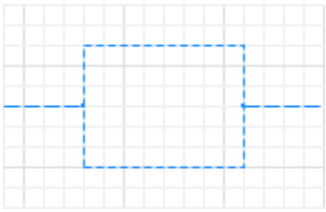
Практическая 24-26

Тема: Имитационное моделирование. AnyLogic разработка типового проекта работы нескольких терминалов с участием нескольких менеджеров и разгрузки ресурсов.. Анализ проекта.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ



Прямоугольный узел

Узлы и пути являются элементами разметки пространства, которые задают местоположение агентов в моделируемом пространстве:

- **Узел** задает место, в котором агенты могут находиться.
- **Путь** графически задает траекторию движения агентов из одного места в другой.

Узлы могут соединяться путями. Вместе они образуют сеть. В сети, узел задает место, где агенты могут останавливаться, тогда как пути, соединяющие узлы, задают маршрут, по которому агенты следуют при движении из одного узла в другой.

Элемент разметки **Прямоугольный узел** задает область прямоугольной формы.

Существует еще два вида узлов, которые вы можете использовать:

Точечный узел. Используйте его для рисования узла транспортировки и переходов в сети. Точечный узел создается автоматически при соединении двух путей.

Многоугольный узел. Используйте его, чтобы нарисовать узел сложной формы.

Чтобы задать конкретные узлы ожидания внутри прямоугольного или многоугольного узла, используйте аттракторы.

Service

Захватывает для агента заданное количество ресурсов, задерживает их, а затем освобождает захваченные им ресурсы. Эквивалентен последовательности блоков **Seize**, **Delay**, **Release** (и сам реализован именно таким способом) и должен использоваться в тех случаях, когда все, что требуется — это задержать захваченные ресурсы на заданное время, а затем их отпустить. Большинство параметров этих вложенных блоков вынесены в интерфейс блока **Service**.

Один набор ресурсов может выполнять задачи с разными приоритетами, заданные в нескольких блоках.

- Если задачи имеют одинаковый приоритет, они будут выполняться последовательно в соответствии с настройками времени выполнения каждой из них. Если освободившихся единиц ресурса хватает, то такие задачи могут выполняться одновременно. Если у двух задач одинаковый приоритет, но одна из них по какой-то причине была ранее приостановлена, то выбирается приостановленная задача.[1]
- Если для задач не задан механизм вытеснения, они будут выполняться последовательно в соответствии с настройками времени выполнения каждой из них. Если освободившихся единиц ресурса хватает, то такие задачи могут выполняться одновременно.
- Начало выполнения задачи, заданной в одном блоке, не сбрасывает приоритеты задач, заданных в других блоках.

Существует несколько способов указать ресурсы, которые следует захватить в блоке **Service**. Более подробную информацию об этом вы можете найти в статье про использование ресурсов.

Параметры

Захватить

Здесь вы можете выбрать, требуются ли для сервиса **ресурсы одного типа** или ресурсы разных типов **((альтернативный) набор ресурсов)**. Детальное описание см. в статье про использование ресурсов.

Набор(ы) ресурсов

[Параметр виден, если **Захватить: (Альтернативный) набор ресурсов**]
Здесь вы можете задать требуемые наборы ресурсов (блоки **ResourcePool**). Вы можете добавить несколько наборов с помощью кнопки **Добавить список**. Ресурсы набираются согласно их доступности. Детальное описание см. в статье про использование ресурсов.

Тип ресурсов

[Параметр виден, если **Захватить: Ресурсы одного типа**]
Блок **ResourcePool**, задающий ресурсы, которые требуются для обслуживания агента. Детальное описание см. в статье про использование ресурсов.

Количество ресурсов

[Параметр виден, если **Захватить: Ресурсы одного типа**]

Выражение, возвращающее требуемое количество ресурсов для агента.

Вместимость очереди

[Параметр виден, если не выбрана опция **Максимальная вместимость**]

Вместимость вложенной очереди queue.

Максимальная вместимость

Если опция выбрана (true), то вместимость очереди queue будет максимально возможной (ограничена константой Integer.MAX_VALUE).

