

Министерство транспорта Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский университет транспорта (МИИТ)»(РУТ(МИИТ))

Институт транспортной техники и систем управления  
Кафедра «Управление и защита информации»

## **Отчет**

на тему:

«Изучение современных средств моделирования систем AnyLogic»

Выполнил: ст. гр. ВУЦ-421

Полунин С.К.

Проверил: к.т.н., доц.

Логинова Л. Н.

Москва 2024

Отчёт выполнен в 1 части и содержит: страниц – 46, иллюстраций – 66, в отчёте использовано источников – 2.

## Содержание

<b>Введение.....</b>	<b>4</b>
<b>Основная часть отчета .....</b>	<b>5</b>
<b>1. Постановка задачи .....</b>	<b>5</b>
<b>2. Модель потребительского рынка .....</b>	<b>5</b>
<b>2.1 Создание популяции агентов.....</b>	<b>5</b>
<b>2.2 Задание поведения потребителей.....</b>	<b>12</b>
<b>2.3 Добавление графика для визуализации результатов моделирования .....</b>	<b>16</b>
<b>2.4 Добавление эффекта рекомендации .....</b>	<b>20</b>
<b>2.5 Учёт повторных продаж продукта .....</b>	<b>25</b>
<b>2.6 Учёт времени доставки продукта .....</b>	<b>26</b>
<b>2.7 Моделирование отказов от покупки .....</b>	<b>29</b>
<b>3. Модель распространения эпидемии .....</b>	<b>34</b>
<b>3.1 Создание диаграммы потоков и накоплений.....</b>	<b>34</b>
<b>3.2 Добавление графика для визуализации динамики процесса .....</b>	<b>39</b>
<b>3.3 Эксперимент варьирования параметров.....</b>	<b>41</b>
<b>Заключение .....</b>	<b>45</b>
<b>Список использованных источников.....</b>	<b>46</b>

## **Введение**

Программное обеспечение AnyLogic предоставляет возможности моделирования для организаций в сферах транспорта, производства, логистики, добычи полезных ископаемых, цепочек поставок, здравоохранения и многих других.

AnyLogic – инструмент для имитационного моделирования различными методами, разработанный «The AnyLogic Company», обеспечивает улучшенную эффективность и снижение рисков при решении сложных задач в различных отраслях, а так же позволяет исследовать сценарии "что если" с помощью имитационного 2D- и 3D-моделирования.

Встроенные библиотеки анимации, охватывающие различные отрасли, обеспечивают возможность моделирования сложных систем с различными уровнями детализации. Модели AnyLogic позволяют аналитикам, инженерам и менеджерам глубже понять взаимосвязанные процессы внутри и вокруг организаций, оптимизируя сложные системы и процессы в широком спектре отраслей.

AnyLogic позволяет создавать имитационные модели с использованием различных методологий и языков моделирования, таких как дискретно-событийное, агентное, системное, стохастическое моделирование, блок-схемы процессов, диаграммы состояний и диаграммы действий. Пользователи могут создавать визуальные модели с графическими объектами, представляющими транспортные средства, персонал, оборудование, здания и другие объекты в соответствии с требованиями бизнеса.

## **Основная часть отчета**

### **1. Постановка задачи**

Разработать модель потребительского рынка и модель распространения эпидемии.

### **2. Модель потребительского рынка**

Для изучения процесса вывода нового продукта на рынок создадим агентную модель.

- 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.

#### **2.1 Создание популяции агентов**

Для начала создадим новую модель (рисунок 1), в который мы непосредственно будем работать.

В ней указываем следующие параметры:

- Имя модели: Market
- Java пакет: market
- Единицы модельного времени – указываем «дни»

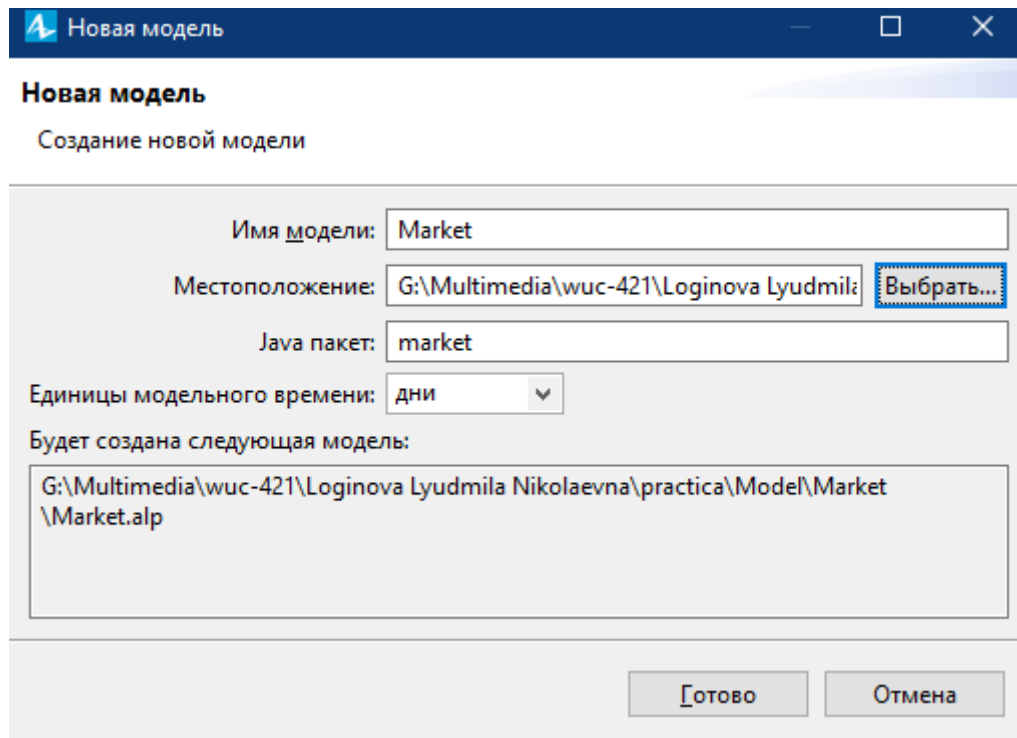


Рисунок 1 – Создание новой модели

Добавим Агента на диаграмму Main (рисунок 2).

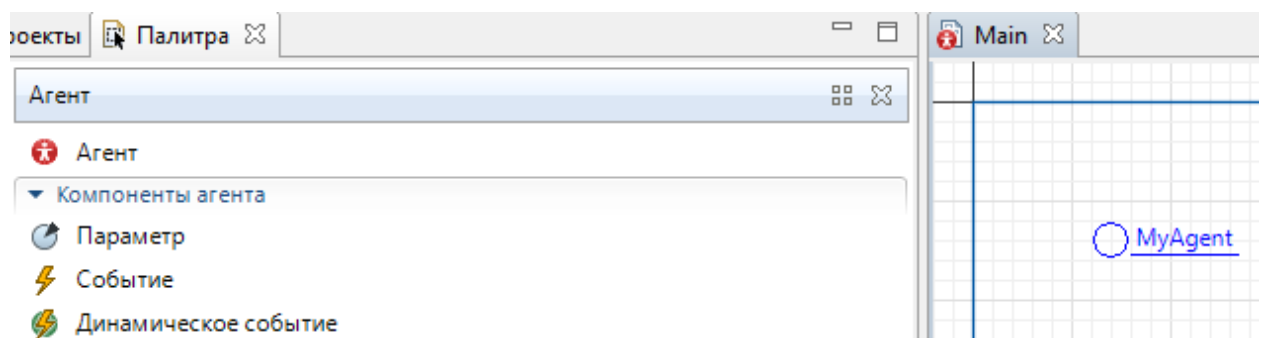


Рисунок 2 – Добавление Агента на диаграмму Main

После добавления откроется мастер создания агентов «Новый агент». Нам необходимо создать большое количество агентов одного типа, поэтому на «Шаг 1» мастера выберем опцию «Популяция агентов» (рисунок 3).

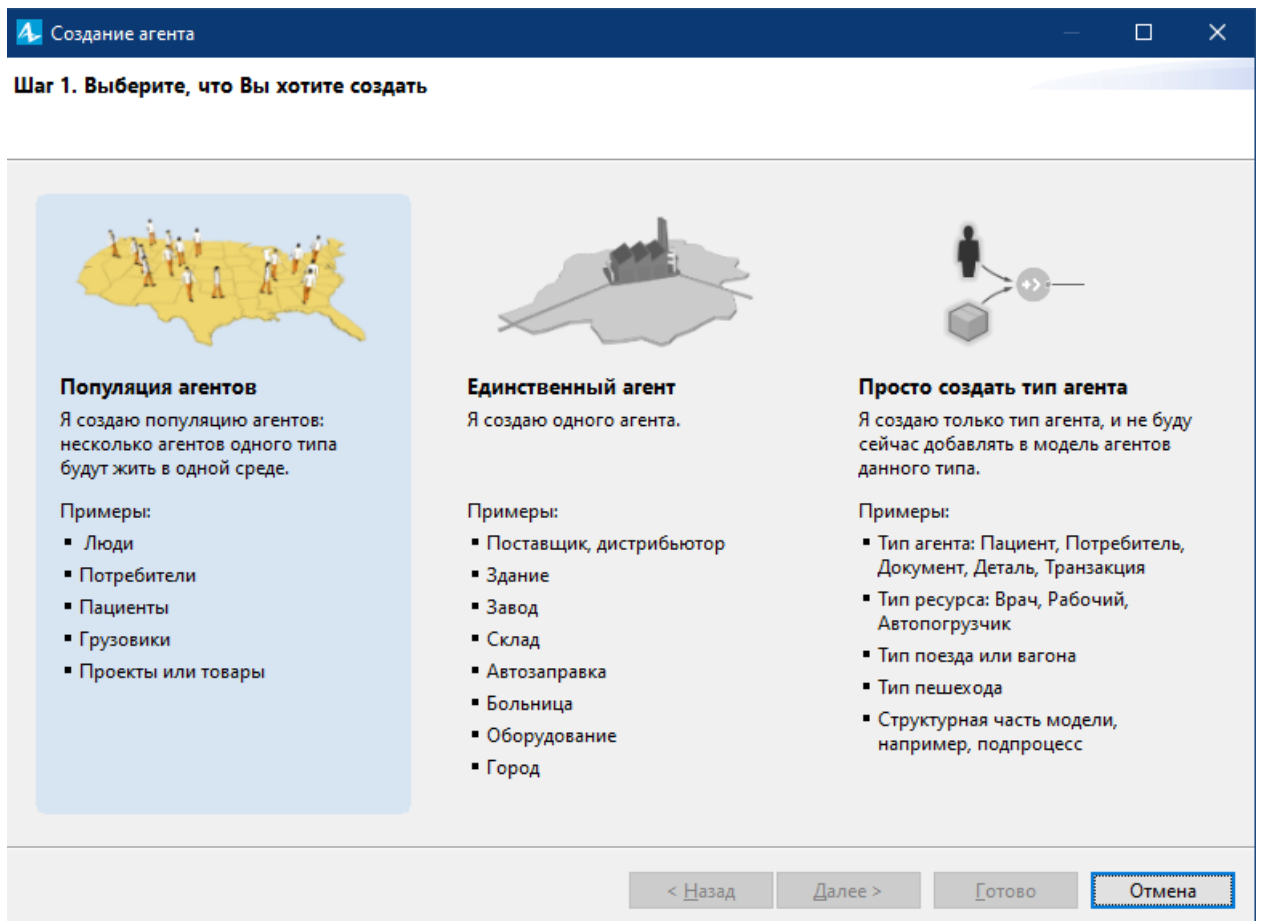


Рисунок 3 – Выбор опции

На странице мастера «Шаг 2», заполним поле «Имя нового типа», вводим «Consumer» (Потребитель) (рисунок 4) и содержание поля «Имя популяции» автоматически изменится на «consumers».

**Создание агента**

**Шаг 2. Создание нового типа агента**

Имя нового типа:

Имя популяции:

☒ Создать новый тип агента "с нуля"

☐ Использовать таблицу базы данных  
Я хочу считать значения параметров агентов из базы данных

☐ Агент будет использоваться в диаграммах процессов

Рисунок 4 – Ввод имени

На странице мастера «Шаг 3» выберем фигуру анимации агента. Поскольку мы создаем простую модель с двумерной анимацией, выбираем опцию «2D» и затем выбираем первую фигуру (Человек) из расположенного ниже списка (рисунок 5).

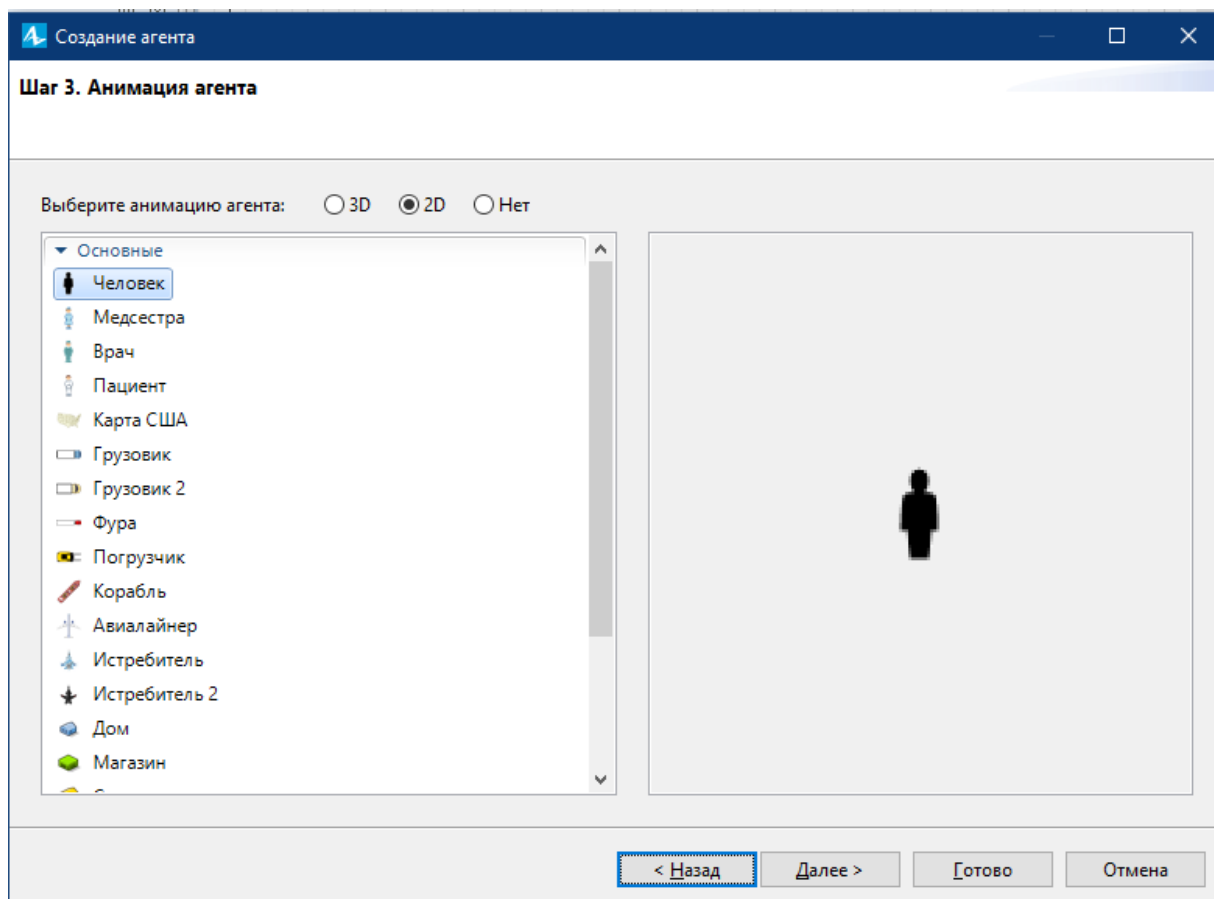


Рисунок 5 – Выбор фигуры

На следующей странице мастера «Шаг 4» 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.

В поле «Параметр» изменяем заданное по умолчанию имя параметра на AdEffectiveness. Выбираем 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 (рисунок 6).



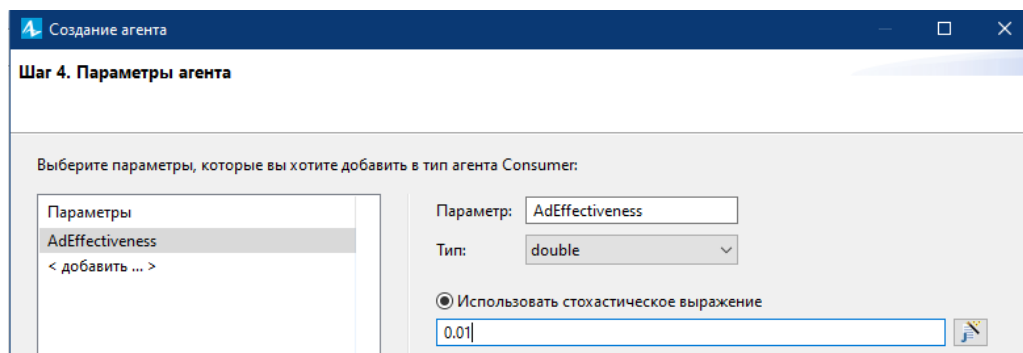


Рисунок 6 – Настройка параметров агента

На следующей странице мастера «Шаг 5» в поле «Создать популяцию с агентами» вводим значение 5000, чтобы создать 5000 агентов типа Consumer (рисунок 7). Каждый агент, живущий в создаваемой нами популяции, будет моделировать отдельного агента-потребителя.

