

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ БЕЛГОРОДСКОЙ ОБЛАСТИ  
ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«АЛЕКСЕЕВСКИЙ КОЛЛЕДЖ»

УТВЕРЖДАЮ:

Заместитель директора

 И.А. Злобина

31 августа 2021 г.

**Комплект  
контрольно-оценочных средств**

по МДК

**МДК.06.04 Интеллектуальные системы и технологии**

для специальности

**09.02.07 Информационные системы и программирование**

## РАССМОТРЕНО

на заседании предметно-цикловой комиссии  
общепрофессиональных дисциплин и профессиональных модулей  
специальностей 09.02.04 Информационные системы (по отраслям) и 09.02.07  
Информационные системы и программирование  
Протокол № 1 от 31 августа 2021 г.

Председатель  И.В. Косинова

Комплект контрольно-оценочных средств разработан на основе  
Федерального государственного образовательного стандарта среднего  
профессионального образования по специальности 09.02.07 Информационные  
системы и программирование

Составитель: Рогачева Олеся Николаевна, преподаватель

# 1. Паспорт комплекта оценочных средств

## 1.1 Область применения комплекта оценочных средств

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу МДК.06.04 Интеллектуальные системы и технологии

КОС включают контрольные материалы для проведения промежуточной аттестации в форме дифференцированного зачета.

КОС разработан на основании рабочей программы МДК.06.04 Интеллектуальные системы и технологии

## 1.2 Система контроля и оценки освоения программы МДК

Контроль и оценка результатов освоения МДК.06.04 Интеллектуальные системы и технологии осуществляется преподавателем в процессе проведения теоретических и практических занятий, дифференцированного зачета.

<b>Результаты обучения (освоенные умения, усвоенные знания), с учетом личностных результатов, профессионального стандарта и стандарта компетенции Ворлдскиллс</b>	<b>Формы и методы контроля и оценки результатов обучения</b>
<p><b>умения:</b></p> <ol style="list-style-type: none"><li>1) осуществлять настройку информационной системы для пользователя согласно технической документации;</li><li>2) применять основные правила и документы системы сертификации Российской Федерации;</li><li>3) применять основные технологии экспертных систем;</li><li>4) разрабатывать обучающие материалы для пользователей по эксплуатации информационных систем;</li></ol> <p><b>знания:</b></p> <ol style="list-style-type: none"><li>1) регламенты и нормы по обновлению и техническому сопровождению обслуживаемой информационной системы;</li><li>2) политику безопасности в современных информационных системах;</li><li>3) достижения мировой и отечественной информатики в области интеллектуализации информационных систем;</li><li>4) принципы работы экспертных систем.</li></ol>	<p>Экспертное наблюдение и оценка при выполнении практической работы, проверка домашнего задания.</p> <p>Тестирование, защита практической работы, устный и письменный опрос, дифференцированный зачет</p> <p>Экспертное наблюдение и оценка при выполнении практической работы, проверка домашнего задания.</p> <p>Тестирование, защита практической работы, устный и письменный опрос, дифференцированный зачет</p>

## 2. Комплект оценочных средств

### 2.1. Контрольные вопросы к дифференцированному зачету

1. Виды интеллектуальных систем.
2. Области применения интеллектуальных систем.
3. Основные модели интеллектуальных систем.
4. Архитектура интеллектуальных информационных систем.
5. Типовая схема функционирования интеллектуальной системы.
6. Пример интеллектуальных систем с коммутативными способностями.
7. Пример интеллектуальных экспертных систем.
8. Примеры самообучающихся интеллектуальных систем.
9. Примеры адаптивных интеллектуальных систем.

### 2.2. Типовые задания для оценки освоения МДК.06.04

#### Интеллектуальные системы и технологии

##### Задание 1. Моделирование интеллектуальных систем

###### Моделирование одноканальной смо с очередью.

**Постановка задачи.** В банковский офис обращаются клиенты. Офис представляет собой автоматизированный пункт обслуживания, в котором установлен банкомат. Банкомат обслуживает одновременно одного клиента. Клиенты прибывают по экспоненциальному закону с интенсивностью  $\lambda=0,67$ . Одновременно в офисе может находиться не более 15 клиентов. Интервал времени работы банкомата подчиняется треугольному закону распределения с параметрами  $x_{\min}=0,8$ ,  $x_{\max}=1,3$  предпочтительное значение 1.

**Построение модели.** Модель строится с «нуля». Банковский офис представляет собой систему массового обслуживания (СМО). Построение модели такой системы выполняется с помощью элементов библиотеки Enterprise Library Для построения СМО используются элементы:

- Source – источник заявок.
- Queue – очередь ожидающих обслуживания заявок.
- Delay – Элемент моделирующий узел обслуживания.
- Sink – Элемент принимающий отработанные заявки.

Общий вид модели СМО банковского офиса показан на рисунке 3.6.

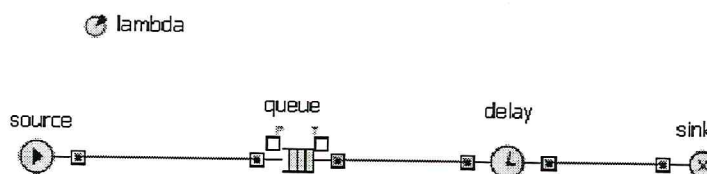


Рис. 3.6. Модель офиса

##### Источник заявок

Заявки – клиенты офиса пребывают с интенсивностью  $\lambda=0,67$ .

Источник заявок обладает следующими настройками:

- Заявки пребывают согласно интенсивности.
- Интенсивность прибытия равна  $\lambda$ . **Lambda** – параметр (панель «Основная» пункт **параметр**). Значение соответствует интенсивности потока клиентов, и равно 0,67.

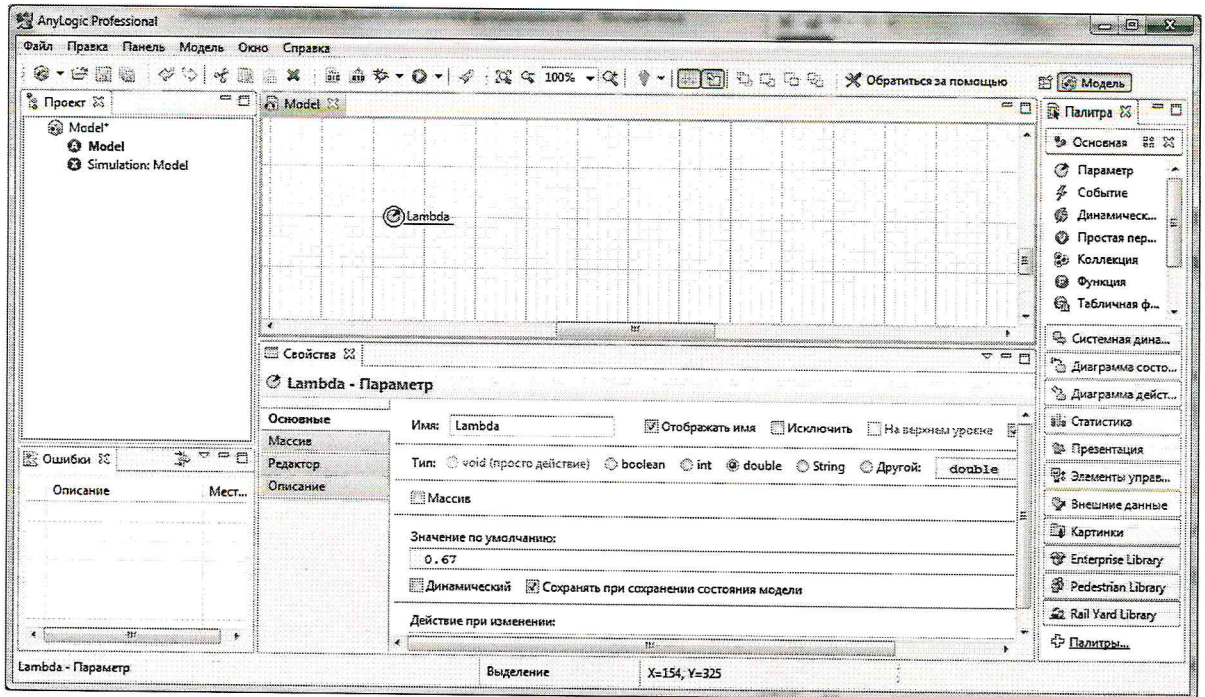


Рис. 3.7. Задание параметра интенсивности заявок.