Время задержки

Выражение, вычисляющее время задержки для агента.

Пересылать захваченные ресурсы

Если опция выбрана (true), захваченные ресурсы будут пересылаться в указанное местоположение.

Место назначения

[Параметр виден, если выбрана опция **Пересылать захваченные ресурсы**]

Задает место, куда будут пересылаться ресурсы. Ресурсы могут быть отправлены в следующее место назначения:

К агенту — ресурсы пересылаются в местоположение указанного агента

Узел сети — ресурсы пересылаются в указанный узел сети

Аттрактор — ресурсы пересылаются в указанный аттрактор

Узел

[Параметр виден, если **Место назначения: Узел сети**]

Узел сети, куда отправляются агенты, созданные этим блоком.

Аттрактор

[Параметр виден, если **Место назначения: Аттрактор**]

Аттрактор, куда отправляются агенты, созданные этим блоком.

По окончании, движущиеся ресурсы

Выберите, движущиеся ресурсы **Возвращаются в базовую точку** (если их сразу же не захватывает другой агент) или **Остаются на месте**.

Место агентов (queue)

Фигура разметки (узел или путь), где располагаются агенты, пока они находятся во вложенном блоке queue.

Место агентов (delay)

Фигура разметки (узел или путь), где располагаются агенты, пока они находятся во вложенном блоке delay.

ResourcePool

Задает набор ресурсов, которые могут захватываться и освобождаться агентами с помощью блоков **Seize**, **Release**, **Assembler** и **Service**.

Типы ресурсов

Ресурсы могут быть трех типов:

- **Статические** ресурсы привязаны к определенному местоположению (например, узлу) внутри сети и не могут быть перемещаться или быть перемещены. Примером статического ресурса может быть рентгеновский кабинет или весы-платформа.
- **Движущиеся** ресурсы могут перемещаться сами по себе, они могут представлять персонал, транспорт, и т.д.
- **Переносные** ресурсы могут быть перемещены агентами или движущимися ресурсами. Переносное ультразвуковое устройство или кресло-каталка могут быть примерами переносных ресурсов.

Движущиеся и переносные ресурсы имеют базовое местоположение, куда они могут при необходимости вернуться или быть возвращены. Ресурсы одного типа могут иметь индивидуальные свойства, отображаться на презентации, хранить статистику своего использования и т.д. Вы можете создать свои собственные типы ресурсов. Агент использует имя типа, чтобы ссылаться на ресурсы, и может выбирать конкретный ресурс, анализируя его свойства.

Любой ресурс может быть либо **свободен**, либо **занят**. Блок собирает статистику занятости ресурсов (непрерывная статистика процента занятых ресурсов от их общего числа). Отдельные ресурсы собирают индивидуальную статистику загруженности. Также вы можете собрать статистику времени, которое отдельный ресурс потратил на обслуживание, перерывы, поломки и дополнительные задачи.

Движущиеся ресурсы не служат препятствиями друг для друга, поэтому их движение не в полной мере соответствует реальности. Если при создании модели вы хотите учитывать возможные столкновения и

разрешение навигационных конфликтов, более подходящим инструментом для этого будут транспортеры либо двигающиеся по заданному пути, либо с произвольной навигацией.

Блоки библиотеки, которые работают с ресурсами

Библиотека Моделирования Процесса поддерживает разнообразные операции с ресурсами: агент может захватить один или несколько ресурсов (блок **Seize**), освободить ресурсы (блок **Release**), отправить захваченные ресурсы в определенное местоположение (блок **ResourceSendTo**), прикреплять ресурсы так, чтобы они перемещались вместе с агентом (блок **ResourceAttach**), и отсоединять прикрепленные ресурсы (блок **ResourceDetach**). Совершая операции над множеством ресурсов, укажите их список в параметре **ResourcePool**; например, чтобы захватить двух медсестер и один рентгеновский кабинет, укажите: Nurse, Nurse, XRay.

Если запросы от блоков **Seize** и **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.

Один набор ресурсов может выполнять задачи с разными приоритетами, заданные в нескольких блоках.