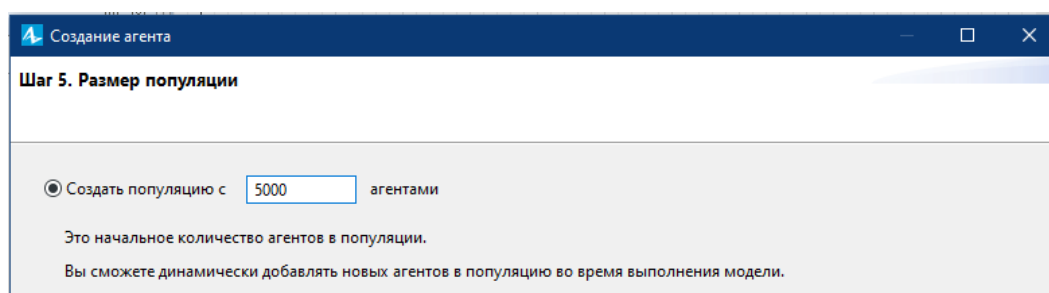


Рисунок 7 – Размер популяции

На странице мастера «Шаг 6» оставляем 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. Выберем опцию «Применить случайное расположение» (рисунок 8).

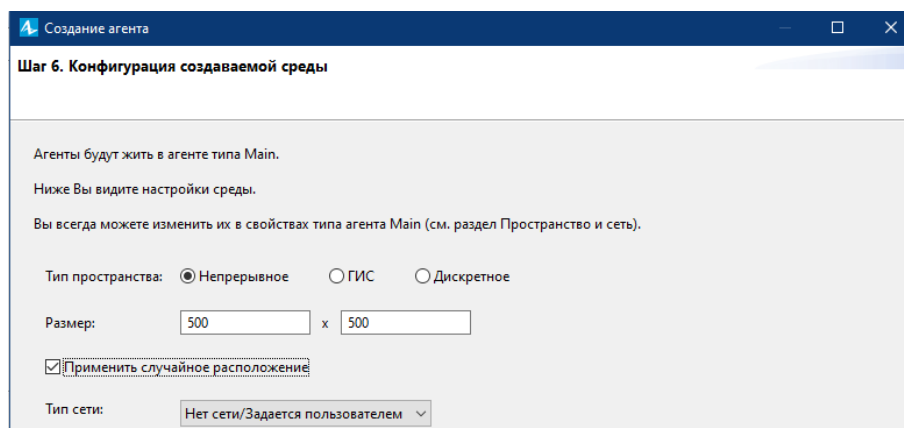


Рисунок 8 – Конфигурация создаваемой среды

После завершения настройки агентов, можно заметить, что у нашей модели два типа агентов: «Main» и «Consumer» (рисунок 9).

- 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).

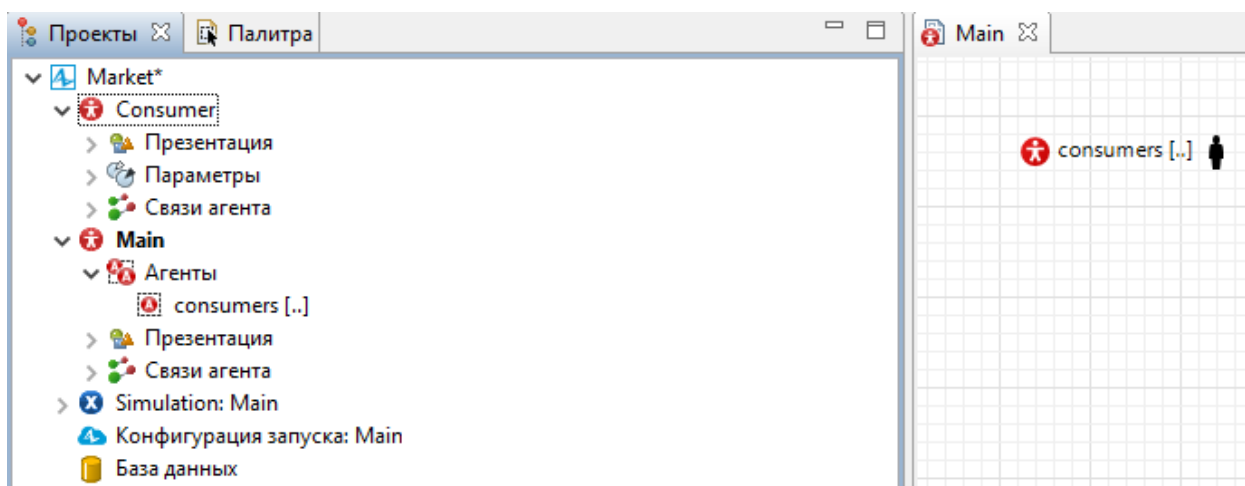


Рисунок 9 – Отображение новых типов Агентов

Выделяем фигуру анимации популяции агентов, расположенную на диаграмме Main, откроем секцию ее свойств «Специфические» и выберем опцию «Рисовать агента со сдвигом от данной точки» (рисунок 10).

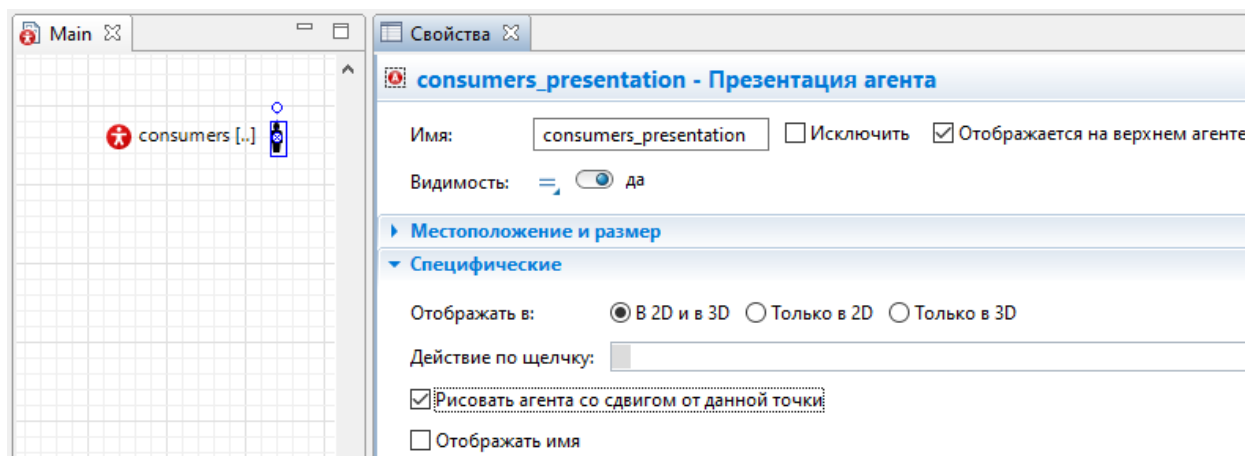


Рисунок 10 – Настройка отображение агентов

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

Нажимаем на «Построить модель», затем на «Запустить», появится окно модели. Увидим презентацию модели - 5000 фигур анимации агентов популяции «consumers». Так как мы пока не задавали правила поведения агентов, на анимации больше ничего не происходит (рисунок 11).

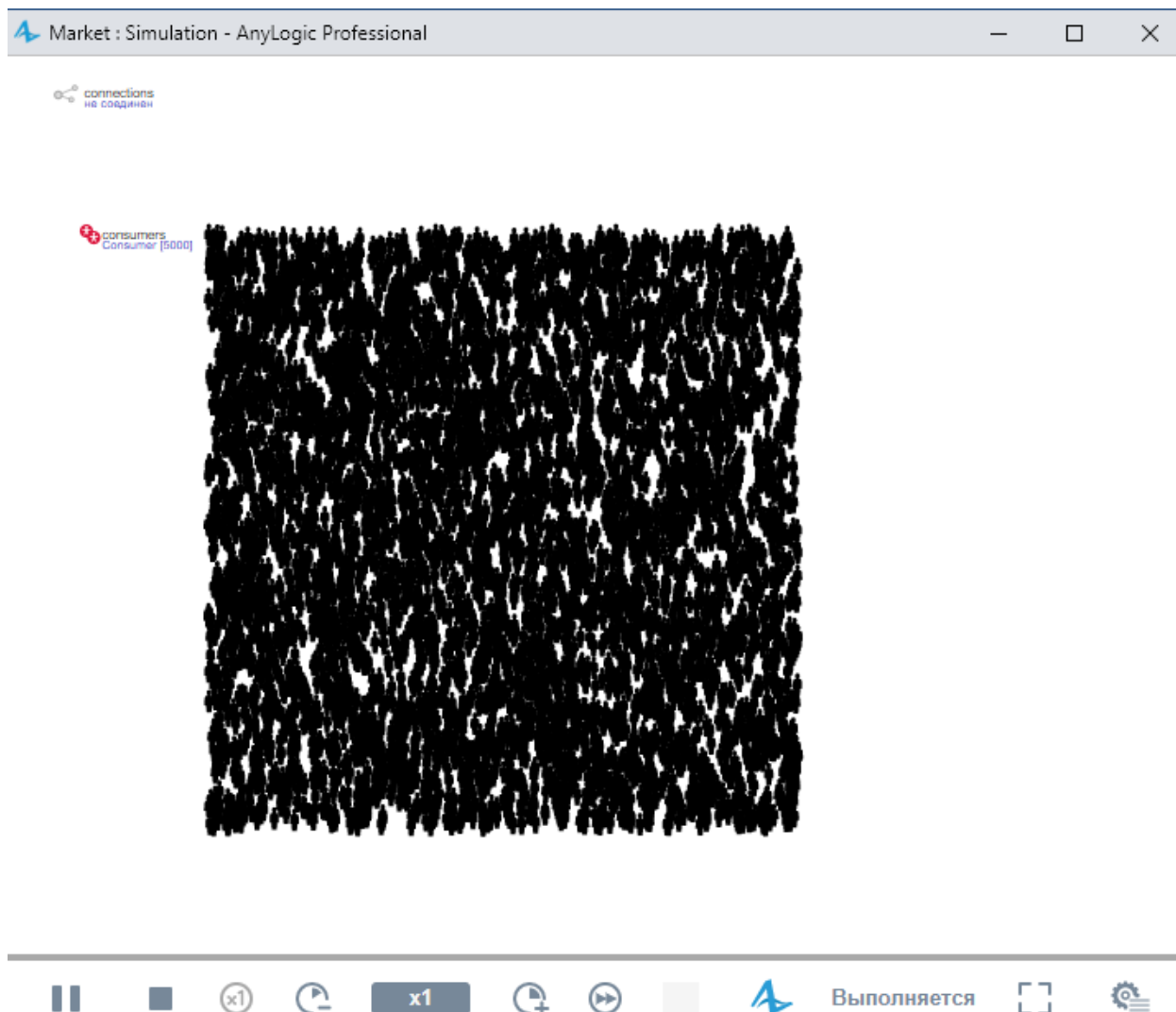


Рисунок 11 – Результат работы нашей модели

## 2.2 Задание поведения потребителей

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

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.

Выбрав на панели управления «Consumer», добавляем на диаграмму «Начало диаграммы состояний» (рисунок 12).

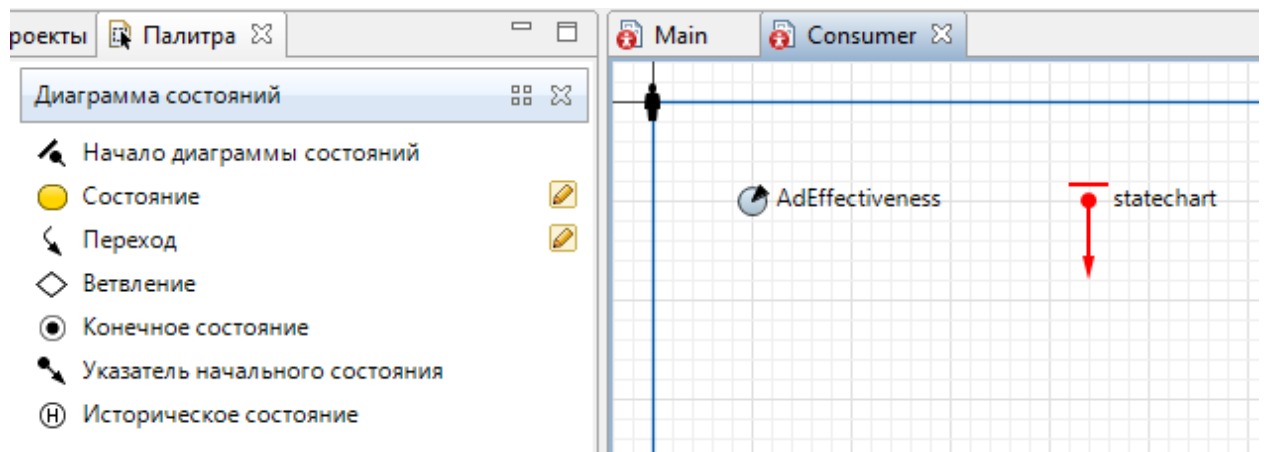


Рисунок 12 – Добавление «начало диаграммы состояний»

Затем добавим элемент «Состояние» и соединим с началом диаграммы (рисунок 13):

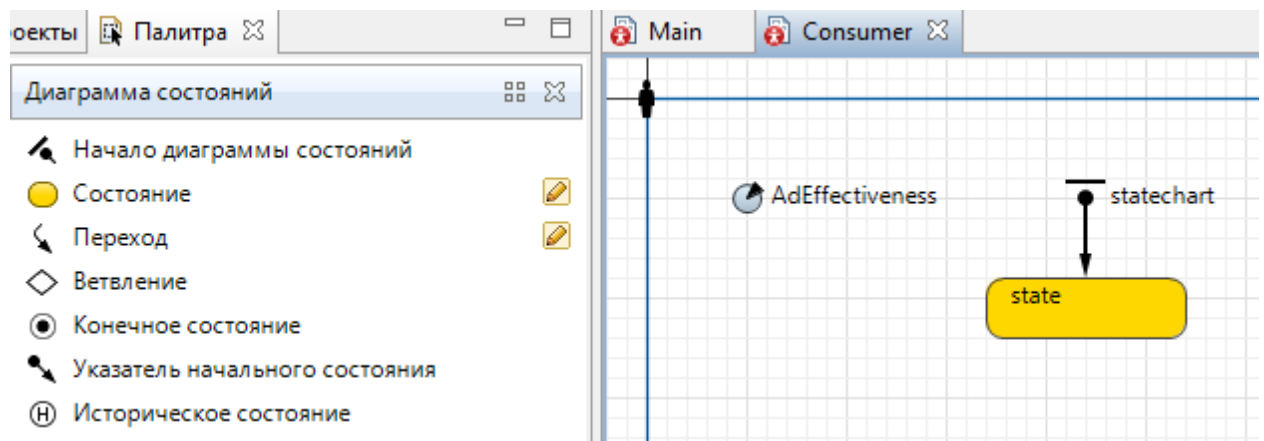


Рисунок 13 – Добавление элемента «состояние»

Выделяем «состояние» и изменяем его свойства: назовем состояние «PotentialUser» и поменяем цвет заливки на «lavender». После впишем в графу «Действие при входе:» `shapeBody.setFillColor(lavender);`, где `shapeBody` является именем фигуры анимации потребителя, которую мы выбрали в мастере создания агентов (рисунок 14).

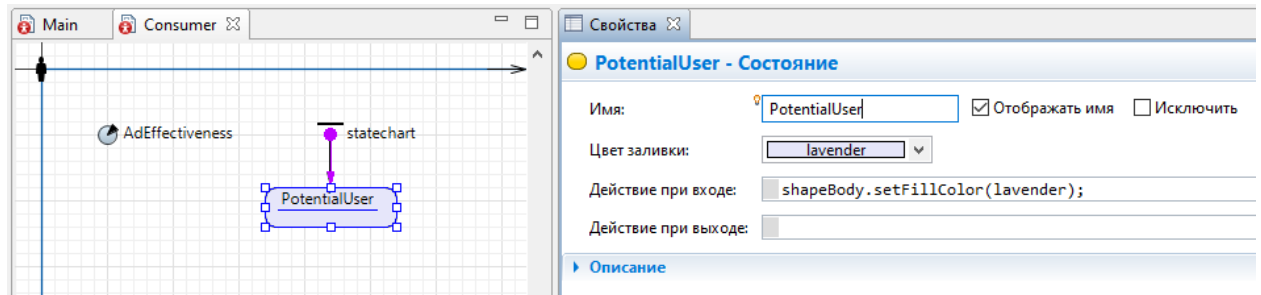


Рисунок 14 – Изменение элемента «состояние»

Добавим еще одно состояние в диаграмму состояний потребителя, где будут изменены следующие свойства: имя – «User», цвет заливки – «yellowGreen», действие при входе – «`shapeBody.setFillColor(yellowGreen);`» (рисунок 15).