Закон распределения потока заявок можно задать в свойстве *interarrivalTime* на вкладке *Параметры* для объекта *source*. По умолчанию распределение случайного потока заявок подчиняется экспоненциальному закону. –*exponential()*.

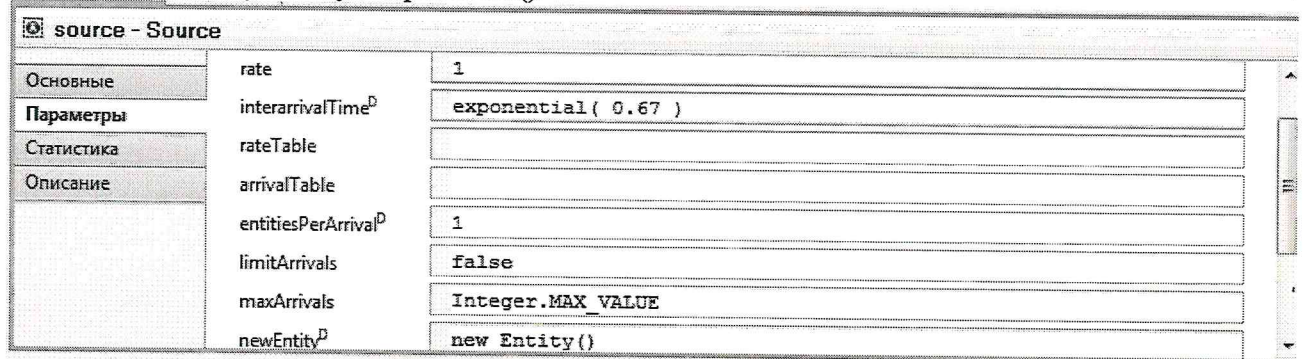


Рис. 3.8. Задание закона распределения потока заявок.

AnyLogic предоставляет функции и других случайных распределений, таких как:

- нормальное с дисперсией  $\sigma$  и мат. ожиданием  $m$  –  $\text{normal}(\sigma, m)$ ;
- равномерное на отрезке  $[a, b]$  –  $\text{uniform}(a, b)$ ;
- треугольное с минимальным значением  $a$ , средним значением  $b$  и максимальным  $c$  –  $\text{triangular}(a, b, c)$ ;

и т.д.

- Количество заявок пребывающих за один раз равно единице.

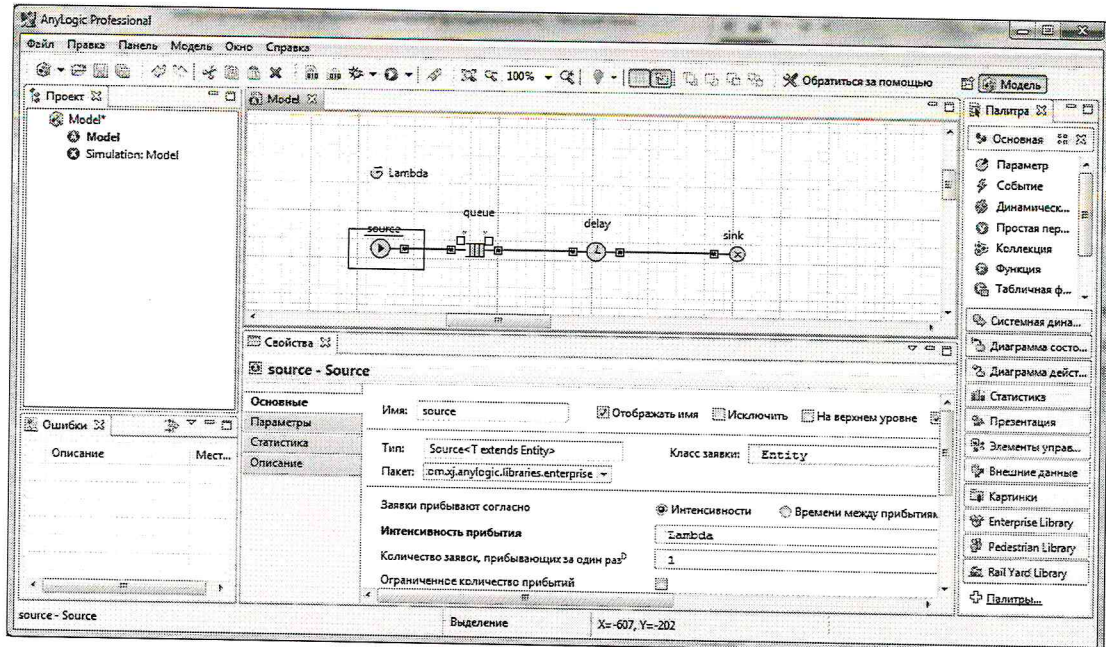


Рис. 3.9. Параметры источника заявок.

### Очередь

Этот элемент характеризуется параметрами:

- Вместимость очереди равна 15.
- Включить сбор статистики – да.

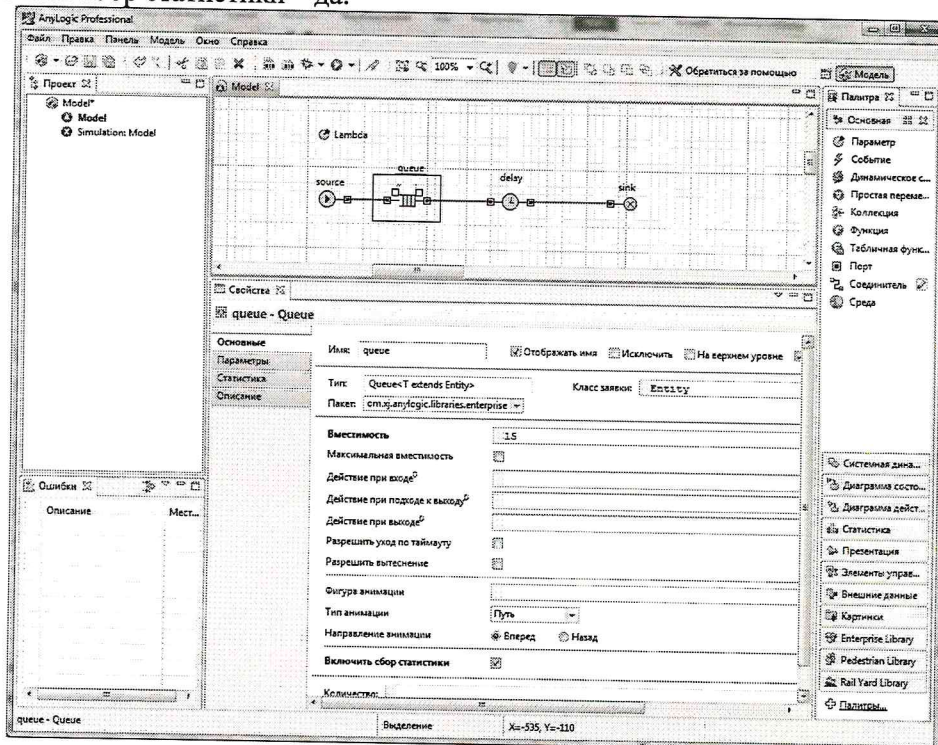


Рис. 3.10. Параметры очереди.

### Узел обслуживания

Параметры элемента:

- Задержка задается явно.
- Время задержки равно:  $\text{triangular}(0.8, 1.3, 1)$ .
- Вместимость узла – один клиент.
- Включить сбор статистики- да.

