



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ
Кафедра программных систем

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Лабораторная работа

Моделирование непрерывных
событийно-стохастических систем
в AnyLogic

Методические указания

Самара

2025

УДК 681.3.066

ББК 32.973.26-018.2

Баландин А.В. Моделирование непрерывных событийно-стохастических систем в AnyLogic. Методические указания. - Самар. ун-т. Самара, 2025. 53 с.

Методические указания предназначены для выполнения лабораторной работы по освоению специализированных средств программной системы AnyLogic для имитационного моделирования непрерывных событийно-стохастических систем (процессов обслуживания), а также средств сбора статистики и статистической обработки результатов компьютерных экспериментов с моделью. Приведены индивидуальные варианты настройки свойств элементов разработанной модели, для которых требуется оценить характеристики процессов обслуживания.

Методические указания предназначены учащимся направления 02.03.02 – Фундаментальная информатика и информационные технологии, изучающим дисциплину «Моделирование информационных процессов и систем».

ОГЛАВЛЕНИЕ

НЕПРЕРЫВНЫЕ СОБЫТИЙНО-СТОХАСТИЧЕСКИЕ СИСТЕМЫ	4
ПРОСТЕЙШИЙ ПРОЦЕСС ОБСЛУЖИВАНИЯ	5
ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ОБСЛУЖИВАНИЯ	8
<i>Библиотека моделирования процессов.....</i>	<i>8</i>
<i>Средства сбора статистики и статистического анализа</i>	<i>12</i>
ЛАБОРАТОРНАЯ РАБОТА	18
ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ.....	18
ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	18
УПРАЖНЕНИЕ 1	19
<i>Создание диаграммы.....</i>	<i>19</i>
<i>Запуск модели</i>	<i>21</i>
УПРАЖНЕНИЕ 2.....	24
<i>Сбор и анализ в модели статистических данных.....</i>	<i>24</i>
УПРАЖНЕНИЕ 3.....	29
<i>Моделирование абстрактного ресурса.....</i>	<i>29</i>
<i>Моделирование параллельного процесса обслуживания.....</i>	<i>30</i>
<i>Презентация процесса параллельного обслуживания</i>	<i>32</i>
УПРАЖНЕНИЕ 4.....	33
<i>Создание нестандартных типов агентов.....</i>	<i>33</i>
<i>Создание нестандартных типов ресурсов.....</i>	<i>34</i>
<i>Ссылка на параметры агентов созданных типов</i>	<i>35</i>
УПРАЖНЕНИЕ 5.....	36
<i>Вычисление характеристик процесса обслуживания.....</i>	<i>36</i>
<i>Сбор статистики времени обслуживания клиента</i>	<i>37</i>
<i>Презентация эксперимента модели с гистограммой.....</i>	<i>40</i>
<i>Оценка вероятности отказа в обслуживании</i>	<i>41</i>
<i>Оценка абсолютной пропускной способности банковского отделения</i>	<i>44</i>
<i>Оценка относительной пропускной способности банковского отделения.....</i>	<i>45</i>
ОЦЕНКА АДЕКВАТНОСТИ ИТОГОВОЙ МОДЕЛИ	46
ОЦЕНКА ХАРАКТЕРИСТИК ЗАДАННОГО ВАРИАНТА ПРОЦЕССА ОБСЛУЖИВАНИЯ.....	48
ОФОРМЛЕНИЕ РЕЗУЛЬТАТОВ РАБОТЫ	48
ОЦЕНОЧНЫЕ ВОПРОСЫ.....	49
ПРИЛОЖЕНИЕ 1	51
ПРИЛОЖЕНИЕ 2	53

Непрерывные событийно-стохастические системы

Предмет моделирования S можно формализовать в виде непрерывной событийно-стохастической системы, если его можно представить как стохастический конечный автомат, функционирующий в непрерывном времени - $M_T(t_0, \delta t)$, который в случайные моменты непрерывного времени случайным образом мгновенно переходит из текущего состояния в некоторое другое состояние в результате возникновения контролируемых во времени случайных событий из конечного множества возможных случайных событий, и остаётся в новом состоянии до возникновения в случайный момент непрерывного времени очередного случайного события. При этом с увеличением значения непрерывного времени уменьшается вероятность сохранить стохастическому конечному автомату в текущий момент времени текущее состояние, а вероятность случайного перехода автомата в текущий момент времени из текущего состояния в одно из возможных других состояний увеличиваются, "заставляя" его покинуть текущее состояние.

Значительную долю непрерывных событийно-стохастических систем составляют системы, которые по своей сути реализуют процесс обслуживания некоторых абстрактных заявок на обслуживание. Системы обслуживания функционируют во времени непрерывно, а изменение состояния системы связывают с событиями, характерными для систем обслуживания (поступление заявки на обслуживание, начало обработки заявки, завершение обработки, завершение обслуживания заявки системой). Для систем обслуживания характерным является случайное появление во времени заявок на обслуживание, и завершение обслуживания заявок в случайные моменты времени. А основными оценками эффективности системы обслуживания являются время обслуживания заявок и её пропускная способность.

Так как реализация процессов обслуживания является характерной чертой информационных систем, то выделяемые при их моделировании отношения между параметрами зачастую могут быть интерпретированы как отношения

обслуживания (например, отношение, связывающее обслуживание потока заявок от удалённых терминалов на обработку приложениями в узлах компьютерной сети). Для анализа характеристик информационных систем как сложных систем обслуживания на практике используются методы и средства имитационного моделирования .

Простейший процесс обслуживания

При моделировании отношения $r(p_1, p_2, \dots, p_{n_r})$ как сложной системы процессов обслуживания полагают, что оно может быть структурно представлено в виде взаимосвязанных по входу/выходу приборов, реализующих простейшие процессы обслуживания (Рисунок 1).

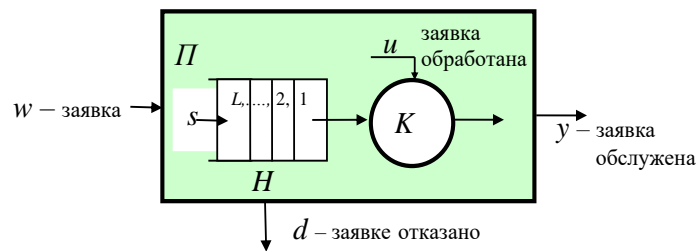


Рисунок 1 Простейший процесс обслуживания

Прибор простейшего обслуживания Π включает в себя ограниченный накопитель заявок H и канал обработки заявки - K . Поступающая в Π в некий момент непрерывного времени заявка w может получить отказ в обслуживании, если накопитель H полон, и тогда покидает прибор как не обслуженная заявка - d . Результаты обработки заявок y на выходе одного прибора обслуживания могут поступать в качестве заявок на обслуживание w на вход другого прибора обслуживания, связанного с ним по входу/выходу, образуя более сложную систему обслуживания.

Относительно поступившей заявки на обслуживание прибор Π реализует простейший процесс обслуживания, в котором выделяют следующие основные события:

- отказ заявке в обслуживании - d ;
- приём заявки w на обслуживание - s ;

- завершение ожидания заявкой s начала обработки;
- завершение обработки заявки s каналом;
- выдача u как результата обслуживания заявки s прибором Π .

Если отказ d не последовал, то прибор Π начинает простейший процесс обслуживания заявки s , а именно:

- принимает заявку s в накопитель заявок H – это очередь к каналу обработки K , в которой могут одновременно находиться $l = 0, 1, \dots, L$ заявок (L - ёмкость накопителя H),
- при освобождении канала K он сразу становится готовым к обработке очередной заявки s ;
- из не пустого накопителя очередная заявка s мгновенно поступает на обработку.

В результате текущее состояние $z \in Z$ прибора Π определяется парой состояний $\langle H, K \rangle$, H - *состояние накопителя*, K - *состояние канала*, которое изменяется скачком, когда в текущий момент непрерывного времени возникают случайные события поступления в прибор Π заявок на обслуживание w :

- событие отказа заявке w в обслуживании прибором Π по причине отсутствия свободных мест в накопителе H – *поток отказов в обслуживании* - d .
- событие приёма заявки w в накопитель H прибора Π на обслуживание – *поток принятых на обслуживание заявок* - s ;
- событие завершения обработки заявки s каналом K - *поток обработанных заявок* - i ;
- событие завершения обслуживания заявки s прибором Π - *поток обслуженных заявок* - u ;

В итоге входной поток заявок на обслуживание w разделяется на поток заявок, получивших отказ в обслуживании – d , и поток обслуженных заявок - u .

Если заявки могут покидать процесс обслуживания не обслуженными, то важной характеристикой системы является её продуктивность, количественно оцениваемая в виде абсолютной и относительной пропускной способности.

Абсолютная пропускная способность – это среднее число обслуженных заявок, покидающих прибор обслуживания в единицу времени. Эта характеристика позволяет оценить текущую продуктивность прибора обслуживания.

Относительная пропускная способность – это отношение абсолютной пропускной способности к среднему числу заявок, поступающих в прибор за единицу времени (к *интенсивности* потока заявок). Эта характеристика оценивает долю, которую составляют заявки, получившие обслуживание, от всех поступивших заявок. В общем случае часть заявок будет получать отказ в обслуживании.

Заметим, что в соответствии с определением абсолютной пропускной способности, если заявки не поступают в прибор обслуживания, то его текущая продуктивность (абсолютная пропускная способность) будет равна нулю. По мере увеличения *интенсивности потока заявок* будет расти и продуктивность прибора обслуживания. При этом рост абсолютной пропускной способности будет ограничен предельной способностью прибора обслуживать заявки в единицу времени (*предельная интенсивность обслуживания*, как внутреннее свойство прибора обслуживания).

Если интенсивность потока заявок не превышает интенсивности обслуживания прибора, то относительная пропускная способность равна единице. Когда интенсивность потока заявок начинает превышать интенсивность обслуживания прибора, относительная пропускная способность начинает стремиться к нулю. Чем она ближе к нулю, тем длиннее очередь заявок, ждущих освобождения прибора, а при ограниченной очереди увеличивается количество заявок, получающих отказ. Таким образом, абсолютная и относительная пропускные способности являются основными характеристиками оценки качества работы системы массового обслуживания.

Имитационное моделирование процессов обслуживания

Библиотека моделирования процессов

Эффективным способом моделирования процессов обслуживания является разработка компьютерных моделей, имитирующих структуру и функционирование процессов обслуживания. Такие модели называют *имитационными* моделями. Выполнение натуральных экспериментов с имитационной моделью позволяет проанализировать поведение и исследовать характеристики процессов обслуживания любой сложности. Эффективность имитационного моделирования как средства интерпретации и анализа процессов обслуживания существенно зависит от наличия и использования специализированных программных систем, являющихся инструментальными средствами построения имитационных моделей. Одним из таких средств имитационного моделирования является программная система AnyLogic, предоставляющая графо-символический язык построения имитационных моделей процессов обслуживания, основанный на использовании специальной программной Библиотеки Моделирования Процессов (Enterprise Library).

Построение в AnyLogic имитационной модели выражается в изображении структуры процесса обслуживания в виде диаграммы, представляющей схему поступления и обслуживания потоков заявок. Диаграмма выражается средствами библиотеки моделирования процессов, представленной на панели **Палитра**. Панель **Палитра** содержит различные наборы конструктивных элементов графического редактора диаграмм активного объекта, которые могут быть использованы при построении диаграммы процесса обслуживания (Рисунок 2, а)).

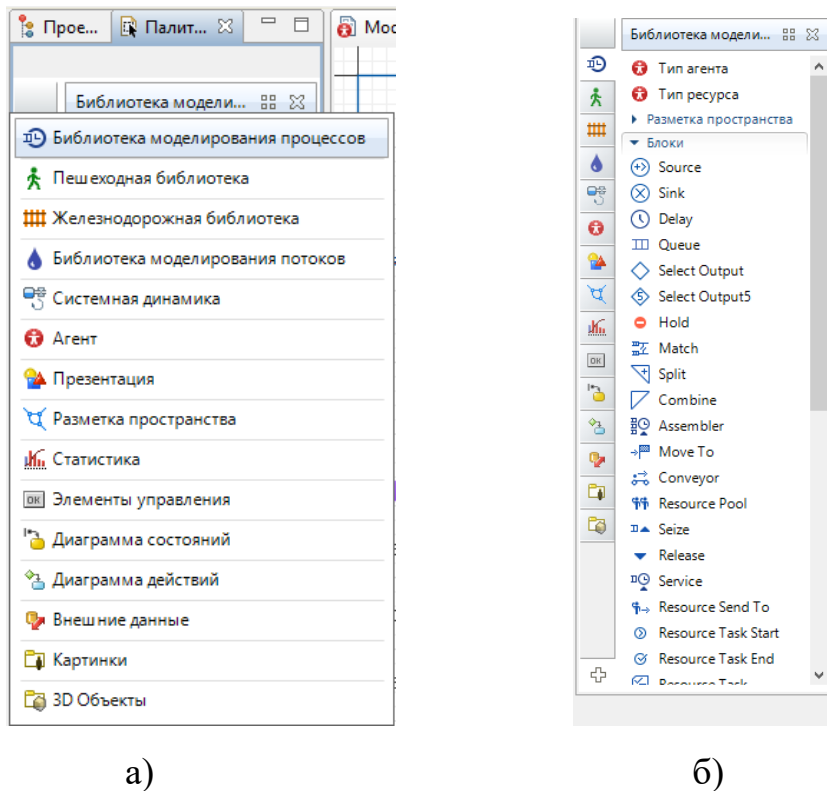


Рисунок 2 Палитра библиотеки моделирования процессов

Чтобы добавить на диаграмму модели необходимый блок, нужно щелчком мыши раскрыть на панели **Палитра** (Рисунок 2, а)) содержимое набора Библиотека моделирования процессов, выделить в нём нужный блок (Рисунок 2, б)) и перетащить его мышью в рабочую область графического редактора. При этом его свойства будут отображены в окне **Свойства**. В этом окне можно задавать свойства блока в соответствии с особенностями моделируемого процесса обслуживания. При необходимости для изменения свойств блока его нужно будет выделять щелчком мыши на диаграмме или в дереве проекта.

Для установления на диаграмме структурных связей между блоками они наделены входными и выходными *портами*. Блоки можно соединять путём перетаскивания с помощью мыши порта одного блока на порт другого. Для придания линии соединения необходимой формы можно щёлкнуть мышью по линии и потянуть за появившуюся точку изгиба. Можно создавать необходимое количество точек изгиба. Ниже коротко рассмотрены блоки, которые потребуются при выполнении лабораторной работы. Подробное описание всех блоков

библиотеки моделирования процессов представлено в справочной системе AnyLogic в интернете¹ (Рисунок 3):

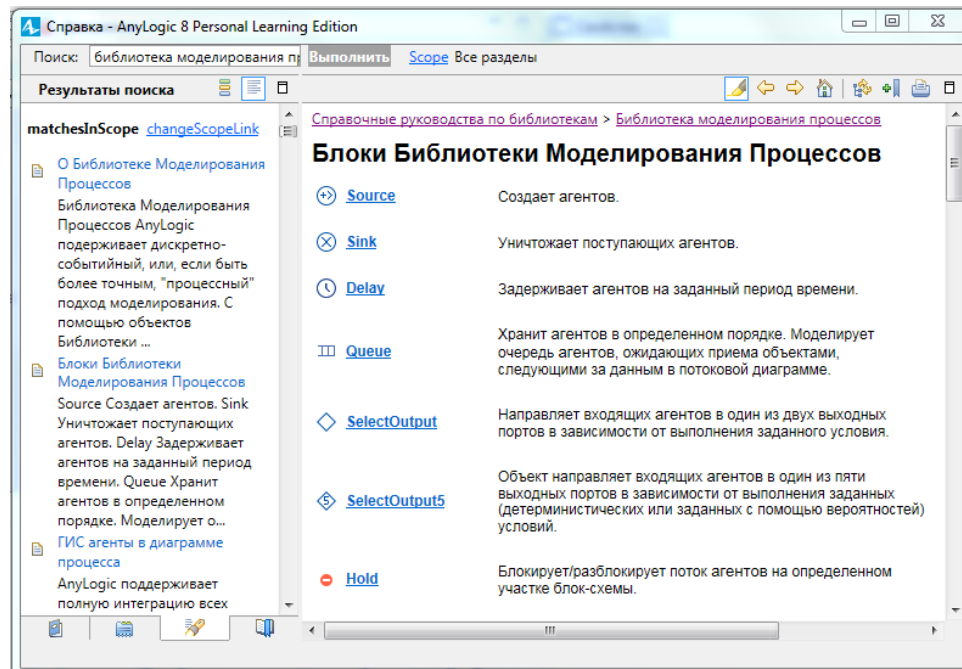



Рисунок 3 Справочная система AnyLogic

Ниже представлены краткие описания блоков, использование которых потребуется при выполнении лабораторной работы.