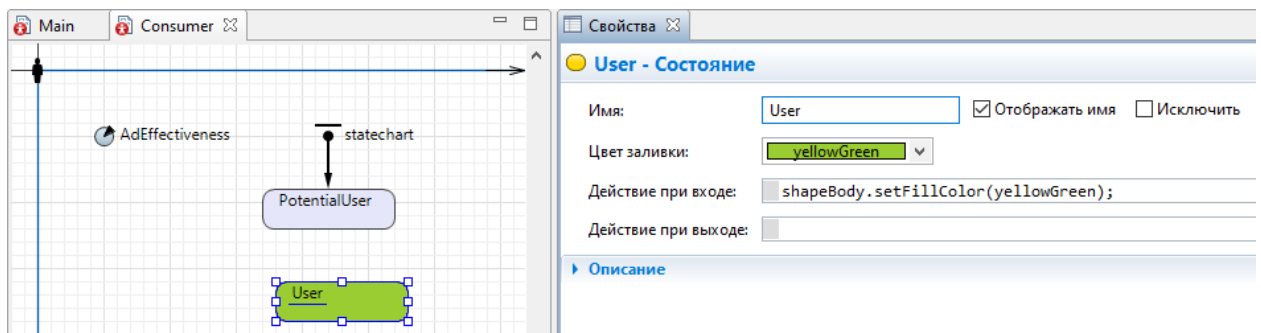


Рисунок 15 – Добавление нового состояния

Нарисуем переход из состояния PotentialUser в состояние User, чтобы промоделировать, как человек приобретает продукт (и становится его потребителем). Для этого воспользуемся функцией «Переход» в палитре Диаграмма состояний (рисунок 16).

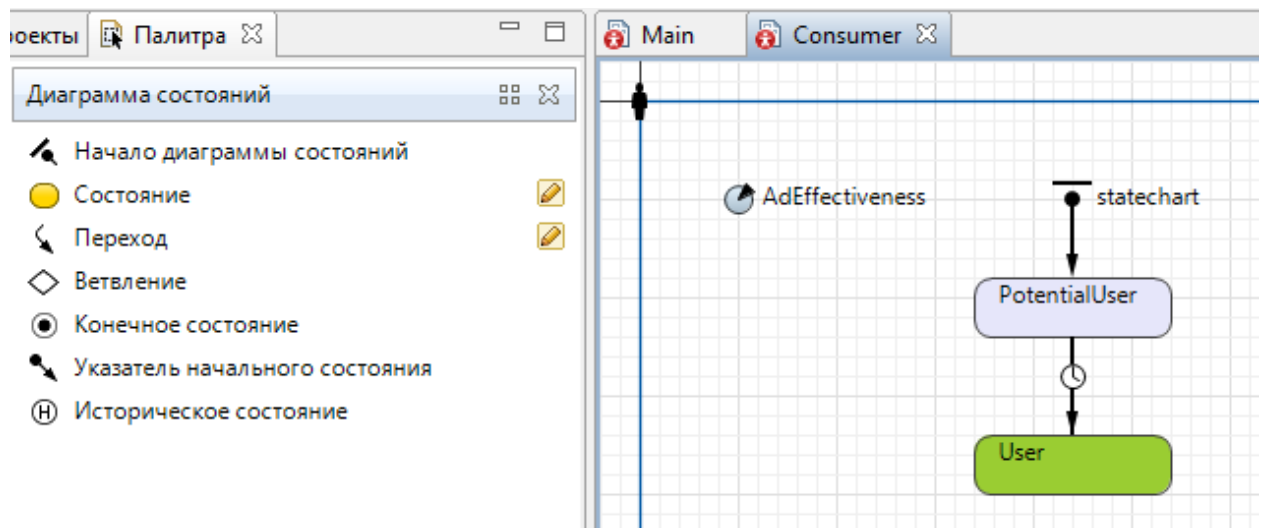


Рисунок 16 – Добавление функции «переход»

Назовём переход Ad, потому что он представляет действие рекламы, от английского слова «advertising» - “реклама”. Поставим галочку в свойстве «Отображать имя», чтобы имя перехода показывалось в графическом редакторе.

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».

В итоге наш значок поменяется, что будет отображать тип срабатывания перехода (рисунок 17).

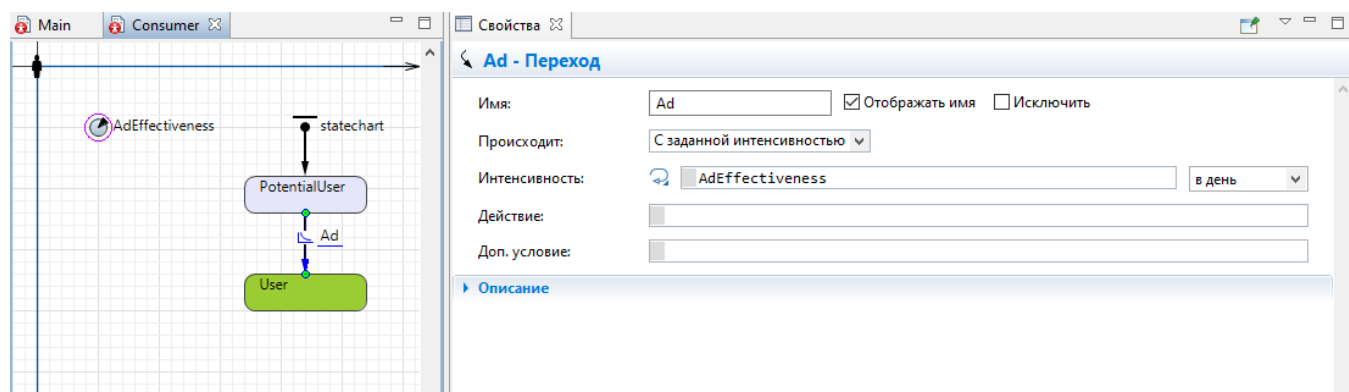


Рисунок 17 – Настройка функции «переход»

Теперь зададим единицы модельного времени. Чтобы изменить настройки модели, переключаемся из «Палитры» в панель «Проекты», и затем выбираем элемент модели в дереве (самый верхний уровень дерева, элемент «Market»). Перейдем в панель «Свойства» и выберем дни в качестве «Единиц модельного времени» (рисунок 18).

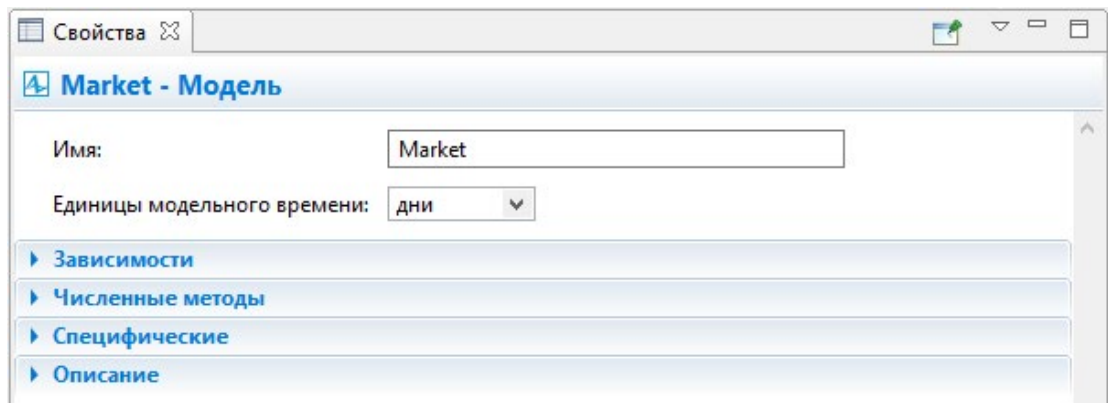


Рисунок 18 – Задача единиц модельного времени

Запускаем модель. Популяция агентов постепенно окрашивается в зеленый цвет (изменение, к которому приводит эффект рекламы), пока каждый потенциальный потребитель не купит продукт (рисунок 19).



Рисунок 19 – Результат моделирования

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 светлозеленый (yellowGreen). Чем больше людей покупают продукт, тем больше появляется зеленых фигур анимации агентов.

### **2.3 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.

Сначала зададим функцию, которая будет считать количество потенциальных потребителей. Чтобы добавить новую функцию подсчета статистики по популяции агентов, откроем диаграмму агента «Main», выделим популяцию агентов «consumers» и перейдем в раздел свойств «Статистика».

Нажимаем на «чтобы добавить элемент данных..», указываем имя – «NPotential», задаем функцию типа «кол-во» и вводим в условии «item.inState(Consumer.PotentialUser)» (рисунок 20), где

- 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.



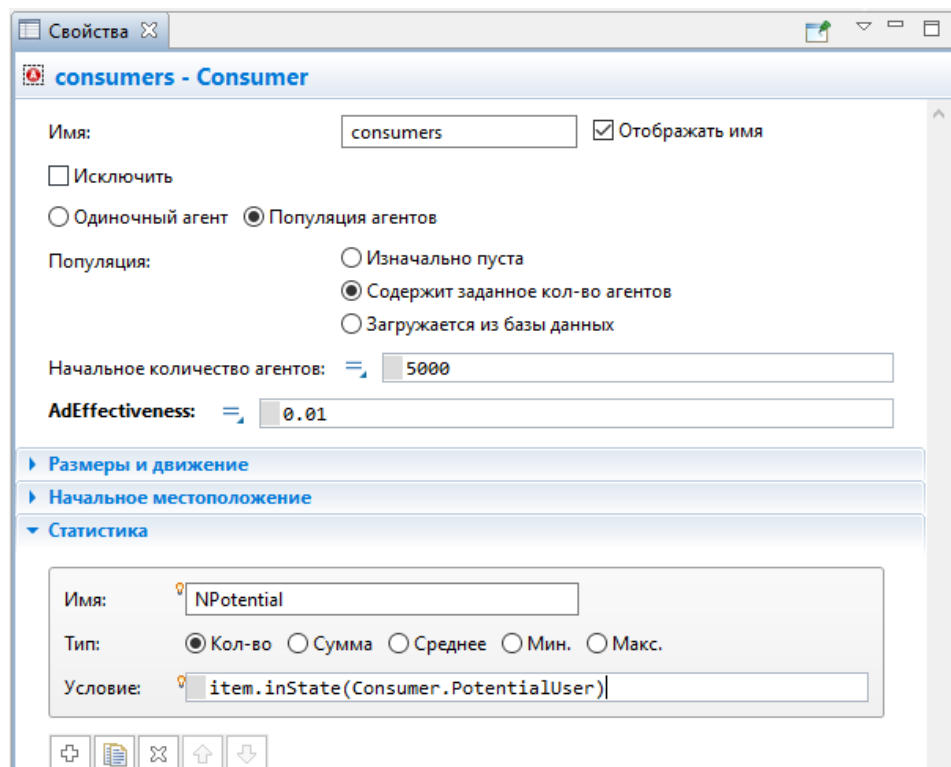


Рисунок 20 – Заполнение раздела статистики

Задайте вторую функцию статистики для подсчета потребителей продукта. Назовите ее «NUser». Пусть она считает количество агентов, для которых выполняется «Условие» `item.inState(Consumer.User)` (рисунок 21).

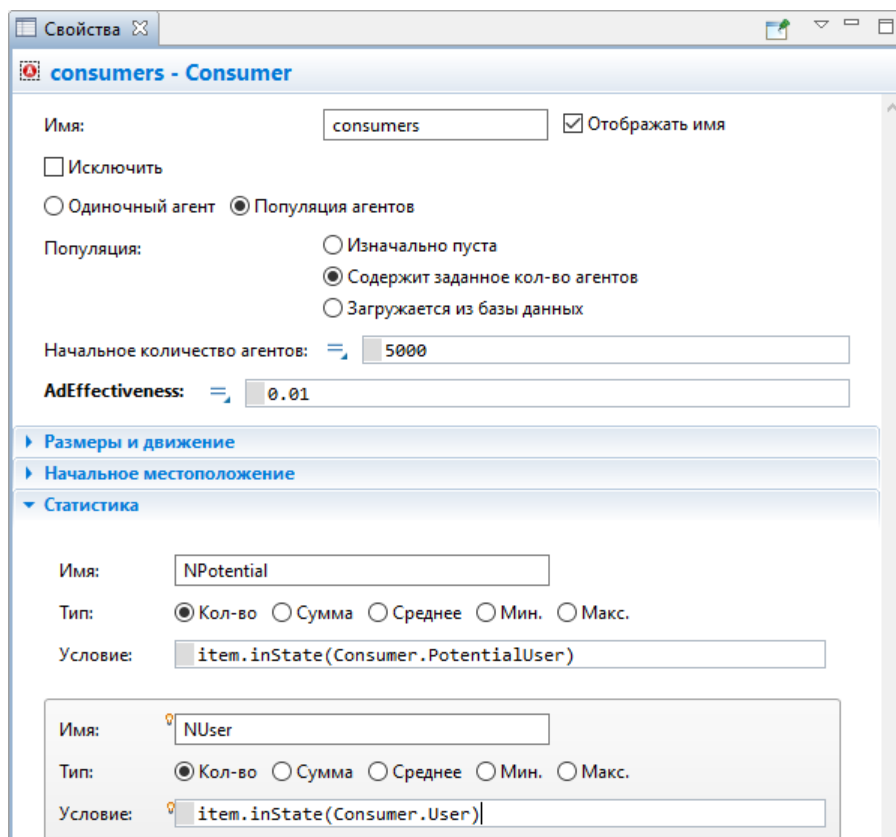


Рисунок 21 – Добавление второй функции раздела статистики

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

На диаграмме «Main», переместите вправо фигуру презентации популяции агентов «consumers».

Откроем палитру «Статистика» и перетащим элемент «Временная диаграмма с накоплением» из палитры на диаграмму «Main», чтобы создать график, который будет отображать динамику изменений числа потенциальных потребителей и владельцев продукта (рисунок 22).

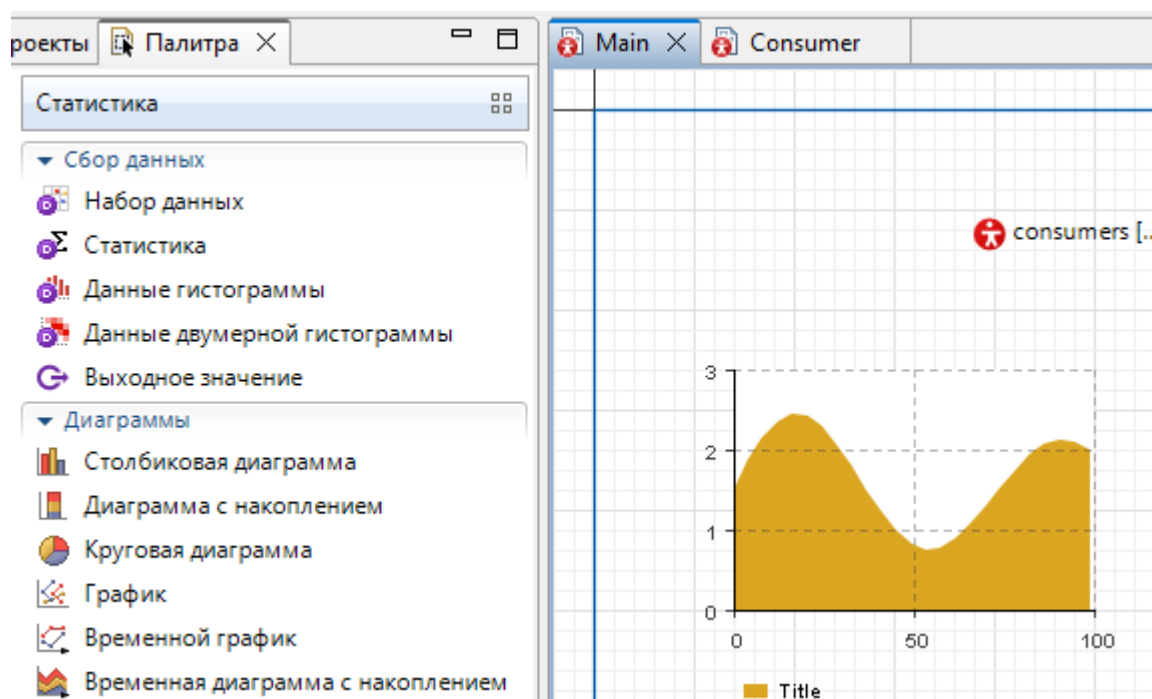
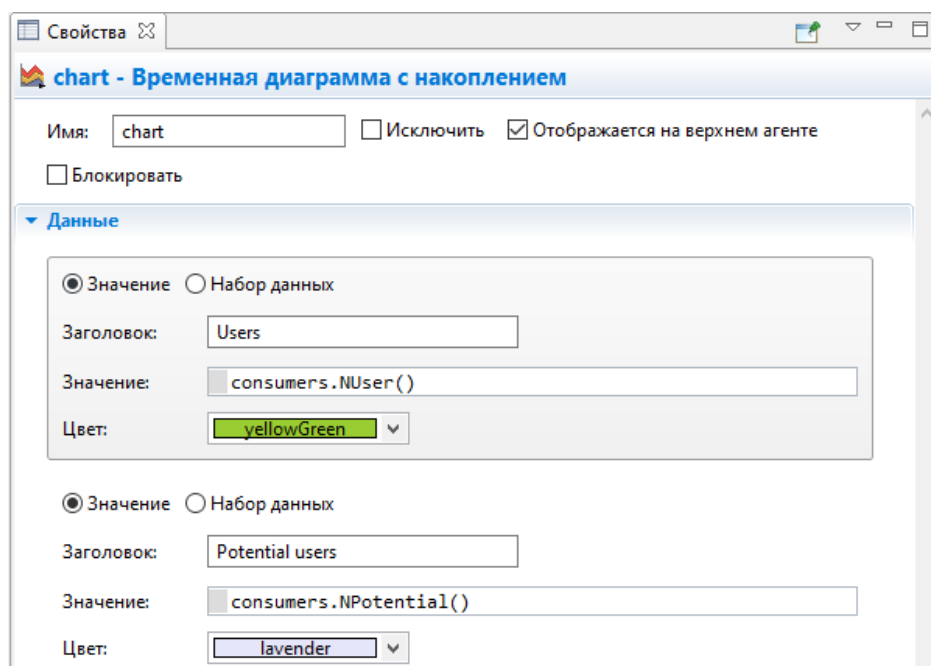


Рисунок 22 – Добавление временной диаграммы

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

В секции «Данные» свойств временной диаграммы с накоплением, измените свойства элемента данных следующим образом: заголовок – «Users», цвет – «yellowGreen», значение – «consumers.NUser()», где consumers - это имя нашей популяции агентов, а NUser() - это функция сбора статистики, которую мы задали ранее.