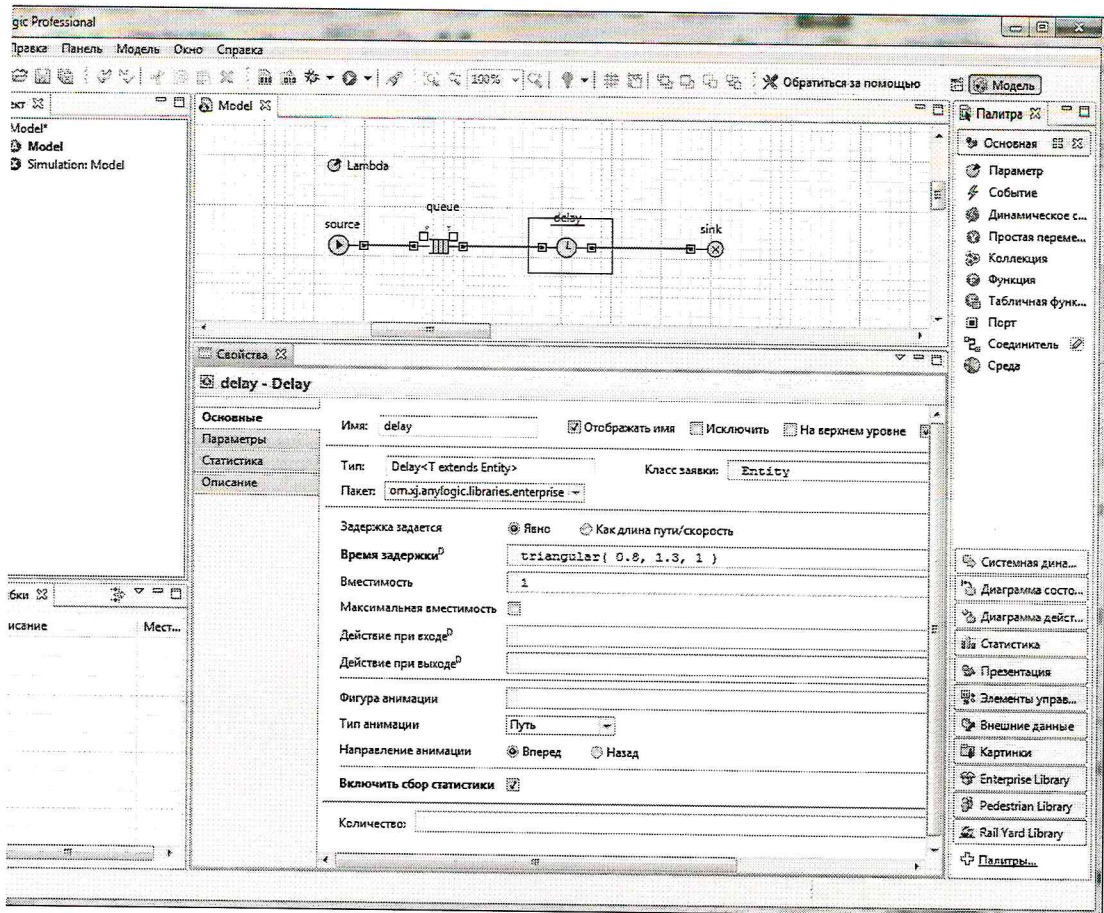


Рис. 3.11. Параметры узла обслуживания.

Элемент, принимающий заявки обладает параметрами настройки по умолчанию.

Настройте эксперимент модели:

- Модельное время – минуты.
- Время остановки модели не задано.
- Задайте Режим выполнения со скоростью 8.

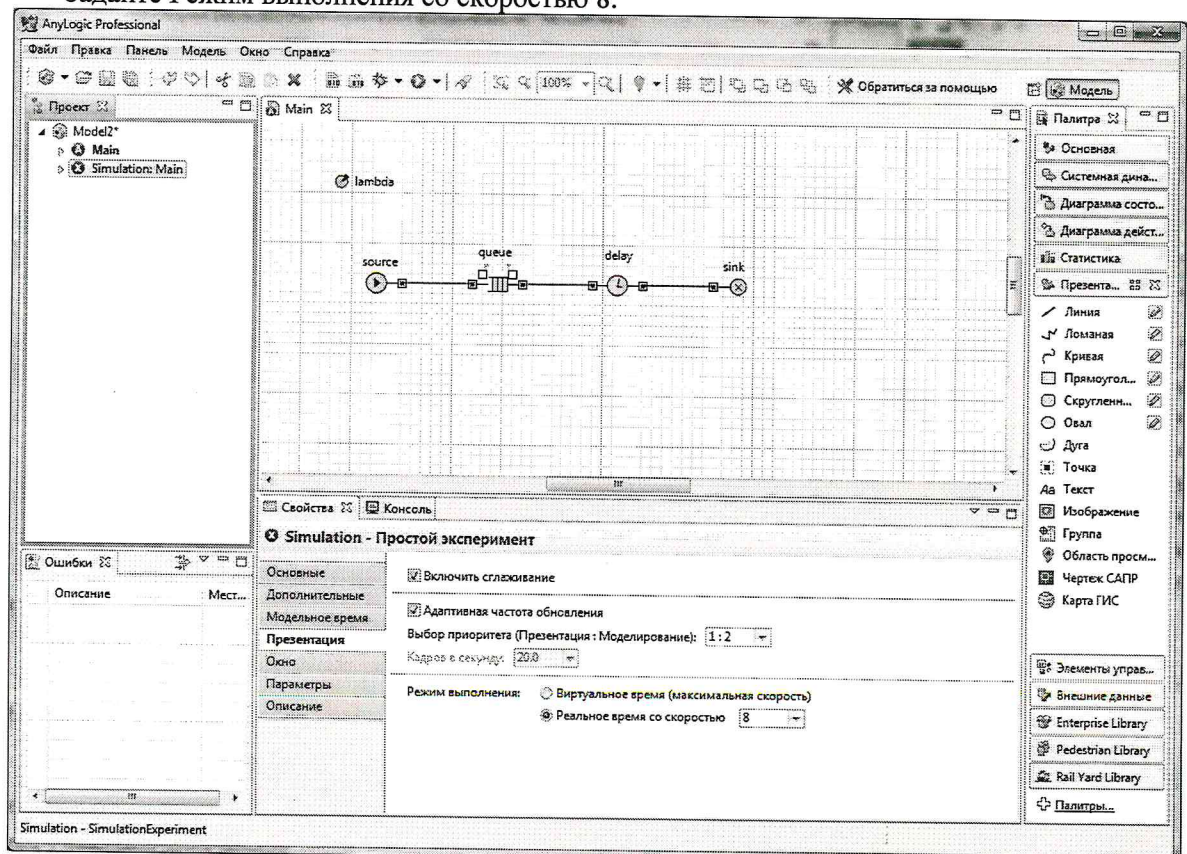


Рис. 3.12. Настройка параметров эксперимента.

Для запуска модели щелкните мышью по кнопке **Запустить**. Откроется окно с презентацией запущенного эксперимента. AnyLogic автоматически помещает на презентацию каждого простого эксперимента заголовок и кнопку, позволяющую запустить модель и перейти на презентацию.

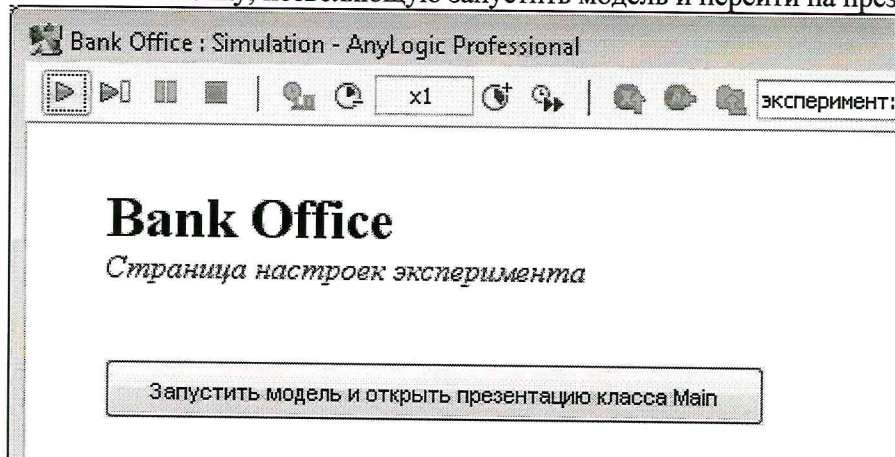


Рис. 3.13. Запуск эксперимента.

