Лабораторная работа № 2

ИСПОЛЬЗОВАНИЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ

Лабораторная работа состоит из 2х частей. В первой части рассматривается модель простой СМО с очередью, на примере работы магазина с терминалом самообслуживания, во второй части — модель дополняется кассирами, которые обслуживают покупателей с большим количеством товаров или с оплатой наличными.

ЦЕЛЬ РАБОТЫ: научиться использовать средства имитационного моделирования для подбора эффективных параметров работы магазина на основе теории массового обслуживания и законов распределения случайных величин.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Случайная величина — это величина, принимающая в результате испытания одно из возможных значений, при этом появление того или иного значения является случайным событием.

Законом распределения случайной величины называется всякое соотношение, устанавливающее связь между возможными значениями случайной величины и соответствующими им вероятностями.

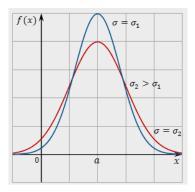
Математическое ожидание случайной величины (M(x)=m) – среднее значение случайной величины, равное сумме произведений всех возможных значений случайной величины на их вероятности

Дисперсия случайной величины $(D(x)=\sigma^2)$ — математическое ожидание квадрата ее отклонений от математического ожидания.

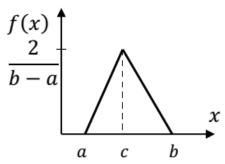
Среднее квадратическое отклонение случайной величины (σ)- корень квадратный из ее дисперсии

Нормальное распределение - это неограниченное непрерывное распределение. Иногда его называют Гауссовым распределением или колоколообразной кривой. Поскольку оно способно описать возрастающую сумму малых независимых ошибок, нормальное распределение используется во множестве статистических расчетов. Однако существует большое количество случаев его неоправданного использования.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-m)^2}{2\sigma^2}}.$$



Треугольное распределение - это непрерывное распределение, ограниченное с обеих сторон.



Функция плотности вероятностей треугольного распределения имеет вид треугольника с основанием аb и точкой с, лежащей где-то на этом основании и задается выражением:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{(x-a)^2}{(b-a)(c-a)} & x \in [a,c] \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & x \in [c,b] \\ 1 & x > b \end{cases}$$

Математическое ожидание

$$M(x) = \frac{a+b+c}{3}$$

Дисперсия:

$$D(x) = \frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$$

Треугольное распределение часто применяется при недостаточной информации или полном ее отсутствии. Оно редко может точно представить набор значений. Несмотря на это, благодаря простоте использования оно употребляется в качестве функциональной формы представления областей с размытой логикой.

Непрерывная случайная величина имеет **равномерное распределение** на отрезке [a, b], если на этом отрезке плотность распределения случайной величины постоянна, а вне его равна нулю:

$$f(x) = \begin{cases} 0, & \text{при } x < a; \\ \frac{1}{b-a}, & \text{при } a \le x \le b; \\ 0, & \text{при } x > b. \end{cases}$$

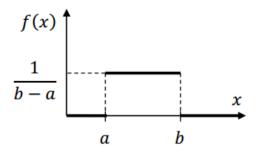
Математическое ожидание

$$m_{x} = \frac{a+b}{2};$$

Дисперсия

$$D_x = \frac{(b-a)^2}{12}$$

Равномерное распределение используется для задания случайной переменной, которая может принимать любое значение в интервале между а и b с равной вероятностью. Обратите внимание, что вероятность максимального значения равна 0.



Экспоненциальное распределение - это непрерывное распределение, ограниченное снизу. Его форма всегда остается неизменной: оно начинается с конечного значения при минимальном значении аргумента λ и непрерывно уменьшается при увеличении х. С увеличением х скорость уменьшения экспоненциального распределения возрастает.

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} &, x \ge 0, \\ 0 &, x < 0. \end{cases}$$
$$f(x) = \lambda e^{-\lambda x}, \quad x \ge 0.$$
$$f(x) \uparrow \lambda$$

Математическое ожидание

$$M(X) = \frac{1}{\lambda};$$

 \boldsymbol{x}

Дисперсия

$$D(X) = \frac{1}{\lambda^2}$$

Экспоненциальное распределение часто используется для представления промежутка времени между случайными событиями, например, времени между прибытиями заявок в модели СМО или времени между отказами в моделях надежности. Оно применялось также для описания времени выполнения определенной операции. Кроме этого, оно может служить явным способом описания временных зависимостей шума. Во всех этих моделях явным образом используется отсутствие зависимости экспоненциального распределения от предыстории: при сдвиге во времени значения определяемых им вероятностей не изменяются. Даже в тех случаях, когда известно, что экспоненциальные модели описывают ситуацию неточно, на начальном этапе основную роль играет удобство их математической обработки. В дальнейшем могут рассматриваться более сложные распределения, такие как Эрланга или Вейбулла.

Функции вероятностных распределений, используемые в среде AnyLogic:

AnyLogic поддерживает большое количество разных вероятностных распределений. Чтобы получить случайное значение, сгенерированное согласно закону вероятностного распределения, нужно вызвать соответствующий метод, например, exponential (0.6).

normal(double sigma, double mean) - генерирует значение согласно нормальному распределению.

Параметры

Имя	Тип	Описание		
sigma double		параметр формы = стандартное отклонение		
mean double		параметр сдвига = среднее значение		

triangular(double min, double max, double mode) - генерирует значение согласно треугольному распределению.

Параметры

Имя	Тип	Описание		
min	double	минимальное значение х		
max double максимальное значение х		максимальное значение х		
mode double наг		наиболее вероятное значение х		

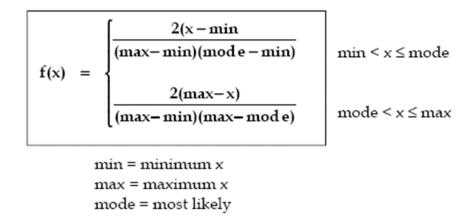
Эта функция автоматически проверяет, принадлежит ли наиболее вероятное значение х (mode) определенному интервалу (min, max). При превышении указанного максимального значения функция рассматривает его как максимальное, и наоборот: max рассматривается как mode, то есть обращение к функции triangular (1, 5, 10) эквивалентно triangular (1, 10, 5): при этом генерируется вариант треугольного распределения с минимальным значением 1, максимальным значением 10 и наиболее вероятным значением 5.

triangular(double min, double max) - генерирует значение согласно треугольному распределению с параметром mode=(min max)/2. Эквивалентно triangular(min, (min max)/2, max).

Параметры

Имя	Тип	Описание	
min	double	минимальное значение х	
max double максимальное значение х		максимальное значение х	

Форма треугольного распределения может быть очень асимметричной, включая отрицательную асимметрию. В исключительных случаях, когда mode (наиболее вероятное значение х) равно min или max, треугольное распределение принимает вид прямоугольного треугольника.



uniform(double min, double max) - генерирует случайное значение, равномерно распределенное в интервале [min, max), верхняя граница интервала не включена. Плотность вероятности не зависит от значения х.

Параметры

Имя	Тип	Описание
min	double	минимальное значение х
max	double	максимальное значение х

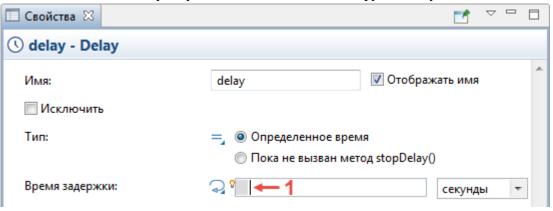
exponential(double lambda, double min) - генерирует значение согласно экспоненциальному распределению.