Добавим ещё один элемент данных, его свойства: заголовок – «Potential users», цвет – «lavender», значение – «consumers.NPotential()» (рисунок 23).



Свойства

chart - Временная диаграмма с накоплением

Имя:  ☐ Исключить ☒ Отображается на верхнем агенте

☐ Блокировать

Данные

☒ Значение ☐ Набор данных

Заголовок:

Значение:

Цвет:

☒ Значение ☐ Набор данных

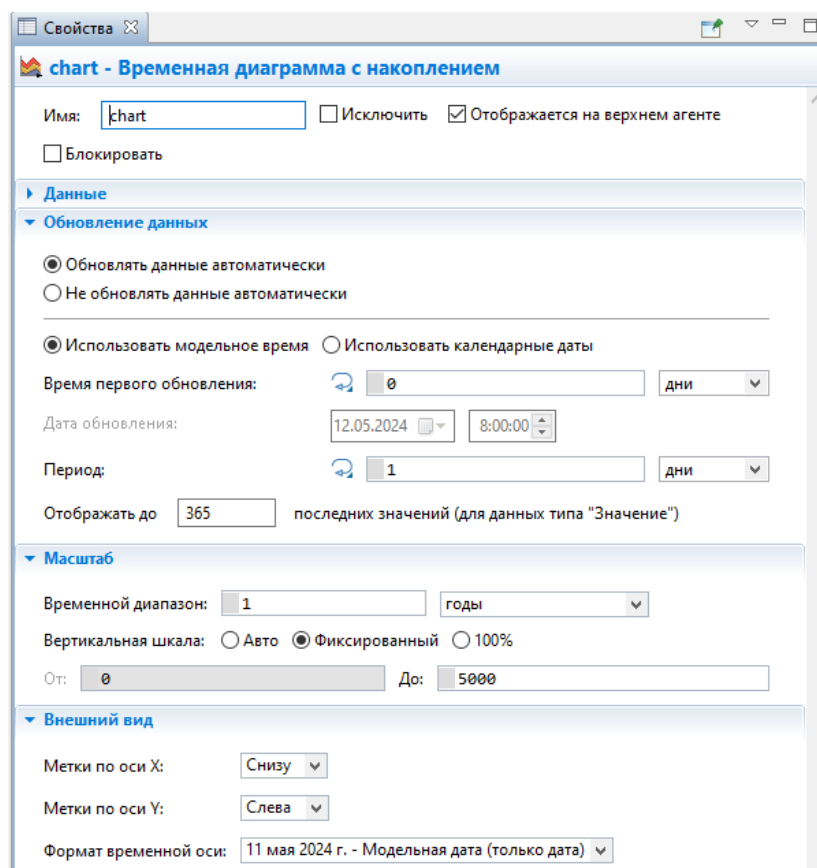
Заголовок:

Значение:

Цвет:

Рисунок 23 – Временная диаграмма с внесенными данными

Изменим следующие параметры в нашей диаграмме: «обновление данных», «масштаб» и «внешний вид» (рисунок 24):



Свойства

chart - Временная диаграмма с накоплением

Имя:  ☐ Исключить ☒ Отображается на верхнем агенте

☐ Блокировать

Данные

Обновление данных

☒ Обновлять данные автоматически ☐ Не обновлять данные автоматически

☒ Использовать модельное время ☐ Использовать календарные даты

Время первого обновления:  дни

Дата обновления:

Период:  дни

Отображать до  последних значений (для данных типа "Значение")

Масштаб

Временной диапазон:  годы

Вертикальная шкала: ☐ Авто ☒ Фиксированный ☐ 100%

От:  До:

Внешний вид

Метки по оси X:

Метки по оси Y:

Формат временной оси:

Рисунок 24 – Изменение параметров

Запускам нашу модель, рисунок 25:



Рисунок 25 – Работа модели

## 2.4 Добавление эффекта рекомендация

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  $\text{AdoptionFraction} = 0.01$ .

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

Выполним следующее: откроем диаграмму типа агента «Consumer» добавим параметр, который будет задавать среднее количество контактов потребителя с другими людьми в течение дня. Перетащим элемент «Параметр» из палитры «Агент» на диаграмму агента «Consumer». Назовем параметр «ContactRate». В данной модели средняя интенсивность контактов равна одному контакту в день. Перейдем в свойства этого параметра и введем «1» в поле «Значение по умолчанию». Добавим еще один параметр, «AdoptionFraction», который задает вероятность приобретения продукта в результате общения с пользователем этого продукта. В свойствах данного параметра зададим «Значение по умолчанию: 0.01» (рисунок 26).

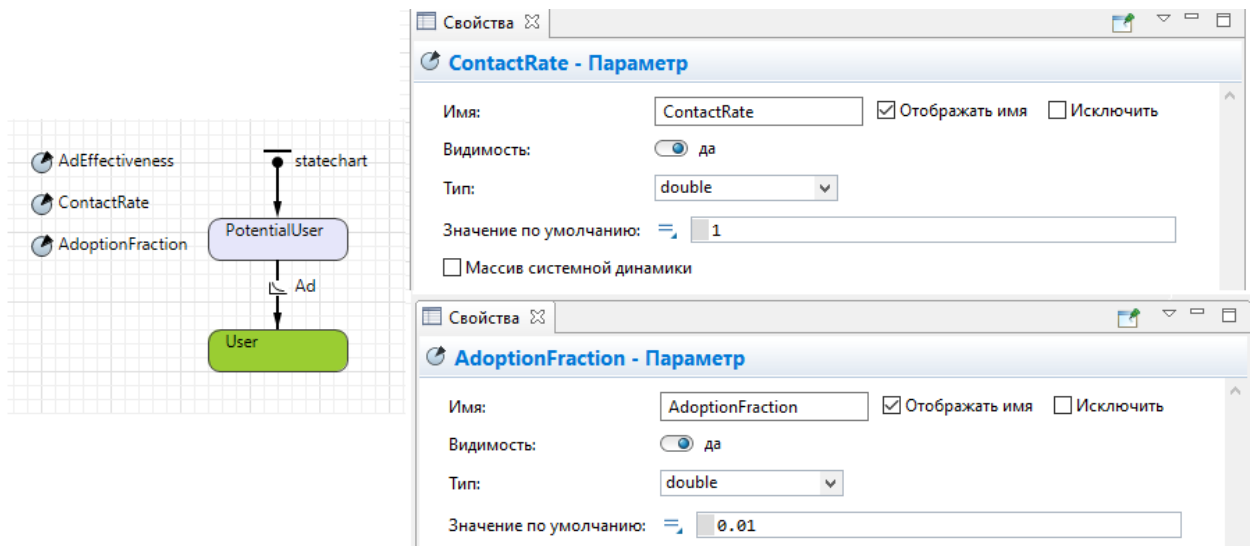


Рисунок 26 – Добавление новых параметров

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.

Откроем диаграмму агента «Consumer» и изменим размер состояния User, чтобы в него поместился внутренний переход, который мы сейчас добавим.

Нарисуйте переход внутри состояния User (рисунок 27).

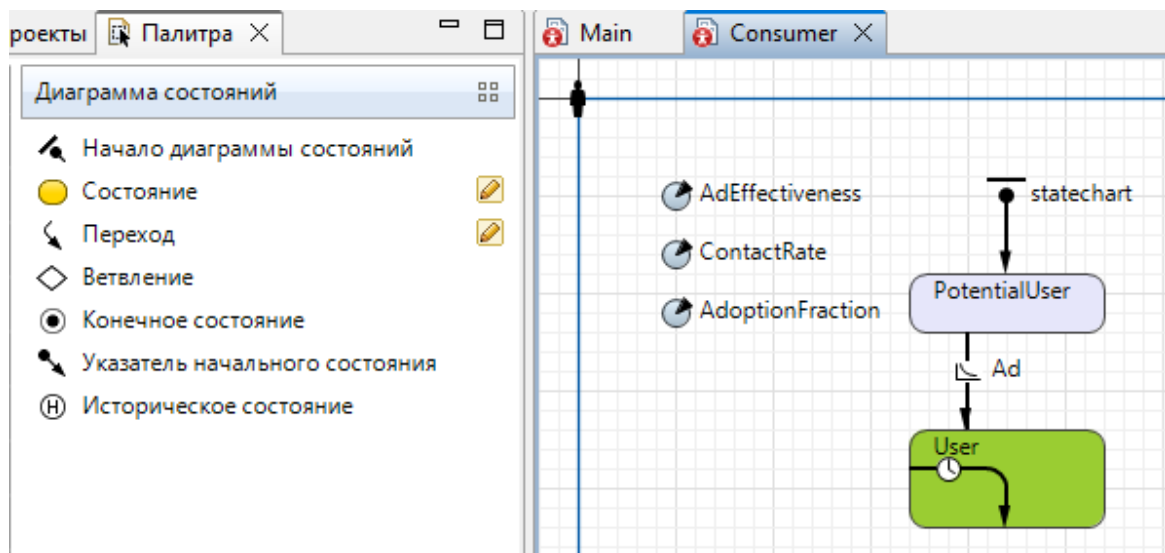


Рисунок 27 – Отображение внутреннего перехода

Изменим свойства перехода. Этот переход будет происходить «С заданной интенсивностью», равной значению параметра «ContactRate». Назовем переход «Contact» и включим отображение имени этого перехода в графическом редакторе (рисунок 28). Укажем «Действие», которое должно выполняться при срабатывании перехода: `sendToRandom("Buy")`.

The screenshot shows the 'Свойства' (Properties) window for a transition named 'Contact'. The window has a title bar 'Свойства' and a close button. The main area is titled 'Contact - Переход'. It contains several fields and checkboxes:
 

- Имя:** A text field containing 'Contact' and a checkbox 'Отображать имя' (Show name).
- Исключить:** A checkbox.
- Происходит:** A dropdown menu set to 'С заданной интенсивностью' (With given intensity).
- Интенсивность:** A section with a circular arrow icon, a text field containing 'ContactRate', and a dropdown menu set to 'в день' (per day).
- Действие:** A text field containing the code `sendToRandom("Buy")`.
- Доп. условие:** An empty text field.

Рисунок 28 – Изменение свойств

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 нарисуем сейчас).

Нарисуем еще один переход из состояния PotentialUser в состояние User и назовем его «WOM». Этот переход будет моделировать покупку продукта в результате рекомендаций других людей (рисунок 29).

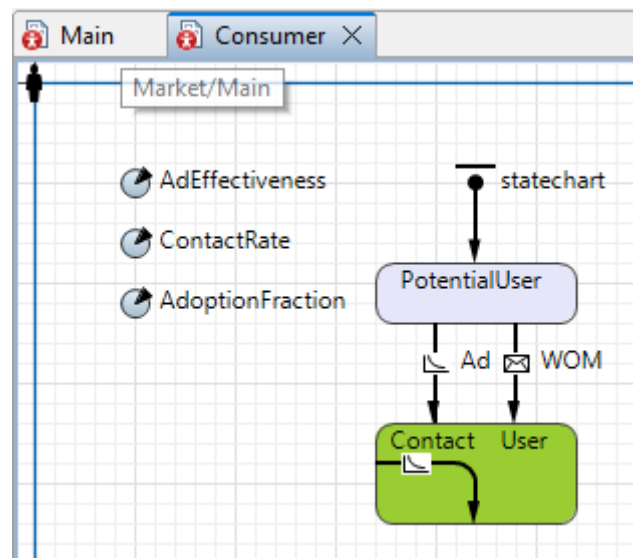


Рисунок 29 – Добавление нового перехода

Добавим параметры к переходу «WOM» (рисунок 30). 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)`

The screenshot shows the 'WOM - Переход' (WOM - Transition) configuration window. The 'Имя' (Name) field is set to 'WOM'. The 'Исключить' (Exclude) checkbox is unchecked. The 'Происходит' (Occurs) dropdown is set to 'При получении сообщения' (On message received). The 'Тип сообщения' (Message type) dropdown is set to 'Object'. Under 'Осуществлять переход' (Perform transition), the 'При получении заданного сообщения' (On receiving the specified message) radio button is selected. The 'Сообщение' (Message) field contains 'Buy'. The 'Действие' (Action) field is empty. The 'Доп. условие' (Additional condition) field contains the code `randomTrue(AdoptionFraction)`.

Рисунок 30 – Изменение свойств нового параметра



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.

Сохраняем нашу модель и запускаем её (рисунок 31).

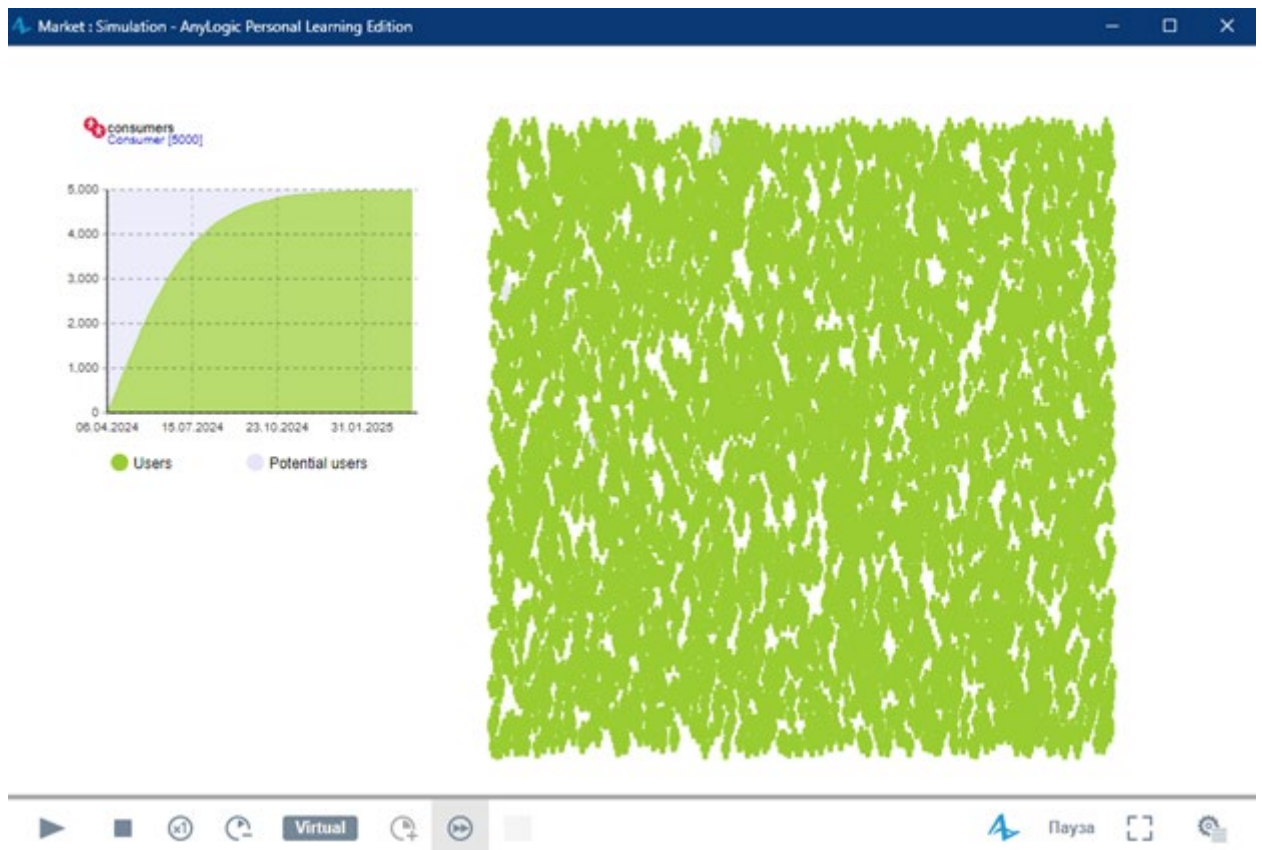


Рисунок 31 – Работа нашей модели

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



## 2.5 Учёт 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).

Откроем диаграмму агента «Consumer» и добавим на нее параметр «DiscardTime», укажем следующие изменения в нем, рисунок 32.

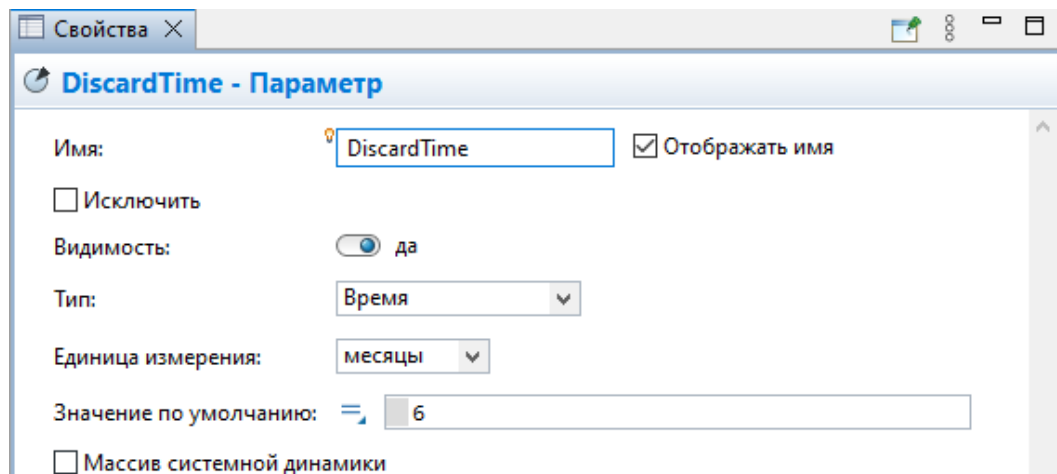


Рисунок 32 – Добавление нового параметра

Нарисуем переход из состояния User в состояние PotentialUser, чтобы промоделировать истечение срока службы товара. Назовем переход «Discard», пусть он срабатывает по таймауту, равному значению параметра «DiscardTime» (рисунок 33).