Блок  **Source** - используется в качестве начальной точки создания *заявок* на обслуживание в локальной переменной блока **agent** в виде объекта стандартного типа **Agent** со значением по умолчанию. Такие агенты, как объекты типа **Agent**, имеют "пустое" значение (компоненты отсутствуют). При моделировании процессов обслуживания стандартные агенты могут использоваться лишь для имитации потока абстрактных заявок. Чтобы блоком создавались агенты-заявки со значениями отличного от стандартного типа **Agent**, необходимо явно выбрать и установить этот тип значений, указав конструктор нужного типа в параметре **Новый агент** блока **Source**. Если выбранный тип агента имеет компоненты, то при создании блоком **Source** очередного агента можно оформить его значение программным кодом над агентом с именем **agent**, заданным

¹ <https://anylogic.help/ru/library-reference-guides/process-modeling-library/pml-blocks.html>.


в поле действия При выходе, которое выполняется перед тем, как созданный агент покинет блок **Source**.


Агент, поступивший в любой блок, доступен в программном коде, формируемым в свойствах блока, под системным локальным именем **agent**, которое необходимо использовать для доступа к компонентам заявки-агента.

Блок **Source** можно настроить на генерацию агентов согласно:


- заданной интенсивности (количество заявок в единицу времени),
- времени между прибытиями, заданному с помощью табличной функции (таблица интенсивности),
- расписанию, задающему точные времена и количество создаваемых агентов,
- "вручную" путём вызова функции блока - **inject()**.


Момент времени создания очередного агента может быть детерминированным, случайным, зависящим от каких-то дополнительных данных, и т.д. Поток заявок может быть, например, специфицирован как пуассоновский поток с заданной интенсивностью поступления заявок, или со случайным интервалом времени между прибытиями заявок, подчиняющимся, например, экспоненциальному или иному другому закону распределения.


Блок  **Queue** - моделирует в диаграмме накопитель, в котором формируется очередь задержанных агентов (например, как заявок, ожидающих своей очереди обработки). Свойством этого блока является вместимость очереди (значение по умолчанию - 100).

Блок  **Delay** - задерживает агента на заданный период времени. Одновременно могут быть задержаны сразу несколько агентов (по умолчанию - 1). Время задержки вычисляется динамически, может быть случайным, зависеть от компоненты текущего агента или от каких-то других условий. Агенты задерживаются в блоке независимо друг от друга, время задержки вычисляется отдельно для каждого агента. Как только время задержки некоторого агента

истекает, он тут же покидает блок. Если блок **Delay** заполнен полностью, то он не будет принимать новых агентов до тех пор, пока их количество в блоке не станет меньше значения вместимости блока. Для сохранения нового пришедшего агента нужно использовать один из блоков буферизации, например, блок **Queue**.

Блок  **Sink** - уничтожает поступивший агент. В диаграмме используется в качестве конечной точки (поглотителя) потока агентов. Для того чтобы агент удалялся из модели и уничтожался, нужно чтобы все пути движения потока агентов завершались блоком **Sink**.

Блок  **ResourcePool** - основной способ создания специальных типов агентов - *ресурсов*, моделирующих сущности, которые не могут быть созданы в модели как обычные агенты через блок **Source** и подобные блоки. Ресурсы в контексте модели необходимы другим агентам для выполнения определённых задач. Ресурсы могут захватываться и освобождаться агентами в блоках **Service**, **Release**, **Assembler** или **Seize**. Общее количество создаваемых в блоке **ResourcePool** ресурсов ограничено. Агенты, которым требуются ресурсы, могут их присваивать и освобождать в ограниченных количествах. Если запросы на ресурсы от блоков **Seize** или **Service** не могут быть удовлетворены в текущий момент времени, эти запросы помещаются в очередь блока **ResourcePool**. Эта очередь может быть либо обычной очередью FIFO, либо учитывать приоритеты запросов.

Блок  **Service** - захватывает для вошедшего в блок агента заданное количество ресурсов, удерживает их в течение заданного времени, а затем освобождает захваченные им для агента ресурсы.

Средства сбора статистики и статистического анализа

AnyLogic позволяет производить сбор различной статистики, связанной с работой блоков диаграммы. Некоторые блоки могут самостоятельно производить сбор основной статистики. Например, для блока **Queue** в модели можно собирать

статистику по *длине очереди*, а для блока **Delay** - статистику по *коэффициенту использования*. Для этого нужно лишь включить у блока режим сбора статистики (если он отключён для повышения скорости выполнения модели). Чтобы для блока включить сбор статистики, следует установить для него переключатель **Включить сбор статистики** на вкладке **Специфические** панели свойств блока (рисунок 1).

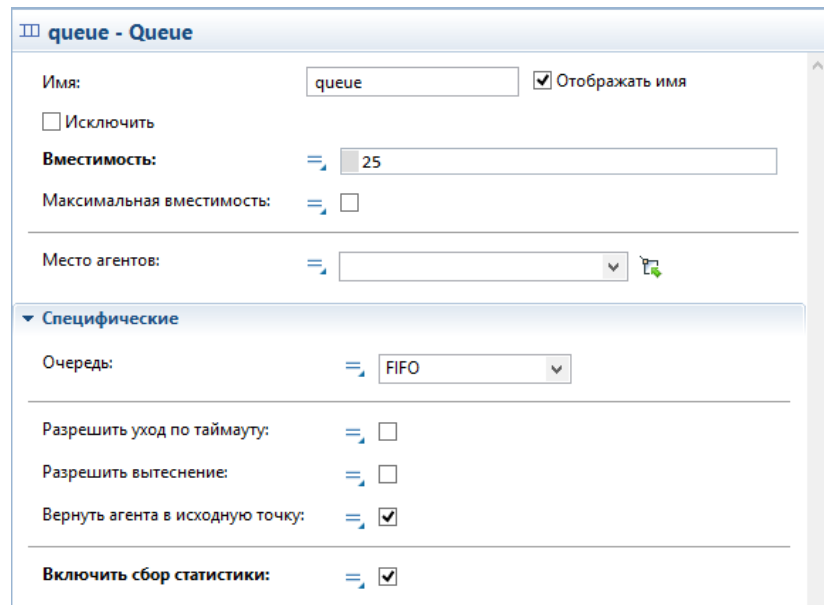


Рисунок 1

Статистику работы модели можно получать и отображать различными способами. Например, в процессе выполнения модели текущую статистику, связанную с работой блоков **queue** и **delay**, можно просматривать, просто открыв их окно инспекта. Однако универсальным способом сбора и обработки любой статистики является использование специального набора объектов **Статистика** на панели **Палитра** (Рисунок 4).

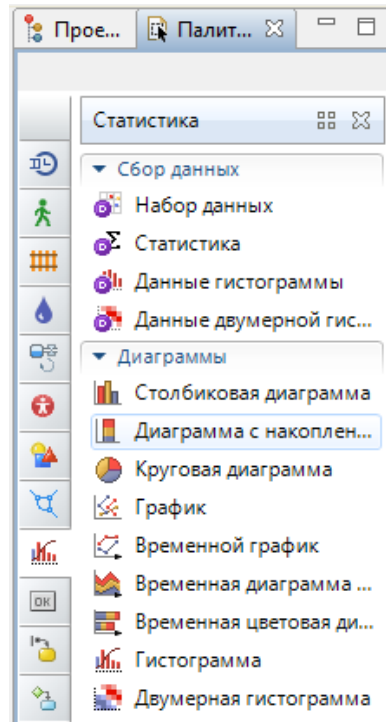



Рисунок 4


В наборе объектов **Статистика** представлены специальные объекты для сбора результатов моделирования и вычисления статистических значений, а также для отображения результатов с помощью различных гистограмм, диаграмм и графиков. Кроме того, результаты моделирования отображаются в открытом окне инспекта блоков набора **библиотеки процессов** или объектов набора **Статистика**.

Рассмотрим основные объекты набора **Статистика**, которые будут использоваться при выполнении лабораторной работы.

Объект  **Набор данных** - представляет собой двумерный массив измерений (конечного размера), предназначенный для сохранения значений указанных данных, полученные в последовательные моменты модельного времени. Каждое измерение, внесённое в набор данных, имеет два значения: x и y (типа `double`). Обычно значение y является значением наблюдаемого атрибута модели. В качестве значения x набора данных можно использовать либо другую наблюдаемую величину, либо указать модельное время. В первом случае сохраняются данные о связи значений одного атрибута от значений другого - такие наборы данных называются *фазовыми*. Если в качестве значения x указывать модельное время, т.е. связывать наблюдаемое значение y с моментом

времени, в который оно было измерено, то такой набор данных называется *временным*. Примером величины, для которой может собираться временной набор данных, может быть длина очереди.

Набор данных не предоставляет полной статистической информации по собираемым данным, он позволяет лишь узнать *минимальное* и *максимальное* из хранимых в данный момент значений (отдельно по x и по y). Добавление нового измерения в набор данных, который уже содержит максимально допустимое количество измерений, вызовет потерю самого старого измерения, хранимого в этом наборе данных. Если это значение было минимальным или максимальным, то заново вызовется процедура поиска нового минимального/максимального значения, что может занять значительное время при большом количестве хранимых измерений. Поэтому рекомендуется в качестве максимально допустимого количества измерений стараться задавать значение заведомо большее, чем фактическое количество измерений, которое предположительно будет добавлено в этот набор данных.

Объект  **Статистика** - вычисляет основную статистическую информацию (*минимум, максимум, среднее значение, среднеквадратическое отклонение* и др.) для последовательности измеренных значений (типа `double`). Блок работает по-разному в зависимости от того, считаются ли собираемые данные *непрерывными* или *дискретными* (Рисунок 5).

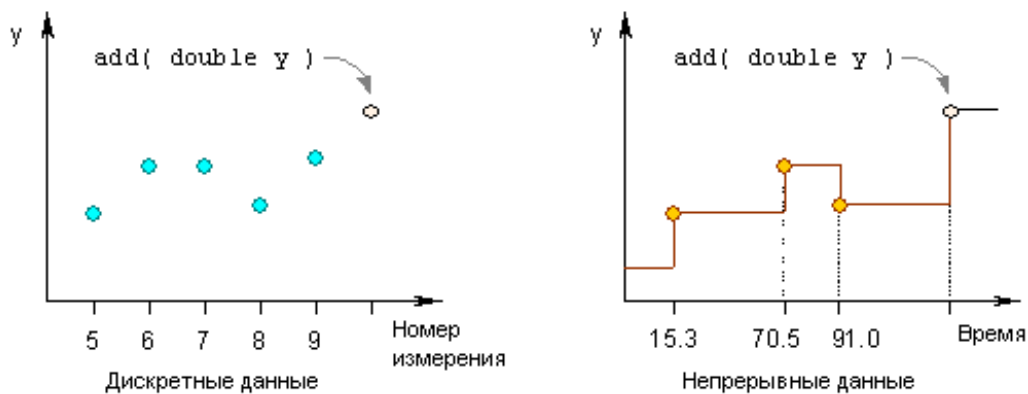



Рисунок 5

Если собираемые данные считаются дискретными значениями некой величины y , которая в модели по смыслу не связана со временем (например, длительности интервалов поступления заявок), то в блоке **Статистика** следует выбирать режим Дискретная (неупорядоченный во времени набор данных). Статистические характеристики таких величин вычисляются по полученному набору данных обычными статистическими методами (например, среднее значение является просто суммой измерений, поделённой на их общее количество).


Если собираемые данные считаются значениями некой величины y , которая в модели является непрерывной, то в блоке **Статистика** следует выбирать режим Непрерывная (упорядоченный во времени набор данных). Тогда набор данных будет собираться в виде временного ряда, в который измерения добавляются с возрастающими значениями временных меток. Использование набора данных y в виде временного ряда позволяет при необходимости получать его значение в любой заданный момент времени. Получение статистических характеристик таких величин основано на их интегрировании. Например, вычисление среднего значения \bar{y} на отрезке $[0, T]$ будет осуществляться по формуле:

$$\bar{y} = \frac{\int_0^T y(t) dt}{T}.$$

Заметим, что методы обработки временных рядов подразумевают, что последнее ранее добавленное значение является актуальным значением y в момент времени вызова метода.

Объект  **Данные гистограммы** - для заданного набора данных производит построение функции плотности распределения вероятности и интегральной функции распределения на фиксированном или автоматически выбирающемся наборе интервалов и выполняет обычный статистический анализ добавляемых значений (вычисляет среднее значение, минимум, максимум, дисперсию, средний

доверительный интервал и др.). Собранные статистика может быть отображена с помощью блока **Гистограмма**.

Объект  **Гистограмма** - отображает данные, собранные объектом **Данные гистограммы** (на одной гистограмме могут одновременно отображаться данные сразу нескольких таких объектов). Ось X всегда масштабируется таким образом, чтобы вместить все данные. Масштаб по оси Y также выбирается автоматически, таким образом, чтобы высота самого высокого столбца была равна высоте области диаграммы. При необходимости на гистограмме могут быть отображены функция плотности вероятности, функция распределения и линия среднего значения. Для этого в свойствах объекта **Данные гистограммы** следует отметить признак **Считать CDF**.

Функция плотности вероятности отображается в виде гистограммы - набора вертикальных столбцов, каждый из которых соответствует определённому интервалу. Высота столбца пропорциональна плотности (или количеству) значений, попавших в этот интервал.

Интегральная функция распределения отображается в виде ломаной линии поверх плотности вероятности, а среднее значение – в виде вертикальной линии, проходящей через точку с координатой, равной среднему значению.

Лабораторная работа

Цель лабораторной работы

Целью лабораторной работы является изучение и практическое использование средств библиотеки моделирования процессов для имитационного моделирования процессов обслуживания, а также освоение средств системы AnyLogic, предназначенных для статистической оценки вероятностных характеристик процессов обслуживания.

При выполнении лабораторной работы необходимо *изучить* предназначение и основные свойства блоков библиотеки моделирования процессов, задействованных при построении модели процессов обслуживания, а также овладеть следующими навыками:

- создание имитационной модели процессов обслуживания в виде диаграммы взаимосвязанных блоков, участвующих в управлении заявками на обслуживание;
- проведение экспериментов с имитационной моделью процессов обслуживания;
- использование при построении модели средств палитры AnyLogic для сбора статистических данных и их вероятностных оценок.

Порядок выполнения работы

Лабораторная работа заключается в последовательном выполнении ряда упражнений, направленных на разработку имитационной модели процессов обслуживания и их анализ. Необходимо последовательно выполнить все упражнения, используя свой вариант параметров процессов обслуживания, заданный в методических указаниях (см. Приложение 1).

В учебных целях в качестве системы массового обслуживания будем рассматривать гипотетическое банковское отделение. Будем полагать, что в банковском отделении находятся **банкомат** и стойки обслуживания клиентов кассирами - **операторы**, которые предназначены для обслуживания клиентов

банка. Стандартные операции с наличностью клиенты банка производят с помощью банкомата, а операции любой сложности – с помощью операторов. Необходимо построить имитационную модель банковского отделения и на основе результатов экспериментов с моделью осуществить оценку эксплуатационных характеристик его функционирования.

Упражнение 1

Создание диаграммы

Создайте в AnyLogic проект новой модели, и назовите её **Модель_обслуживания**. Переименуйте класс **Main** в **Процесс_обслуживания**. Создайте диаграмму модели процесса обслуживания, которая будет состоять пока только из одного банкомата. Для этого перетащите из палитры в окно графического редактора блоки библиотеки моделирования процессов и соедините их так, как показано ниже (Рисунок 6):

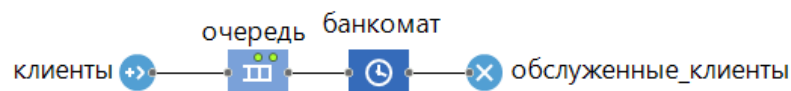


Рисунок 6

Далее задайте необходимые свойства объектам, соответствующих классам блоков.