Щелкните по этой кнопке. AnyLogic переключится в режим работы модели. С помощью визуализированной блок-схемы вы можете проследить, сколько человек находится в очереди, сколько человек в данный момент обслуживается и т.д.

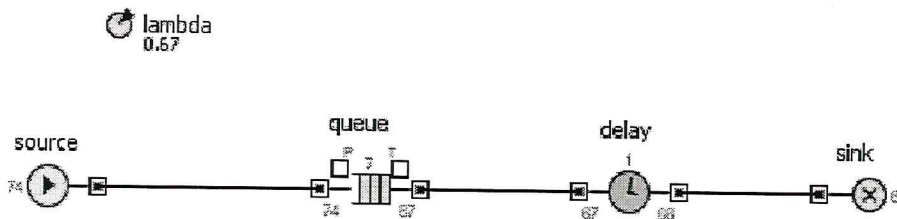


Рис.3.14. Вид работающей модели

### Анимация модели

Покажем процесс обслуживания клиентов в виде анимации очереди, ведущей к банкомату, так как это показано на рисунке 3.15.

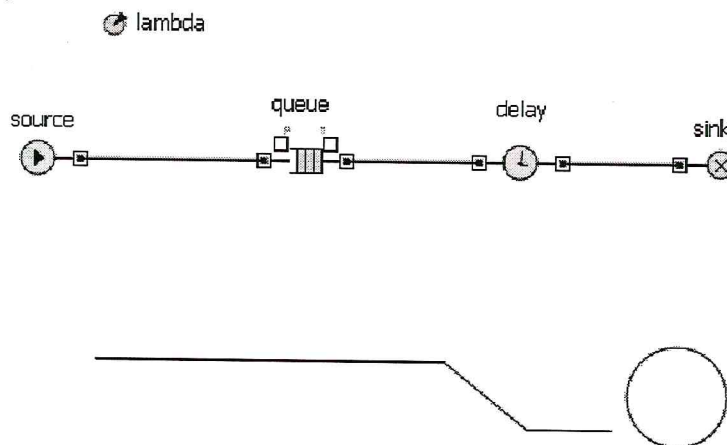


Рис.3.15. Анимация очереди

Банкомат представим в виде окружности. Когда клиент находится в банкомате, окружность будет окрашена в красный цвет, при свободном банкомате окружность закрашивается в зеленый цвет. С помощью элемента «Овал» палитры «Презентация» разместите окружность и присвойте ей имя ServicePoint. Цвет заливки должность изменяться динамически:

`delay.size()>0 ? Color.red: Color.green`

Здесь `size()` – метод объекта `delay`, который возвращает количество заявок-клиентов в приборе обслуживания.



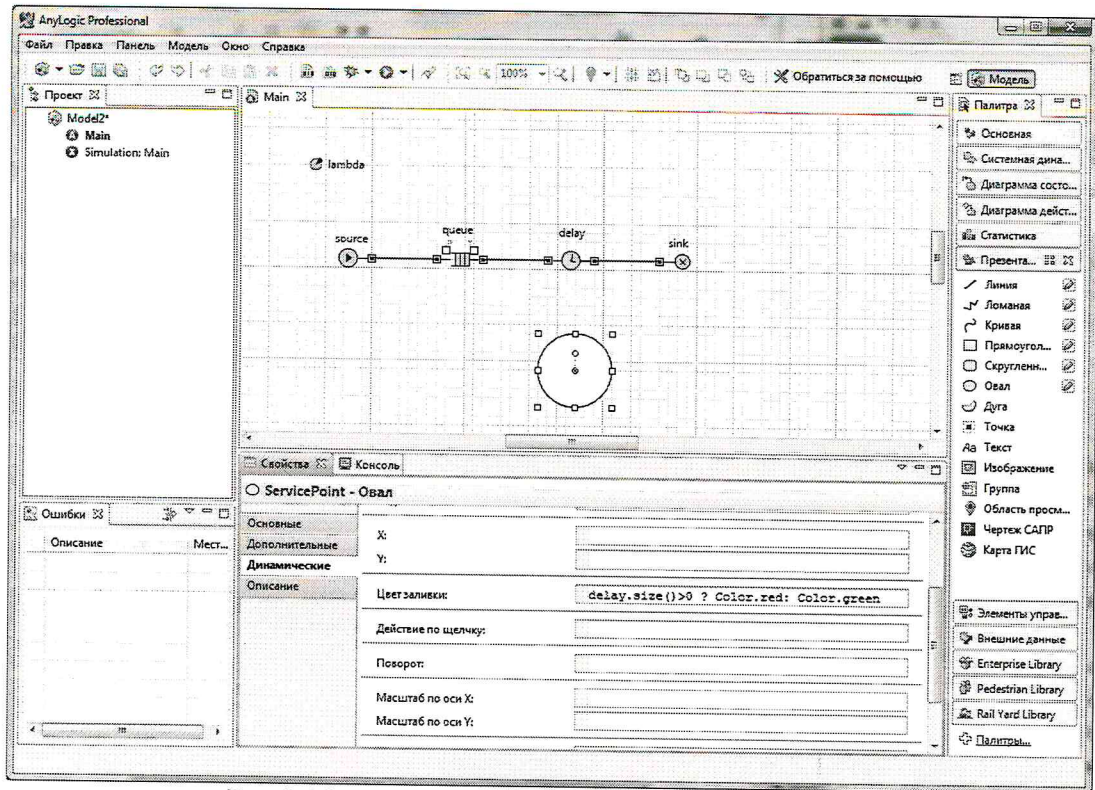


Рис.3.16. Задание свойств окружности – банкомата.

Для отображения очереди следует нарисовать ломаную линию (см. рисунок 3.15), используя элемент «Ломаная» из палитры «Презентация». Режим рисования включается после выполнения двойного щелчка по пиктограмме

Рисование ломаной нужно выполнять по направлению движения клиентов к банкомату: слева на право. Ломаной присвойте имя GoToService.