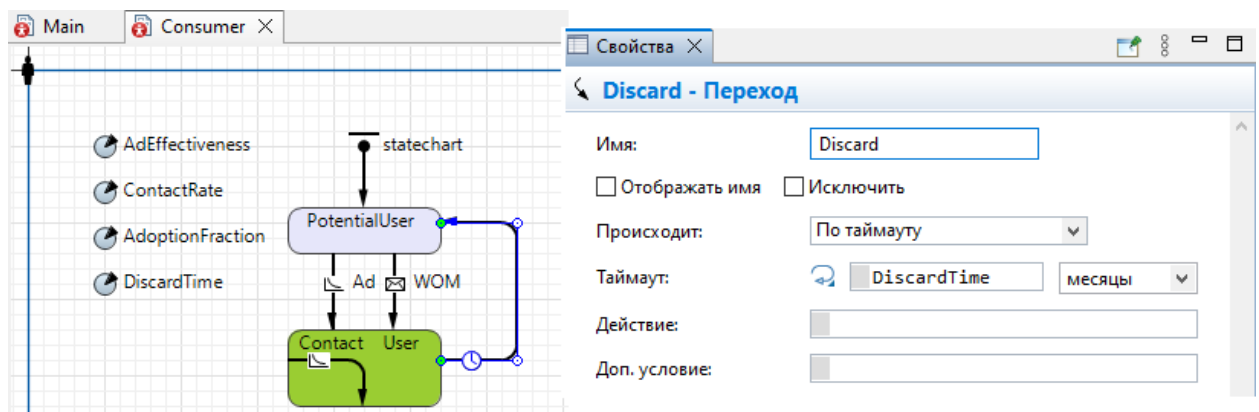


Рисунок 33 – Добавление нового перехода

Запускаем модель. Можем заметить изменение динамики продаж продукта: по прошествии определенного времени рост насыщения рынка сменится локальным спадом (рисунок 34).



Рисунок 34 – Результат моделирования

## 2.6 Учёт 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.

Подготовим место для нового состояния между состояниями PotentialUser и User, отсоединим состояние User от переходов и добавим новое «Состояние» из палитры «Диаграмма состояний» в середину диаграммы состояний потребителя, назовем его «WantsToBuy». Потребители в этом состоянии решили купить продукт, но продукт пока еще не приобрели

(рисунок 35). Изменим параметры нового состояния: цвет заливки – «gold», Действие при входе – «shapeBody.setFillColor(gold);».

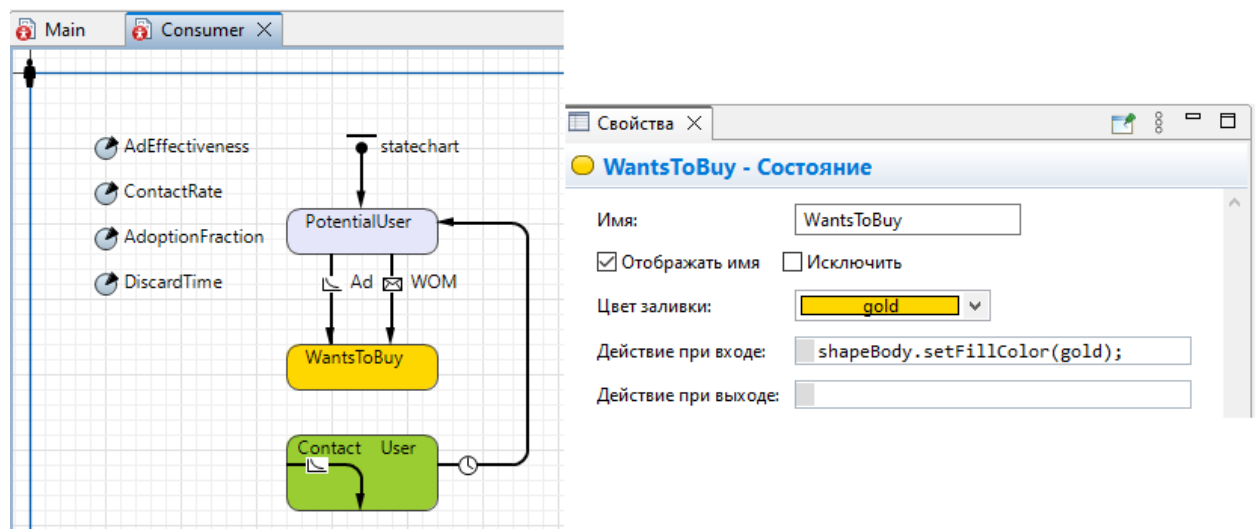


Рисунок 35 – Добавление нового состояния

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, задаем для перехода «Purchase» таймаут длительностью в 2 дня (рисунок 36).

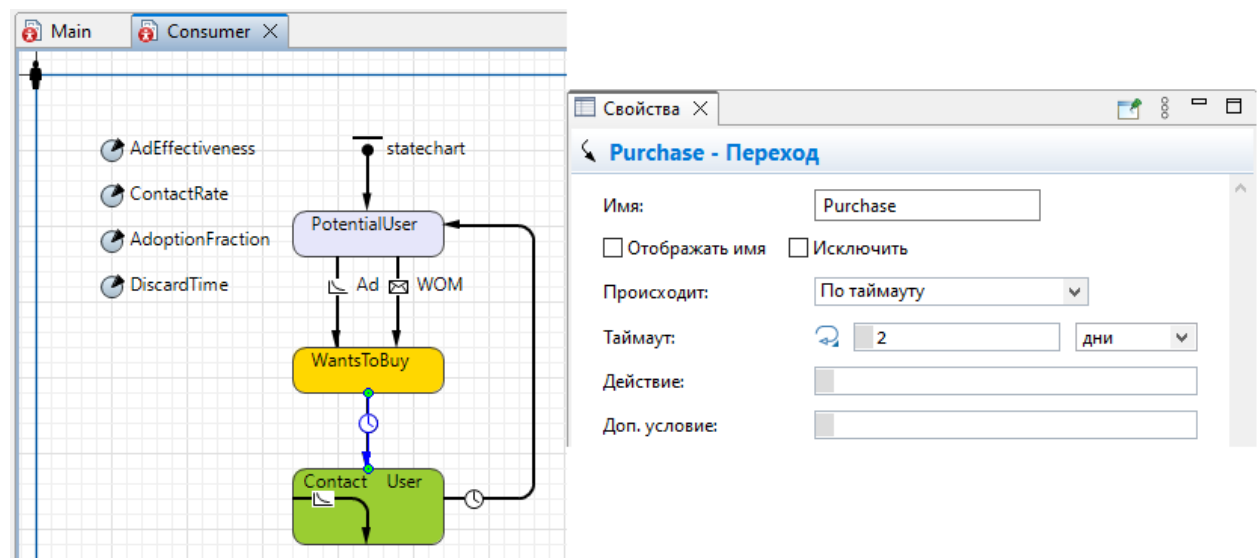


Рисунок 36 – Добавление нового перехода

Создадим еще одну функцию сбора статистики, чтобы вести учет заявок на приобретение товара (рисунок 37).

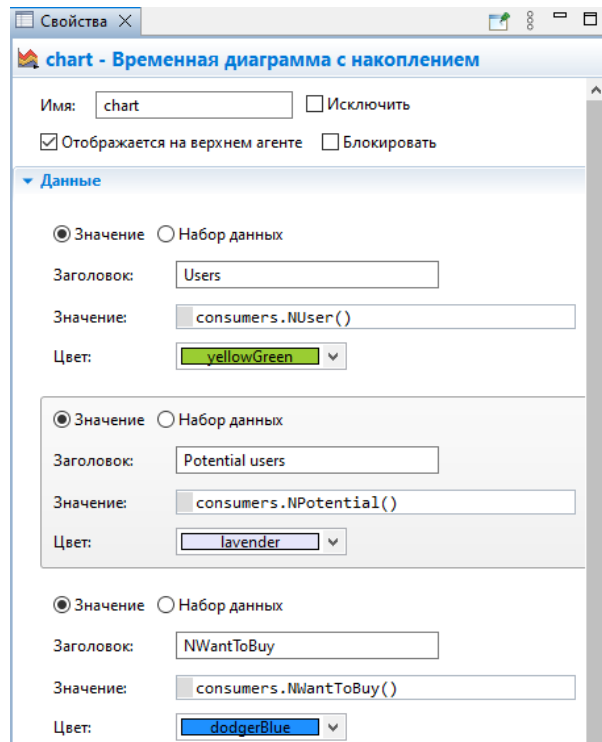


Рисунок 37 – Создание новой функции

Далее на нашем графике добавим еще один элемент данных для отображения и переместим его по середине нашего списка, рисунок 38.

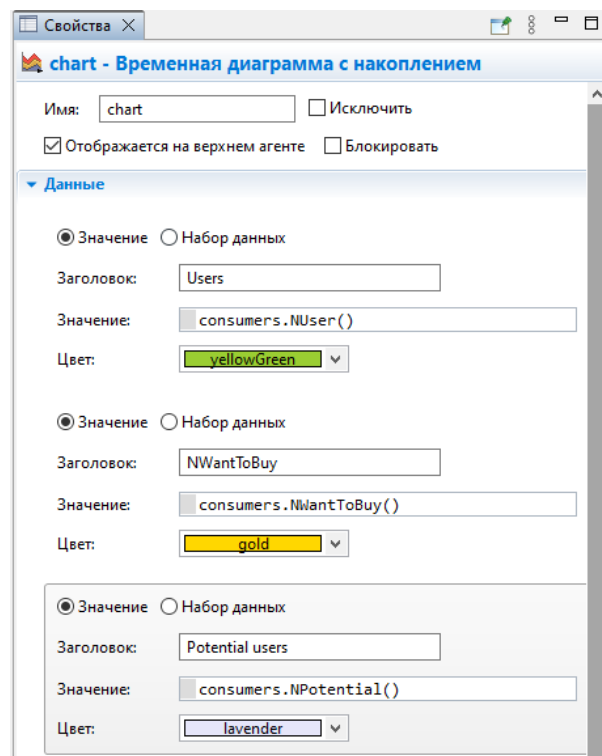


Рисунок 38 – Добавление данных на график

Запустим нашу модель, люди, ожидающие доставки товара, будут отображаться на графике и на анимации желтым цветом (рисунок 39).



Рисунок 39 – Результат запуска модели

## 2.7 Моделирование отказов от покупки

Теперь 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. 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.

Создадим два параметра (рисунок 40). Параметр «MaxWaitingTime» задает максимальное время, в течение которого потребитель готов ждать доставки продукта (в нашем случае - семь дней), другой параметр, «MaxDeliveryTime», задает максимально возможное время доставки товара. Зададим значение этого параметра равным 25 дням.

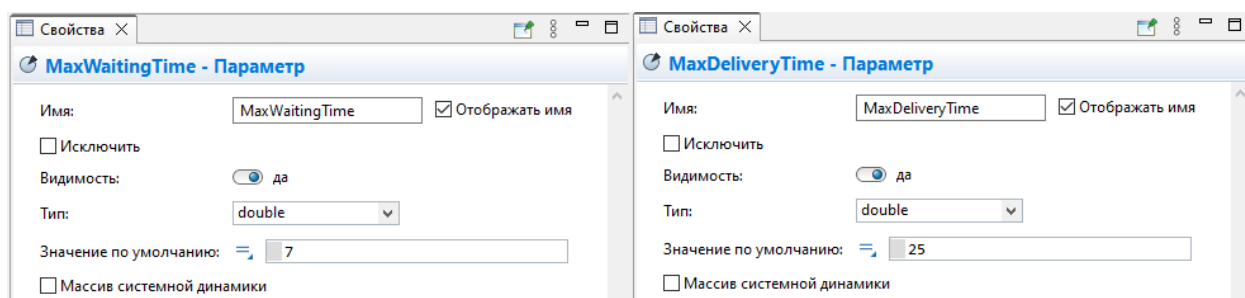


Рисунок 40 – Создание двух новых параметров

Таким образом, доставка товара может длиться от одного до 25 дней, в среднем же доставка занимает два дня. Изменим значение времени доставки с фиксированного периода, равного двум дням, на стохастическое выражение, которое использует вышеуказанный диапазон значений (рисунок 41).

Откроем диаграмму агента «Consumer» и выделим переход «Purchase». Мы хотим изменить значение таймаута, по которому срабатывает переход. Затем, выделив значение «таймаута», нажимаем на «Выберете распределение вероятностей...». В данном окне вводим следующие значения:

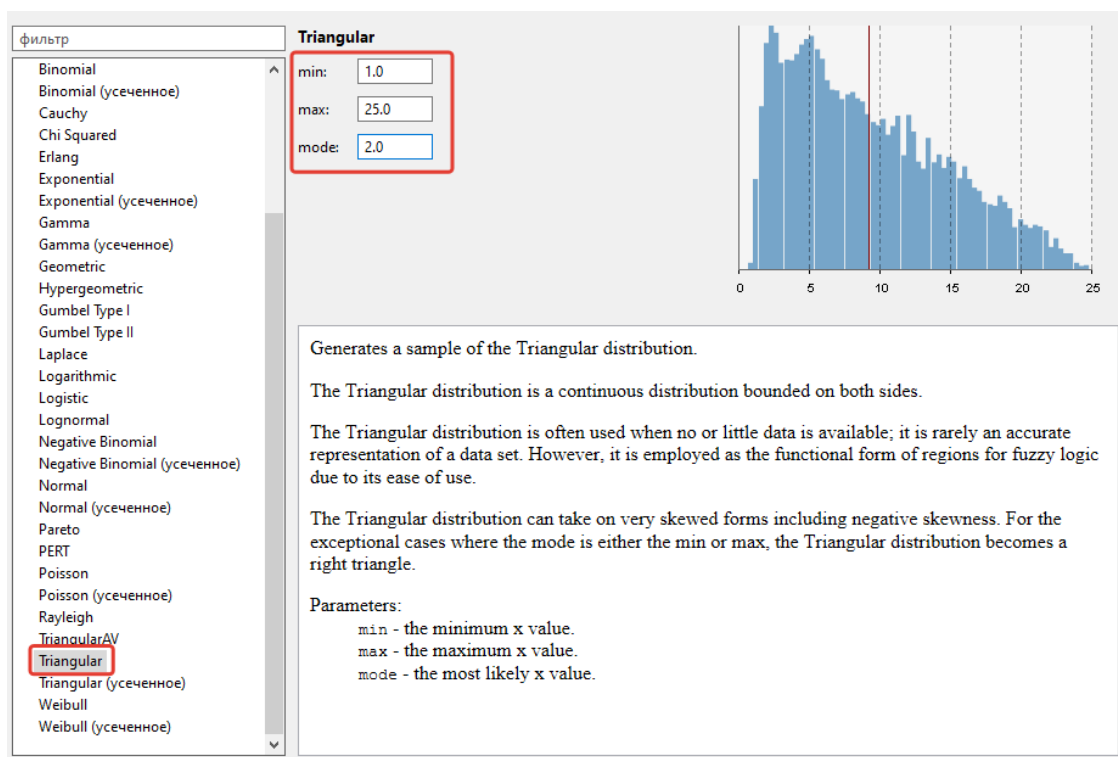


Рисунок 41 – Заполнение значений

В поле задания значения таймаута будет автоматически вставлено выражение «`triangular(1, 25, 2)`». Изменим эту строку на выражение «`triangular(1, main.MaxDeliveryTime, 2)`».

Нарисуем 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.

Помимо этого, изменим следующие параметры, рисунок 42.

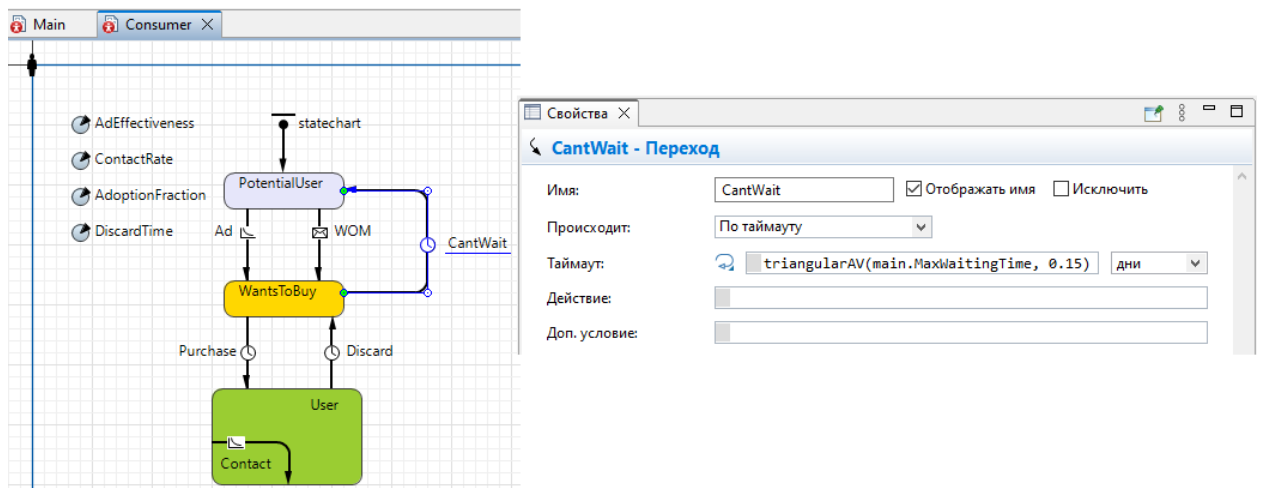


Рисунок 42 – Добавление нового перехода

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.