Параметры

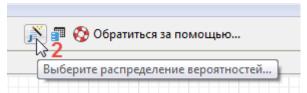
Имя	Тип	Описание
lambda	double	параметр формы
min	double	минимальное значение х

Чтобы вставить вызов функции вероятностного распределения с помощью мастера необходимо:

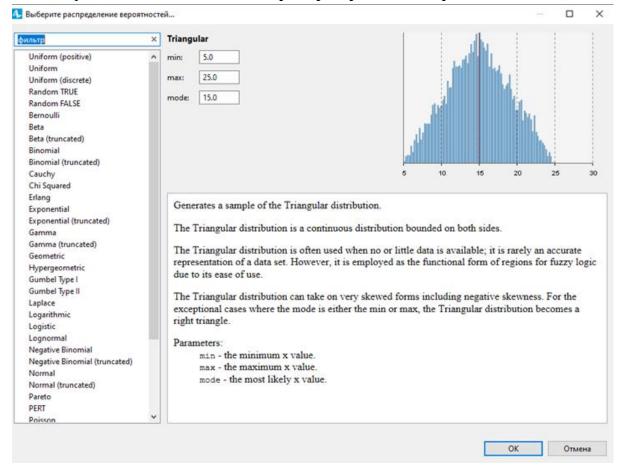
1. Щелкнуть в поле, в которое вы планируете вставить вызов функции вероятностного распределения. Если в поле уже указан вызов какой-либо функции, удалите его.



2. Нажать на кнопку панели инструментов **Выберите распределение** вероятностей...:



3. Откроется диалоговое окно Выберите распределение вероятностей...:



ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

- 1. Разобраться с теоретическими основами теории массового обслуживания и законами распределения случайных чисел для моделирования потоков СМО.
- 2. Выполнить общее задание из примера, а затем индивидуальное задание по варианту, предлагаемому преподавателем.
- 3. Оформить отчет с описанием хода выполнения работы и копиями экранов, поясняющих действия. Сделать выводы по работе построенной модели.
 - 4. Продемонстрировать работу преподавателю. Ответить на контрольные вопросы.

Лабораторная работа №2.1

МОДЕЛИРОВАНИЕ ПРОСТЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С ОЧЕРЕДЬЮ

ЦЕЛЬ РАБОТЫ: научиться средствами среды AnyLogic строить имитационную модель простой системы массового обслуживания и исследовать эффективность ее работы на примере модели магазина с терминалом самообслуживания.

ПОСТАНОВКА ЗАДАЧИ

В магазине находится один терминал самообслуживания, который предназначен для быстрого и эффективного обслуживания посетителей магазина. Покупатели прибывают в магазин с интенсивностью равной λ в минуту и становятся в очередь. Длина очереди к терминалу максимальная. Предположим, что время, которое покупатель тратит в терминале самообслуживания распределено по заданному в исходных данных (табл. 2.1) закону распределения случайных чисел.

Необходимо:

- построить имитационную дискретно-событийную модель в среде Anylogic согласно индивидуальным исходным данным (табл. 2.1), предусмотрев анимацию;
- проанализировать работу терминала самообслуживания, согласно использованного вероятностного распределения времени обслуживания покупателей;
 - проанализировать поведение модели на входящем в магазин 50-м и 200-м покупателе.
- произвести оценку целесообразности установки дополнительных терминалов самообслуживания, для уменьшения скопления людей в магазине*.

*Время проведения исследования можно варьировать от 2 до 6 часов работы магазина, исходя из ограничений учебной версии Anylogic.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ В СРЕДЕ ANYLOGIC

Постановка задачи

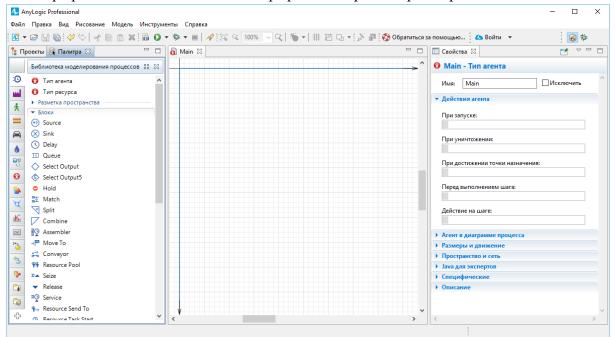
Необходимо в среде Anylogic построить имитационную дискретно-событийную модель магазина, в котором находится терминал самообслуживания, предназначенный для быстрого и эффективного обслуживания посетителей магазина. Покупатели прибывают в магазин с интенсивностью равной 0,3 в минуту. Каждый новый покупатель становятся в очередь, которая не должна превышать 15 человек. Предположим, что время, которое покупатель тратит в терминале самообслуживания распределено по треугольному закону со средним значением, равным 1.5, минимальным - равным 0.8 и максимальным - 3.5 минутам.

Создание новой модели:

- 1. Щелкните мышью по кнопке панели инструментов **Создать .** Появится диалоговое окно **Новая модель**.
 - 2. Задайте имя новой модели. В поле **Имя модели** введите Magazin.
- 3. Выберите каталог, в котором будут сохранены файлы модели. Если вы хотите сменить предложенный по умолчанию каталог на какой-то другой, вы можете ввести путь к нему в поле **Местоположение** или выбрать этот каталог с помощью диалога навигации по файловой системе, открывающегося по нажатию на кнопку **Выбрать**.
 - 4. Выберите минуты в качестве Единиц модельного времени.
 - 5. Щелкните мышью по кнопке Готово, чтобы завершить процесс.

Вы создали новую модель. В ней уже имеется один тип агента Main и эксперимент Simulation. Агенты - это главные строительные блоки модели AnyLogic. В нашем случае агент Main послужит местом, где мы зададим всю логику модели: здесь мы расположим чертеж магазина и зададим диаграмму процесса потока покупателей.

В центре рабочей области находится графический редактор диаграммы типа агента Main.



В левой части рабочей области находятся панель **Проекты** и панель **Палитра**. Панель **Проекты** обеспечивает легкую навигацию по элементам моделей, открытых в текущий момент времени. Поскольку модель организована иерархически, то она отображается в виде дерева. Панель **Палитра** содержит разделенные по палитрам элементы, которые могут быть добавлены на диаграмму типа агента или эксперимента.

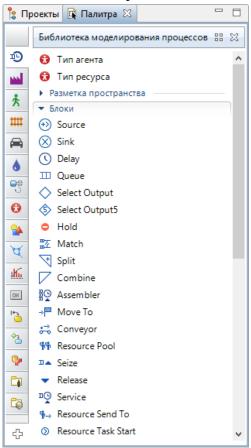
В правой рабочей области будет отображаться панель Свойства. Панель Свойства используется для просмотра и изменения свойств выбранного в данный момент элемента (или элементов) модели. Когда вы выделяете какой-либо элемент, например, в панели Проекты или графическом редакторе, панель Свойства показывает свойства выбранного элемента.

Создание диаграммы процесса

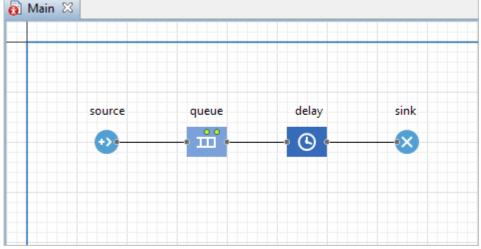
Каждый блок <u>Библиотеки моделирования процессов</u> задает определенную операцию, которая будет производиться над проходящими по диаграмме процесса агентами.