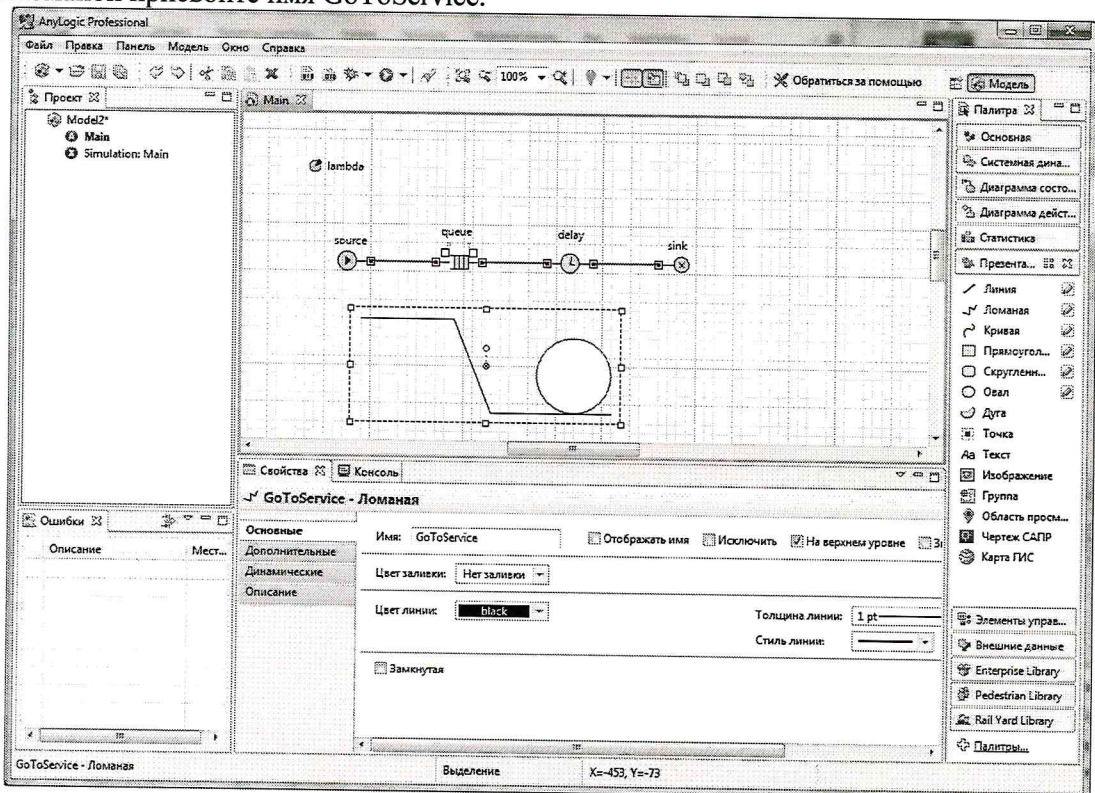


Рис.3.17. Задание свойств ломаной – очереди.

После создания элементов презентации нужно выполнить ряд настроек модели для связи графических элементов с объектами схемы.

Откройте окно свойств элемента очередь (queue) и на вкладке «Основные» задайте настройки так, как это показано на рисунке 3.18.

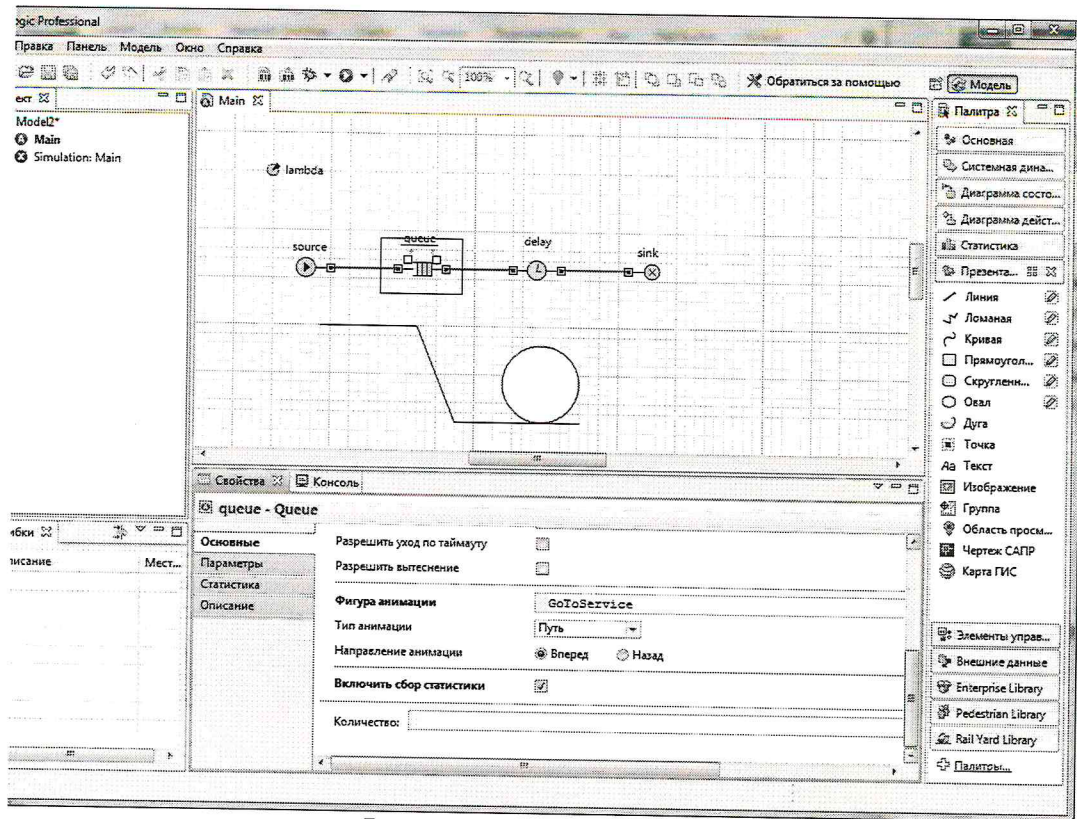


Рис.3.18. Настройка очереди

Откройте прибор обслуживания – элемент delay, и настройте на вкладке «Основные», свойства анимации:

- Фигура анимации: ServicePoint
- Тип анимации: Одиночная

Установите режим скорости исполнения равным 4 и протестируйте модель. На рисунке 3.19. Показан вид работающей модели.

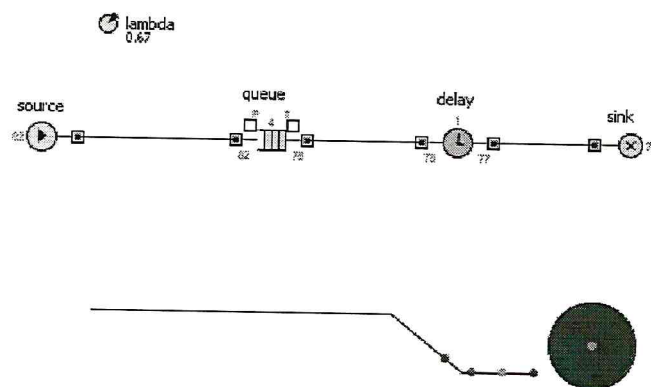


Рис.3.19. Модель с анимацией

### Размещение датчиков.

Чтобы представить процесс загрузки прибора обслуживания и очереди разместим два датчика – столбчатые диаграммы. Первая диаграмма отображает среднее значение клиентов в очереди, а вторая - среднее значение числа обслуженных клиентов в банкомате (приборе обслуживания). Для размещения диаграмм нужно использовать палитру «Статистика» и элемент «Столбиковая диаграмма». Разместите две диаграммы (см. рисунок 3.20).

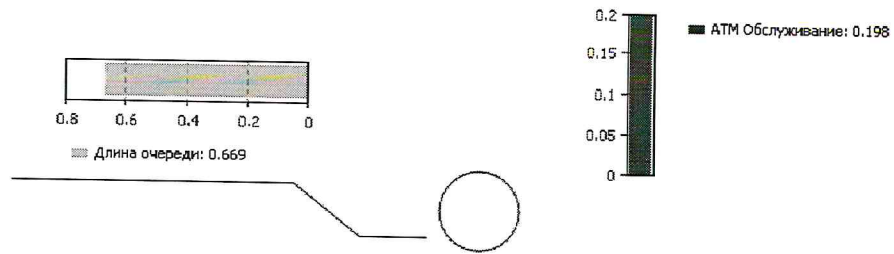


Рис.3.20. Диаграммы загрузки

Первую разместите над очередью. Добавьте элемент данных. Задайте подпись «Длина очереди». Выберите цвет, а затем в качестве значения задайте выражение:

```
queue.statsSize.mean()
```

Здесь метод `mean()` – возвращает среднюю длину очереди.

При вводе выражения можно использовать помощник AnyLogic. Для этого следует нажать комбинацию клавиш CTRL + SPACE.

После ввода выражения нужно сменить ориентацию диаграммы на горизонтальное. Для этого необходимо открыть вкладку «Внешний вид» и изменить направление столбцов (см. рисунок 3.21).