- Если задачи имеют одинаковый приоритет, они будут выполняться последовательно в соответствии с настройками времени выполнения каждой из них. Если освободившихся единиц ресурса хватает, то такие задачи могут выполняться одновременно. Если у двух задач одинаковый приоритет, но одна из них по какой-то причине была ранее приостановлена, то выбирается приостановленная задача.[1]
- Если для задач не задан механизм вытеснения, они будут выполняться последовательно в соответствии с настройками времени выполнения каждой из них. Если освободившихся единиц ресурса хватает, то такие задачи могут выполняться одновременно.
- Начало выполнения задачи, заданной в одном блоке, не сбрасывает приоритеты задач, заданных в других блоках.

[1] Каждый ресурс получает запросы из разных источников. Когда у ресурса есть несколько задач с одинаковым приоритетом (новых или ранее приостановленных), при выборе задачи к исполнению учитываются следующие правила:

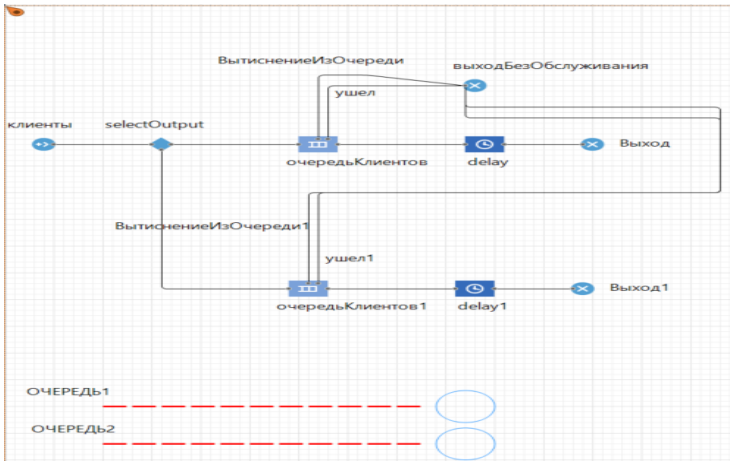
- Самый высокий приоритет — у приостановленных задач из «индивидуальной» очереди единицы ресурса.
  Такие очереди формируются агентами, находящимися в блоках, для которых значение свойства **Правило вытеснения задач** установлено на **Ожидать оригинал ресурса**.
- После этого обрабатываются прерванные задачи из очереди блока **ResourcePool**. Эта очередь формируется агентами, находящимися в блоках, для которых значение свойства **Правило вытеснения задач** установлено на **Захватывать любой ресурс**.

После выполнения задач из этих очередей ресурс сначала выполняет задачи из «индивидуальной очереди», а затем — из очереди блока **ResourcePool**.

Если этот порядок выполнения вам не подходит, вы можете вручную уменьшить приоритет определенной задачи с помощью кода: для этого используйте действие блока **При остановке задачи**.