Диаграмма процесса в AnyLogic создается путем добавления объектов библиотеки из палитры на диаграмму агента, соединения их портов и изменения значений свойств блоков в соответствии с требованиями вашей модели.

1. По умолчанию при создании новой модели в панели **Палитра** открывается **Библиотека моделирования процессов**. Вы можете открывать палитры щелчком по соответствующей иконке на вертикальной панели слева от палитры:



2. Добавьте блоки **Библиотеки моделирования процессов** на диаграмму и соедините их, как показано на рисунке. Чтобы добавить объект на диаграмму, перетащите требуемый элемент из палитры в графический редактор.

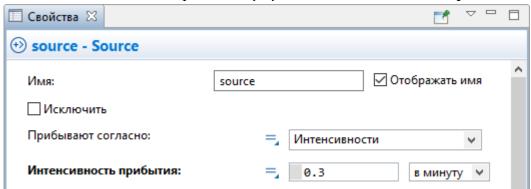


- 3. Когда вы перетаскиваете блоки и располагаете их рядом друг с другом, вы можете видеть, как появляются соединительные линии между блоками. Будьте внимательны, эти линии должны соединять только порты, находящиеся с правой или левой стороны иконок.
- Данная схема моделирует простейшую систему очереди, состоящую из источника агентов, задержки (и очереди перед задержкой) и финального уничтожения агентов:
- ⊕ Объект <u>Source</u> генерирует агентов определенного типа. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток агентов. В нашем примере агентами будут покупатели, а объект **Source** будет моделировать их приход в магазин.

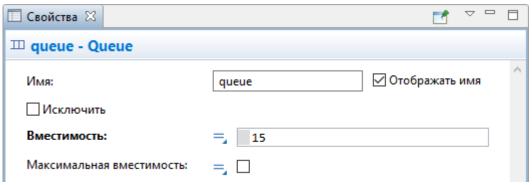
- ^Ⅲ Объект <u>Queue</u> моделирует очередь агентов, ожидающих приема объектами, следующими за данным в диаграмме процесса. В нашем случае он будет моделировать очередь покупателей, ждущих освобождения терминала самообслуживания.
- Объект <u>Delay</u> задерживает агентов на заданный период времени, представляя в нашей модели терминал самообслуживания, у которого покупатель тратит свое время на проведение необходимой ему операции по оплате выбранных товаров.
- ⊗ Объект <u>Sink</u> уничтожает поступивших агентов. Обычно он используется в качестве конечной точки потока агентов (и диаграммы процесса соответственно).

Настройка блоков диаграммы

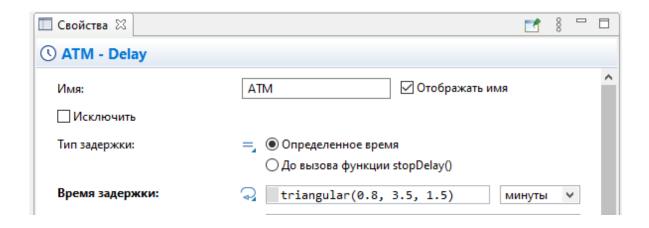
- 1. Чтобы изменить свойства элемента, выделите элемент в графическом редакторе или в панели **Проекты**, щелкнув по нему мышью. Свойства элемента откроются в панели **Свойства**.
- 2. Выделите блок *source*. В панели **Свойства** укажите, как часто должны прибывать покупатели. Введите *0.3* и выберите *в минуту* в поле **Интенсивность прибытия**.



3. Измените свойства блока *queue*. Введите в поле **Вместимость** *15*. В очереди будут находиться не более 15 человек.



4. Измените свойства блока *delay*. Назовите объект *ATM*. Задайте время обслуживания в поле **Время задержки**, распределенное по треугольному закону со средним значением, равным 1.5, минимальным - равным 0.8 и максимальным - 3.5 минутам.



Функция triangular()является стандартной функцией генератора случайных чисел AnyLogic с треугольным распределением. AnyLogic предоставляет функции и других случайных распределений, таких как нормальное (функция normal()), равномерное (функция uniform()), экспоненциальное (функция exponential()) и др.

Запуск модели

Мы закончили моделирование простейшей системы очереди и готовы запустить созданную модель. Сначала постройте вашу модель с помощью кнопки панели инструментов построить модель (при этом в рабочей области AnyLogic должен быть выбран какой-то элемент именно этой модели). Если в модели есть какие-нибудь ошибки, то построение не будет завершено, и в панель Ошибки будет выведена информация об ошибках, обнаруженных в модели. Двойным щелчком мыши по ошибке в этом списке вы можете перейти к месту ошибки, чтобы исправить ее.

После того, как вы исправите все ошибки и успешно построите вашу модель, вы можете ее запустить. Запуская модель, вы автоматически обновляете ее.

1. Щелкните мышью по кнопке панели инструментов **○ Запустить** и выберите из открывшегося списка эксперимент, который вы хотите запустить. Эксперимент этой модели будет называться Magazin/Simulation.

На момент запуска этого конкретного эксперимента наша модель - единственная открытая модель в рабочем пространстве. В дальнейшем будет запускаться тот эксперимент, который запускался вами в последний раз. Чтобы выбрать какой-то другой эксперимент, вам будет нужно щелкнуть правой кнопкой мыши по этому эксперименту в панели **Проекты** и выбрать Запустить из контекстного меню.

Запустив модель, вы увидите <u>окно модели</u>. В нем будет отображена презентация агента верхнего уровня модели. По умолчанию это тип агента Main.

Для каждой модели, созданной с помощью объектов **Библиотеки моделирования процессов**, автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой вы можете изучить текущее состояние модели, например, длину очереди, количество обслуженных человек и так далее.

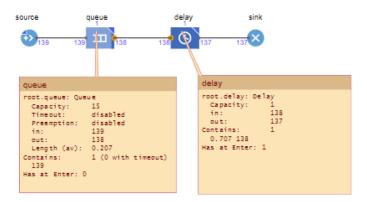


Вы можете изменить скорость выполнения модели с помощью кнопок панели инструментов Замедлить и Ускорить.

Вы можете следить за состоянием любого блока диаграммы процесса во время выполнения модели с помощью окна инспекта этого объекта. Чтобы открыть окно инспекта, щелкните мышью по значку блока. В окне инспекта будет отображена базовая информация по выделенному блоку: например, для блока **Queue** будет отображена вместимость очереди, количество агентов, прошедших через каждый порт объекта, и т.д.

Строка *Содержит* отображает количество агентов, находящихся в данный момент на объекте вместе с ID этих агентов.

На рисунке ниже видно, что 1 человек стоит в очереди, а 138 человек покинули очередь (блок queue), из них 137 обслужили (блок sink), а один еще обслуживается у терминала самообслуживания (блок delay).



Создание анимации модели

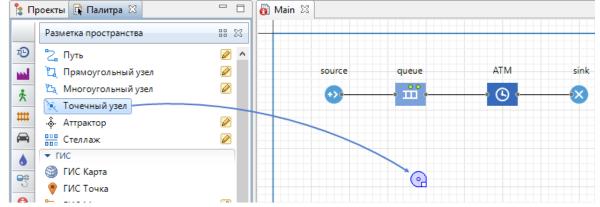
Хотя мы и могли анализировать работу запущенной нами только что модели с помощью диаграммы процесса, но куда удобнее было бы иметь более наглядную анимацию моделируемого нами процесса. В этом примере мы хотим создать визуализированный план магазина.