Столбцы

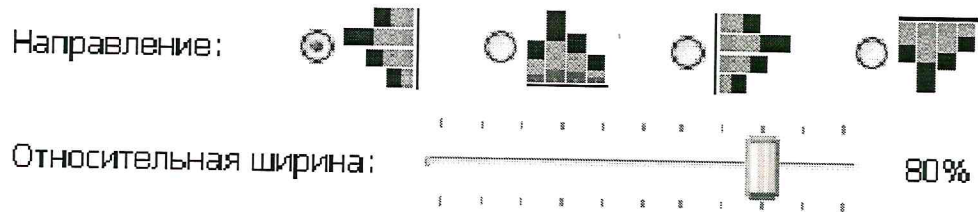


Рис.3.21. Направление столбцов диаграммы

Вторую диаграмму расположите рядом с изображением банкомата. Назовите диаграмму «АТМ Обслуживание», а в качестве значения задайте выражение:

```
delay.statsUtilization.mean()
```

задающее среднее время обслуживания заявки в процессоре. Направление столбцов вертикальное. Вид работающей модели показан на рисунке 3.22.

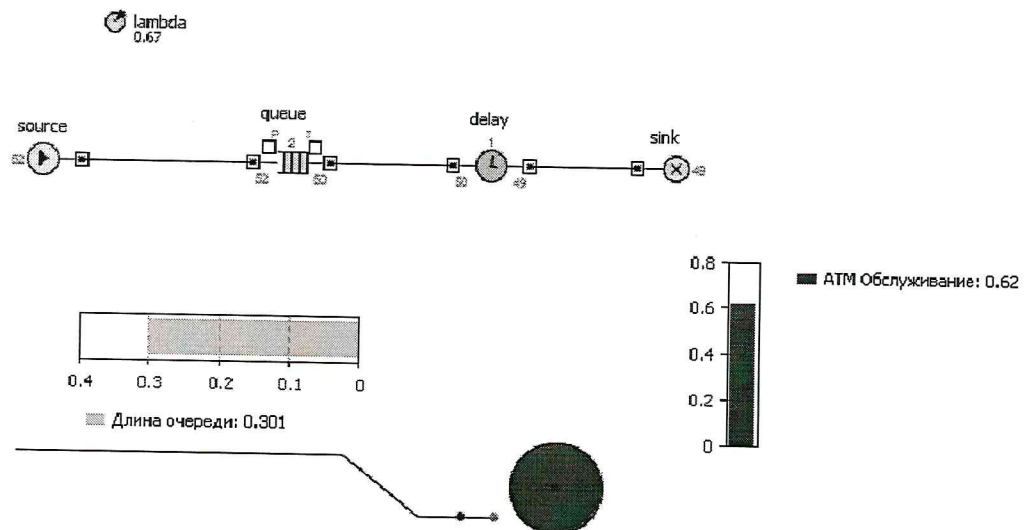


Рис.3.22. Модель с диаграммами

### Моделирование многоканальной смо с очередью.

Усложним модель, добавив в нее банковских кассиров. Можно моделировать число кассиров, как и банкомат, с помощью объектов `delay`. Но куда более удобным представляется моделирование числа кассиров с помощью ресурсов. **Ресурс** – это специальный объект Enterprise Library, который может потребоваться заявке для выполнения какой-то задачи. В нашем примере посетителям банковского отделения (заявкам) необходимо получить помощь у банковских служащих (ресурсов).

Добавьте на диаграмму следующие объекты:

1. *selectOutput* – является блоком принятия решения. В зависимости от заданного вами условия, заявка, поступившая в этот объект, будет поступать на один из двух выходов объекта. Оставьте свойство *selectCondition* –  $uniform() < 0.5$ , тогда к кассирам и банкомату будет приходить примерно равное количество клиентов;
2. *Service* – моделирует занятие заявкой ресурса на определенное время. С помощью этого объекта мы промоделируем обслуживание клиента кассиром. Задайте следующие свойства объекта: назовите объект *tellerLines* (свойство **Имя**); укажите, что в очереди к кассирам может находиться до 20 человек (свойство *queueCapacity*); задайте время обслуживания (свойство *delayTime*). Будем полагать, что время обслуживания имеет треугольное распределение с минимальным средним значением 2.5, средним – 6 и максимальным – 11 минут;

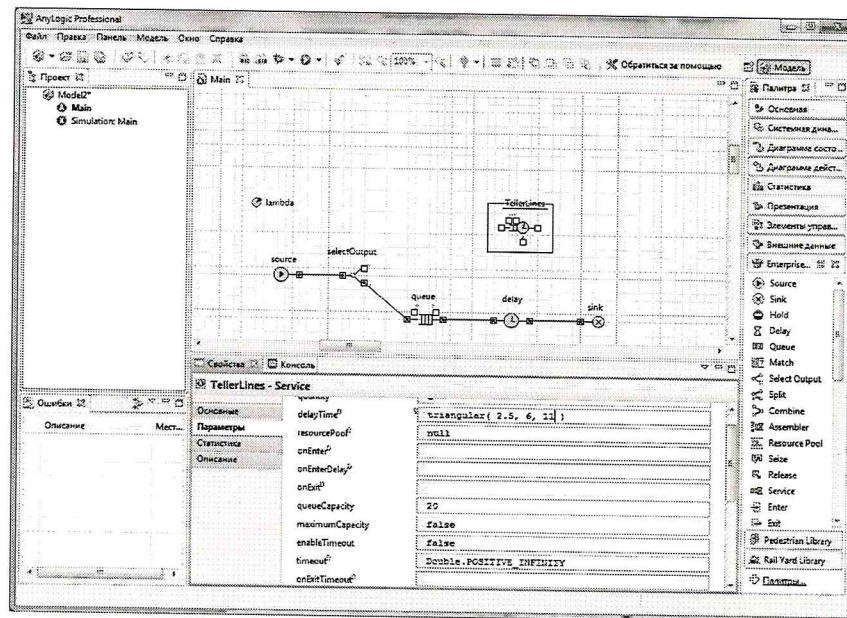


Рис. 3.23. Задание свойств линии касс.

3. *ResourcePool* – задает ресурсы определенного типа. Он должен быть подсоединен к объектам, моделирующим занятие и освобождение ресурсов (в нашем случае это объект *Service*). Задайте следующие свойства объекта: назовите объект *tellers*; задайте число кассиров (свойство *capacity*) – 4.

Измените имя объекта *delay* на *ATM* (банкомат). Соедините объекты соответствующим образом (рис. 3.24).

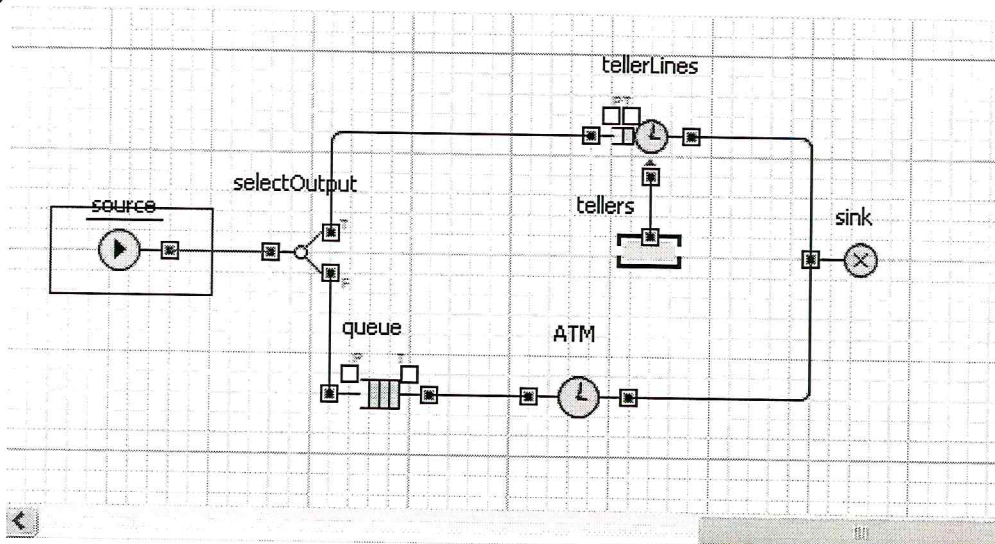


Рис. 3.24. Вид двухканальной СМО.