Объект **клиенты** (Рисунок 7):

клиенты - Source

Имя: Отображать имя Исключить

Прибывают согласно: ▾

Время между прибытиями: ▾

Первое прибытие происходит: ▾

Считать параметры агентов из БД:

За 1 раз создается несколько агентов:

Ограниченное кол-во прибытий:

Рисунок 7 Свойства объекта клиенты

В свойстве объекта **клиенты** класса **Source** - **Прибывают согласно**, необходимо из выпадающего списка выбрать **Времени между прибытиями**. В свойстве **Время между прибытиями** объекта **клиенты** следует указать интервал времени между приходом клиентов (определяет, как часто в отделение приходят клиенты). Будем полагать, что этот интервал является стохастическим со средним значением, равным 5 единицам модельного времени. Полагая что единица модельного времени логически ассоциируется с 1-ой минутой астрономического времени, среднее значение интервала поступления клиентов будет равно 5 мин. Тогда интенсивность прибытия клиентов λ - величина обратная среднему значению интервала, будет равна 0.2, т.е. интенсивность прибытия 0.2 клиента в минуту). Если положить, что случайное значение интервала имеет экспоненциальный закон распределения, то в свойстве **Время между прибытиями** следует задать функцию `exponential(0.2)` (в качестве аргумента задаётся интенсивность). Функция `exponential()` является одной из стандартных функций распределения, реализуемых генератором случайных чисел AnyLogic (например, *нормальное, равномерное, треугольное* и т.д.). За детальным описанием функций и их параметров необходимо обращаться к справочной системе AnyLogic (меню **Справка**, см. тему **Вероятностные распределения**).

Объект **очередь** (Рисунок 8):

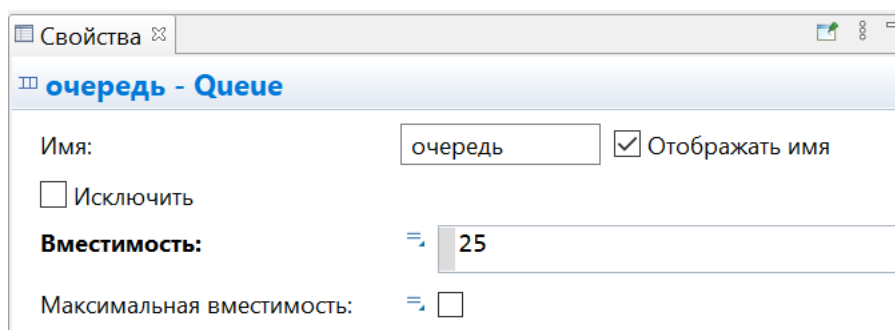


Рисунок 8 Свойства объекта очередь

В свойстве **вместимость** объекта **очередь** класса **Queue** задайте максимальную длину очереди 25.

Объект **банкомат** (Рисунок 9):

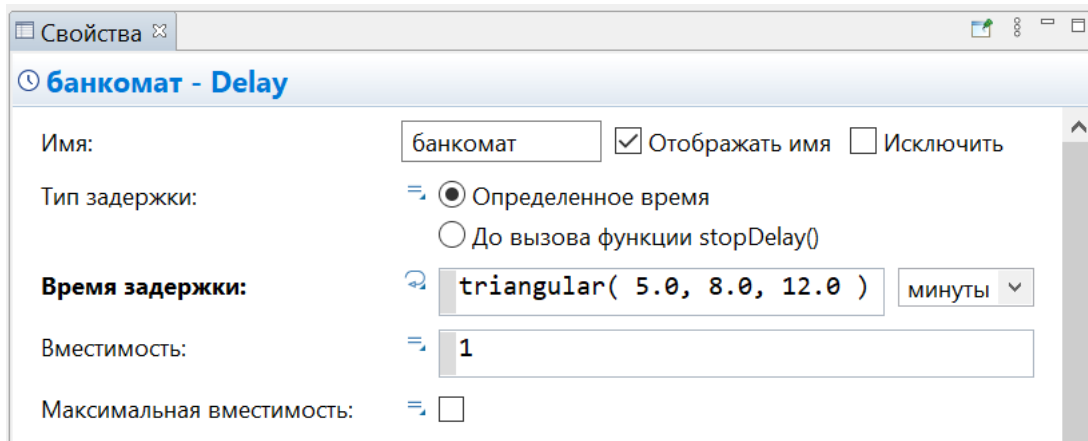


Рисунок 9 Свойства объекта банкомат

В свойстве **Время задержки** объекта **банкомат** задайте время задержки (имитирует время обслуживания клиента). Время обслуживания полагаем случайным распределённым в диапазоне от 5 до 12 минут с наиболее вероятным значением равным 8 минутам в соответствии с *треугольным* распределением – `triangular(5, 8, 12)`.

Объект **обслуженные_клиенты** (Рисунок 10):

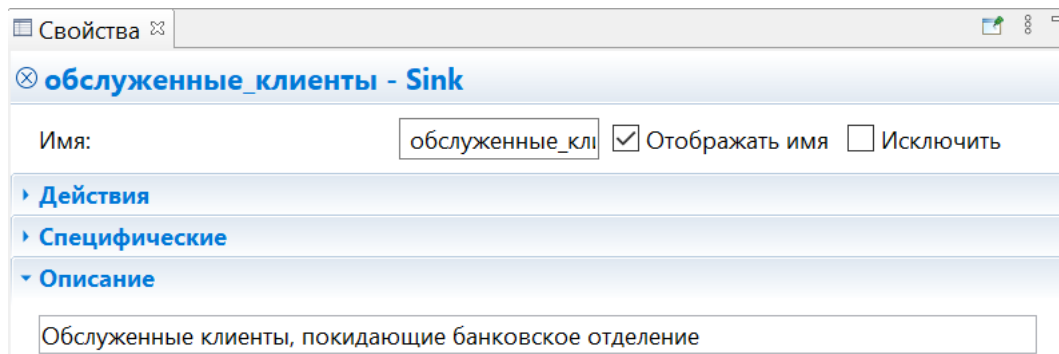


Рисунок 10 Свойства объекта обслуженные_клиенты

В свойствах объекта **обслуженные_клиенты** заполним поле **Описание**. Остальное пока оставим без изменения.

Запуск модели

Предварительно откройте окно свойств модели **Модель_обслуживания**. В свойстве **Модельное время** в качестве единиц модельного времени установите *минуты*, т.е. единица модельного времени будет интерпретироваться как одна минута астрономического времени (Рисунок 11).

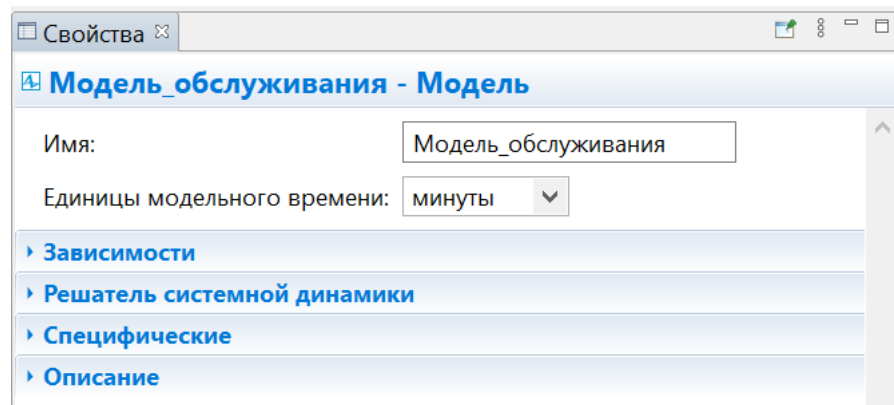


Рисунок 11 Определение единицы модельного времени

В каждом простом эксперименте с моделью автоматически создаётся презентация для наглядной визуализации работы модели в процессе её выполнения, с помощью которой можно наблюдать текущее состояние объектов модели, например, длину очереди, количество обслуженных заявок и т.д. Перейдите в окно свойств простого эксперимента (Рисунок 12).

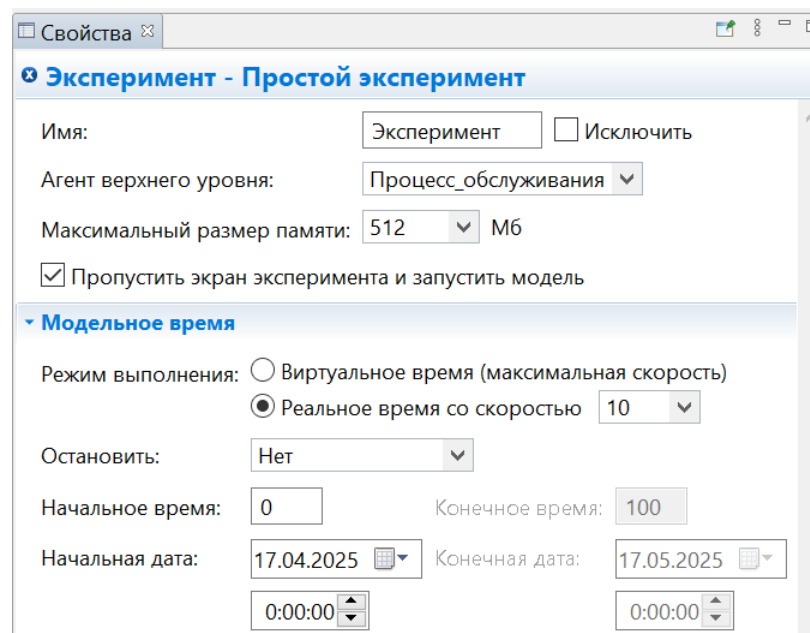


Рисунок 12 Установка свойств простого эксперимента с моделью

В свойствах простого эксперимента с именем Эксперимент с агентом верхнего уровня Процесс_обслуживания на вкладке **Модельное время** можно задать начальный темп модельного времени, задающий скорость выполнения модели и её презентации. Задайте выполнение модели в режиме *реального времени* со скоростью, например, 10 единиц модельного времени в одну секунду физического времени. В результате изменение модельного времени, и

соответственно пересчёт всей модели, будет осуществляться 10 раз в секунду. Управляя таким образом скоростью течения (темпом) модельного времени по отношению к физическому можно добиваться нужной наглядности анимации в процессе эксперимента с моделью, увеличивая или замедляя её скорость. Выбор же в модели режима виртуального времени позволяет выполнять модель с максимальной скоростью, «жертвуя» наглядностью анимации.

Заметим ещё, что в свойствах простого эксперимента можно указать, что при запуске эксперимента следует пропустить экран эксперимента и сразу начинать презентацию выбранного агента модели верхнего уровня.

Запустите модель, откроется окно презентации простого эксперимента с агентом верхнего уровня Процесс_обслуживания² (Рисунок 13).

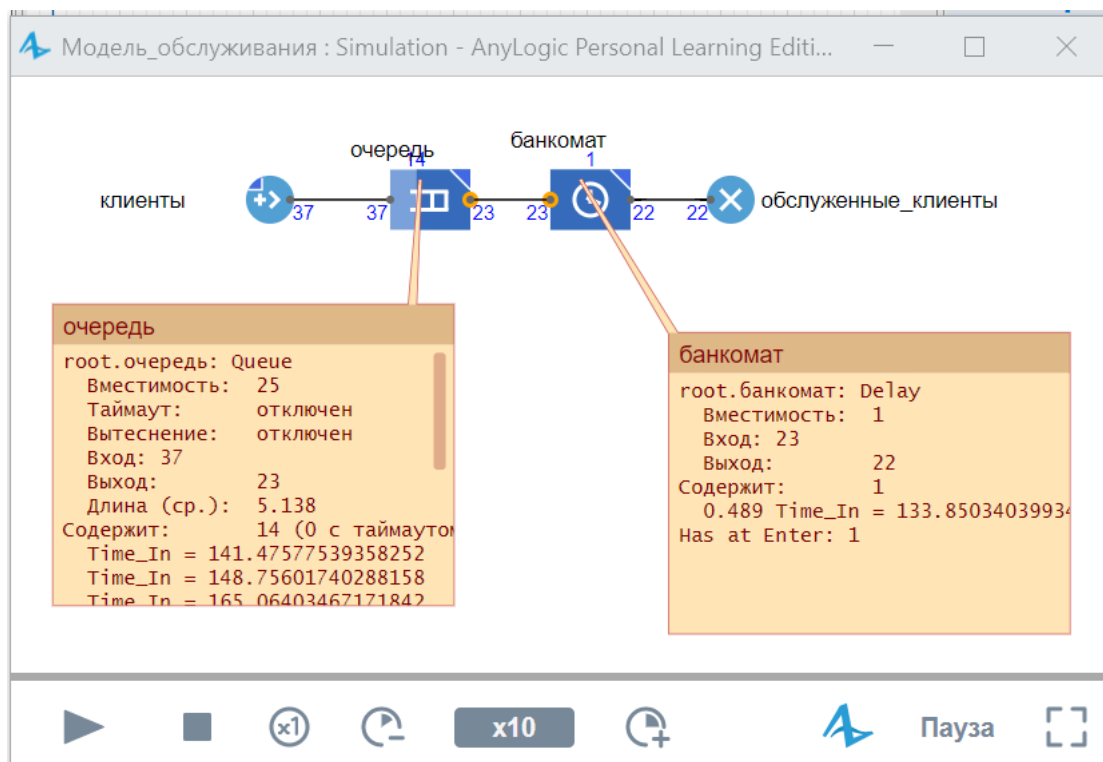


Рисунок 13 Презентация выполнения агента Процесс_обслуживания

Нажав и удерживая левую кнопку мыши можно захватить экран и переместить его содержимое в окне так, чтобы расположение диаграммы было удобным для просмотра. Щелчком левой кнопки мыши по объекту диаграммы можно открыть соответствующий ему инспект, и следить за состоянием объекта.

² В новых версиях AnyLogic вид окна может измениться.

На презентации можно проследить работу диаграммы процесса обслуживания в виде анимации. Можно видеть сколько клиентов находится в очереди, сколько клиентов в данный момент обслуживается и т.д. С помощью кнопок в нижней части окна можно изменять скорость выполнения модели, добиваясь наглядной анимации или ускоряя получение установившихся статистических результатов моделирования.


Упражнение 2

Сбор и анализ в модели статистических данных

Так как получаемые с помощью диаграммы процессов характеристики процесса обслуживания оцениваются статистически (например, *вероятность занятости банкомата* или *средняя длина очереди* клиентов к кассирам и т.п.), то AnyLogic предоставляет специальные средства для сбора по объектам диаграммы процесса статистических данных, а также средства для их анализа и визуализации результатов статистического анализа, представленные в наборе объектов **Статистика** панели **Палитра**. Необходимо воспользоваться этими средствами для оценки вероятности застать клиентом банкомат занятым, и отобразить её в виде столбиковой диаграммой.

Теоретически вероятность застать клиентом банкомат занятым равна среднему значению загрузки банкомата в течение всего времени работы - $\frac{\Delta T}{T}$, где ΔT – суммарное время занятости банкомата работой в течение интервала времени T . У каждого объекта класса **Delay** есть *встроенный объект* - **statsUtilization** - набор данных для сбора и анализа загрузки этого блока во времени (см. справочную информацию о блоке **Delay**). Этот объект имеет метод **mean()**, возвращающий текущее среднее значение загрузки блока в течение всего времени его работы. Объект **statsUtilization** блока **Delay** предоставляет и другие методы для

анализа статистики³, такие как `min()` или `max()` для определения минимального или максимального значения занятости блока поступившей заявкой. Воспользуйтесь методом `mean()` для оценки вероятности застать клиентом банкомат занятым. Для визуализации анализа статистики загрузки объекта **банкомат** в виде столбиковой диаграммы необходимо выполнить следующие действия:

1. Откройте в **Палитре** набор **Статистика**. Перетащите элемент **Столбиковая диаграмма** в область редактора диаграммы процесса обслуживания клиентов, и перейдите на панель **Свойства** созданного объекта **chart**, на вкладку **Данные**.
2. Содержимое поля **Заголовок** по умолчанию измените на **Загрузка банкомата**. Выберите ширину и цвет диаграммы.
3. Введите в поле **Значение** выражение - `банкомат.statsUtilization.mean()`.
4. При необходимости добавления в диаграмму ещё значений можно щёлкнуть мышью по кнопке . Появится секция для задания выражения, значения которого будет отображаться на столбиковой диаграмме в виде дополнительного столбика.
5. На вкладке **Внешний вид** панели **Свойства** можно выбрать расположение легенды относительно диаграммы (например, чтобы она отображалась снизу). Оформите внешний вид и расположение диаграммы столбиковой диаграммы, как показано на рисунке ниже:

В результате панель свойств объекта **chart** типа **Столбиковая диаграмма** будет иметь следующее содержание (Рисунок 14):

³ Полный список методов класса набора данных представлен в справке AnyLogic на странице: `StatisticsContinuous`.