Поскольку в нашем случае нас не интересует конкретное расположение объектов в пространстве, то мы можем просто добавить чисто схематическую анимацию интересующих нас объектов - в нашем случае мы хотим видеть на анимации терминал самообслуживания и ведущую к нему очередь покупателей.

Анимация модели рисуется в той же диаграмме (в графическом редакторе), в которой задается и диаграмма моделируемого процесса.

Задайте фигуру анимации терминала самообслуживания:

- 1. Нарисуем <u>точечный узел</u>, обозначающий термина самообслуживания. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
- 2. Перетащите элемент **Точечный узел** узел налитры **Разметка пространства** в графический редактор и поместите его под блок-схемой процесса.

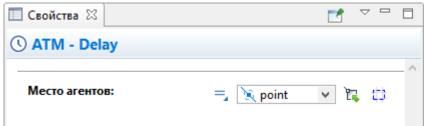


3. Выделите щелчком точечный узел в графическом редакторе, чтобы открыть для него панель Свойства. Мы с вами хотим, чтобы во время моделирования менялся цвет нашей фигуры,

поэтому введите выражение, которое будет постоянно вычисляться заново при выполнении модели, в поле **Цвет**: ATM.size() > 0? red : green . Здесь ATM – это имя нашего объекта *Delay*.

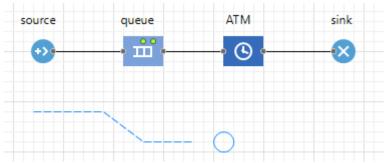
Функция **size**() возвращает число человек, обслуживаемых в данный момент времени. Если терминал самообслуживания занят, то цвет кружка будет красным, в противном случае - зеленым.

- 4. Выделите щелчком блок *delay*, названный нами ATM, в диаграмме процесса, чтобы открыть его свойства.
- 5. Выберите точечный узел *point*, который мы только нарисовали в параметре **Место агентов**. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).



Задайте фигуру анимации очереди к терминалу самообслуживания

- 1. Нарисуем <u>путь</u>, обозначающий очередь к терминалу самообслуживания. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
- 2. Двойным щелчком выделите элемент **Путь 2** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
- 3. Теперь вы можете рисовать путь точка за точкой, последовательно щелкая мышью в тех точках диаграммы, куда вы хотите поместить вершины линии. Чтобы завершить рисование, добавьте последнюю точку пути двойным щелчком мыши.



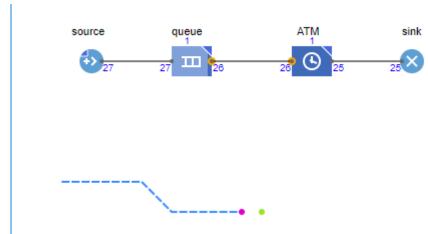
- 4. Выделите щелчком блок *queue* в диаграмме процесса, чтобы открыть для него панель **Свойства**.
- 5. Выберите путь *path*, который мы только нарисовали в параметре **Место агентов**. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).



Теперь вы можете запустить модель и изучить ее поведение. Для ускорения работы модели, переключитесь в режим виртуального времени, щелкнув мышью по кнопке панели инструментов **Реальное/виртуальное время**. В режиме виртуального времени модель будет выполняться максимально быстро, без привязки модельного времени к реальному.

Запуск модели.

Запустите модель. Вы увидите, что у вашей модели теперь есть простейшая анимация — терминал самообслуживания и ведущая к нему очередь покупателей. Цвет фигуры терминала самообслуживания будет меняться в зависимости от того, обслуживается ли покупатель в данный момент времени.



Вы можете самостоятельно, по желанию, добавить свои элементы анимации в данную модель!!

ИНДИВИДУАЛЬНЫЕ ИСХОДНЫЕ ДАННЫЕ К ЛАБОРАТОРНОЙ РАБОТЕ 2.1

ТАБЛИЦА 2.1 – Индивидуальные исходные данные для модели магазина с терминалом самообслуживания.

Вариант	Интенсивность прихода покупателей в магазин, λ	Вероятностное распределение времени обслуживания покупателей терминалом самообслуживания		
1	0,91	Равномерное (2, 6)		
2	0,83	Равномерное (4, 8)		
3	0,77	Равномерное (6, 10)		
4	0,63	Равномерное (5, 9)		
5	0,48	Равномерное (2, 7)		
6	0,40	Равномерное (1, 5)		
7	0,36	Равномерное (4, 8)		
8	0,33	Равномерное (3, 7)		
9	0,29	Экспоненциальное (1, 3)		
10	0,26	Экспоненциальное (2, 4)		
11	0,91	Экспоненциальное (4, 6)		
12	0,83	Экспоненциальное (1, 4)		
13	0,77	Экспоненциальное (1, 2)		
14	0,71	Экспоненциальное (2, 5)		
15	0,67	Экспоненциальное (1, 3)		
16	0,59	Нормальное (1, 5)		
17	0,53	Нормальное (0, 4)		
18	0,45	Нормальное (2, 4)		
19	0,34	Нормальное (4, 8)		
20	0,30	Нормальное (3, 7)		
21	0,28	Нормальное (1, 5)		
22	0,91	Нормальное (2, 6)		
23	0,77	Нормальное (4, 6)		
24	0,71	Треугольное (5, 9, 7)		
25	0,63	Треугольное (3, 7, 5)		
26	0,48	Треугольное (1, 5, 3)		
27	0,45	Треугольное (3, 9, 6)		
28	0,40	Треугольное (2, 6, 4)		
29	0,36	Треугольное (4, 8, 6)		
30	0,34	Треугольное (5, 9, 6)		

Лабораторная работа 2.2

МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С ОЧЕРЕДЬЮ

ЦЕЛЬ РАБОТЫ: научиться создавать дискретно-событийные модели в среде AnyLogic и использовать средства имитационного моделирования для подбора эффективных параметров работы магазина с терминалом самообслуживания и кассирами на основе теории массового обслуживания и законов распределения случайных величин.

ПОСТАНОВКА ЗАДАЧИ

В магазине находятся терминал самообслуживания и стойки кассиров, которые предназначены для быстрого и эффективного обслуживания покупателей магазина. Быстрые операции с безналичным расчетом покупатели производят с помощью терминала самообслуживания, а более сложные операции, такие как покупка большого количества товаров или наличная оплата покупок – с помощью кассиров.

Предположим, что вероятность прихода покупателей в магазин распределено по заданному в исходных данных закону распределения случайных чисел.

Кассиров в магазине — Kas. Ко всем кассирам будет вести одна общая очередь с максимальным количеством человек в этой очереди M_{o} . Вероятность обращения к кассиру $P_{oбp}$. Время обслуживания кассиром покупателя имеет треугольное распределение с заданным интервалом распределения случайной величины.