Запустите модель и изучите ее поведение.

Сбор статистики о времени обслуживания клиента.

Необходимо определить, сколько времени клиент проводит в банковском отделении и сколько времени он теряет, ожидая своей очереди. Соберем эту статистику с помощью специальных

объектов сбора данных и отобразим собранную статистику распределения времени обслуживания клиентов с помощью гистограмм.

Создадим класс сообщения *Customer*. Сообщения этого класса будут представлять клиентов банковского отделения. Выберите базовый класс *Entity*(сообщения), добавьте параметры для хранения информации о проведенном времени:

1. в панели **Проект**, щелкните правой кнопкой мыши по элементу модели и выберите **Создать | Java класс** из контекстного меню (рис. 3.25);

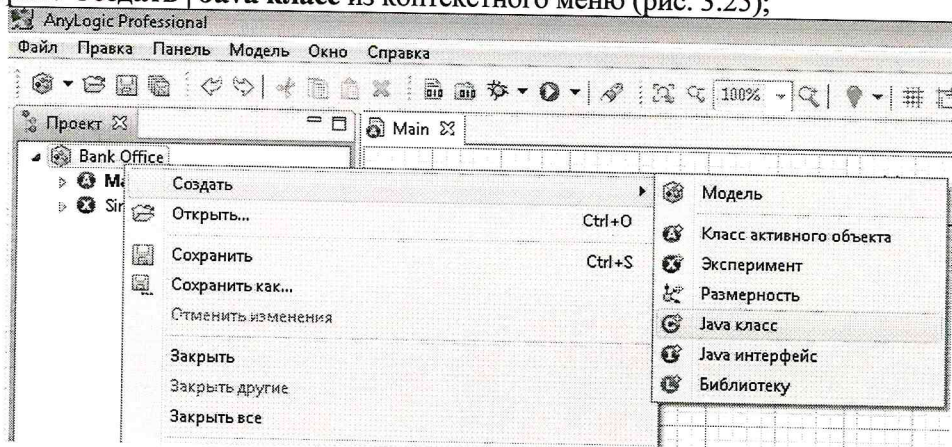


Рис. 3.25 Добавление Java- класса.

2. появится диалоговое окно **Новый Java класс**. В поле **Имя** введите имя нового класса *Customer*;
3. сделайте так, чтобы этот класс наследовался от базового класса заявки *Entity* (рис. 3.26): выберите из выпадающего списка **Базовый класс** полное имя данного класса: *com.xj.anylogic.libraries.enterprise.Entity*;

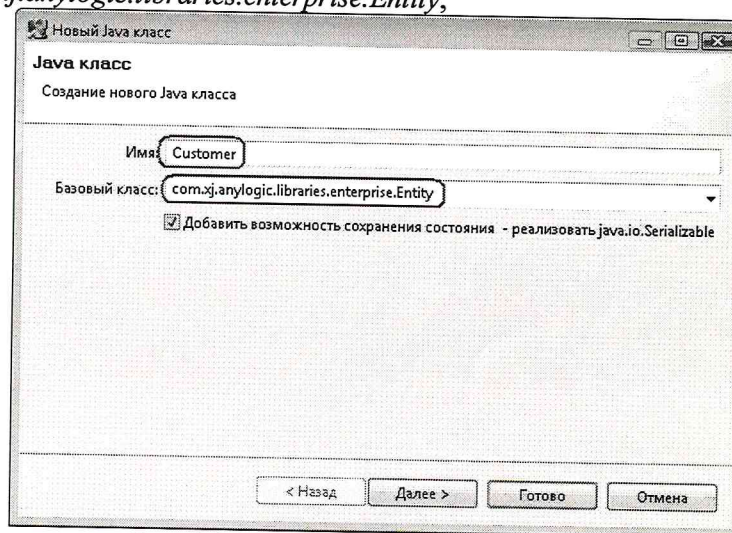


Рис. 3.26 Задание свойств Java- класса.

4. щелкните мышью по кнопке **Далее**. На второй странице Мастера вы можете задать параметры создаваемого Java-класса. Создайте параметры:
    - *enteredSystem* типа *double* для сохранения момента времени, когда клиент пришел в банковское отделение;
    - *startWaiting* типа *double* для сохранения момента времени, когда клиент встал в очередь к банкомату;
  - щелкните мышью по кнопке **Готово**. Вы увидите редактор кода созданного класса. Можете закрыть его, щелкнув мышью по крестику в закладке с его названием.
- Теперь вычислим время, которое тратится персоналом банка на обслуживание клиентов, и время, которое клиенты тратят на ожидание своей очереди.**
- Для этого добавьте элементы сбора статистики по времени ожидания клиентов и времени пребывания клиентов в системе. Эти элементы будут запоминать соответствующие значения времени для каждого клиента и предоставят пользователю стандартную статистическую информацию: среднее, минимальное, максимальное из измеренных значений, среднее квадратичное отклонение, доверительный интервал для среднего и т.п.:

1. чтобы добавить объект сбора данных гистограммы на диаграмму, перетащите элемент **Данные гистограммы** с палитры **Статистика** на диаграмму активного класса;
2. задайте свойства элемента (рис. 3.27).
  - Измените **Имя** на *waitTimeDistr*.
  - Измените **Заголовок** на *Waiting time distribution*.
  - Сделайте **Кол-во интервалов** равным 50.
  - Задайте **Начальный размер интервала**: 0.01;

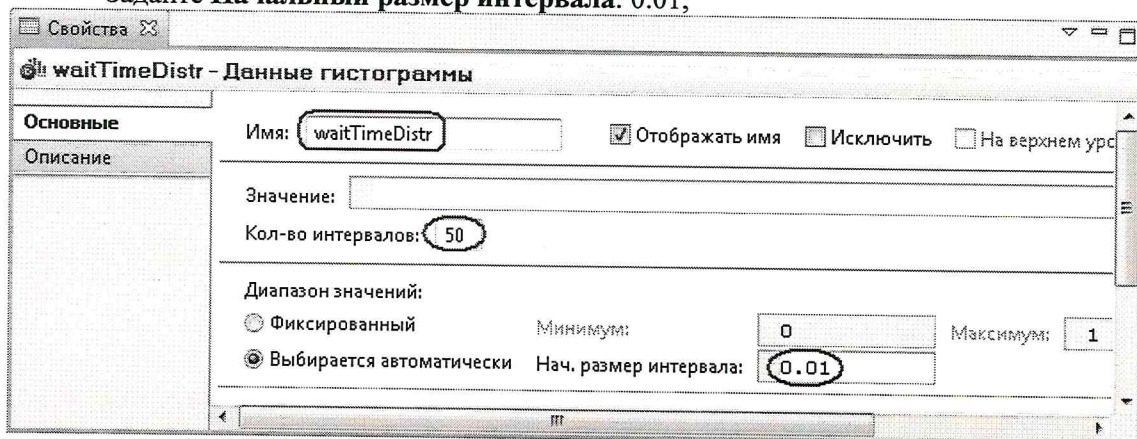


Рис. 3.27 Задание свойств гистограммы.