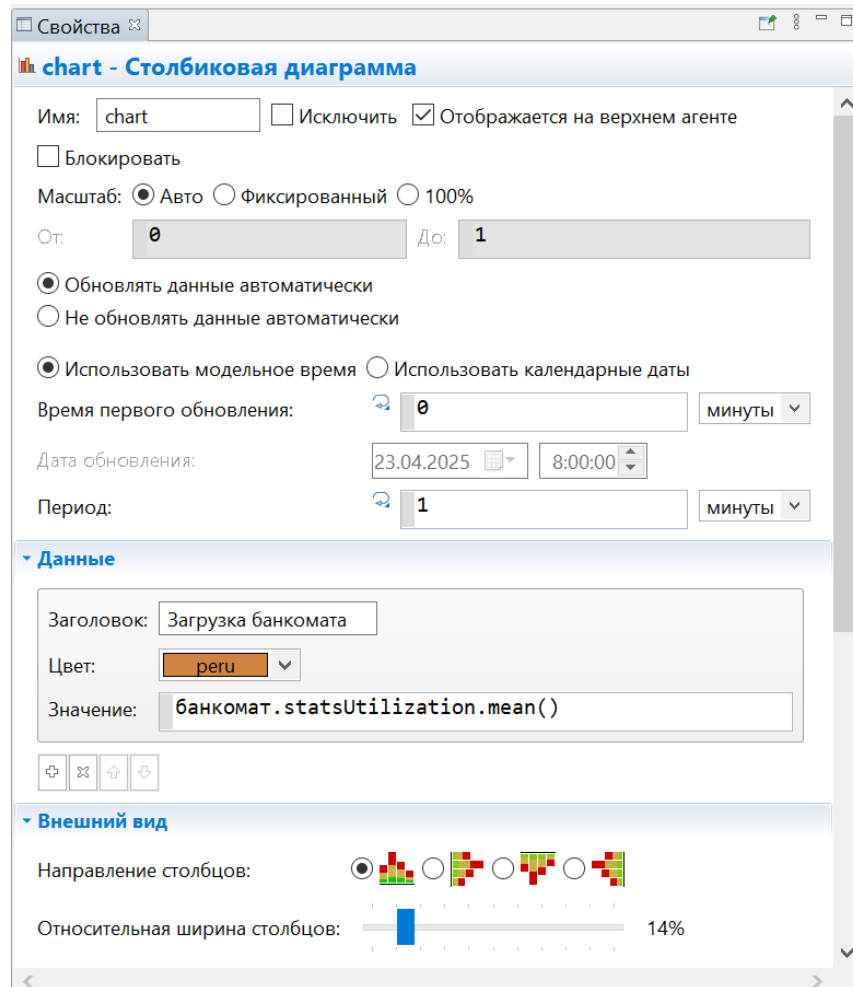


Рисунок 14 Формирование свойств столбиковой диаграммы загрузки банкомата

Аналогичным образом воспользуемся объектом сбора статистики блока **очередь** класса **Queue** для анализа средней длины очереди заявок. Добавьте ещё одну столбиковую диаграмму, но с горизонтальным расположением столбика, для визуализации оценки средней длины очереди заявок – объект **очередь**. Задайте заголовок диаграммы – Средняя длина очереди, а в качестве собираемых для статистики значений задайте значение среднего размера очереди - `очередь.statsSize.mean()`. Здесь `mean()` – метод объекта **statsSize** блока **очередь**, производящего сбор статистики, и вычисляющий в частности значение средней длины очереди. В результате панель свойств столбиковой диаграммы будет иметь следующее содержание (Рисунок 15):

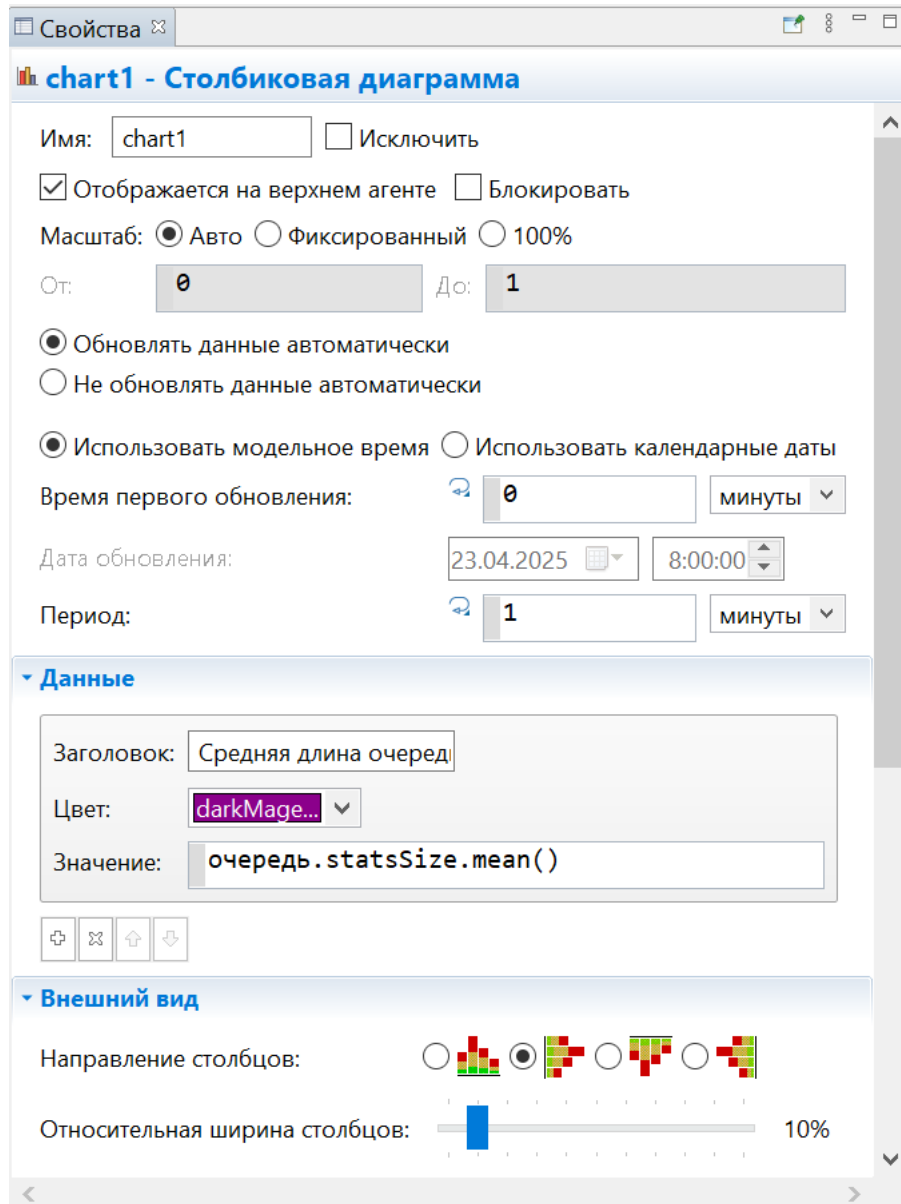


Рисунок 15 Формирование свойств столбиковой диаграммы средне длины очереди

Оформите внешний вид и расположение столбиковых диаграммы в поле редактора агента Процесс_обслуживания, как показано ниже (Рисунок 16):

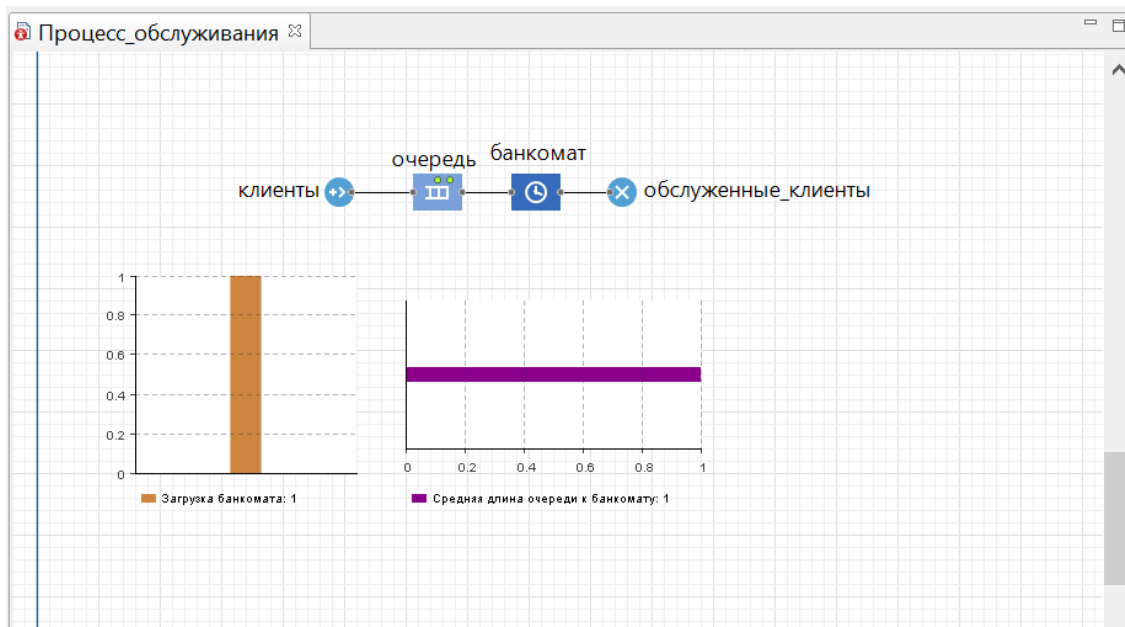


Рисунок 16 Результат формирования столбиковых диаграмм
в окне Процесс_обслуживания

Запустите простой эксперимент с моделью обслуживания и с помощью созданных столбиковых диаграмм проанализируйте динамику изменения занятости банкомата и средней длиной очереди на презентации поведения агента Процесс_обслуживания (**Ошибка! Источник ссылки не найден.**).

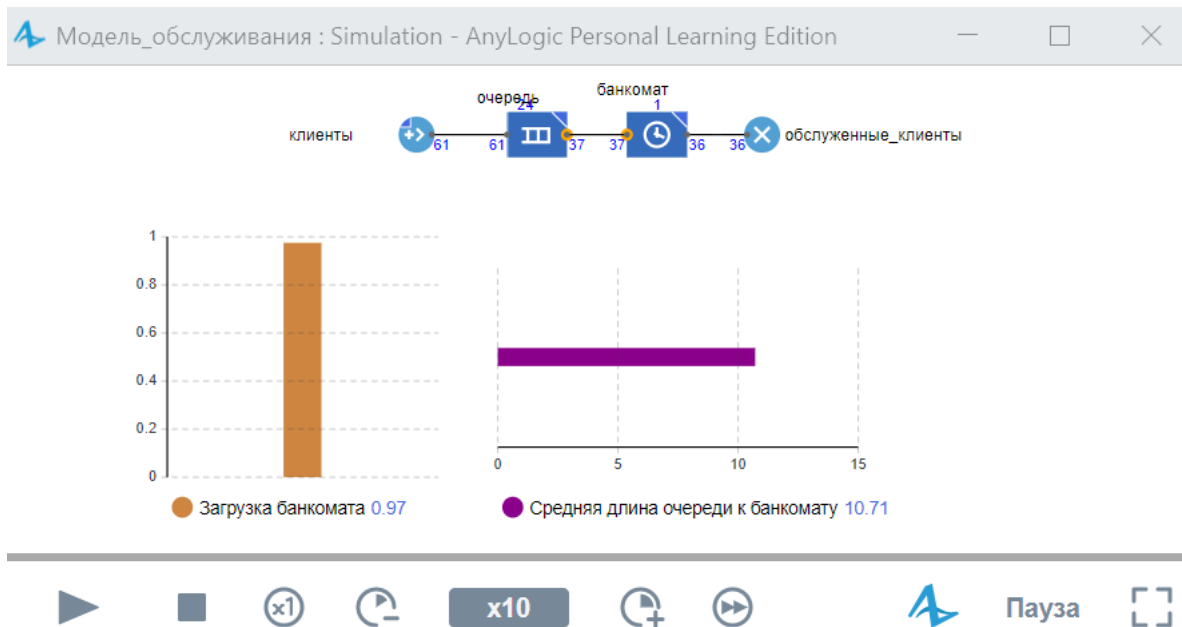


Рисунок 17 Презентация статистики агента Процесс_обслуживания

С помощью столбиковых диаграмм наблюдайте динамику изменения занятости банкомата и средней длиной очереди.

Упражнение 3

Моделирование абстрактного ресурса

Усложним модель, добавив в неё стойки обслуживания кассирами - **операторов**, которые параллельно с банкоматом могут обслуживать клиентов. Очевидно, что работу оператора также можно моделировать блоком **Delay**. Тогда потребуется включить в модель новые блоки **Delay** в количестве добавленных кассиров. Однако, если потребуется изменять количество операторов, то необходимо будет каждый раз менять структуру модели. Существует другая концептуальная возможность моделирования добавляемых операторов, рассматривая их как некий абстрактный ресурс (канал обслуживания), который заявка должна захватить, чтобы быть обслуженной. Ресурсы — это объекты, требуемые агентами для выполнения их задач. В нашем примере агент – это заявка, которой для обслуживания нужно захватить один канал обслуживания (единица ресурса - один оператор). Для реализации концепции канала как ресурса используем блок – **Resource Pool**, из которого можно "черпать" нужное количество операторов.

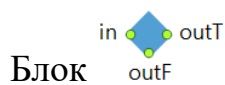
Единственный порт блока **Resource Pool** может быть одновременно соединён с портами других блоков, работающих с ресурсами, которые смогут разделять этот пул, исключительно захватывая и освобождая нужное количество ресурсов. В нашем случае таким блоком целесообразно использовать блок **Service**. Если поступившие от различных блоков запросы ресурса не могут быть удовлетворены в текущий момент времени, эти запросы не теряются, а сохраняются в очереди запросов ресурса блока **Resource Pool**.

Блок **Service** – моделирует захват вошедшей в блок заявкой заданного количества ресурса (находящегося в связанном с ним блоке **Resource Pool**), задерживает заявку на определённое время, а затем освобождает заявку и одновременно захваченные ею ресурсы. Блок располагает и собственной внутренней очередью, куда помещаются агенты, когда в момент их поступления

в блок весь ресурс оказался исчерпанным предыдущими зашедшими в блок клиентами.

Моделирование параллельного процесса обслуживания

Так как теперь банкомат и операторы будут работать параллельно, то поток поступающих клиентов должен как-то разделяться между ними. Будем полагать, что поступающие клиенты имеют возможность выбирать для обслуживания либо банкомат, либо оператора. Это можно реализовать в модели с помощью блока **Select Output**.



Блок **Select Output** - направляет вошедшего агента в один из двух выходных портов в зависимости от выбранного правила и заданного условия отправки агента в выходной порт **outT**: выполнение заданного условия или случайно с заданной вероятностью. В противном случае агент отправляется в порт **outF**.

