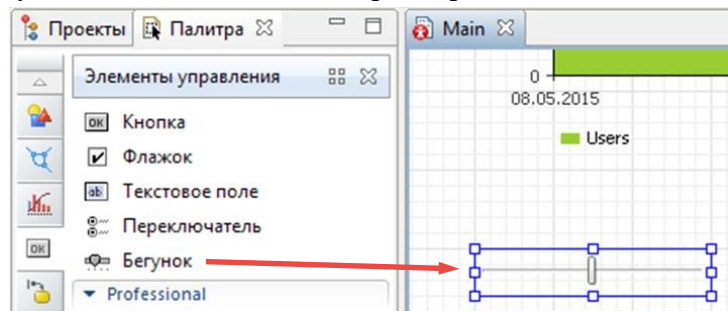
Модели можно сделать интерактивными, добавив в интерфейс модели различные *элементы управления* (кнопки, бегунки, текстовые поля и т.д.). Элементы управления могут использоваться как для задания значений параметров перед началом выполнения модели, так и для изменения модели прямо по ходу ее выполнения.

У элементов управления, имеющих состояние или содержимое (таких, как бегунок, переключатель, текстовое поле и т.д.), есть текущее *значение*, и они могут быть *связаны* с переменными и параметрами, так что, когда пользователь изменяет состояние такого элемента управления, изменяется и значение связанного с ним элемента (но не наоборот). Кроме того, вы можете задать для элемента управления определенное действие, например: вызов функции, планирование события, передачу сообщения, остановку модели и т.д. Действие будет выполняться каждый раз, когда пользователь меняет состояние элемента управления. Значение элемента управления обычно доступно в коде его поля *Действие* как

value, а также возвращается функцией элемента *getValue()*.

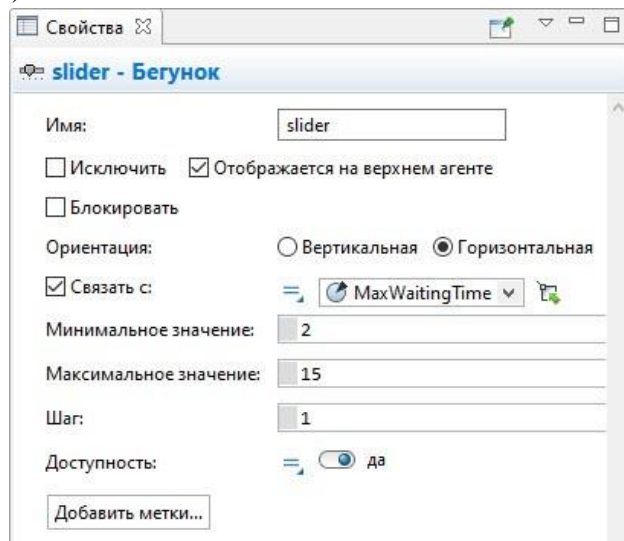
Мы добавим бегунок - элемент управления, который позволяет выбирать числовое значение из определенного интервала. Бегунок часто используется для того, чтобы изменять значения численных переменных и параметров.

12. Вернитесь на диаграмму *Main*. Откройте палитру **Элементы управления**, перетащите элемент **Бегунок** на диаграмму и расположите его под графиком. Сейчас мы свяжем этот бегунок с одним из наших параметров.

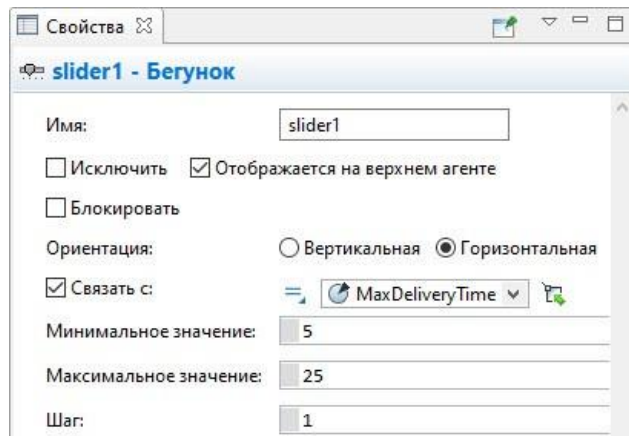


13. Измените свойства бегунка:

- Установите флажок **Связать с** и выберите параметр *MaxWaitingTime* из расположенного справа списка.
- Задайте минимальное и максимальное значения бегунка. Вы сможете варьировать значение параметра в заданном интервале значений. Введите 2 в поле **Минимальное значение** и 15 в поле **Максимальное значение**.
- Чтобы при работе с бегунком выбирались только целые числа, введите 1 в поле **Шаг**.
- Затем щелкните по кнопке **Добавить метки...**, чтобы отображать эти значения бегунка во время моделирования (при этом под бегунком появятся текстовые метки *min*, *value* и *max*).

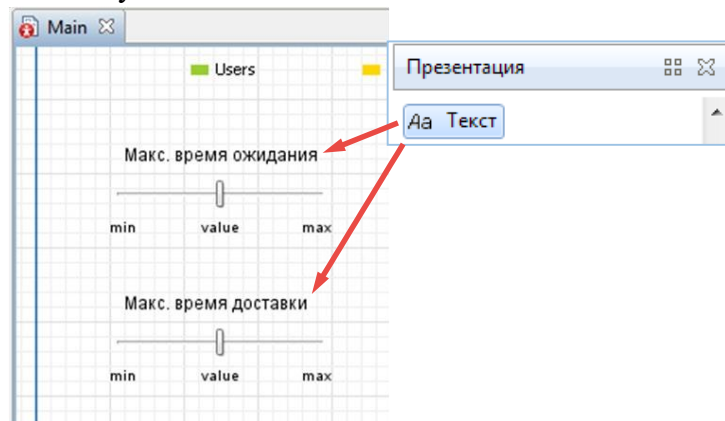


14. Добавьте еще один бегунок под предыдущим и настройте его следующим образом:

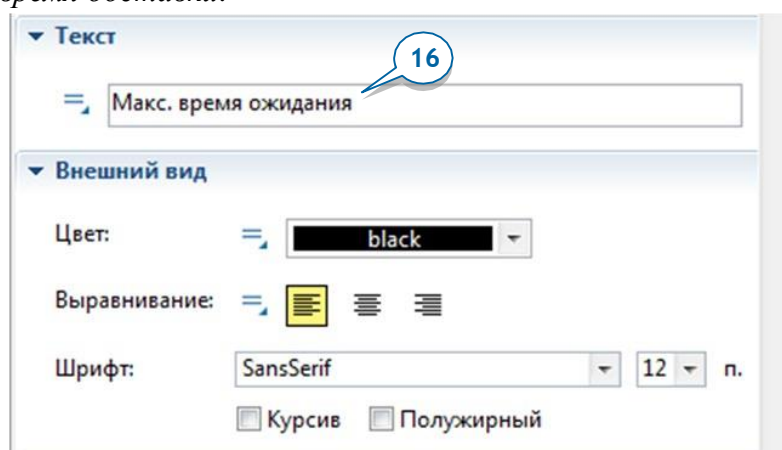


Давайте добавим подписи для каждого созданного бегунка. Для этого воспользуемся фигурой презентации Текст Аа.

15. Откройте палитру Презентация, перетащите две фигуры Текст Аа на диаграмму и расположите их над бегунками.