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.

Вернемся на диаграмму «Main». Откроем палитру «Элементы управления», перетащите элемент «Бегунок» на диаграмму и расположим его под графиком (рисунок 43).

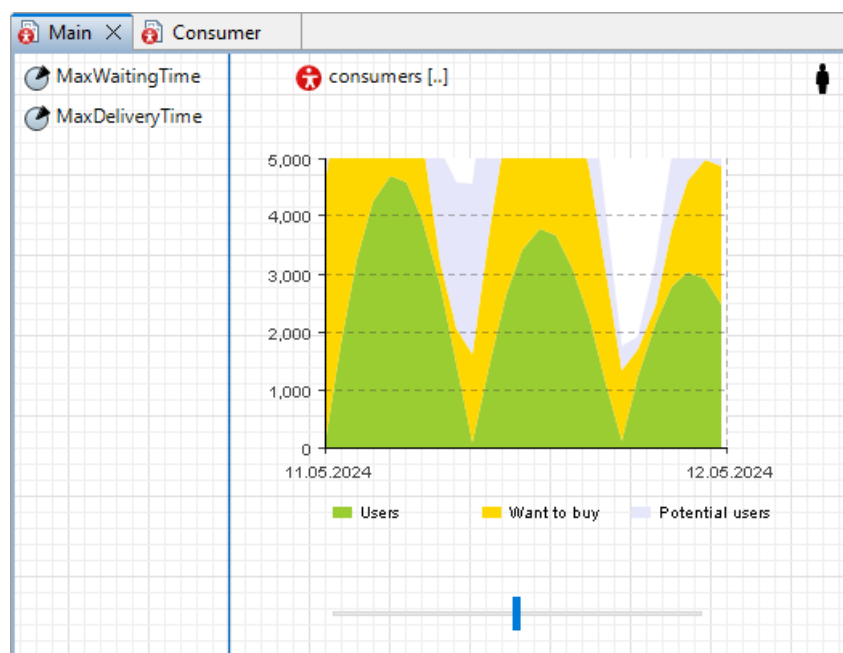


Рисунок 43 – Добавление элемента «Бегунок»



Затем изменим у него следующие параметры, рисунок 44.

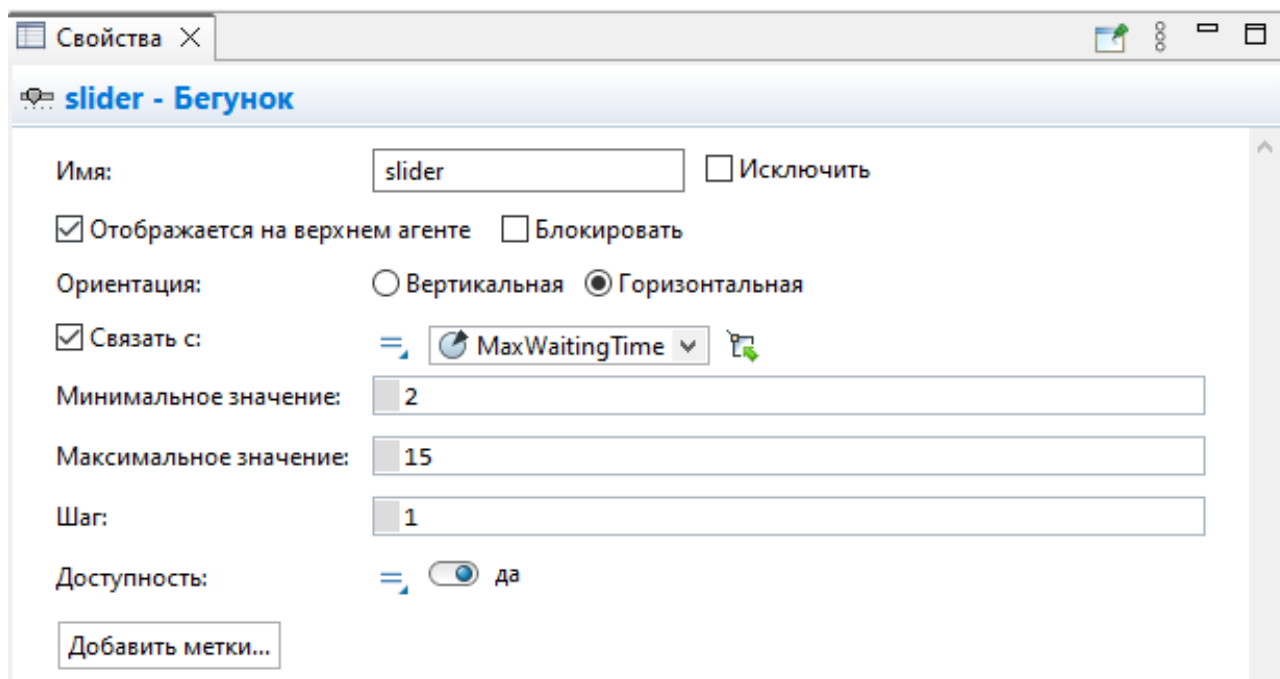


Рисунок 44 – Добавление параметров

Добавим еще один бегунок под предыдущим и настройте его следующим образом, рисунок 45.

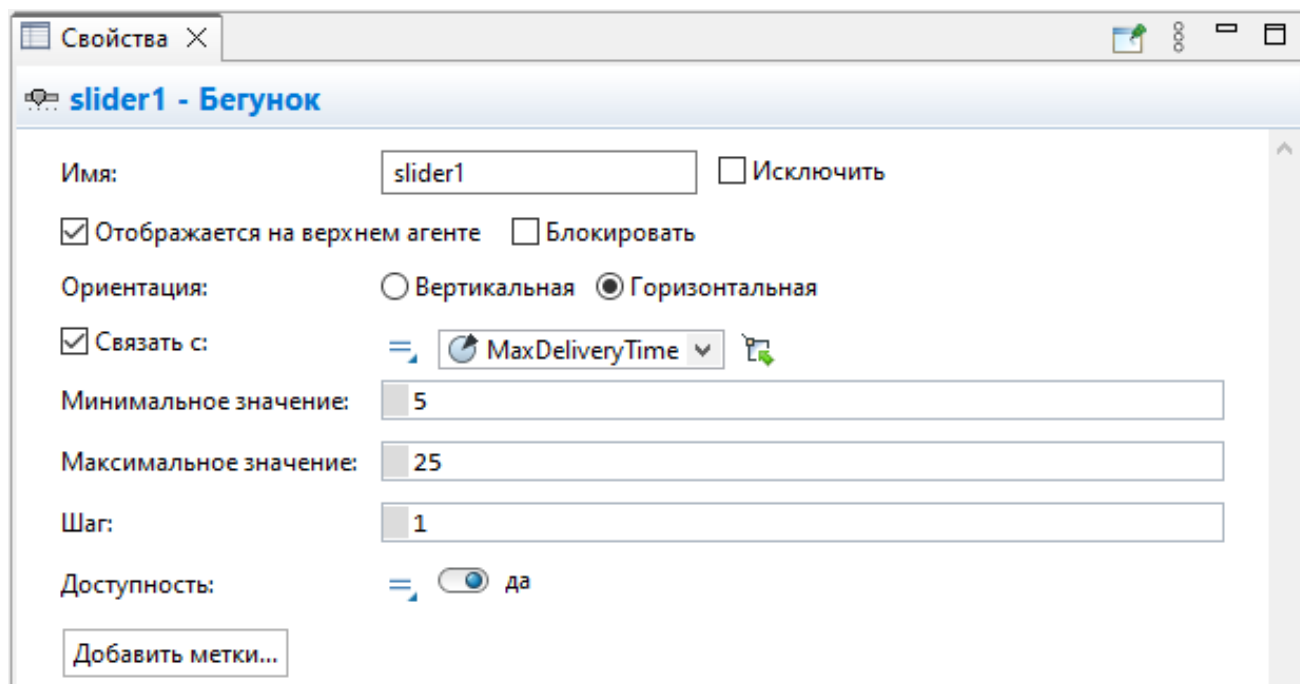


Рисунок 45 – Добавление второго элемента «ползунок» и добавление параметров



После добавим подписи, для этого перенесем две фигуры «Текст», рисунок 46.

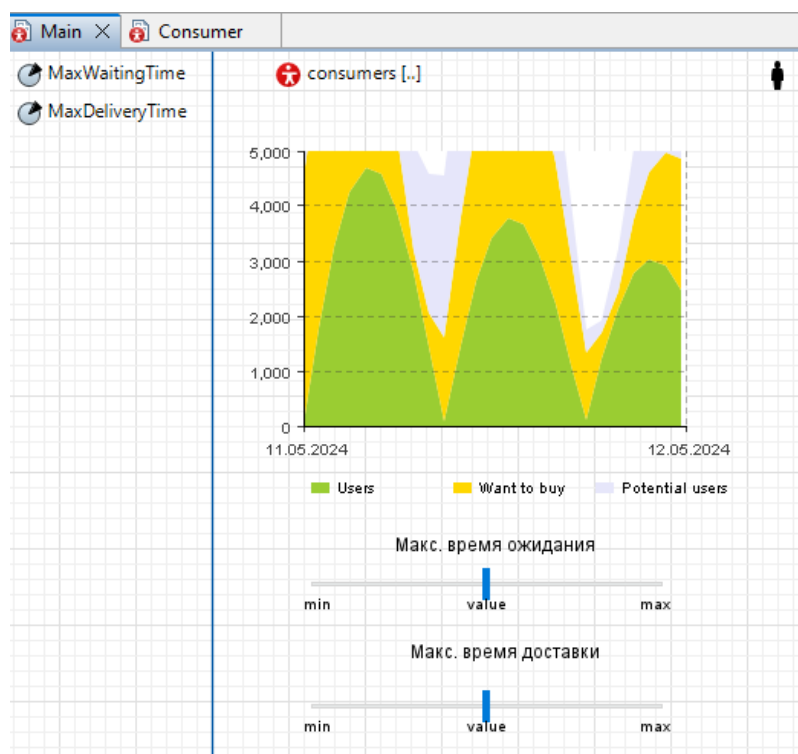


Рисунок 46 – Добавление подписей

Запустим модель и понаблюдаем за ее поведением. Изменяя максимальное время ожидания и максимальное время доставки, можем оценить влияние этих изменений на поведение потребителей и состояние рынка (рисунок 47).

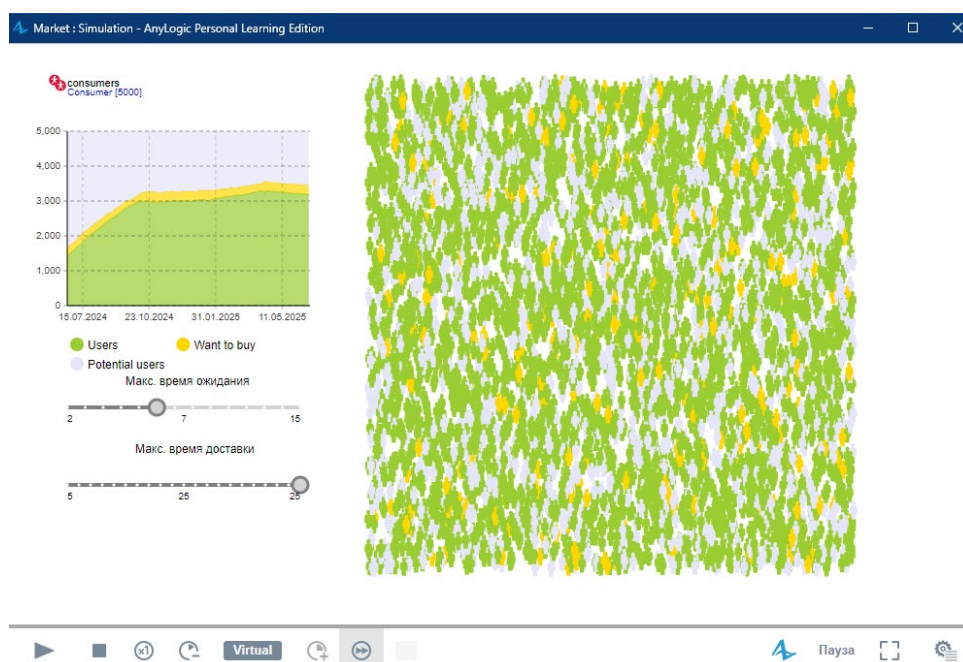


Рисунок 47 – Результат работы модели

### 3. Модель распространения эпидемии

Мы построим модель, изучающую распространение инфекционного заболевания среди населения. Давайте рассмотрим численность населения, равную 10 000 человек, которую обозначим как «TotalPopulation». Вначале заражен только один человек, а все остальные лишь восприимчивы к болезни.

- Во время болезни один человек в среднем контактирует с другими с интенсивностью «ContactRateInfectious», равной 1.25 человека в день. Если заразившийся человек контактирует с восприимчивым к болезни, то вероятность передачи инфекции «Infectivity» равняется 0.6.
- После того, как человек заражается, инкубационный период «AverageIncubationTime» длится 10 дней.
- Средняя длительность болезни после инкубационного периода «AverageIllnessDuration» (другими словами, длительность периода, когда этот человек может заражать других) составляет 15 дней.
- Выздоровевшие люди получают иммунитет к болезни и не могут снова заболеть.

#### 3.1 Создание диаграммы потоков и накоплений

Создадим новую модель, назвав её «SEIR» и выберем дни в качестве единиц модельного времени.

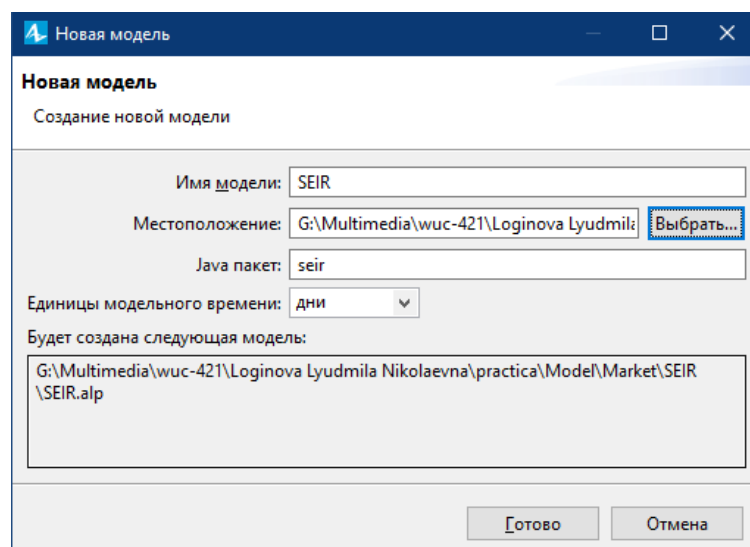


Рисунок 48 – Создание модели

Начнем с того, что нарисуем диаграмму накопителей и потоков. В данной модели мы не будем учитывать все разнообразие населения, а лишь выделим четыре категории людей, имеющие значение для изучаемого нами процесса:

- Susceptible – Восприимчивые к заражению люди, которые еще не были заражены вирусом.
- Exposed – Люди, находящиеся в латентной стадии заражения (они уже заражены, но еще не могут заражать других).
- Infectious – Люди в активной стадии заражения (они могут заражать других людей).
- Recovered – Выздоровевшие люди (они приобрели иммунитет к данному заболеванию).

Откроем палитру «Системная Динамика». Перетащим элемент «Накопитель» из палитры «Системная динамика» на диаграмму «Main» (рисунок 49).

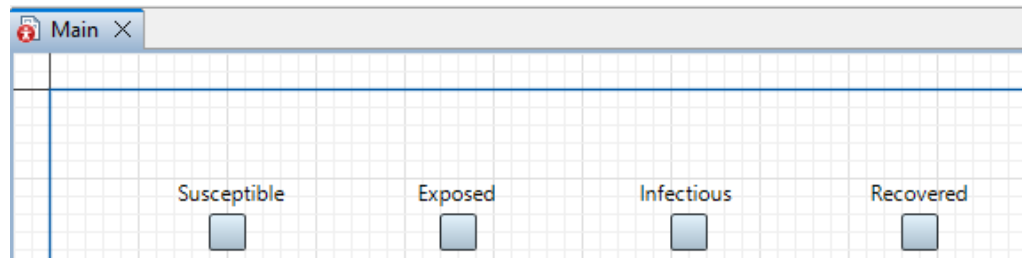


Рисунок 49 – Добавление новых накопителей

Основная логика нашей модели такова: восприимчивые к заболеванию люди подвергаются заражению вирусом, болеют и заражают других, а затем выздоравливают. Чтобы промоделировать перемещение людей между нашими четырьмя накопителями, нам нужно добавить три потока.

Добавим первый поток, который ведет из накопителя «Susceptible» в накопитель «Exposed», и сразу переименуем его в «ExposedRate» (рисунок 50).

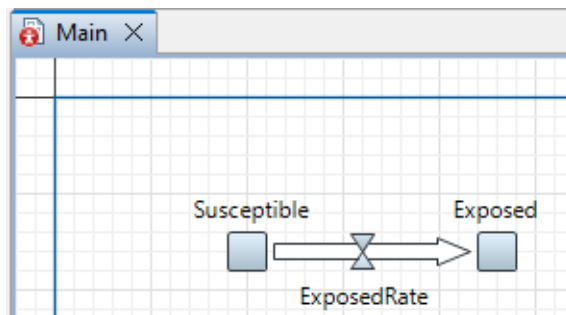


Рисунок 50 – Добавление первого потока

Добавим новые потоки к оставшемуся накопителю (рисунок 51).