Добавьте из палитры в поле редактора агента Процесс_обслуживания объект класса (блока) **Select Output**, и назовите его **выбор**. Добавьте ещё объект класса (блока) **Service**, и назовите его **операторные_стойки**. Свяжите объекты процесса обслуживания соединителями, формируя параллельную линию обслуживания. Добавьте объект **операторы** класса (блока) **ResourcePool**, моделирующего операторов как ресурса, "захватываемого" клиентами, подошедшими к операторным стойкам для обслуживания (Рисунок 18):

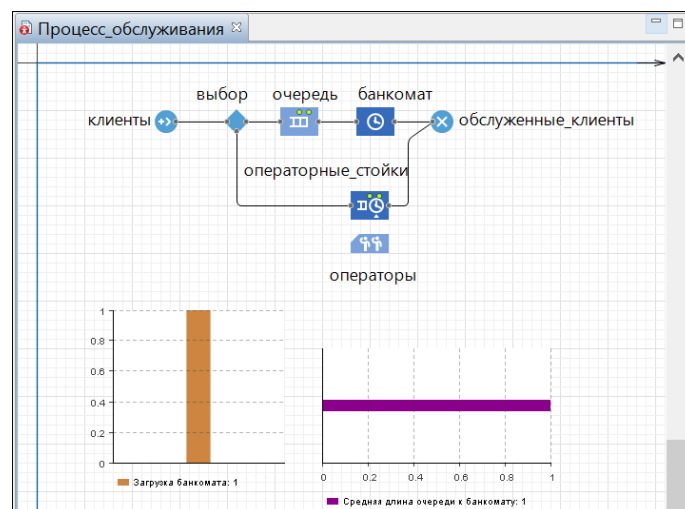


Рисунок 18 Процесс с параллельными линиями обслуживания

В качестве свойства объекта **выбор** установим условие случайного выбора выхода outT. Для этого в поле Условие: зададим функцию `randomTrue(p)` с вероятностью p , которая возвращает истинное значение – `true`, с заданной вероятностью p . Вероятность генерирования значения `false`, соответственно, равна $(1 - p)$. Зададим $p=0.5$, тогда к операторам и банкомату будет приходить примерно равное количество клиентов (Рисунок 19).

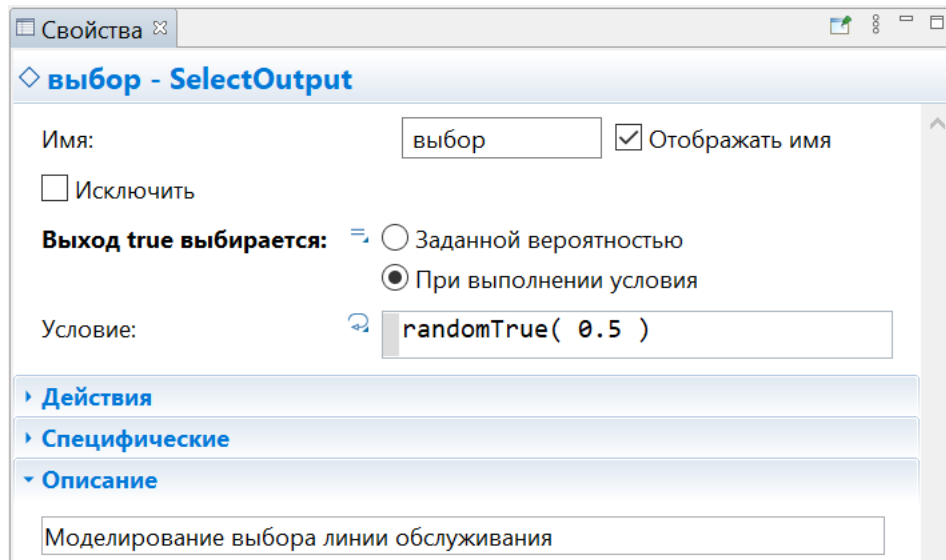


Рисунок 19 Моделирование выбора линии обслуживания

В качестве свойств объекта **операторы** укажите тип ресурса – Статический; Количество ресурсов – 5 (Рисунок 20).

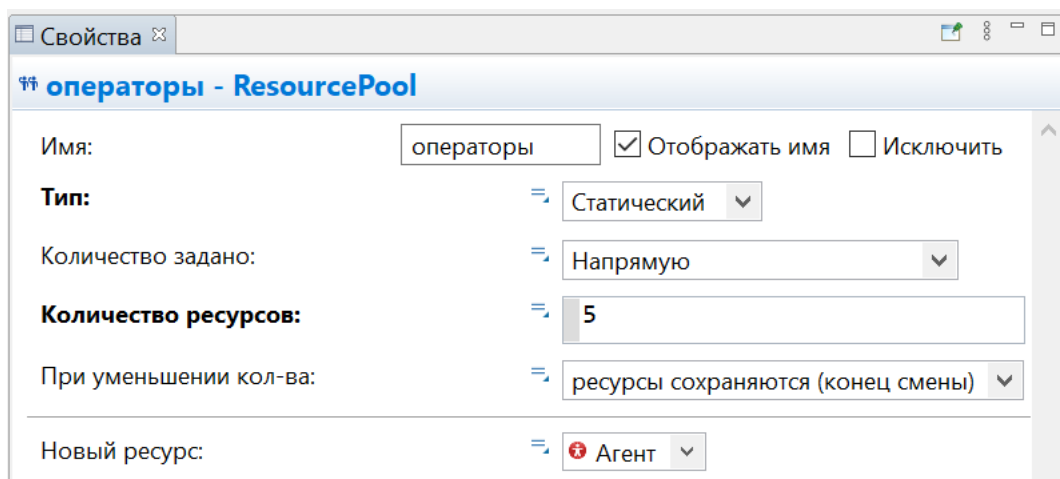


Рисунок 20 Количество операторов операторных стоек

В качестве свойств объекта **операторные_стойки** класса **Service** укажите, что в очереди может находиться до 20 человек (свойство Вместимость очереди); задайте время обслуживания (свойство Время задержки). Будем полагать, что

время обслуживания имеет *треугольное* распределение с минимальным значением 5.5, наиболее вероятным – 10 и максимальным – 15 минут. Для использования операторов в качестве ресурса операторных стоек необходимо для свойства **Захватить** установить ресурсы одного типа. Для свойства **Тип ресурсов** указать имя используемого ресурса, в данном случае операторы (Рисунок 21).

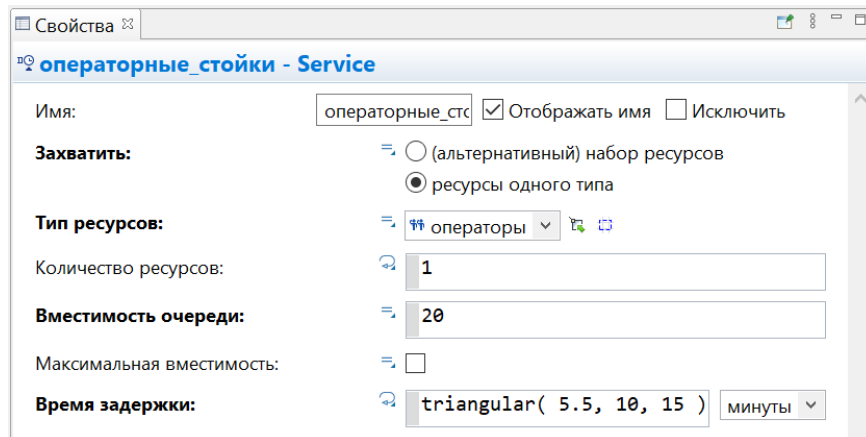


Рисунок 21 Моделирование линии обслуживания операторами

Презентация процесса параллельного обслуживания

Запустите модель и откройте её презентацию простого эксперимента. Проанализируйте работу распределителя потока клиентов, а также загрузку и длину очереди к банкомату и операторным стойкам.

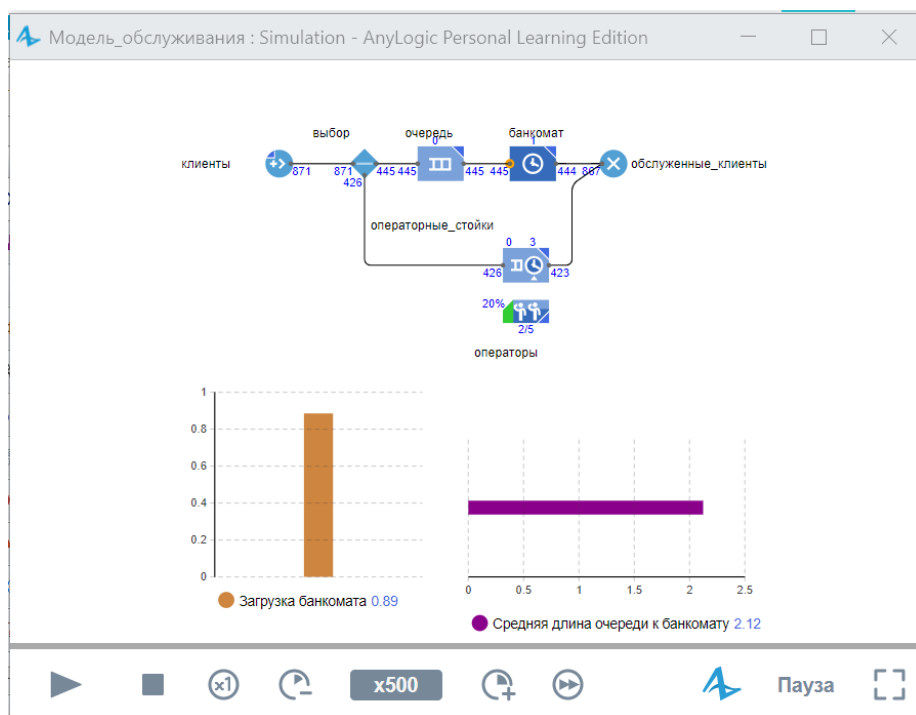


Рисунок 22 Оценка длины очереди и загрузки банкомата

Упражнение 4

Создание нестандартных типов агентов

По умолчанию, создаваемые объектами блоков библиотеки моделирования процессов агенты или ресурсы стандартных типов не имеют прикладных компонентов (параметров, методов), что не позволяет их использовать в прикладных целях, например, для сбора статистики. Для этого Библиотека моделирования процессов предоставляет возможность создавать собственные типы агентов или ресурсов с задаваемым набором необходимых компонент.

Для того, чтобы можно было собирать в модели необходимую статистику для оценки параметров и характеристик эффективности процесса обслуживания клиентов, создайте новый тип агента, который будет иметь параметры, используемые для сбора статистики. Для этого:

1. Перетащите элемент **Тип агента** из палитры Библиотеки моделирования процессов в графический редактор, откроется диалоговое окно Создание агента (Рисунок 23).

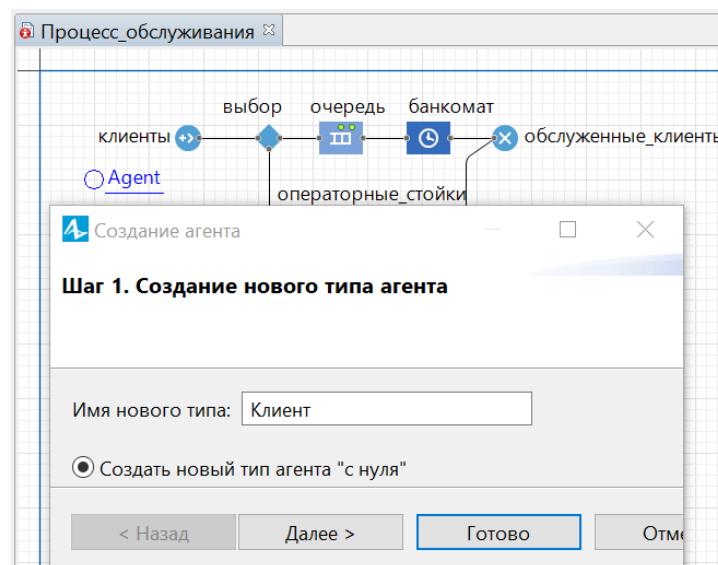


Рисунок 23 Создание нового типа агента

2. В поле **Имя нового типа** введите имя создаваемого агентного типа **Клиент** и щёлкните кнопку **Далее**. Откроется окно выбора фигуры для анимации агента (Рисунок 24).

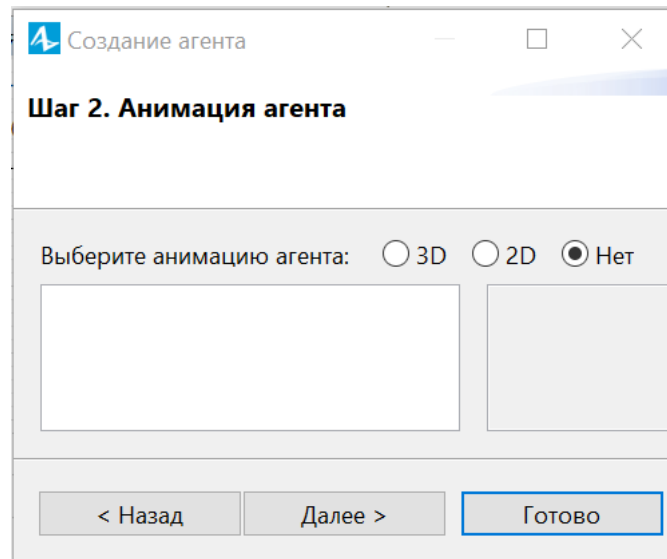


Рисунок 24 Отказ от анимации агента

Выберите опцию Нет, щёлкните кнопку Готово. Откроется окно графического редактора с пустой диаграммой только что созданного агентного типа Клиент. Из палитры добавьте в тип агента Клиент параметр время_прихода и время_обслуживания типа double (Рисунок 25).

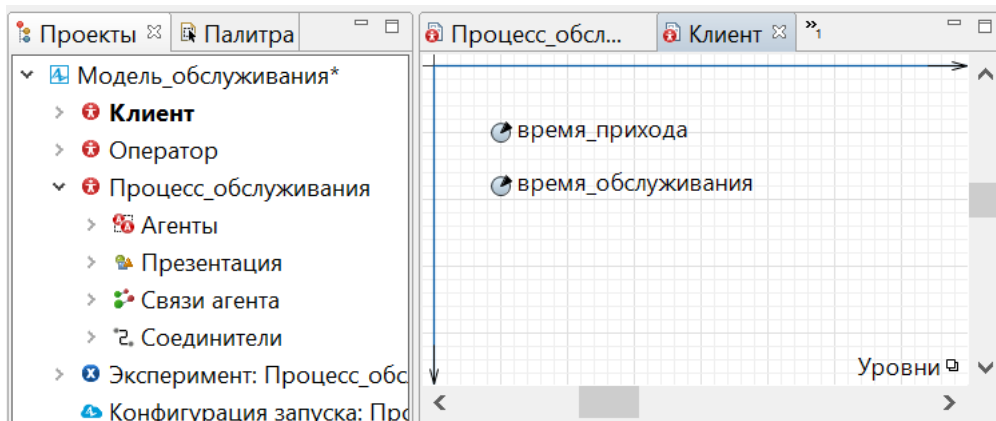


Рисунок 25 Созданный в модели новый тип агента Клиент с параметрами

Создание нестандартных типов ресурсов

AnyLogic рассматривает «ресурсы» как особые агенты, создаваемые и рассматриваемые в модели как абстрактные ресурсы, которые по смыслу могут использоваться соответствующими блоками моделирования процессов в качестве ресурсов. Создание ресурсных типов осуществляется так же, как и создание агентных типов. Заданное имя созданного типа ресурса становится доступным для использования в качестве свойств соответствующих блоков диаграммы процессов.

Формально создайте «пустой» тип единицы ресурса с именем **Оператор** и замените им ранее использованный по умолчанию тип ресурса **Агент** при формировании набора **операторы** как ресурса блока **операторные_стойки** (Рисунок 26).