Необходимо:

- видоизменить имитационную модель лабораторной работы 2.1, **ДОБАВИВ** некоторые условия и параметры, указанные выше, согласно варианта индивидуального задания к лабораторной работе 2.2 (см. табл.2.2). Настройки работы терминала самообслуживания, выполненные по индивидуальным данным в лабораторной работе 2.1, остаются без изменений;
- построить графики и диаграммы переменных, характеризующих поведение модели, и поместите их в отчет;
 - создать анимацию для наглядного наблюдение за работай магазина;
 - проанализировать поведение модели на входящем в магазин 10-м и 100-м покупателе.
- определите оптимальное количество терминалов самообслуживания и кассиров в исследуемом магазине*.

*Время проведения исследования можно варьировать от 2 до 6 часов работы магазина, исходя из ограничений учебной версии Anylogic.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ В СРЕДЕ ANYLOGIC

ПОСТАНОВКА ЗАДАЧИ:

В магазине находятся терминал самообслуживания и стойки кассиров, которые предназначены для быстрого и эффективного обслуживания покупателей магазина. Быстрые операции с безналичным расчетом покупатели производят с помощью терминала самообслуживания, а более сложные операции, такие как покупка большого количества товаров или наличная оплата покупок – с помощью кассиров.

Требуется видоизменить модель из лабораторной работы 2.1, добавив некоторые условия и параметры. Кассиров в магазине — 4. Ко всем кассирам будет вести одна общая очередь с максимальным количеством человек в этой очереди 20. Время обслуживания имеет треугольное распределение с минимальным значением равным 2.5, средним - 6, и максимальным - 11 минутам.

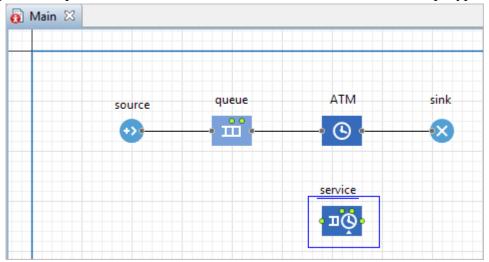
ДОБАВЛЕНИЕ КАССИРОВ

Теперь мы усложним модель, добавив в нее служащих — кассиров. Мы могли бы промоделировать кассиров, как и терминал самообслуживания, с помощью объекта **Delay**. Но куда более удобным представляется моделирование кассиров с помощью ресурсов. Ресурс — это специальный объект **Библиотеки моделирования процессов**, который может потребоваться агенту для выполнения какой-то задачи. В каждый момент времени ресурс может быть занят только одним агентом. В нашем примере покупателям магазина (агентам) необходимо получить обслуживание у кассиров (ресурсов).

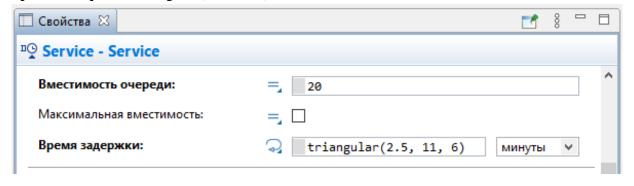
Изменения в диаграмме процесса

Добавьте обслуживание

1. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите на диаграмму Main блок **Service**. Объект **Service** захватывает для агента заданное количество ресурсов, задерживает агента, а затем освобождает захваченные им ресурсы.



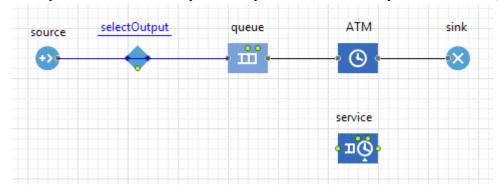
- 2. Перейдите в панель Свойства блока service.
- 3. Измените параметры объекта следующим образом:
- о Ко всем кассирам будет вести одна общая очередь. Задайте максимальное количество человек в этой очереди в поле **Вместимость очереди**: 20.
- о Мы полагаем, что время обслуживания имеет треугольное распределение с минимальным значением равным 2.5, средним 6, и максимальным 11 минутам. Введите в поле **Время задержки**: triangular(2.5, 11, 6)



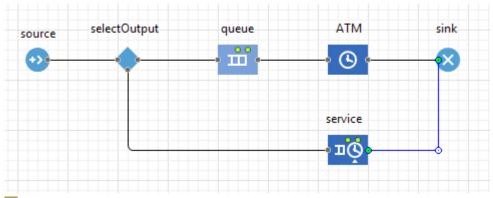
뒬 Смоделируйте выбор клиентов

1. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите на диаграмму процесса Main блок <u>SelectOutput</u> в свободное место между блоками *source* и *queue*. Возможно, вам понадобится переместить несколько блоков, чтобы увеличить длину соединителя между ними. Вы можете выделить несколько блоков диаграммы процесса и переместить их все вместе или перемещать блоки по одному.

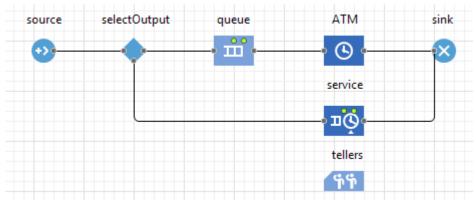
SelectOutput является блоком принятия решения. В зависимости от заданного вами условия, агент, поступивший в объект, будет поступать на один из двух выходных портов объекта.



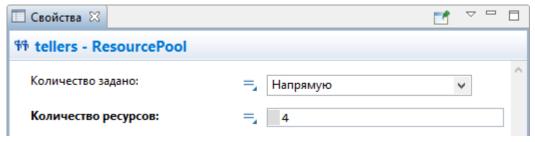
- 2. Выделите блок *selectOutput* в диаграмме процесса. В панели **Свойства** этого блока выберите опцию *При выполнении условия* в параметре **Выход true выбирается**. Убедитесь, что в поле **Условие** стоит выражение randomTrue(0.5). В этом случае к кассирам и банкомату будет приходить примерно равное количество клиентов.
- **randomTrue** возвращает истинное значение (true) с заданной вероятностью р. Эквивалентна функции uniform() < р. Вероятность генерирования значения false, соответственно, равна 1 р.
- 3. <u>Соедините</u> блоки *selectOutput* и *service* с другими блоками так, как показано на рисунке ниже:



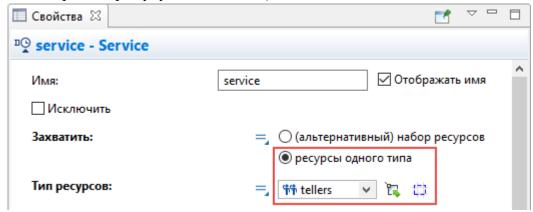
- **Добавьте ресурсы для сервиса**
- 1. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите блок **ResourcePool** на диаграмму агента Main. Объект **ResourcePool** задает ресурсы определенного типа (в нашей модели это будут кассиры магазина).
 - 2. Поместите его, например, под блоком *service* и перейдите в панель **Свойства**.
 - 3. Назовите объект tellers.



4. Задайте число кассиров в поле Кол-во ресурсов: 4.



- 5. Блок **ResourcePool** указывается в объектах, использующих ресурсы, в нашем случае это блок **Service**. Поэтому нам необходимо изменить свойства блока *service* диаграммы процесса.
- 6. Выделите блок service и перейдите в панель Свойства. Выберите опцию Ресурсы одного типа в параметре Захватить ресурсы. Затем укажите блок tellers, который мы добавили на диаграмму, в параметре Блок ResourcePool. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).