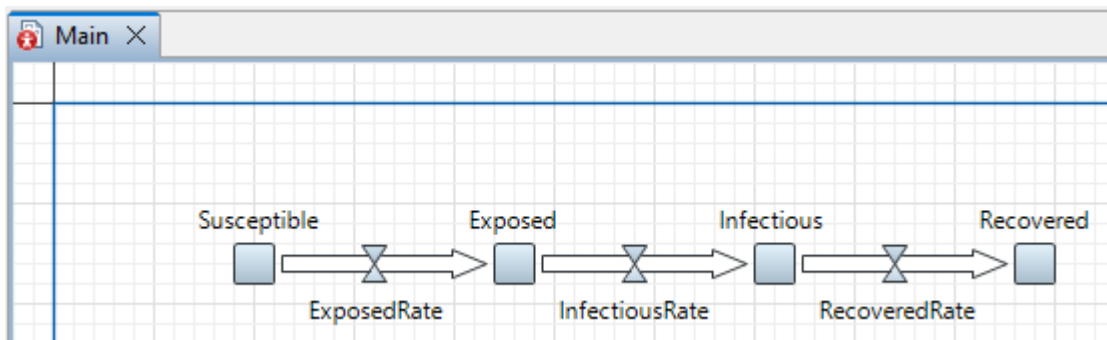


Рисунок 51 – Добавление новых потоков

Теперь зададим параметры и зависимости. Добавим пять элементов «Параметр» (рисунок 52), зададим их имена и значения по умолчанию:

- TotalPopulation = 10 000
- Infectivity = 0.6
- ContactRateInfectious = 1.25
- AverageIncubationTime = 10
- AverageIllnessDuration = 15

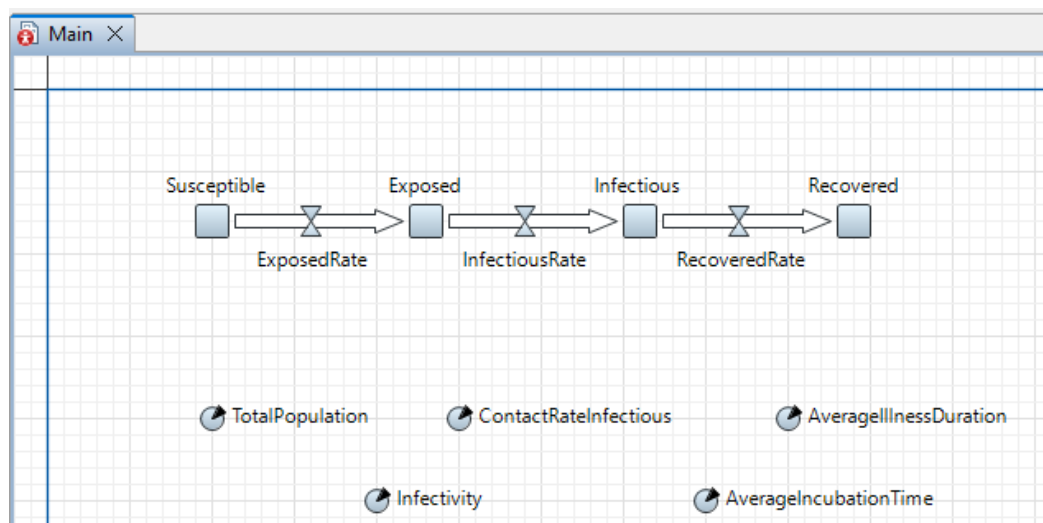


Рисунок 52 – Добавление параметров

Зададим первоначальное количество инфицированных людей, указав значение 1 в качестве «Начального значения» накопителя «Infectious».

Зададим «Начальное значение» накопителя «Susceptible» «TotalPopulation-1».

Нарисуем связь, ведущую из параметра «TotalPopulation» в накопитель «Susceptible» (рисунок 53).

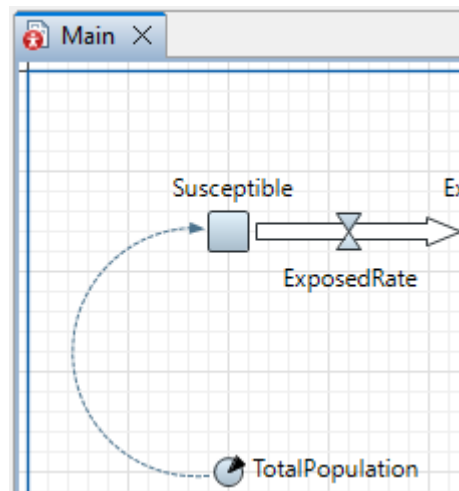


Рисунок 53 – Создание новой связи

Зададим формулу потока «ExposedRate»:  
 $\text{Infectious} * \text{ContactRateInfectious} * \text{Infectivity} * \text{Susceptible} / \text{TotalPopulation}$ ,  
после чего нам необходимо будет нарисовать связи зависимостей, ведущие от указанных в формуле переменных и параметров к этому потоку (рисунок 54).

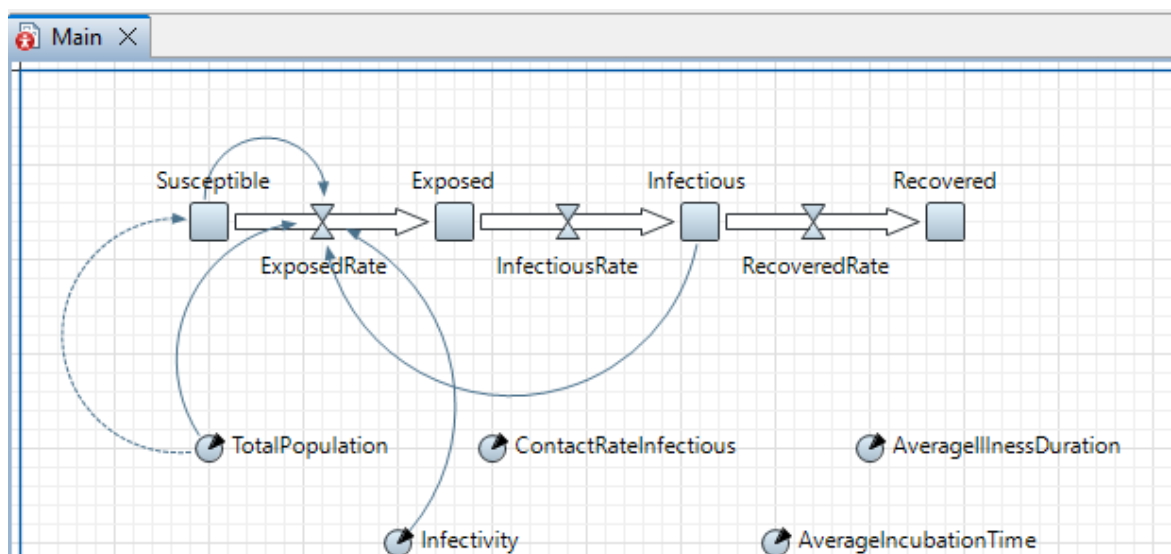


Рисунок 54 – Добавление связей, которые были указаны в формуле

Необходимо задать следующие формулы для потока «InfectiousRate»: «Exposed/AverageIncubationTime», для потока «RecoveredRate»: «Infectious/AverageIllnessDuration», и добавить недостающие связи зависимостей, которые можно выровнить по своему вкусу (рисунок 55).

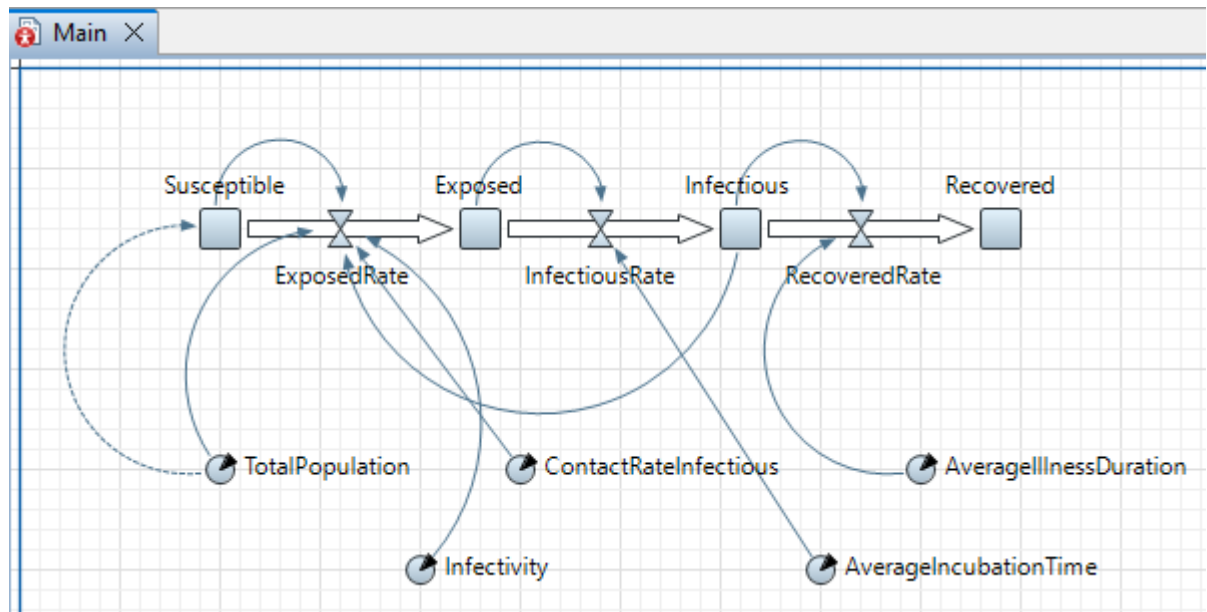


Рисунок 55 – Полученная модель

Запустим модель и исследуем динамику процесса с помощью похожих на виджеты информационных окон этих переменных (рисунок 56).

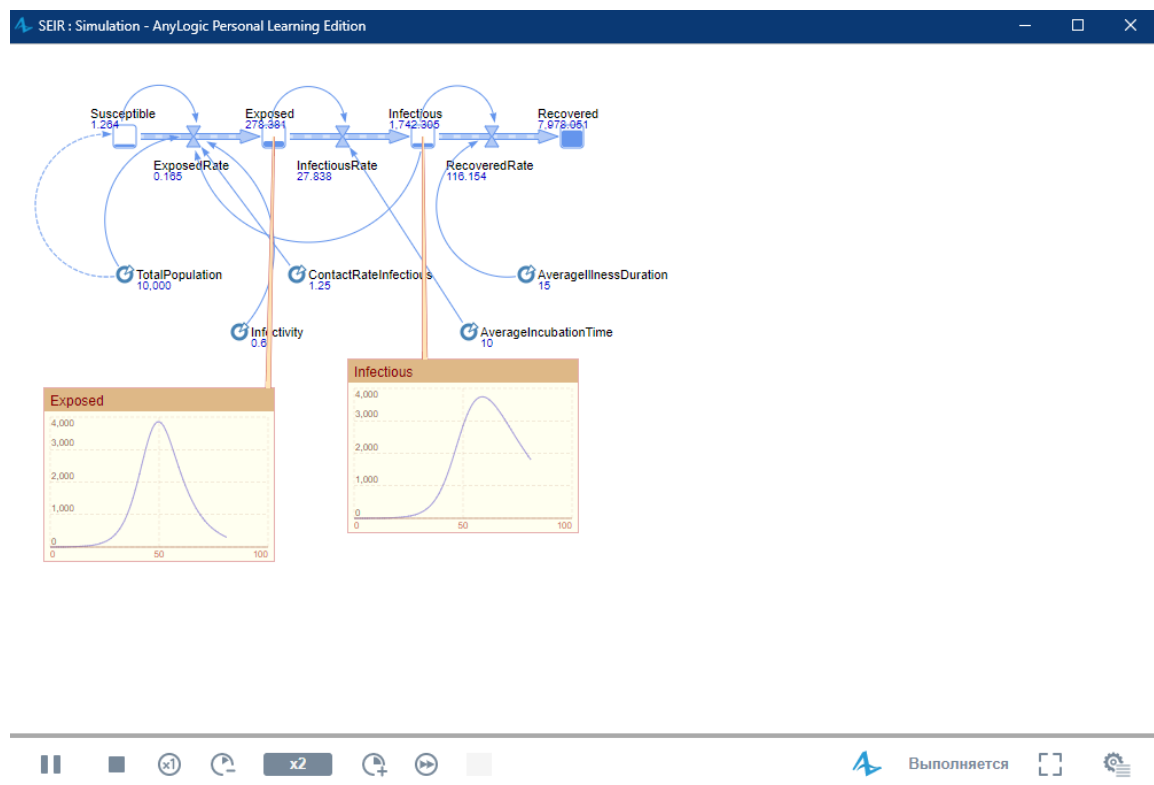


Рисунок 56 – Результат работы модели

### 3.2 Добавление графика для визуализации динамики процесса

Перетащим элемент «Цикл» из палитры «Системная динамика» на диаграмму.

Перейдем в панель «Свойства» и изменим «Тип цикла» на R. Оставим заданное по умолчанию «Направление: по часовой стрелке» и укажем текст, который AnyLogic будет отображать возле значка цикла: Contagion (рисунок 57).

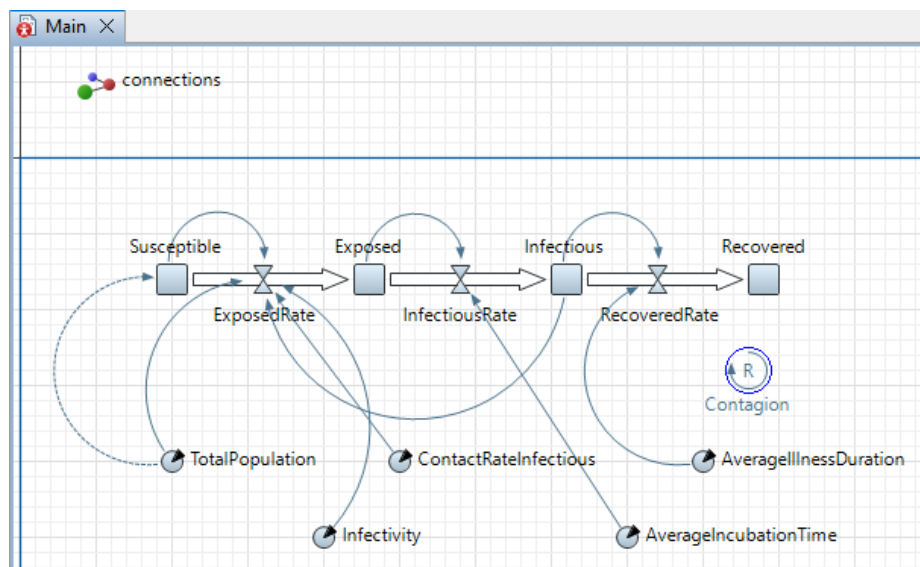


Рисунок 57 – Добавление цикла

Перетащим элемент «Временной график» из палитры «Статистика» на диаграмму и увеличим размер графика (рисунок 58).

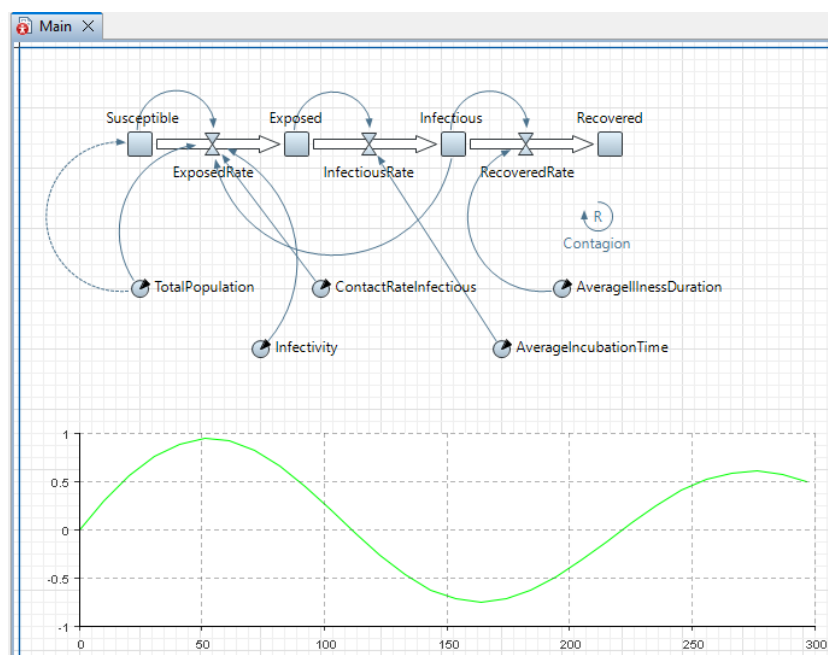


Рисунок 58 – Добавление графика

Изменим свойства элементов данных, созданного для графика по умолчанию, рисунок 59.

The screenshot shows a software window titled "plot - Временной график" with a "Свойства" (Properties) tab. It contains four sections for configuring data series, each with radio buttons for "Значение" (Value) and "Набор данных" (Data Set), and fields for title, value, marker style, line thickness, and color.