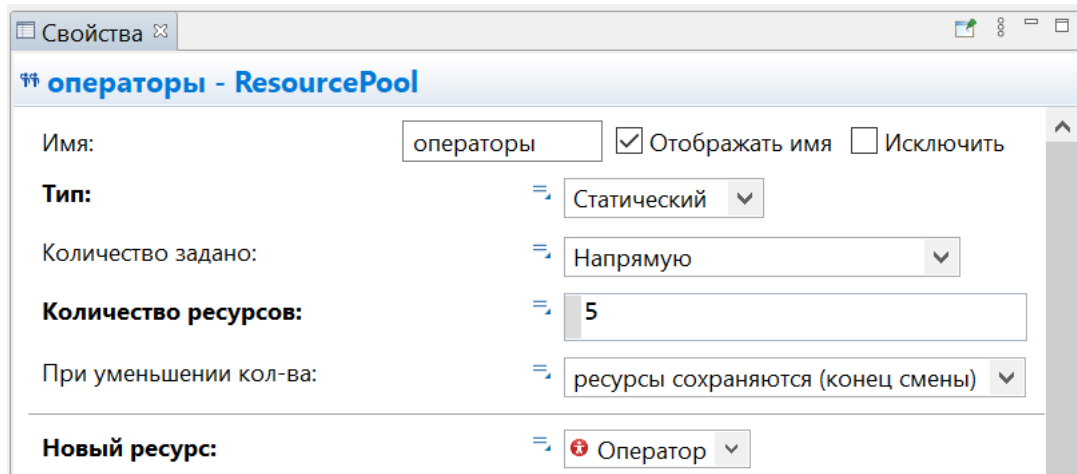


Рисунок 26 Использование созданного нового типа ресурса

Ссылка на параметры агентов созданных типов

Заметим, что в AnyLogic для ссылки на параметры агентов любого типа (в данном случае - типа Клиент) при формировании программного кода в свойствах блоков Библиотеки моделирования процессов необходимо использовать в блоке локальное имя `agent`. При этом в свойствах блока необходимо явно указать тип агента Клиент, к которому будет преобразовываться содержимое локальной переменной `agent`, куда записывается поступившая в блок заявка - экземпляр класса Клиент, и поля заявки, заданные в классе Клиент, будут теперь доступны в программном коде действий блока со ссылкой на `agent.имя_поля`. Но если в свойствах блока тип приходящей заявки явно не указывать, то в программном коде действий блока придётся явно задавать преобразование типа локальной переменной `agent`: `((Клиент)agent).имя_поля`.

Следует, однако, заметить, что явное преобразование типа обязательно потребуется, если какие-то заявки в смешанном потоке заявок могут иметь разные классовые типы (например, `Class1` или - скажем, `Class2`), то в свойствах соответствующего блока в поле Класс заявки придётся в качестве типа оставить

значение базового класса по умолчанию - Agent. В этом случае в программном коде блока потребуется предварительно анализировать тип поступившей заявки, используя оператор instanceof, и явно преобразовывать содержимое локальной переменной agent к выявленному типу. Например, для присвоения параметру Time значения текущего времени time() (в пришедшем агенте либо типа Class1, либо типа Class2) необходимо проанализировать и явно преобразовать текущее значение локальной переменной agent к выявленному типу:

```
if(agent instanceof Class1)
    ((Class1)agent).Time = time();
else
    if(agent instanceof Class2)
        ((Class2)agent).Time = time();
```

Упражнение 5

Вычисление характеристик процесса обслуживания

Важной характеристикой моделируемого процесса обслуживания является оценка времени обслуживания клиентов. Для вычисления времени обслуживания клиента в свойствах блока обслуженные_клиенты в поле задания действия При входе клиента в блок задайте программный код: agent.время_обслуживания=time()-agent.время_прихода.

Для этого в свойствах блока клиенты в поле Тип агента выберите Клиент, в поле задания действия «При выходе» задайте программный код для занесения в параметр время_прихода пришедшего агента типа Клиент значения текущего времени time(): agent.время_прихода=time().

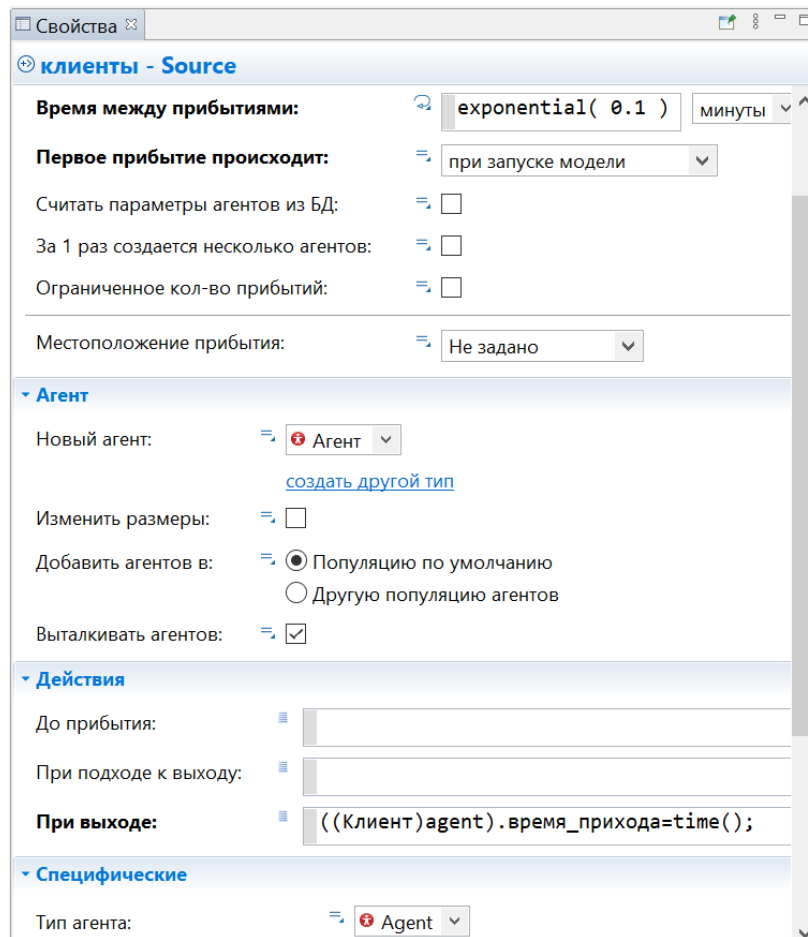




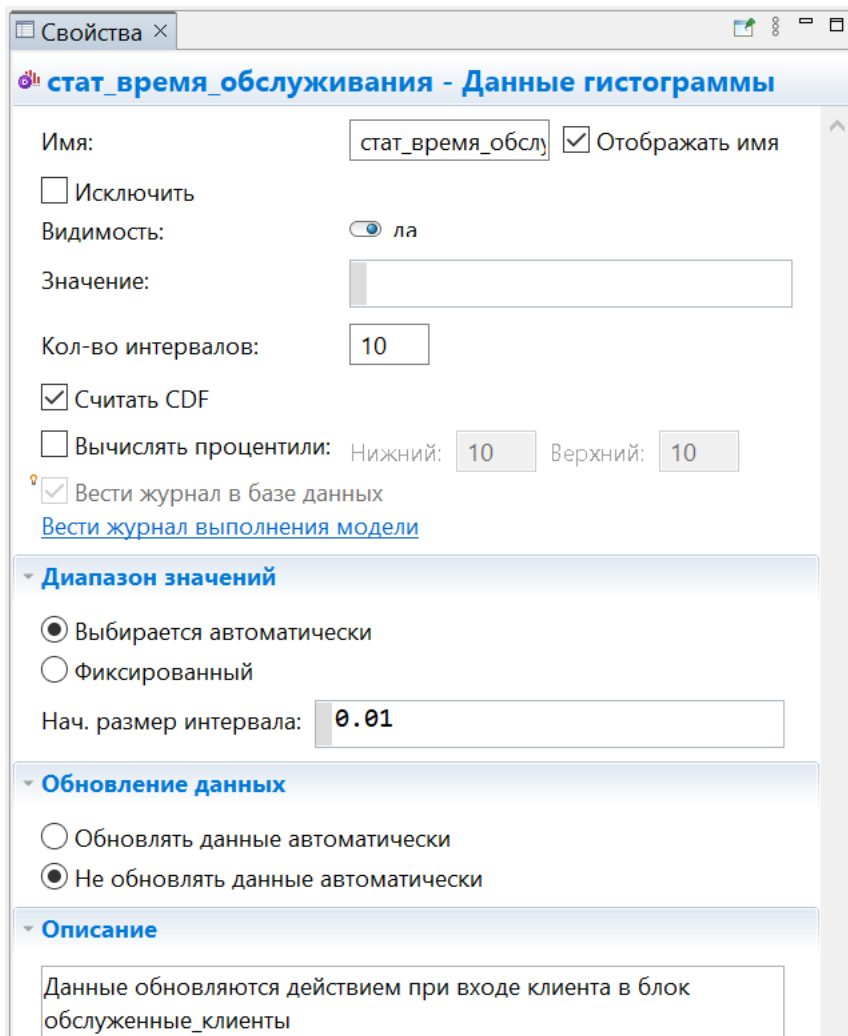
Рисунок 27 Фиксация момента прихода клиента

Сбор статистики времени обслуживания клиента


Так как время обработки клиента банкоматом или оператором является случайным, то время нахождения клиента в процессе обслуживания является случайной величиной. В связи с этим оценка времени обслуживания как случайной величины может быть осуществлена статистическими методами с помощью гистограммы. В модели статистику времени обслуживания можно собрать с помощью специального объекта класса «Данные гистограммы», и затем собранную объектом статистику распределения времени обслуживания клиентов в банковском отделении отобразить с помощью объекта класса «Гистограмма». Для вычисления времени обслуживания клиента в банковском отделении будем использовать параметры объекта клиент класса **Клиент**. В блоке **обслуженные_клиенты** для очередного обслуженного клиента необходимо

вычислять время его нахождения в банковском отделении, сохраняя результат в предварительно размещённом на диаграмме объекте  Данные гистограммы .

Добавьте объект сбора данных для гистограммы на диаграмму, перетащив из набора **Статистика** панели **Палитра** объект  Данные гистограммы в область диаграммы процесса обслуживания, задайте объекту имя `стат_время_обслуживания`, и следующие свойства (Рисунок 28):



Свойства ×

 **стат_время_обслуживания - Данные гистограммы**

Имя: Отображать имя

Исключить

Видимость: ла

Значение:

Кол-во интервалов:

Считать CDF

Вычислять процентиля: Нижний: Верхний:

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ **Диапазон значений**

Выбирается автоматически

Фиксированный

Нач. размер интервала:

▼ **Обновление данных**

Обновлять данные автоматически

Не обновлять данные автоматически

▼ **Описание**

Рисунок 28 Свойства объекта `стат_время_обслуживания`

В свойствах блока `обслуженные_клиенты` в качестве действия при входе в этот блок задайте выражение для добавления в объект `стат_время_обслуживания` значение вычисленного параметра клиента `время_обслуживания`, вошедшего в блок `обслуженные_клиенты`, используя метод `add(<значение>)` объекта `стат_время_обслуживания` (Рисунок 29):

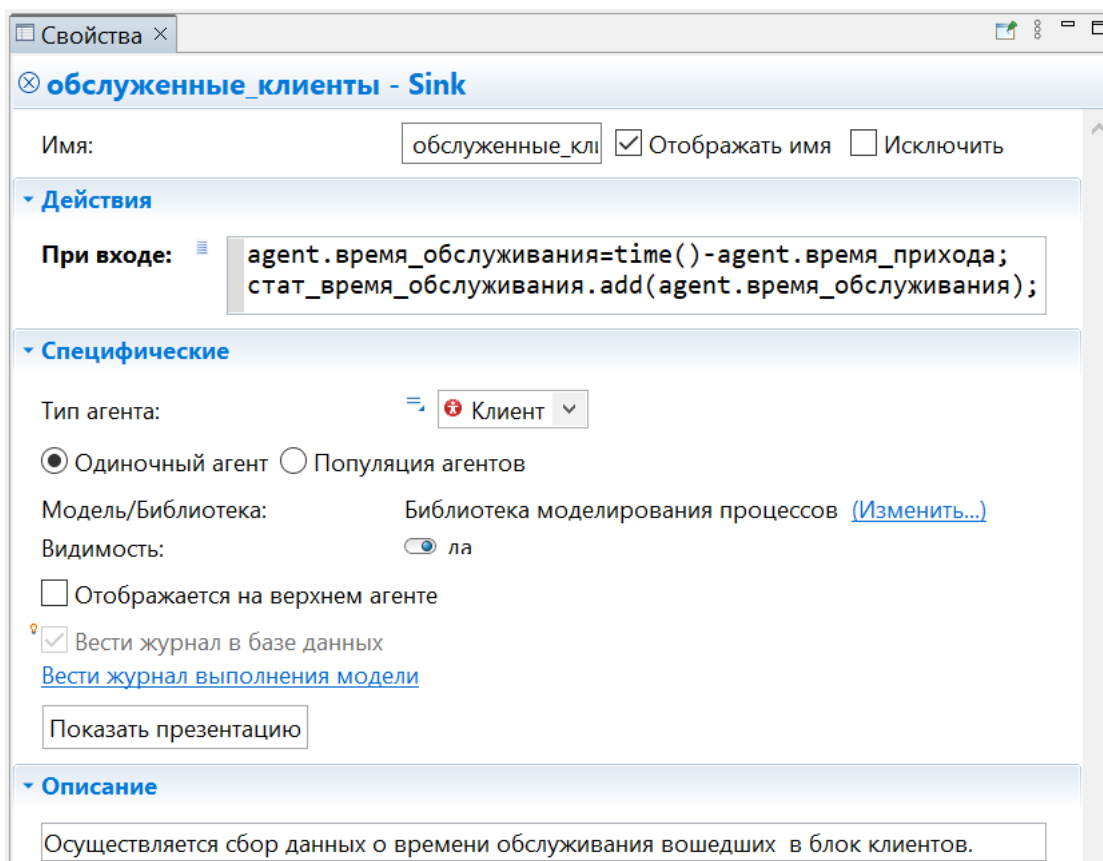


Рисунок 29 Сбор статистики времени обслуживания клиентов

Теперь добавьте в область диаграммы процесса обслуживания гистограмму для отображения распределения времён обслуживания. Чтобы добавить гистограмму, перетащите объект **Гистограмма** из набора **Статистика** в удобное место окна графического редактора и установите свойства гистограммы. В поле **Данные** задайте имя объекта набора статистических данных `стат_время_обслуживания` (Рисунок 30).

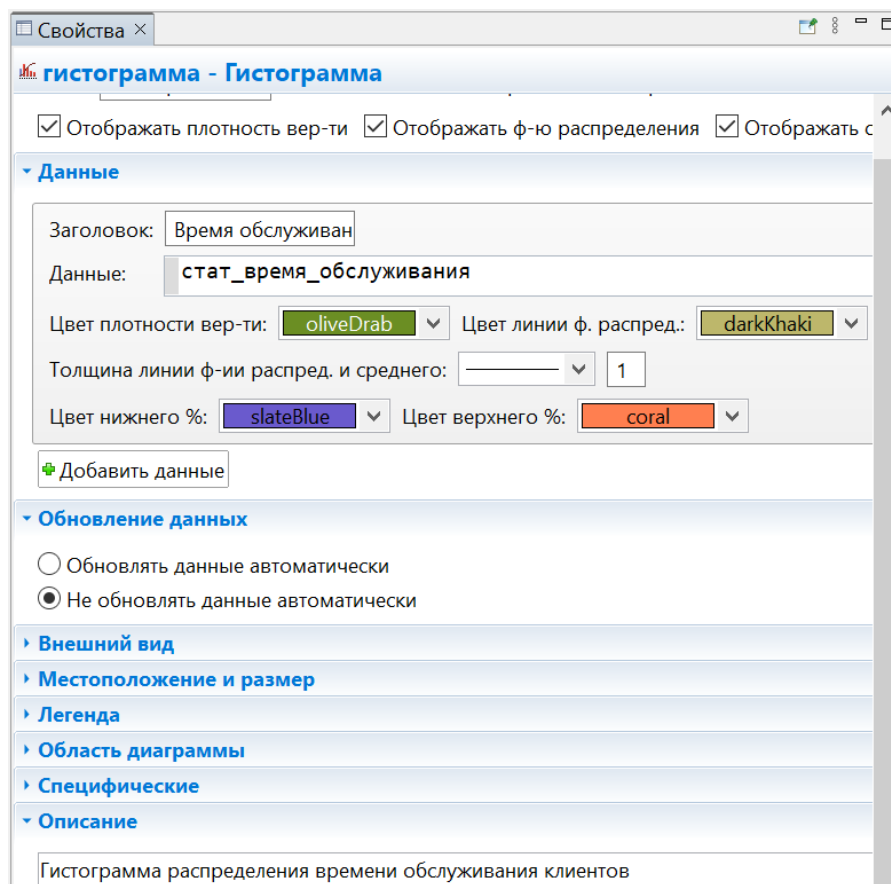


Рисунок 30 Свойства гистограммы времени обслуживания клиентов

Презентация эксперимента модели с гистограммой

Запустите модель. Проанализируйте работу распределителя потока клиентов, а также загрузку операторов и банкомата. Для ускорения выполнения модели переключите эксперимент в режим виртуального времени и посмотрите, как меняется вид гистограммы плотности и функции распределения вероятностей времени пребывания клиента в системе (), например:

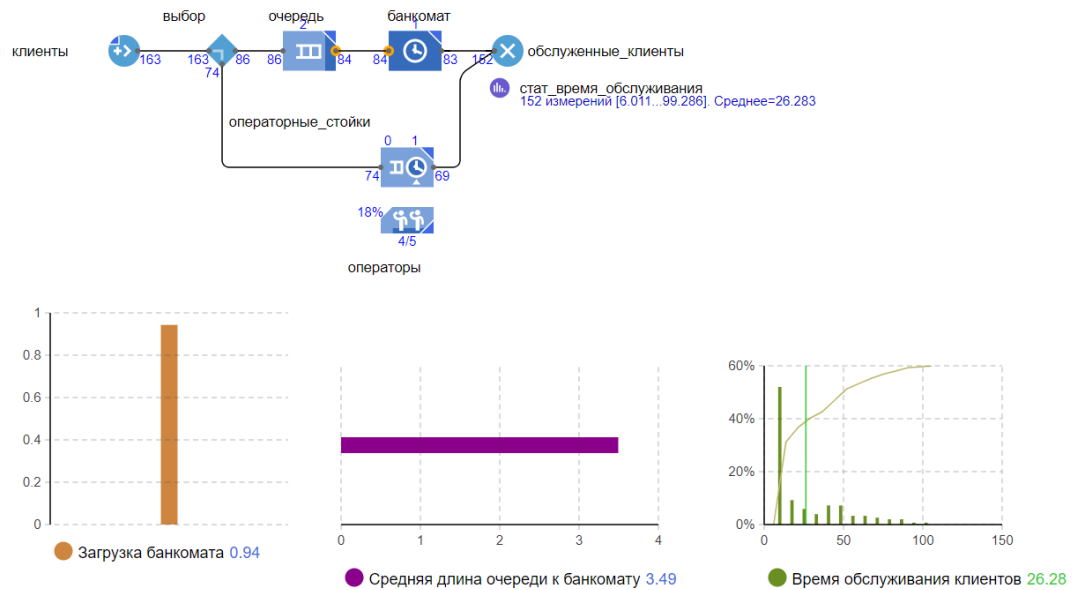


Рисунок 31 Презентация статистических данных процесса обслуживания

Оценка вероятности отказа в обслуживании

Статистическая оценка вероятности некоторого события равна отношению числа зафиксированных при испытаниях возникновений события к общему числу проведённых испытаний. Оценим вероятность того, что посетителю будет отказано в обслуживании (очереди к банкомату и операторным стойкам полны). В этом случае вероятность отказа будет равна отношению числа посетителей, получивших отказ к общему количеству посетителей банковского отделения.

Для подсчёта общего количества посетителей и общего количества отказов в обслуживании добавьте в агент Процесс_обслуживания переменные `общее_количество_посетителей` и `общее_количество_отказов` типа `double` с начальным значением 0.

В блоке **выбор** добавьте в поле действия При входе программный код `общее_количество_посетителей++`; (Рисунок 32):

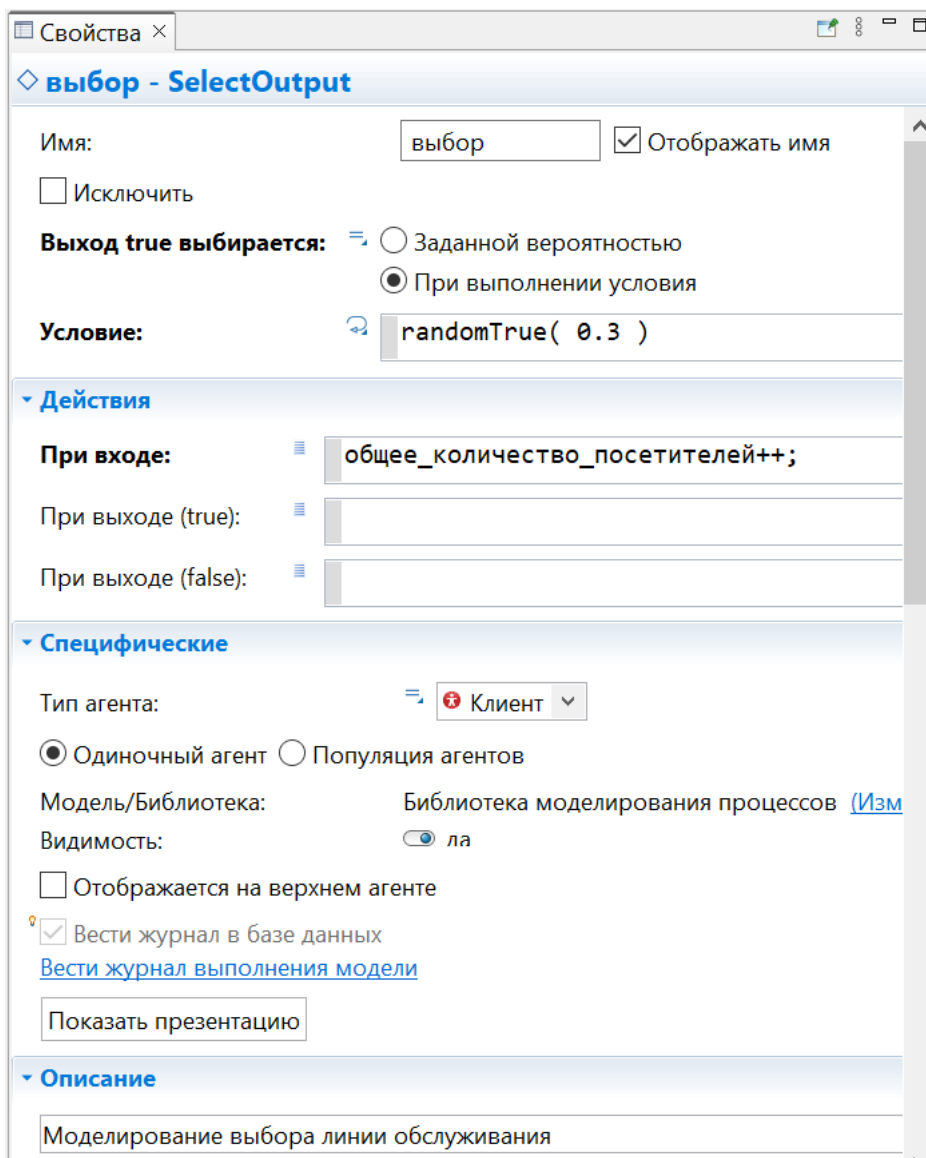


Рисунок 32 Подсчёт общего количества посетителей

В блоке **отказы** добавьте в поле действия При входе программный код `общее_количество_отказов++`; (Рисунок 33):

Текущее число посетителей банковского отделения будет подсчитываться и храниться в переменной `общее_количество_посетителей`, а общее количество отказов в обслуживании в переменной `общее_количество_отказов`.

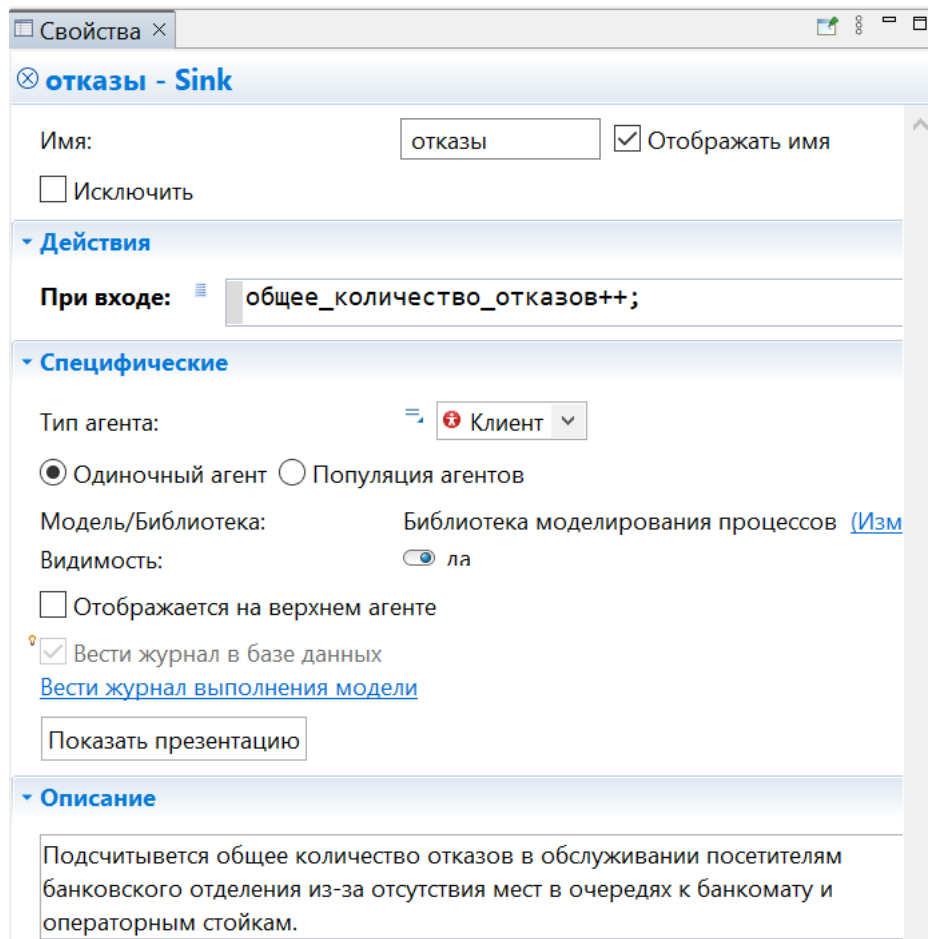
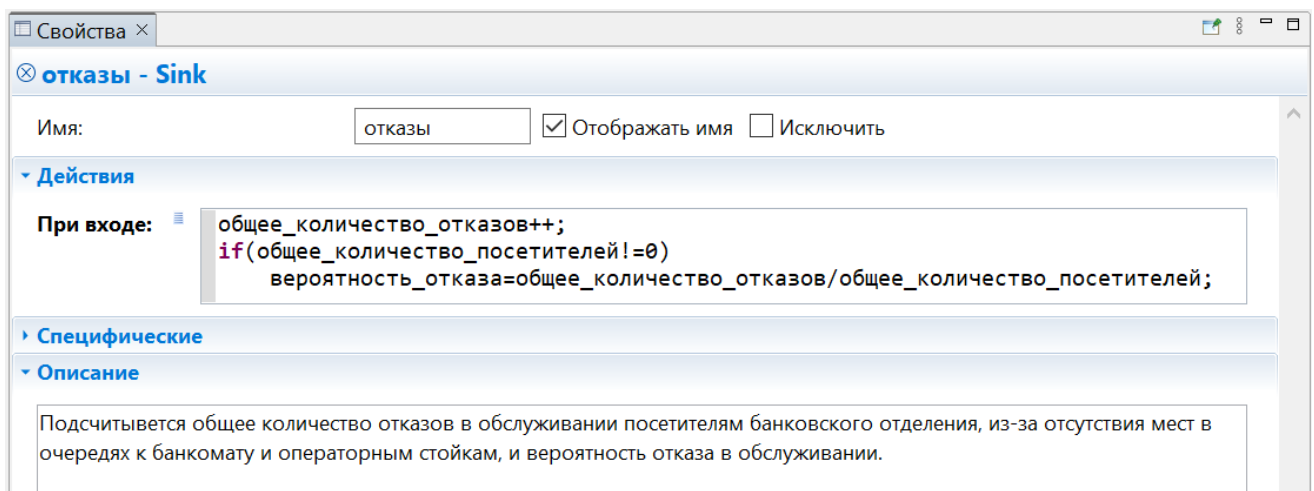


Рисунок 33 Подсчёт общего количества отказов в обслуживании

Теперь имеются все необходимые статистические данные для подсчёта вероятности отказа посетителю в обслуживании. Для подсчёта вероятности отказа создадим переменную `вероятность_отказа`. Значение этой переменной будем присваивать в поле действия **При входе** блока **отказы**.



Заметьте, что во избежание деления на ноль перед подсчётом выполняется проверка: не равно ли общее количество посетителей нулю. Для визуализации вероятности отказа на презентации воспользуйтесь объектом **Столбиковая диаграмма**, в свойствах которой в область **Данные** поместите переменную `вероятность_отказа` (Рисунок 34):

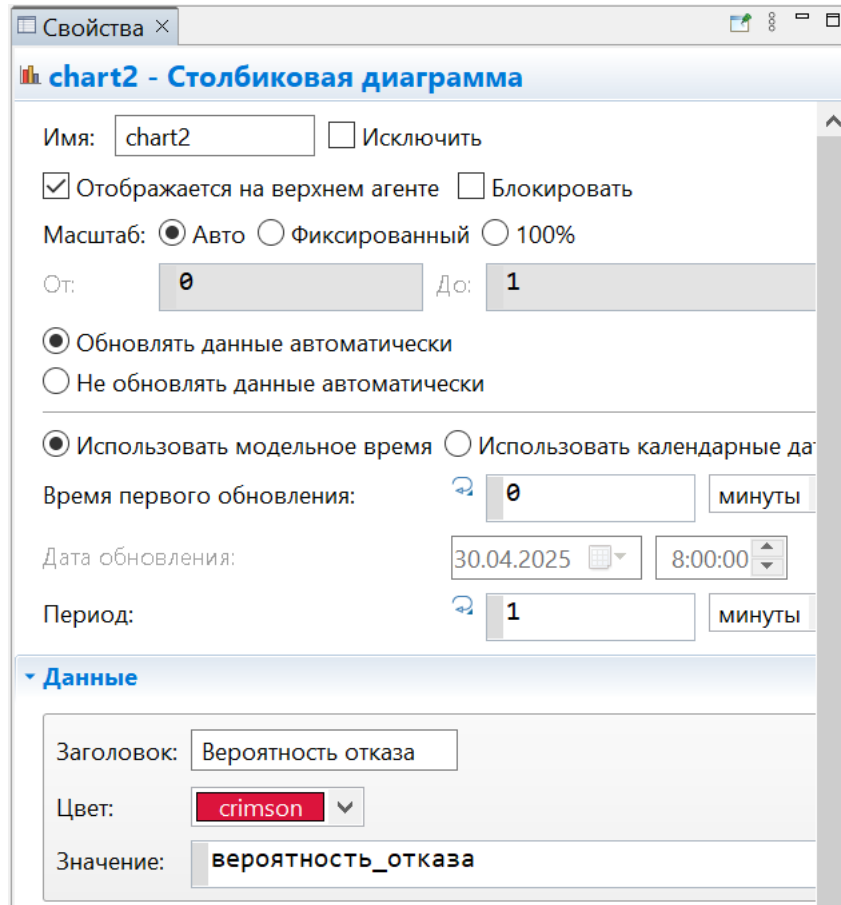


Рисунок 34 Диаграмма вероятности отказа в обслуживании

Оценка абсолютной пропускной способности банковского отделения

Абсолютная пропускная способность банковского отделения — это его текущая продуктивность равная *среднему* числу клиентов, которое оно обслуживает в единицу времени. Клиенты в модели завершают обслуживание в блоке **обслуженные_клиенты** класса **Sink**, там и следует производить расчёты. Создадим переменную `абсолютная_пропускная_способность`, которая будет содержать информацию о текущей продуктивности банковского отделения в момент времени t , которая вычисляется по формуле:

$$A(t) = \frac{\sum_{\Delta t} r}{\Delta t},$$

где Δt – время, истекшее с момента начала эксперимента с моделью, $\sum_{\Delta t} r$ – суммарное количество обслуженных и покинувших банковское отделение заявок в течение времени Δt . Для вычисления этой формулы необходимо выполнять следующее действие При входе клиента в блок **обслуженные_клиенты**:

```
абсолютная_пропускная_способность = обслуженные_клиенты.count()/time();
```

где long count() - метод, который возвращает количество агентов, уничтоженных блоком обслуженные_клиенты класса **Sink**. То есть, абсолютная пропускная способность будет равна отношению общего числа обслуженных и уничтоженных блоком обслуженные_клиенты клиентов к величине истекшего интервала времени моделирования time(). Для визуализации результата можно воспользоваться столбиковой диаграммой.