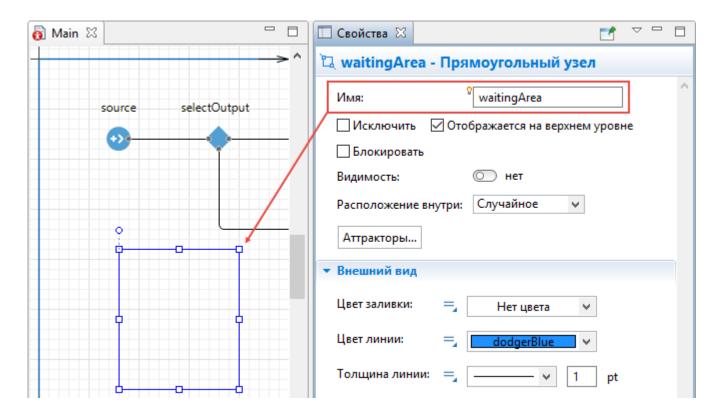


7. Поскольку модель изменилась, мы должны изменить и ее анимацию.

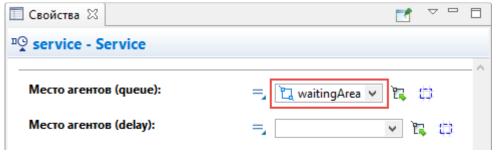
ДОБАВЛЕНИЕ ФИГУР РАЗМЕТКИ ПРОСТРАНСТВА

Теперь давайте нарисуем область для ожидания и место обслуживания клиентов кассирами.

- 뒬 Задайте фигуру разметки для электронной очереди
- 1. В этот раз мы будем рисовать место ожидания покупателями, используя прямоугольный узел. Вначале откройте палитру Разметка пространства панели Палитра.
- 2. Двойным щелчком выделите элемент **Прямоугольный узел** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
- 3. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел достигнет нужной формы. Вы можете редактировать фигуру и после того, как ее рисование завершено
 - 4. Назовите эту область waitingArea.
- 5. Переключите элемент управления **Видимость** в положение **нет**. Таким образом разметка пространства не будет видна на анимации во время выполнения модели.

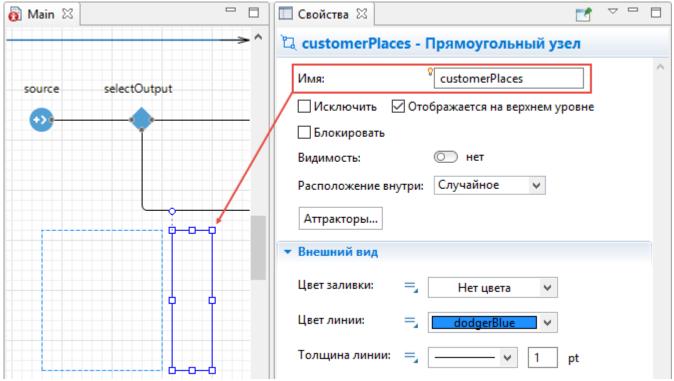


- 5. Выделите щелчком блок *service block* в диаграмме процесса и перейдите в его свойства.
- 6. Выберите только что нарисованный нами узел waitingArea в параметре **Mecto** агентов (queue)

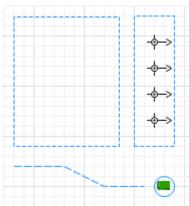


뒬 Задайте фигуру разметки места обслуживания клиентов

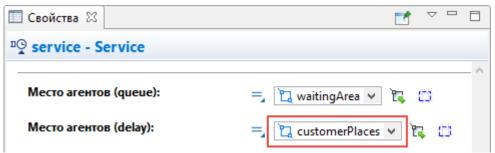
- 1. Покупателям требуется место, на котором они могли бы находиться во время обслуживания у кассиров. Мы нарисуем такую область, используя прямоугольный узел.
 - 2. Вначале откройте палитру Разметка пространства панели Палитра.
- 3. Двойным щелчком выделите элемент **Прямоугольный узел ¬** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
- 4. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел имеет нужную форму. Вы можете редактировать фигуру и после того, как ее рисование завершено.
 - 5. Назовите эту область *customerPlaces*.
 - 6. Переключите элемент управления Видимость в положение нет.



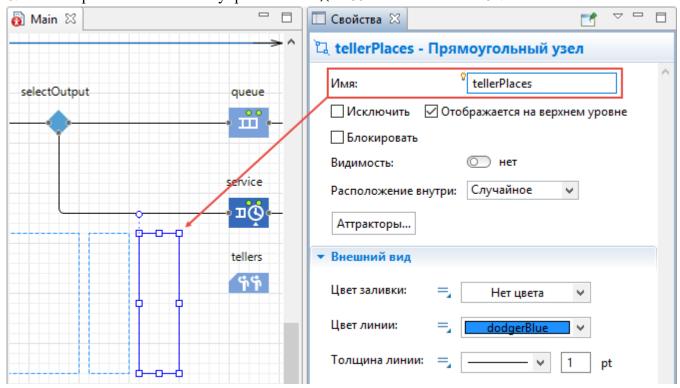
6. Мы будем использовать <u>аттракторы</u>, чтобы задать местоположение тех покупателей, которые будут обслуживаться у кассиров. Выделите узел *customerPlaces* в графическом редакторе и щелкните кнопку **Аттракторы...** в свойствах узла. В открывшемся окне **Аттракторы** укажите число аттракторов 4 в режиме создания **Количество аттракторов**, затем щелкните **ОК**. Вы увидите, что четыре аттрактора появились в узле *customerPlaces* на равном расстоянии друг от друга.



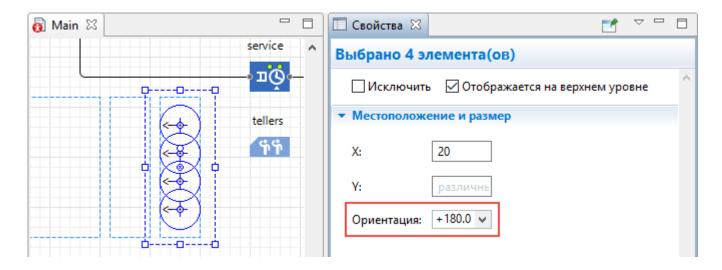
- 7. Теперь нам необходимо сослаться на эту фигуру в диаграмме процесса. Щелкните блок *service* и перейдите в панель **Свойства** этого блока.
- 8. Выберите нарисованный нами узел customerPlaces в параметре **Место агентов** (delay)



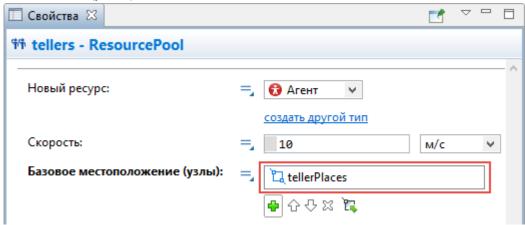
- 1. Кассирам магазина также требуется место, где они могли бы находиться во время обслуживания клиентов. Мы нарисуем такую область, используя прямоугольный узел.
 - 2. Вначале откройте палитру Разметка пространства панели Палитра.
- 3. Двойным щелчком выделите элемент **Прямоугольный узел** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
- 4. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел имеет нужную форму. Вы можете редактировать фигуру и после того, как ее рисование завершено.
 - 5. Назовите эту область tellerPlaces.
 - 6. Переключите элемент управления Видимость в положение нет.