Series Name	Value	Color
Susceptible people	Susceptible	dodgerBlue
Exposed people	Exposed	darkOrange
Infectious people	Infectious	magenta
Recovered people	Recovered	lime

Рисунок 59 – Актуальные свойства графика



Установили «Временной диапазон» на «300 единиц мод. времени» и запустили нашу модель (рисунок 60).

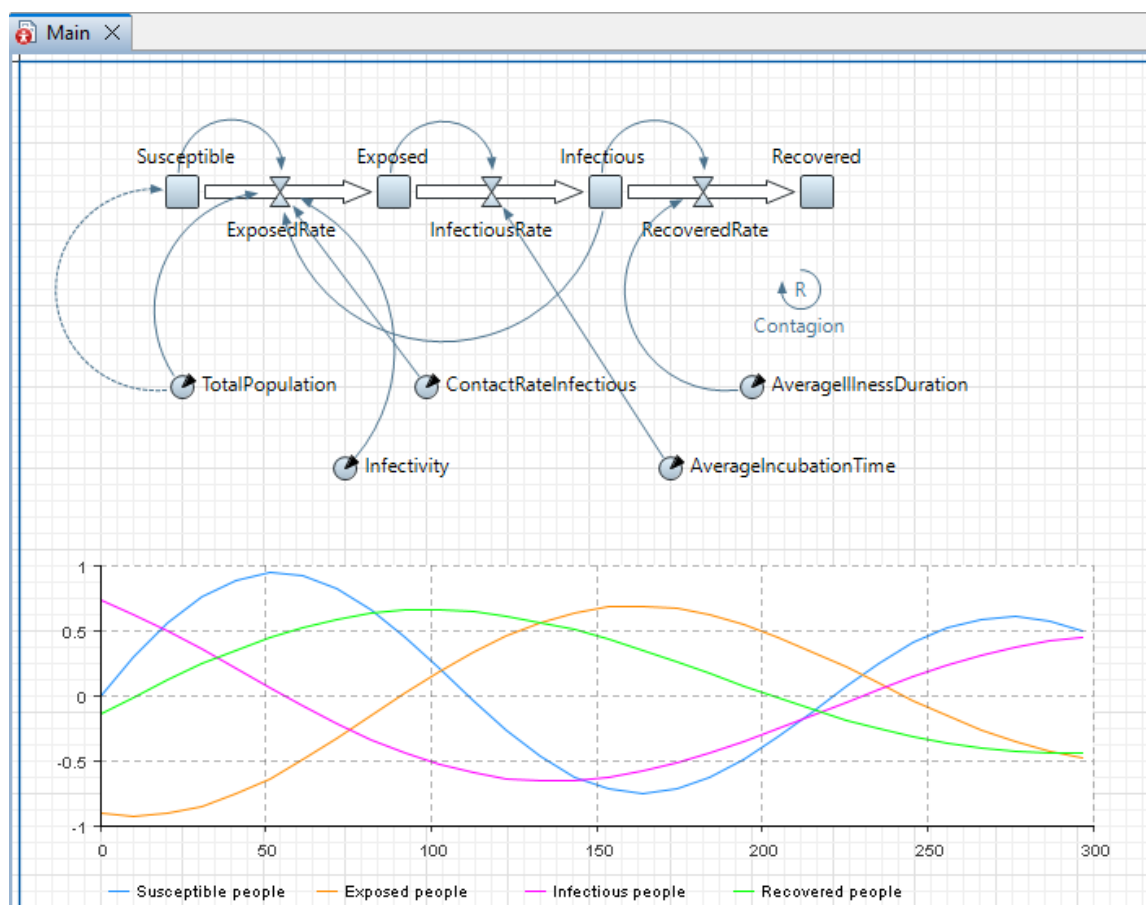


Рисунок 60 – Результат работы модели

### 3.3 Эксперимент варьирования параметров

Теперь изучим, как меняется динамика распространения эпидемии при различных значениях интенсивности контактов между людьми, воспользовавшись экспериментом варьирования параметров. Мы запустим этот эксперимент в облачном сервисе («облаке») AnyLogic Cloud.

Открываем «Конфигурация запуска». Здесь можно указать входные и выходные данные модели перед тем, как загрузить ее в облако AnyLogic. Выберем все параметры из списка «Входные данные» и «Выходные данные» (рисунок 61).

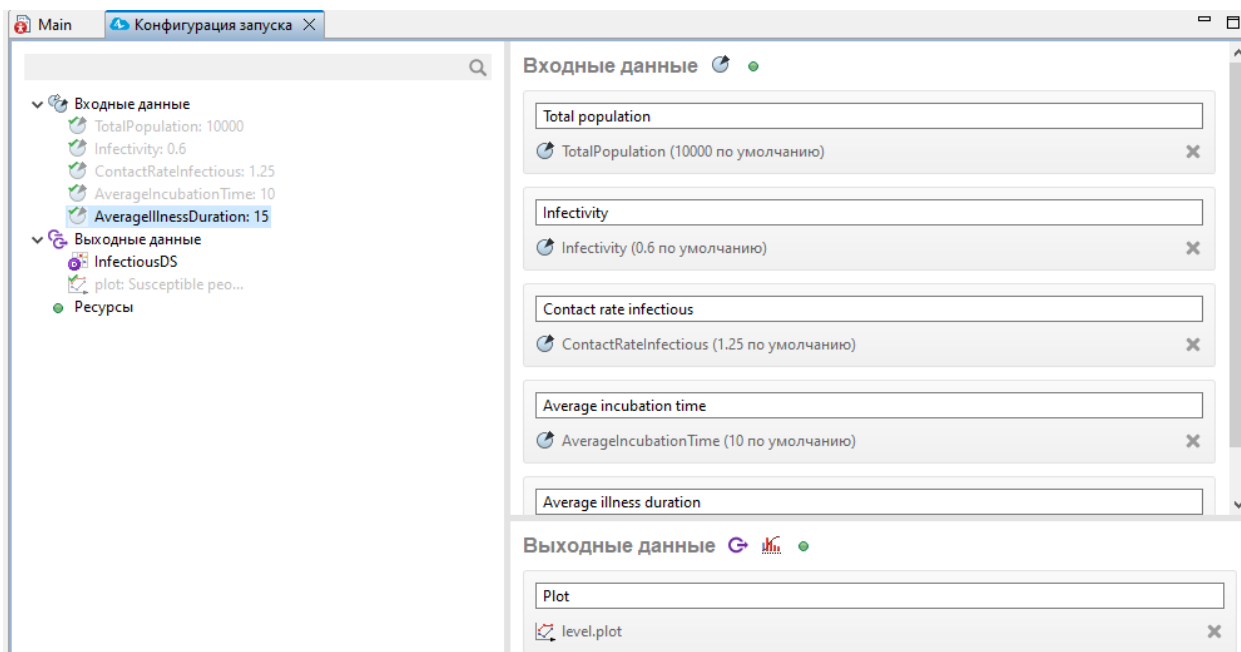


Рисунок 61 – Заполнение конфигурации запуска

Необходимо настроить эксперимент так, чтобы он моделировал ровно 300 дней (рисунок 62).

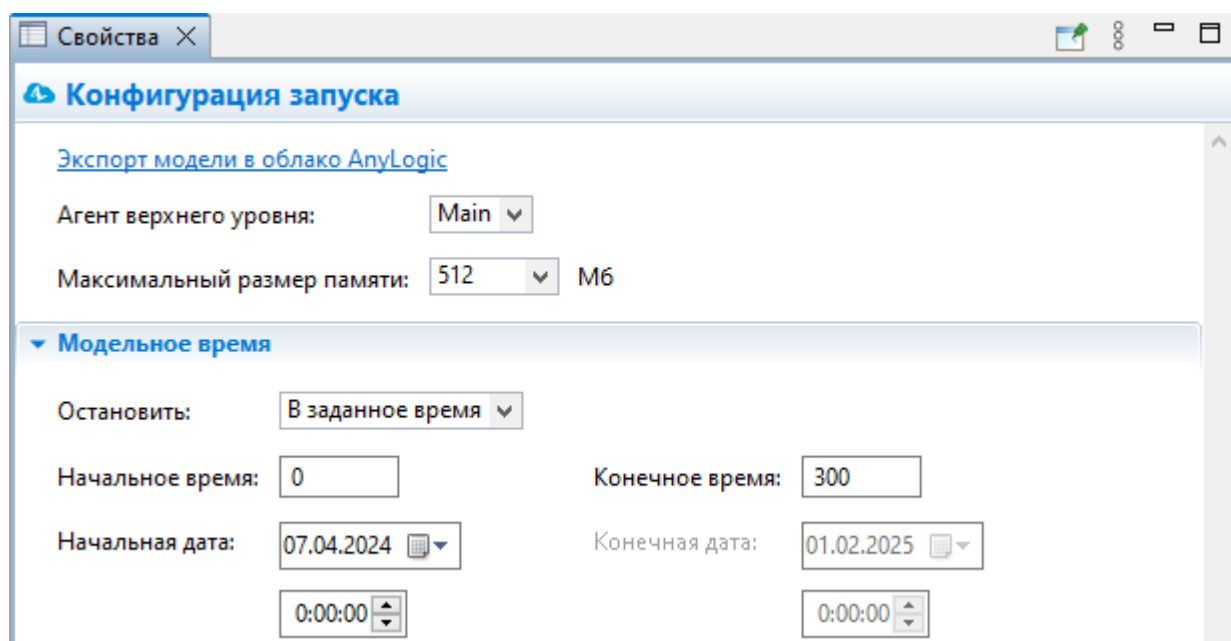


Рисунок 62 – Моделирование времени

Экспортируем нашу модель в «облако AnyLogic», проходим регистрацию и нас перенаправляет в браузер (рисунок 63).

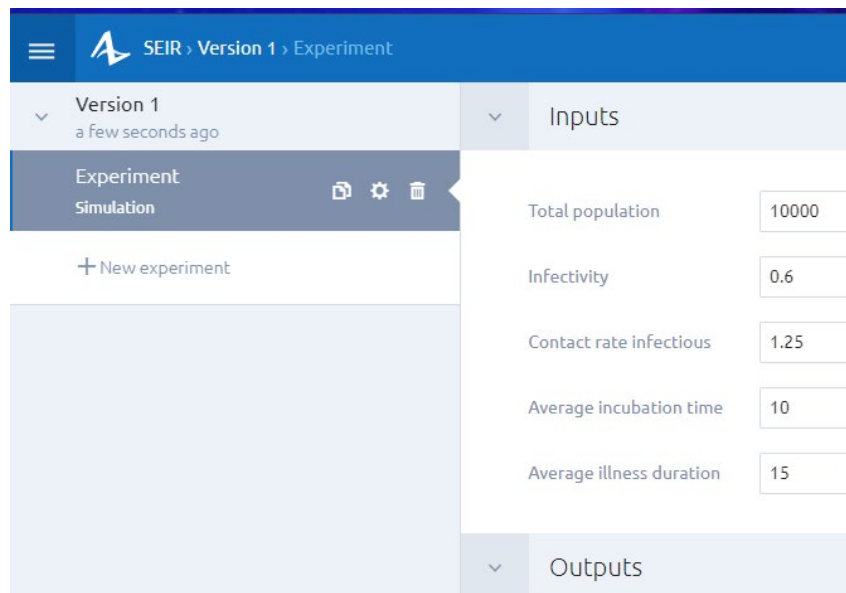


Рисунок 63 – Облако AnyLogic

После запуска модели в облаке, появится такой же график, который мы получили ранее (рисунок 64).

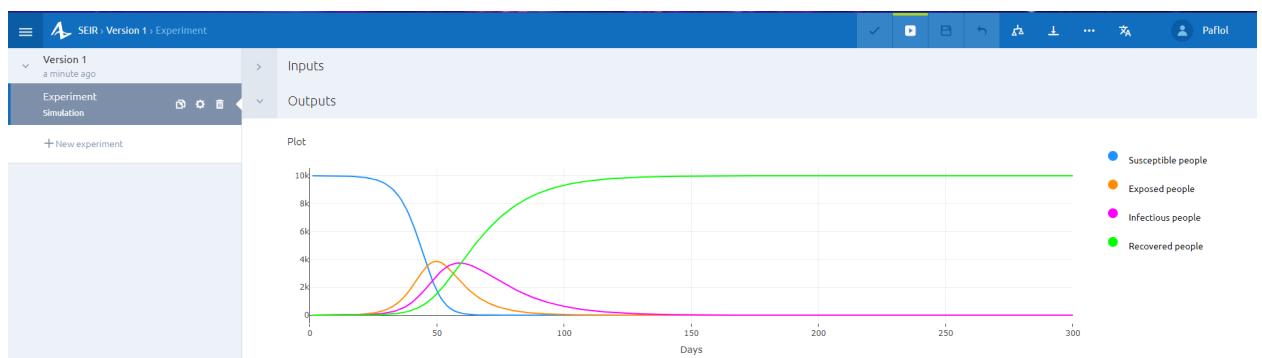


Рисунок 64 – Полученный результат

Создадим другой эксперимент в Cloud (рисунок 65).

The screenshot shows the 'New experiment' dialog box. It has a title bar with a close button (X). The main content area contains the following fields and options:

- Version «Version 1»**: A dropdown menu showing the selected version.
- Experiment name**: A text input field containing 'ContactRateInfectious'.
- Experiment type**: A dropdown menu showing 'Variation with replications'.
- Description**: A text area containing 'A Variation experiment with replications performed for each iteration.'

At the bottom of the dialog, there are two buttons: 'Cancel' and 'Add'.

Рисунок 65 – Создание нового эксперимента

Найдем параметр «Contact rate infectious» и изменим его тип на «Варьировать в диапазоне в выпадающем списке».

Укажем 0.3 в качестве минимального значения параметра, а 2 — в качестве максимального. В поле step введем шаг: 0.1.

Мы готовы запустить эксперимент и понаблюдать за изменением динамики распространения заболевания на нескольких прогонах с помощью графиков.

Эксперимент варьирования параметров произведет несколько прогонов модели с отличающимися значениями параметра «Contact rate infectious» и выведет результаты моделирования на графиках (рисунок 66).

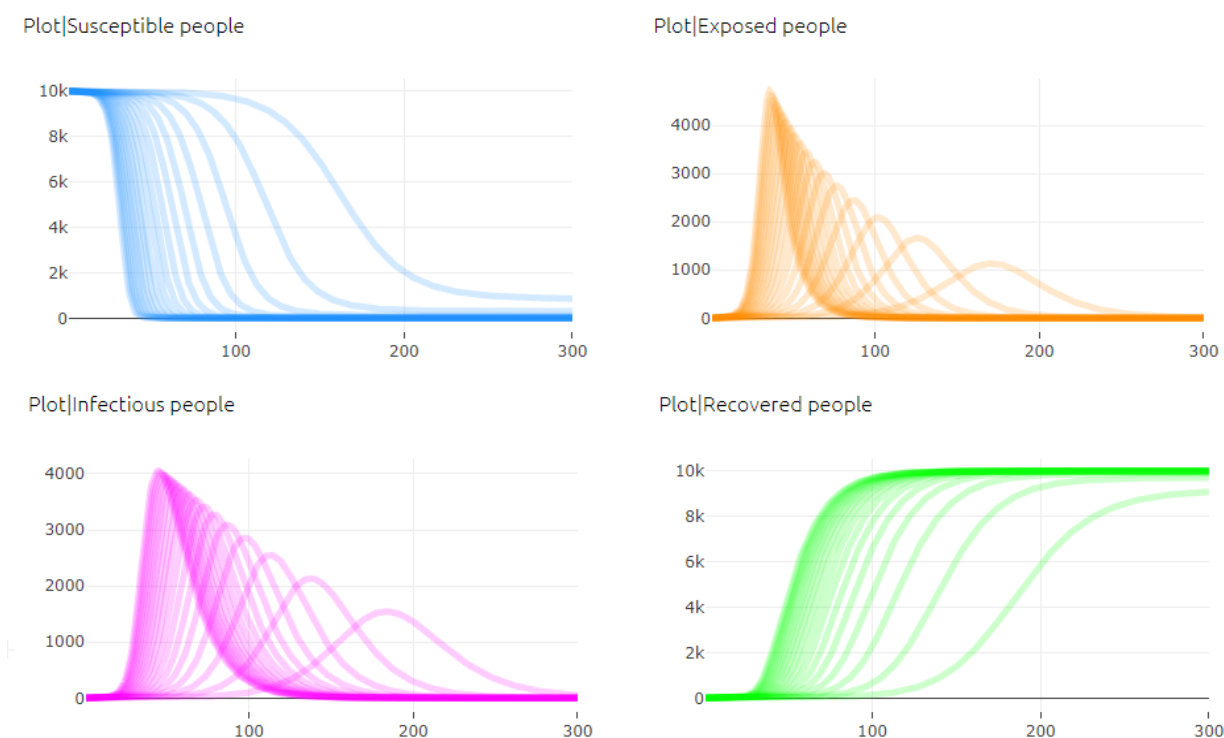


Рисунок 66 – Результат моделирования

Каждый график включает результаты нескольких прогонов (по одной кривой на запуск) — всего 18. Другими словами, мы видим 18 сценариев заболеваемости для разных показателей интенсивности контактов, варьирующихся от 0.3 до 2. Эти сценарии отражают 18 шагов в рамках диапазона значений для параметра, который мы задали ранее.

## **Заключение**

Было освоено программное обеспечение «AnyLogic» на основе двух моделей. Данные модели являются мощными инструментами для изучения сложных систем. Модель потребительского рынка может помочь компаниям оптимизировать свои маркетинговые стратегии, а модель распространения эпидемии может помочь органам здравоохранения разработать эффективные меры по контролю заболеваний.

### **Список использованных источников.**

1. Григорьев Илья. Anylogic за три дня. – 2023. – С. 5-122.
2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.