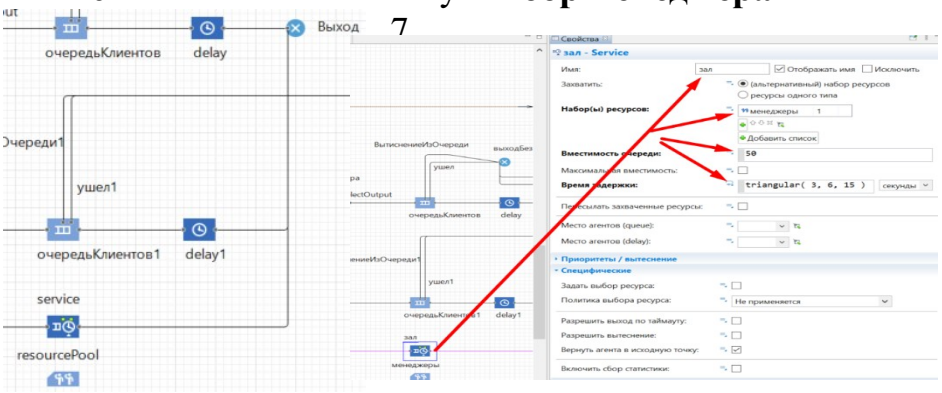
ХОД РАБОТЫ

Задание1



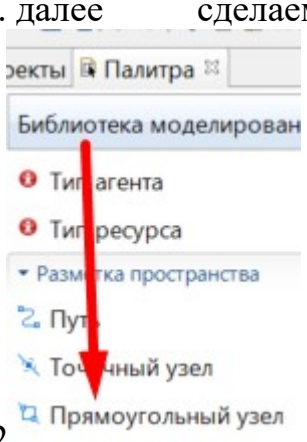
1. **запустите ваш проект, который я вам говорила сохранить под новым названием** или
2. удалите элемент выбора с 5 портами и добавьте элемент с двумя портами, как делали ранее.
3. Соедините элемент ветвления с очередями, путям и точечным узлам укажите очереди и элементы delay
4. Установите равнозначное распределение элементу.

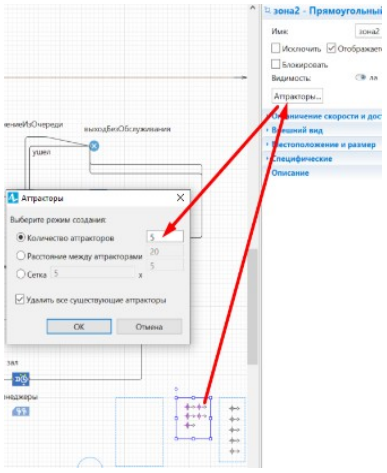
5. **Добавьте еще один элемент ветвления с двумя выходами выборМнеджера.** Создаем для того, чтобы клиенты, те которые приходили, могли выбрать пойти к менеджеру или сразу к терминалу.
6. Установим элементу **ВыборМенеджера** в свойствах **вероятность 0.6.**



7 Добавим новый элемент **service** Service , он вам поможет распределить ресурсы с разными задержками по времени.
Добавим элемент **resourcePool** Resource Pool ,задает ресурсы определённого типа (движущиеся, статичные, переносные), соединим его с элементом и выход соединим с элементом Выход.

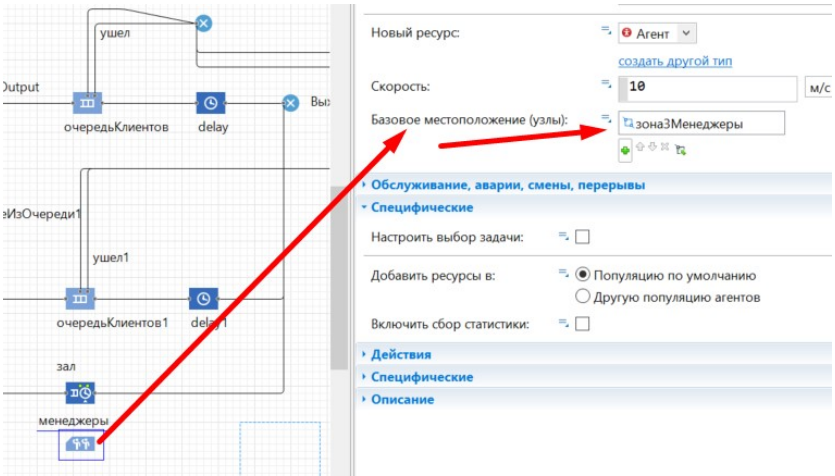
9. Выход 1, удалите элемент и связь от dilay1 и установите связь на Выход
10. **resourcePool** переименуем в **Менеджеры**, количество в свойствах данного элемента укажем **5**
11. **service** переименуем в **зал** и в свойствах данного элемента зададим **Наборы ресурсов-Менеджеры, вместимость очереди** укажем **50, время задержки укажите** `triangular(36,15)`
12. далее сделаем анимацию для данных элементов:



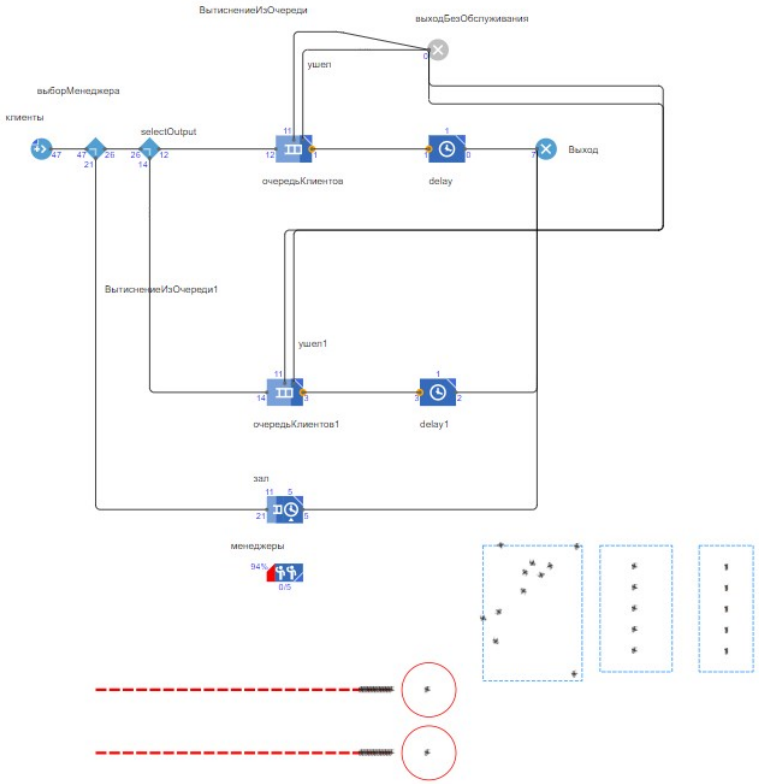


добавим элемент **прямоугольный узел** Многоугольный узел (это зоны различные), мы создадим 3 зоны, а значит добавьте 3 элемента.
Переименуйте их на Зона1, Зона2, Зона3.
У данного элемента, есть свойство Аттракторы, они создают место положение агентов.
Установите по 5 Аттракторы зоне2 и зоне3
12.1 Аттракторы из зоны 3 поверните в левую сторону (так как мы создаём имитацию общения клиентов и менеджеров, а они общаются лицом к лицу.) Для этого, нажмите на каждый элемент внутри зала3 и стрелку разверните в лево.
12.5 Переименуйте **зона1** в **зону1Ожидание**, **зона2**-в **зона2Обслуживание и зона3** в **зону3Менеджеры**, для более понятного поиска элементов из списка, для переименование используем CTRl+Enter

12.6 Выберите элемент **Зал** и в свойствах установите ему: место агентов очередь -Зона1Ожидание, место агентов заявки – зона2Обслуживание



12.7 Далее нажмите элементменеджеры и перейдите в его свойтва и укажите Базовое местоположение(узлы-Зона3Менеджеры.)
12.8 Добавьте новый тип ресурса на схему и назовите его **персонал**,нажмите далее из списка



картинок выберите **Служащий** и нажмите далее,**готово**

12.9 **Перейдите на основную рабочую**

область

12.10 Нажмите на элемент **менеджеры** и в его свойствах **новый ресурс** укажите из списка **персонал**

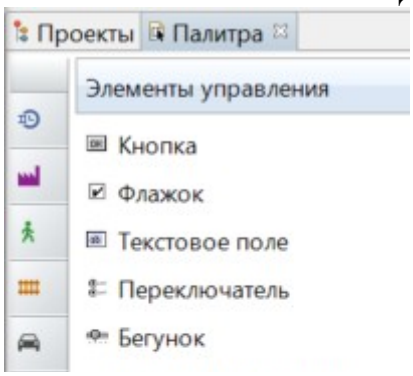
12.11 Результат**.**

Задаине2

1. **Добавьте** новый элемент **Бегунок** из **библиотеке элементов управления**
2. В свойствах элементА **Бегунок** нажмите галочку (связать с) и

укажите **менеджеры,** максимальное значение установите **6, добавьте метки,** данный элемент позволит в системе увеличивать и уменьшать количество менеджеров.