- 6. Мы будем использовать <u>аттракторы</u>, чтобы задать местоположение кассиров. Выделите узел *tellerPlaces* в графическом редакторе и щелкните кнопку **Аттракторы...** в свойствах узла. В открывшемся окне **Аттракторы** укажите число аттракторов 4 в режиме создания **Количество аттракторов**, затем щелкните **ОК**.
- 7. Вы увидите, что четыре аттрактора появились в узле *tellerPlaces* на равном расстоянии друг от друга, но они направлены не в ту сторону. Выделите все аттракторы, зажав клавишу Shift и щелкнув по ним мышью, и потом выберите *180.0* в параметре **Ориентация** секции свойств **Местоположение и размер**.



- 8. Щелкните объект tellers в диаграмме процесса и перейдите в его свойства.
- 9. Выберите нарисованный нами узел *tellerPlaces* в параметре **Базовое** местоположение (узлы)



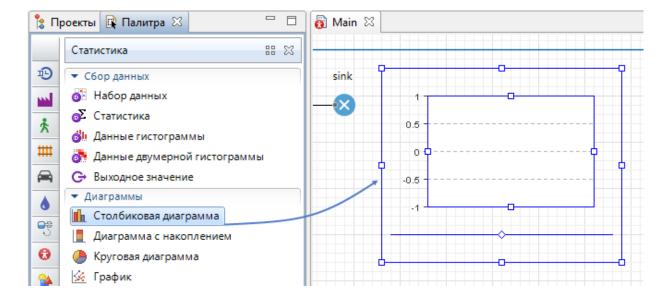
Вы можете запустить модель и наблюдать, как покупатели обслуживаются у терминалов самообслуживания и проходят к кассирам.

СБОР СТАТИСТИКИ

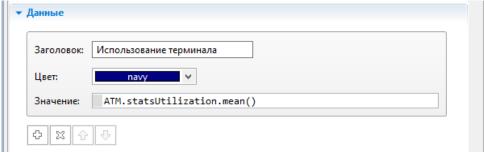
AnyLogic предоставляет пользователю удобные средства для сбора статистики по работе блоков диаграммы процесса. Блоки Библиотеки моделирования процессов самостоятельно производят сбор основной статистики. Мы можем, например, просмотреть интересующую нас статистику (скажем, статистику занятости терминала самообслуживания и длины очереди) с помощью диаграмм.

Сбор статистики использования ресурсов

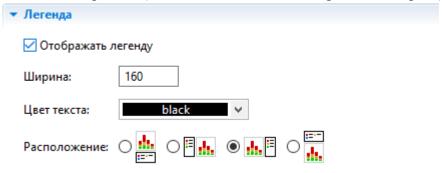
- **Добавьте** диаграмму для отображения средней занятости терминала самообслуживания
- 1. Откройте палитру **Статистика**. Эта палитра содержит элементы сбора данных и статистики, а также диаграммы для визуализации данных и результатов моделирования. Перетащите элемент **Столбиковая диаграмма** из палитры **Статистика** на диаграмму:



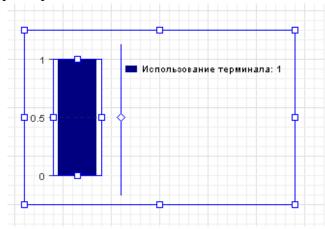
- 2. Перейдите в секцию Данные свойств столбиковой диаграммы.
- 3. Измените Заголовок на Использование терминала.
- 4. Введите ATM.statsUtilization.mean() в поле Значение. Здесь ATM это имя нашего объекта Delay. У каждого объекта Delay есть встроенный набор данных statsUtilization, занимающийся сбором статистики использования этого объекта. Функция mean() возвращает среднее из всех измеренных этим набором данных значений. Вы можете использовать и другие методы сбора статистики, такие, как min() или max().



5. Перейдите в секцию **Легенда** панели **Свойства**. Измените расположение легенды относительно диаграммы (мы хотим, чтобы она отображалась справа).

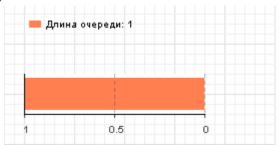


6. И измените ее размер:

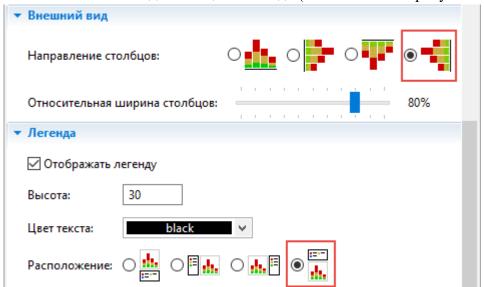


Е Добавьте диаграмму для отображения средней длины очереди

1. Аналогичным образом добавьте еще одну столбиковую диаграмму. Измените ее размер так, как показано на рисунке:



2. Перейдите в секцию **Внешний вид** панели **Свойства** и выберите последнюю опцию параметра **Направление столбцов**, чтобы столбцы столбиковой диаграммы росли влево. Также измените положение легенды в секции **Легенда** (как показано на рисунке ниже).

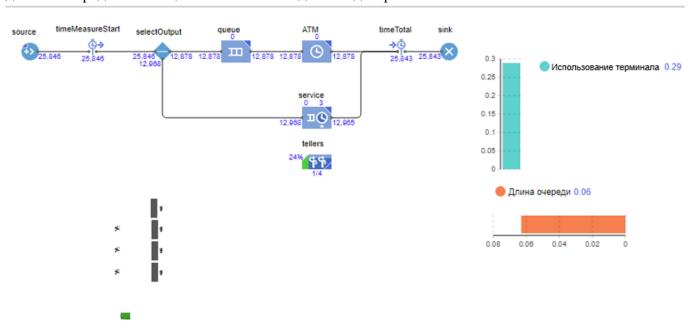


3. В секции **Данные** свойств диаграммы измените **Заголовок** на **Длина** *очереди*. Задайте **Значение**: queue.statsSize.mean()

3десь statsSize - это имя объекта типа "статистика" StatisticsContinuous, производящего сбор статистики размера очереди объекта Queue.

Данные		
Заголовок:	Длина очереди	
Цвет:	coral	
Значение:	queue.statsSize.mean()	

Запустите модель и понаблюдайте за занятостью терминала самообслуживания и средней длиной очереди с помощью только что созданных диаграмм.



© Самостоятельно создайте Столбчатую диаграмму для сравнения количества покупателей, обслуженных терминалами и кассирами за время работы магазина:



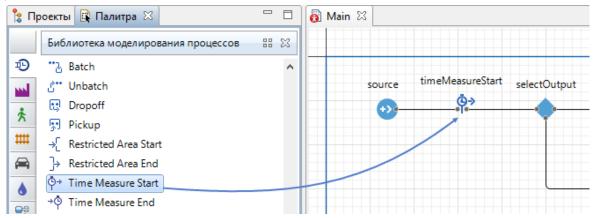
Сбор статистики по времени обслуживания

Мы хотим знать, сколько времени покупатель проводит в магазине. Мы соберем эту статистику с помощью блоков <u>TimeMeasureStart</u> и <u>TimeMeasureEnd</u> из **Библиотеки моделирования процессов** и отобразим собранную статистику распределения времен обслуживания покупателей с помощью гистограммы. Чтобы измерить время, проведенное агентами на определенном отрезке диаграммы процесса, мы должны разместить эти блоки соответственно на входе в интересующий нас отрезок и на выходе из него. Первый блок хранит

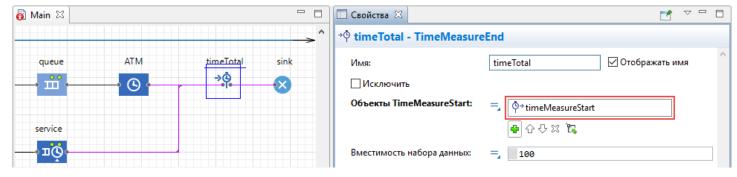
время прохождения агента через блок, а второй измеряет время, которое агент провел на отрезке диаграммы процесса после того, как покинул первый блок.