Оценка относительной пропускной способности банковского отделения

Относительная пропускная способность банковского отделения равна отношению абсолютной пропускной способности к среднему числу заявок, поступающих в банковское отделение за единицу времени. Абсолютная пропускная способность была посчитана в предыдущем пункте. Для вычисления относительной пропускной способности не хватает только среднего числа клиентов, поступающих в банковское отделение в единицу времени (интенсивность поступления), найдём его.

Добавте в область графического редактора агента Процесс_обслуживания переменную интенсивность_потока_посетителей – среднее число посетителей приходящих в банковское отделение в единицу времени. Для управления значением переменной интенсивность_потока_посетителей ей следует при входе в блок **обслуженные_клиенты** в качестве действия выполнять присвоение значения:

```
интенсивность_потока_посетителей = клиенты.count() / time();
```

Это следует выполнить именно в блоке **обслуженные_клиенты**, а не **клиенты**, так как в блоке **клиенты** в начальный момент времени функция `time()` вернёт 0, что приведёт к ошибке. Кроме того, это полезно тем, что будет достигаться согласованность значений относительной пропускной способности с абсолютной, которые вычисляются в одном блоке **обслуженные_клиенты**.

Теперь имеется всё необходимое для подсчёта блоке **обслуженные_клиенты** относительной пропускной способности. В результате при входе в блок **обслуженные_клиенты** будут выполняться все необходимые действия (Рисунок 35).

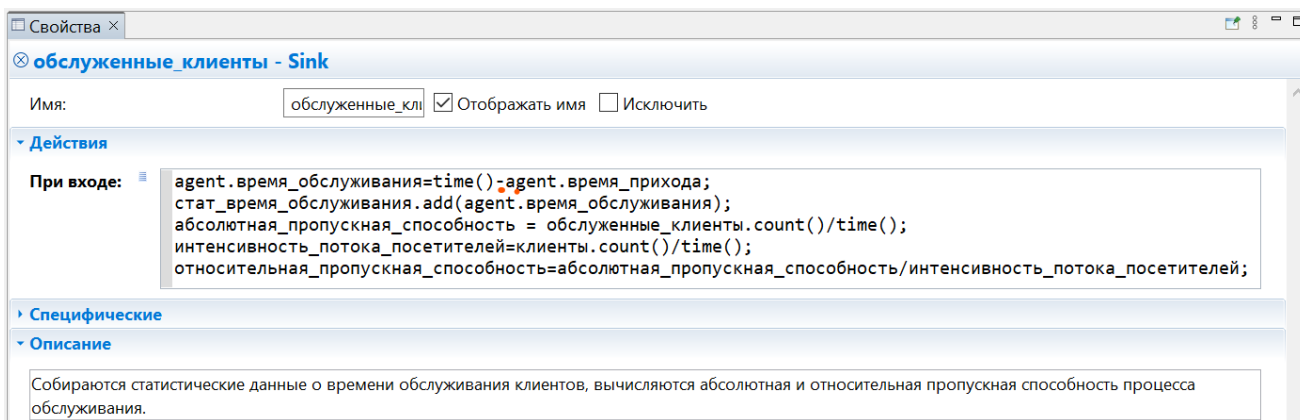


Рисунок 35 Действия при входе в блок обслуженные_клиенты

Значение относительной пропускной способности можно на презентации представить столбиковой диаграммой.

Оценка адекватности итоговой модели

Результат выполнения упражнений – это построенная модель банковского отделения с банкоматом и операторными стойками. В результате выполнения всех этапов построения модели состав компонентов агента Процесс_обслуживания будет выглядеть следующим образом (Рисунок 36):

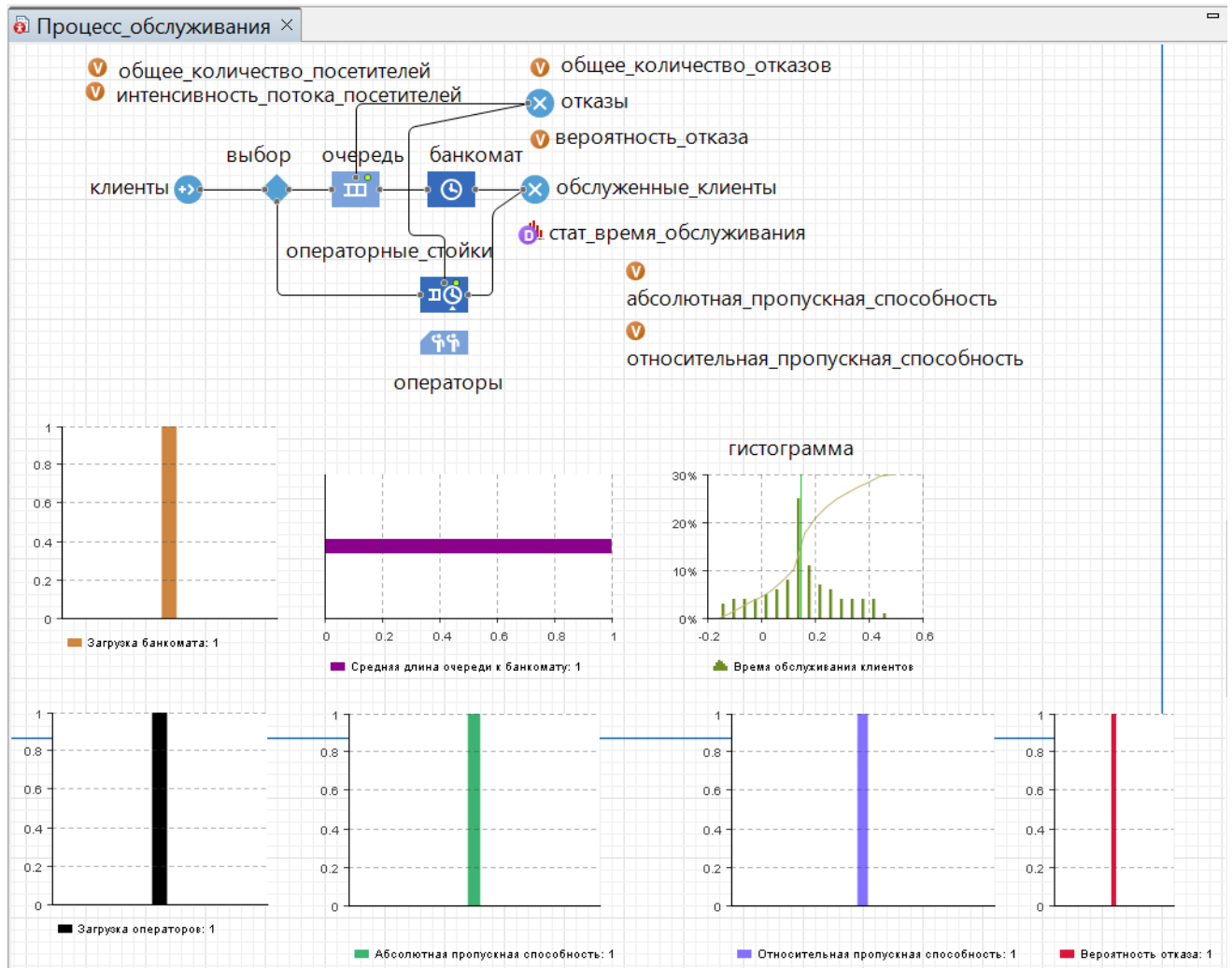


Рисунок 36 Итоговая модель процесса обслуживания

Теперь необходимо проверить адекватность модели, проверив выполнение известных теоретических свойств процессов обслуживания. Для этого необходимо:

1. Установить в блоке клиенты интенсивности потока посетителей равной нулю и проверить, что у банковского отделения относительная пропускная способность равна единице, абсолютная пропускная способность равна нулю, очереди клиентов равны нулю, вероятность отказа в обслуживании равна нулю.

2. Убедиться, что при увеличении интенсивности потока посетителей абсолютная пропускная способность начинает расти, при этом относительная пропускная способность до определённой интенсивности сохраняет значение равное единице, а длины очередей близки к нулю, вероятность отказа в обслуживании равна нулю.

3. Убедиться, что дальнейшее увеличение интенсивности прибытия посетителей в итоге приводит к стабилизации абсолютной пропускной способности, относительная пропускная способность становится меньше единицы и стремится к нулю, растёт количество клиентов в очереди, стремясь заполнить все места, вероятность отказа в обслуживании отлична от нуля и стремится к единице.

Оценка характеристик заданного варианта процесса обслуживания

Теперь необходимо самостоятельно изменить свойства модели в соответствии с полученным индивидуальным вариантом свойств (Приложение 1) и оценить характеристики процесса обслуживания:

- *вероятность отказа* клиенту в обслуживании банковским отделением;
- *время обслуживания* клиентов;
- *средняя длина очереди* к банкомату;
- *абсолютная пропускная способность* банковского отделения;
- *относительная пропускная способность* банковского отделения.

Оформление результатов работы

Отчёт о выполнении лабораторной работы должен содержать следующие разделы:

1. Титульный лист (см. приложение).
2. Вариант задания.
3. Диаграмма модели (скриншот).
4. Результаты статистической оценки характеристик процесса обслуживания посетителей банковским отделением, полученные на модели, модифицированной в соответствии с заданным вариантом свойств (скриншот).
5. Выводы относительно эффективности работы банковского отделения.

Оценочные вопросы

1. Характеристика непрерывных событийно-стохастических моделей класса Q-схем.
2. Основные понятия процессов обслуживания. Модели потоков событий.
3. Модель простейшего прибора обслуживания, характеристики эффективности его функционирования. Абсолютная и относительная пропускная способность прибора обслуживания.
4. Основные блоки панели **Палитра** AnyLogic для построения имитационных моделей процессов обслуживания.
5. Предназначение и характеристики блока **Source**.
6. Предназначение и характеристики блока **Queue**.
7. Предназначение и характеристики блока **Delay**.
8. Предназначение и характеристики блока **Sink**.
9. Роль связей между блоками в модели процессов обслуживания.
10. Использование окна инспекта для просмотра текущей статистики блоков модели.
11. Организация сбора и анализа статистических данных в AnyLogic.
12. Объекты **Статистика** панели **Палитра** для сбора и обработки статистики.
13. Предназначение и использование объекта **Данные гистограммы**.
14. Отображение статистики с помощью блока **Гистограмма**.
15. Средства панели **Палитра** для отображения статистических характеристик на презентации модели.
16. Понятие ресурсов и их использование в модели.
17. Предназначение и характеристика блока **Resource Pool**.
18. Использование ресурсов блоком **Service**.
19. Предназначение и характеристика блока **Select Output**.
20. Использование программного кода в диаграмме процессов обслуживания.
21. Вычисление в диаграмме абсолютной пропускной способности банковского отделения.

22. Вычисление в диаграмме относительной пропускной способности банковского отделения.
23. Оценка адекватности модели процессов обслуживания.

Приложение 1

ВАРИАНТЫ ПАРАМЕТРОВ МОДЕЛИ

Таблица 1

Вариант	Распределение вероятности периодов прихода клиентов в банк (мин)	Соотношение выбора кассир/банкомат $\text{randomTrue}(p)^4$	Распределение вероятности времени обслуживания клиента банкоматом (мин)	Распределение вероятности времени обслуживания клиента кассиром (мин)	Количество кассиров
1.	Экспоненциальное: $\mu=1/\lambda=1.0$	1/1	Эрланга: $\text{erlang}(5,2,0)$	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=3$;	2
2.	Экспоненциальное $\mu=1/\lambda=1.5$	1/2	Эрланга: $\text{erlang}(0,10,1)$	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=4$;	3
3.	Экспоненциальное $\mu=1/\lambda=2.0$	2/1	Равномерное $\text{min}=2$; $\text{max}=7.5$;	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=5$;	4
4.	Экспоненциальное $\mu=1/\lambda=2.5$	1/3	Равномерное $\text{min}=3$; $\text{max}=8.5$;	Треугольное $\text{min}=3$; $\text{max}=6.5$; $\text{mode}=4$;	5
5.	Экспоненциальное $\mu=1/\lambda=3.0$	3/1	Равномерное $\text{min}=4$; $\text{max}=6.5$;	Треугольное $\text{min}=2$; $\text{max}=8.5$; $\text{mode}=2.5$;	4
6.	Нормальное $\mu=2.0$; $\sigma=0.5$	1/1	Равномерное $\text{min}=3$; $\text{max}=6.5$;	Экспоненциальное $\mu=1/\lambda=3.0$	3
7.	Нормальное $\mu=2.5$; $\sigma=0.6$	1/2	Равномерное $\text{min}=2$; $\text{max}=6.5$;	Экспоненциальное $\mu=1/\lambda=2.5$	2
8.	Нормальное $\mu=3.0$; $\sigma=0.7$	2/1	Треугольное $\text{min}=2$; $\text{max}=8.5$; $\text{mode}=2.5$;	Экспоненциальное $\mu=1/\lambda=2.0$	3
9.	Нормальное $\mu=3.5$; $\sigma=0.8$	1/3	Треугольное $\text{min}=3$; $\text{max}=6.5$; $\text{mode}=4$;	Экспоненциальное $\mu=1/\lambda=1.5$	4
10.	Нормальное $\mu=4.0$; $\sigma=0.9$	3/1	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=5$;	Экспоненциальное $\mu=1/\lambda=1.0$	5
11.	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=3$;	1/1	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=4$;	Эрланга: $\text{erlang}(0,10,1)$	4
12.	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=4$;	1/2	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=3$;	Эрланга: $\text{erlang}(5,2,0)$	3
13.	Треугольное $\text{min}=2$; $\text{max}=6.5$; $\text{mode}=5$;	2/1	Нормальное $\mu=4.0$; $\sigma=0.9$	Равномерное $\text{min}=2$; $\text{max}=7.5$;	2
14.	Треугольное $\text{min}=3$; $\text{max}=6.5$; $\text{mode}=4$;	1/3	Нормальное $\mu=3.5$; $\sigma=0.8$	Равномерное $\text{min}=3$; $\text{max}=8.5$;	3
15.	Треугольное $\text{min}=2$; $\text{max}=8.5$; $\text{mode}=2.5$;	3/1	Нормальное $\mu=3.0$; $\sigma=0.7$	Равномерное $\text{min}=4$; $\text{max}=6.5$;	4
16.	Равномерное $\text{min}=2$; $\text{max}=6.5$;	1/1	Нормальное $\mu=2.5$; $\sigma=0.6$	Равномерное $\text{min}=3$; $\text{max}=6.5$;	5

⁴ Использовать в блоке **Select Output**. Генерирует True с вероятностью p.

17.	Равномерное min=3; max=6.5;	1/2	Нормальное $\mu=2.0$; $\sigma=0.5$	Равномерное min=2; max=6.5;	4
18.	Равномерное min=4; max=6.5;	2/1	Экспоненциальное $\mu=1/\lambda=3.0$	Треугольное min=2; max=6.5; mode=4;	3
19.	Равномерное min=3; max=8.5;	1/3	Экспоненциальное $\mu=1/\lambda=2.5$	Треугольное min=2; max=8.5; mode=2.5;	2
20.	Равномерное min=2; max=7.5;	3/1	Экспоненциальное $\mu=1/\lambda=2.5$	Треугольное min=2; max=6.5; mode=5;	3
21.	Эрланга: erlang(5,2,0)	1/1	Экспоненциальное $\mu=1/\lambda=1.0$	Треугольное min=2; max=6.5; mode=4;	4
22.	Эрланга: erlang(0,10,1)	1/2	Экспоненциальное $\mu=1/\lambda=1.5$	Треугольное min=2; max=6.5; mode=3;	5



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

**ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ
Кафедра программных систем**

**Дисциплина
Моделирование информационных процессов и систем**

**ОТЧЕТ
по лабораторной работе**

**Моделирование непрерывных
событийно-стохастических систем
в AnyLogic**

Вариант № _____

Выполнил: Фамилия И.О., группа № 6XXX-XXXXXXX

Проверил: Фамилия И.О.

Самара 2025