3. ЗАПУСТИТЕ МОДЕЛЬ И ПРОКОМЕНТИРУЙТЕ МОДЕЛЬ, СДЕЛАЙТЕ ПОЛНЫЙ АНАЛИЗ И ВЫАОДЫ ПО МОДЕЛИ, РЕЗУЛЬТАТЫ ЗАПИШИТЕ.



4. При анализе попробуйте изменять входные параметры интенсивности ресурсов и их обработки
5. Выводы и результаты разных входных параметров запиши в вывод и прикрепите скриншоты.

Задание 3

Создать еще один узел обслуживания клиентов, по типу кассиров.

1. Для этого добавьте еже один элемент сервис **service** на рабочую область

1.1 Назовите элемент **service** Service **касса**

2. Добавьте на рабочую область еще один элемент **resourcePool** Resource Pool

2.1 переименуйте элемент **resourcePool** на **кассиры.**

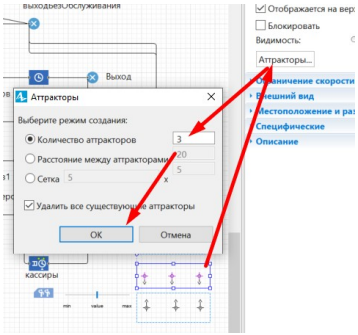
2.1 установите в свойствах элемента **кассиры** количество ресурса **3**

3. Добавьте элемент выбора с двумя выходами **selectOutput**

3.1 задайте элементу **selectOutput вероятность** оплаты на кассе **30%**

3.2 свяжите один выход из элемента **selectOutput** напрямую с **Выход** схемы напрямую.

3.3 Второй выход из элемента **selectOutput** свяжите с добавленным вашим элементом **кассиры** и выход уже из элемента кассиры присоедините к **Выход** из схемы.