🧏 Добавьте блоки измерения времени в диаграмму процесса

- 1. Измените диаграмму процесса так, чтобы между блоком *source* и блоком *selectOutput* появилось место для нового блока.
- 2. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и добавьте блок **TimeMeasureStart** в освободившееся место. Убедитесь, что порты блока соединены к соседним блокам.

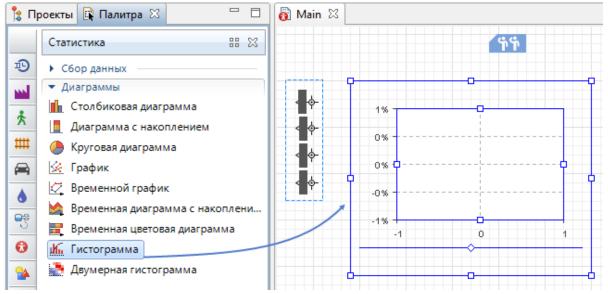


- 3. Перетащите блок *sink* вправо.
- 4. Перетащите блок **TimeMeasureEnd** из **Библиотеки моделирования процессов** на графический редактор и разместите его перед блоком *sink*.
- 5. Убедитесь, что входной порт блока соединен с блоками *ATM* и *service*, а выходной с блоком *sink*.
- 6. Чтобы рассчитать распределения времен для агентов, в свойствах каждого блока **TimeMeasureEnd** должен быть указан как минимум один блок **TimeMeasureStart**. Откройте свойства вашего блока и задайте блок **TimeMeasureStart** в параметре **Объекты TimeMeasureStart**.

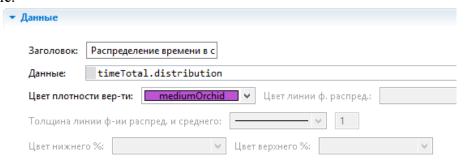


🛂 Добавьте гистограмму для отображения собранной статистики

1. Чтобы добавить гистограмму на диаграмму агента, перетащите элемент Гистограмма из палитры Статистика в графический редактор. Измените ее размер.

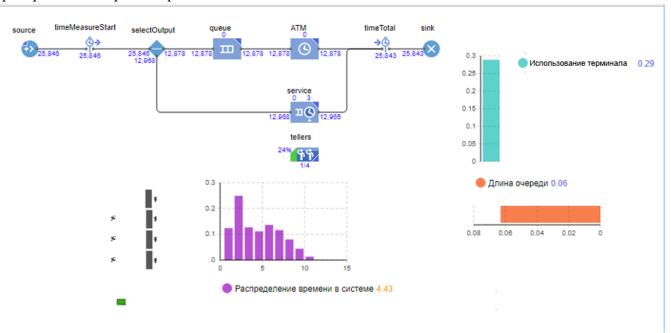


- 2. Откройте секцию Данные свойств гистограммы
- 3. Измените Заголовок отображаемых данных на Распределение времени в системе.
- 4. Введите timeTotal.distribution в поле Данные. timeTotal имя блока TimeMeasureEnd, который собирает распределение времен пребывания агентов в системе.



ЗАПУСК МОДЕЛИ.

Запустите модель. Включите режим виртуального времени и понаблюдайте за тем, какой вид примет распределение времен пребывания клиентов в системе.



ИНДИВИДУАЛЬНЫЕ ИСХОДНЫЕ ДАННЫЕ К ЛАБОРАТОРНОЙ РАБОТЕ 2.2

ТАБЛИЦА 2.2 - Индивидуальные исходные данные для модели магазина с терминалом самообслуживания и кассирами.

Вари-	Распределение	Робр	Время	M_o	Kas
ант	вероятности прихода	(вероятность	обслуживания	(максимальная	(количе-
	покупателей в магазин	обращения к	покупателя	очередь к	ство кас-
	(интенсивность	кассиру)	кассиром	кассирам), чел	сиров),
	прибытия в минуту)		(треугольное		чел.
			распределение), в		
			минутах		
1.	Экспоненциальное (1, 3)	0,55	3 ± 1	10	1
2.	Экспоненциальное (2, 4)	0,33	3 ± 2	15	2
3.	Экспоненциальное (4, 6)	0,67	3.5 ± 1	20	3
4.	Экспоненциальное (1, 4)	0,25	3.5 ± 2	25	4
5.	Экспоненциальное (1, 2)	0,75	4 ± 2	30	5
6.	Экспоненциальное (2, 5)	0,80	4 ± 1	20	3
7.	Экспоненциальное (1, 3)	0,20	4 ± 2.5	15	2
8.	Нормальное (1, 2)	0,55	3 ± 1	10	1
9.	Нормальное (0, 2)	0,33	3 ± 2	15	2
10.	Нормальное (1, 3)	0,67	3.5 ± 1	20	3
11.	Нормальное (2, 3)	0,25	3.5 ± 2	25	4
12.	Нормальное (2, 4)	0,75	4 ± 2	30	5
13.	Нормальное (1.5, 3)	0,80	4 ± 1	20	3
14.	Нормальное (1.2, 2)	0,17	4 ± 2.5	15	2
15.	Нормальное (1.4, 3)	0,20	3.5 ± 1.5	20	3
16.	Треугольное (1.5, 3, 4)	0,55	3 ± 1	10	1
17.	Треугольное (1.3, 4, 2)	0,33	3 ± 2	15	2
18.	Треугольное (1, 4, 3)	0,67	3.5 ± 1	20	3
19.	Треугольное (1.3, 3, 2)	0,25	3.5 ± 2	25	4
20.	Треугольное (1.2, 4, 3)	0,75	4 ± 2	30	5
21.	Треугольное (1.4, 5, 3)	0,80	4 ± 1	20	3
22.	Треугольное (1.5, 5, 2.5)	0,20	4 ± 2.5	15	2
23.	Равномерное (1, 3)	0,55	3.5 ± 1.5	10	1
24.	Равномерное (2, 3)	0,33	3 ± 1	15	2
25.	Равномерное (2, 4)	0,67	3 ± 2	20	3
26.	Равномерное (1, 3)	0,25	3.5 ± 1	25	4
27.	Равномерное (2, 3.5)	0,75	3.5 ± 2	30	5
28.	Равномерное (1, 2.5)	0,80	4 ± 2	20	3
29.	Равномерное (2.3, 4)	0,20	3.5 ± 1.5	15	2
30.	Равномерное (2.2, 5)	0,17	4 ± 2.5	20	3

КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1. Случайные величины и законы их распределений.
- 2. Понятие имитационного моделирования
- 3. Системы массового обслуживания (СМО). Основные понятия и характеристики.
- 4. Простейший поток событий и его свойства.
- 5. Показатели эффективности одноканальной СМО с очередью.
- 6. Показатели эффективности многоканальной СМО с очередью.
- 7. Влияние применяемого закона распределения случайных чисел на потоки событий и на работу СМО в целом.