3. создайте еще один элемент сбора данных гистограммы (**Ctrl** + перетащите только что созданный объект данных гистограммы, чтобы создать его копию). Измените **Имя** этого элемента на *timeInSystemDistr*, а **Заголовок** на *Time in system distribution*.  
Измените свойства блоков вашей диаграммы процесса. Задайте следующие свойства объектов диаграммы:
  1. блок *source*, свойство **Новая заявка** – введите `new Customer()`. Введите `Customer` в поле **Класс заявки**. Это позволит напрямую обращаться к полям класса заявки *Customer* в коде динамических параметров этого объекта. Введите `entity.enteredSystem = time()`; в поле **Действие привходе**. Этот код будет сохранять время создания заявки-клиента в переменной *enteredSystem* нашего класса заявки *Customer*. Функция `time()` возвращает текущее значение модельного времени;
  2. блок *tellerLines* (блок *Service*) – введите `Customer` в поле **Класс заявки**. Добавьте код в поля: **Действие при входе**: `entity.startWaiting = time()`;  
**Действие при выходе**: `waitTimeDistr.add(time() - entity.startWaiting)`;
  3. блок *queue* – введите `Customer` в поле **Класс заявки**. Добавьте код в поля: **Действие при входе**: `entity.startWaiting = time()`;  
**Действие при выходе**: `waitTimeDistr.add(time()-entity.startWaiting)`

Данный код добавляет время, в течение которого клиент ожидал обслуживания в объект сбора данных *waitTimeDistr*;

4. блок *ATM* (блок *delay*) – введите `Customer` в поле **Класс заявки**;

5. блок *sink* – введите `Customer` в поле **Класс заявки**. Напишите следующий код, чтобы сохранить в наборах данных данные о клиенте, покидающем банковское отделение (**Действие при входе**):

```
timeInSystemDistr.add(time()-entity.enteredSystem);
```

Данный код добавляет полное время пребывания клиента в банковском отделении в объект сбора данных гистограммы *timeInSystemDistr*.

Добавьте две гистограммы для отображения распределений времен ожидания клиента и пребывания клиента в системе.

Чтобы добавить гистограмму на диаграмму класса активного объекта, перетащите элемент **Гистограмма** из палитры **Статистика** в то место, куда вы хотите ее поместить. Укажите, какой элемент сбора данных хранит данные, которые хотите отображать на гистограмме: щелкните мышью по кнопке **Добавить данные** и введите в поле **Данные** имя соответствующего элемента – *waitTimeDistr*.

Аналогичным образом добавьте еще одну гистограмму и расположите ее под ранее добавленной. В поле **Данные** введите *timeInSystemDistr*. Измените заголовки отображаемых данных.

Запустите модель. Включите режим виртуального времени и посмотрите, какой вид примет распределение времени ожидания и времени пребывания клиента в системе.

Индивидуальные варианты заданий.

Внесите изменения в модель банковского отделения согласно варианту (параметры законов распределения задайте произвольно).

Вариант	Распределение вероятности прихода клиентов в банк	Вероятность обращения к кассиру/к банкомату	Время обслуживания клиента кассиром	Количество кассиров
1	Экспоненциальное	1/1	4±2	1
2	Экспоненциальное	1/2	6±2	2
3	Экспоненциальное	2/1	8±2	3
4	Экспоненциальное	1/3	7±2	4
5	Экспоненциальное	3/1	9±2	5
6	Нормальное	1/1	8±2	1
7	Нормальное	1/2	7±2	2
8	Нормальное	2/1	9±2	3
9	Нормальное	1/3	4±2	4
10	Нормальное	3/1	6±2	5
11	Треугольное	1/1	2±2	1
12	Треугольное	1/2	7±2	2
13	Треугольное	2/1	9±2	3
14	Треугольное	1/3	4±2	4
15	Треугольное	3/1	6±2	5
16	Равномерное	1/1	4±2	1
17	Равномерное	1/2	6±2	2
18	Равномерное	2/1	7±2	3
19	Равномерное	1/3	8±2	4
20	Равномерное	3/1	9±2	5

Проанализируйте поведение модели. Постройте графики и диаграммы.

## Критерии оценивания

**«5» «отлично» или «зачтено»** – студент показывает глубокое и полное овладение содержанием программного материала по МДК, в совершенстве владеет понятийным аппаратом и демонстрирует умение применять теорию на практике, решать различные практические и профессиональные задачи, высказывать и обосновывать свои суждения в форме грамотного, логического ответа (устного или письменного), а также высокий уровень овладения общими и профессиональными компетенциями и демонстрирует готовность к профессиональной деятельности;

**«4» «хорошо» или «зачтено»** – студент в полном объеме освоил программный материал по МДК, владеет понятийным аппаратом, хорошо ориентируется в изучаемом материале, осознанно применяет знания для решения практических и профессиональных задач, грамотно излагает ответ, но содержание, форма ответа (устного или письменного) имеют отдельные неточности, демонстрирует средний уровень овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности;

**«3» «удовлетворительно» или «зачтено»** – студент обнаруживает знание и понимание основных положений программного материала по МДК, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения практических и профессиональных задач, не умеет доказательно обосновать свои суждения, но при этом демонстрирует низкий уровень овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности;

**«2» «неудовлетворительно» или «не зачтено»** – студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, беспорядочно и неуверенно излагает программный материал по МДК, не умеет применять знания для решения практических и профессиональных задач, не демонстрирует овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности.



### 3. Информационное обеспечение

перечень учебных изданий, электронных изданий, электронных и Интернет-ресурсов, образовательных платформ, электронно-библиотечных систем, веб-систем для организации дистанционного обучения и управления им, используемые в образовательном процессе как основные и дополнительные источники.

#### Основные источники:

3. Разработка программных модулей программного обеспечения для компьютерных систем. (2-е изд., стер.) учебник/Федорова Г.Н. –М. ИЦ Академия,2017- 336 с.
4. Технология разработки программных продуктов (11-е изд., стер.). Учебник/ Рудаков А.В. –М. ИЦ Академия,2017 г.-208 с.
5. Разработка программных модулей программного обеспечения для компьютерных систем. Федорова Г.Н. – М.: Академия, 2017. – 336 с.

#### Дополнительные источники:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб.пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; Под ред. Л. Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2017.-400 с.
2. Богомазова Г.Н. Модернизация программного обеспечения персональных компьютеров, серверов, периферийных устройств и оборудования: учебник.- М.: ИЦ Академия, 2016.-192 с.
3. Черпаков И.В. Основы программирования: Учебник и практикум для СПО.- М.: Юрайт,2017.-219 с
4. Калайда В.Т., Романенко В.В. Технология разработки программного обеспечения: учебное пособие.-Томск: Томский межвузовский центр дистанционного образования, 2007.-257 с.

#### Электронные издания (электронные ресурсы):

1. НОУ ИНТУИТ: <http://www.intuit.ru/department/se/devis/>
  2. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. [http://real.tepkom.ru/Real\\_OM-СМ\\_A.asp](http://real.tepkom.ru/Real_OM-СМ_A.asp)
  3. Цифровая образовательная среда СПО PROОбразование
- Вичугова, А. А. Инструментальные средства разработки компьютерных систем и комплексов : учебное пособие для СПО / А. А. Вичугова. — Саратов : Профобразование, 2017. — 135 с. — ISBN 978-5-4488-0015-3. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROОбразование : [сайт]. — URL: <https://profspo.ru/books/66387> (дата обращения: 04.09.2020). — Режим доступа: для авторизир. пользователей
- Куликова, Т. А. Инструментальные средства разработки мультимедийных приложений : учебное пособие (лабораторный практикум) / Т. А. Куликова, Н. А. Поддубная. — Ставрополь : Северо-Кавказский федеральный университет, 2019. — 148 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL:

<https://profspo.ru/books/99423> (дата обращения: 12.11.2020). — Режим доступа: для авторизир. пользователей

- Сосновиков, Г. К. Средства разработки реляционных баз данных в СУБД Access 2010 : учебное пособие / Г. К. Сосновиков, Л. А. Воробейчиков. — Москва : Московский технический университет связи и информатики, 2017. — 129 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://profspo.ru/books/92481> (дата обращения: 05.11.2020). — Режим доступа: для авторизир. пользователей

**Электронно-библиотечная система:**

IPR BOOKS

- <http://www.iprbookshop.ru/66387.html>.
- <http://www.iprbookshop.ru/99423.html>.

**Веб-система для организации дистанционного обучения и управления им:**

Система дистанционного обучения ОГАПОУ «Алексеевский колледж»  
<http://moodle.alcollege.ru/>