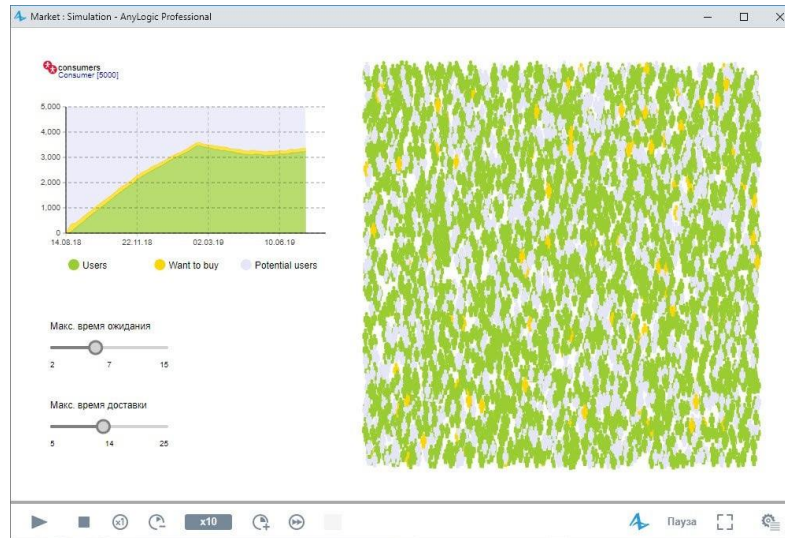
16. В свойствах текстовых меток, в секции Текст, задайте текст, который вы хотите отображать с помощью этих меток. Для верхней метки введите *Макс. время ожидания*, для нижней - *Макс. время доставки*.



В секции свойств Внешний вид вы можете отформатировать текст, изменив его цвет, выравнивание, шрифт и размер.

Подписи под бегунками также являются фигурами Текст Аа. Вы можете редактировать их, как любую другую текстовую фигуру AnyLogic.



17. Запустите модель и наблюдайте за ее поведением. Изменяя максимальное время ожидания и максимальное время доставки, вы можете оценить влияние этих изменений на поведение потребителей и состояние рынка.



Контрольные вопросы

1. В чем причины появления агентного моделирования?
2. Поясните сущность агентного моделирования?
3. Как может быть задана структура агентной модели?
4. Как можно задать поведение агентной модели?
5. Какими именно свойствами должен обладать объект, чтобы называться агентом?
6. Приведите примеры агентов.
7. Опишите пользовательский интерфейс AnyLogic, включая графический редактор и панели?
8. Каким образом осуществляется навигация по элементам модели в панели

Проекты?

9. Какой тип агента создается по умолчанию?
10. Как создать популяцию агентов?
11. Какие типы пространства могут быть у популяции агентов?
12. Какие типы агентов были созданы в построенной модели?
13. Что из себя представляет среда обитания агентов?
14. Где можно изменить настройки среды обитания?
15. Как скомпилировать модель?
16. Как запустить модель?
17. Опишите кнопки управления выполнением модели.
18. Что такое диаграмма состояний?
19. Опишите следующие инструменты: .
20. Как работать с мастером подстановки кода?
21. Какие параметры можно задать для элемента **Состояние** .
22. Какие типы срабатывания элемента переход вы можете назвать?
23. Где можно назвать единицы модельного времени?
24. Поясните сущность понятия «модельное время»?
25. Опишите режимы выполнения модели и кнопки панели **Модельное время**?

26. Какая функция проверяет, является ли для этого агента активным указанное состояние диаграммы состояний?
27. Назовите и опишите типы диаграмм в Anylogic?
28. Как можно осуществить подбор масштаба графика в Anylogic?
29. Какими способами можно организовать Взаимодействие агентов в Anylogic?
30. Опишите различия между внутренними и внешними переходами.
31. В каком случае Anylogic выделяет переходы красным цветом?
32. Опишите какие инструменты используются в Anylogic для исправления ошибок?
33. Назовите наиболее часто используемые функции распределения вероятностей?
34. Каким образом в Anylogic можно выбирать распределения вероятностей?
35. Какие элементы управления можно добавить в модель?
36. Зачем используются элементы управления в модели?
37. Подготовьте детальный рассказ об изменениях вносимых в модель и настройках основных элементов.