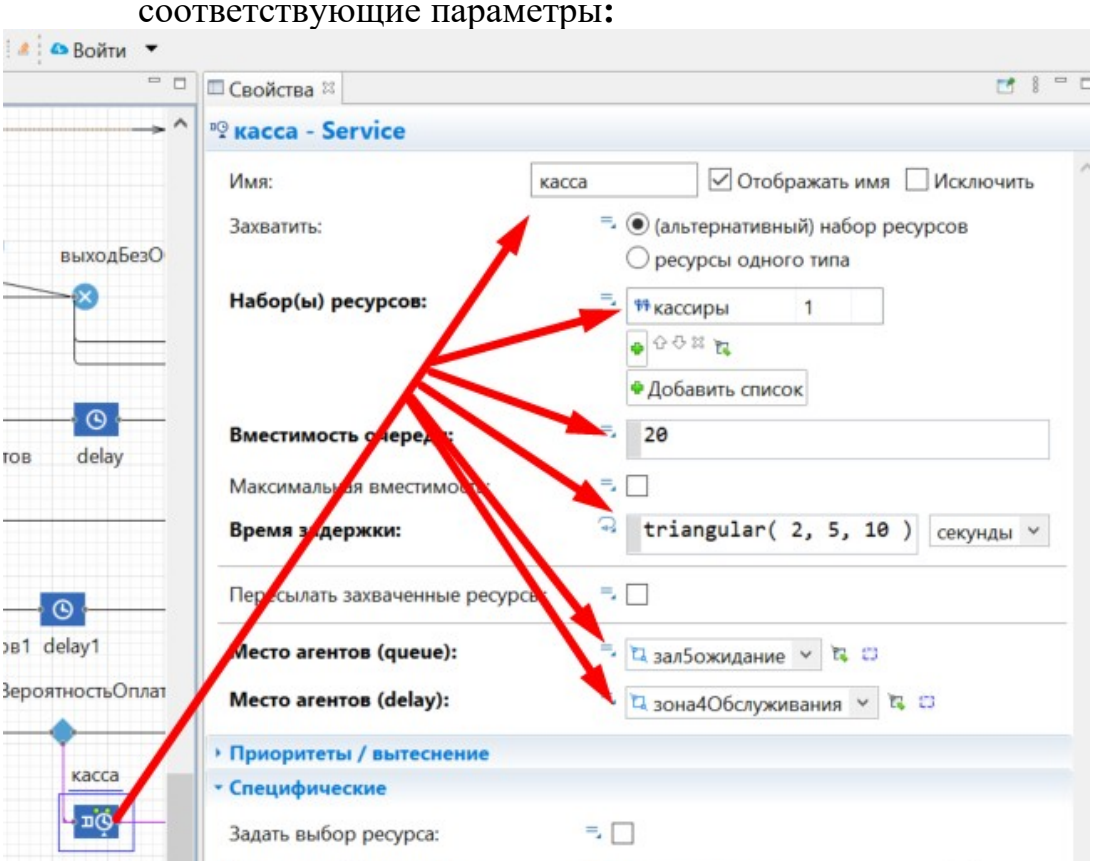


4. Добавьте 3 элемента **прямоугольный узел** Многоугольный узел на рабочую область

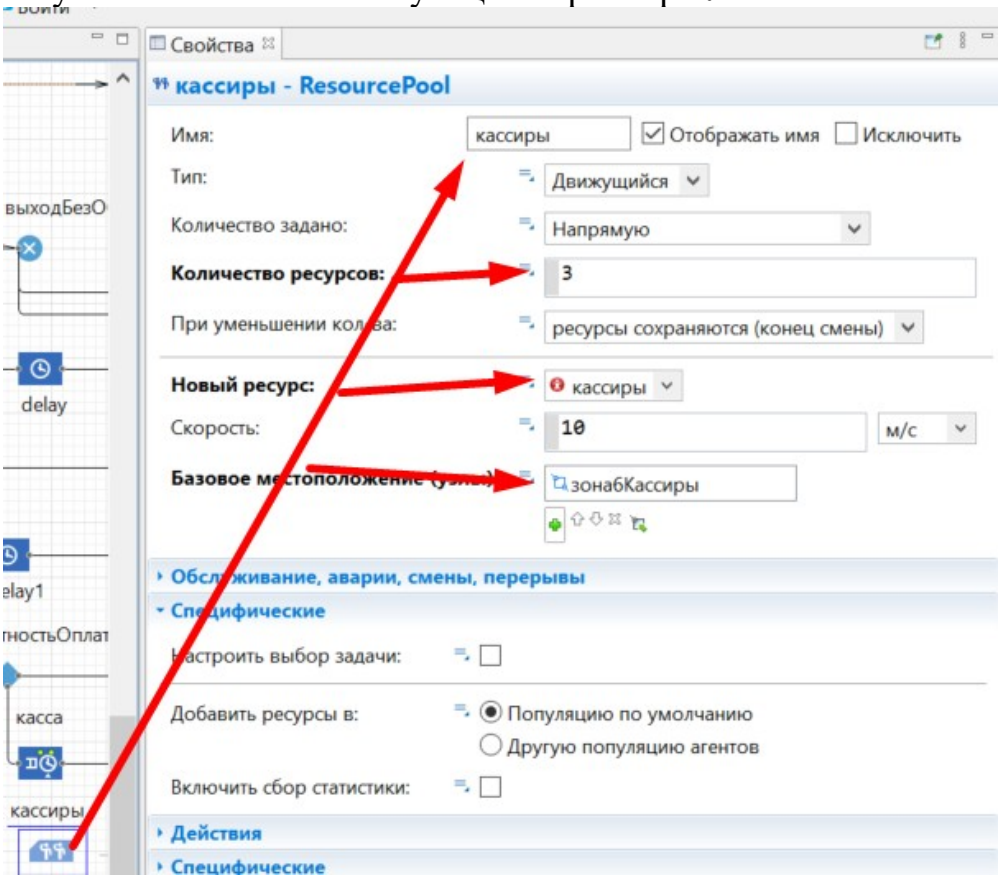
4.1 двум из них добавьте по 3 **аттрактора** в каждую зону

4.2 переименуйте каждую зону логическим названием

5. добавьте на рабочую область новый **тип ресурса** Тип ресурса , дайте название **Кассиры**.
6. Нажмите элемент **service** Service **касса** перейдите в его свойства и установите соответствующие параметры**:**



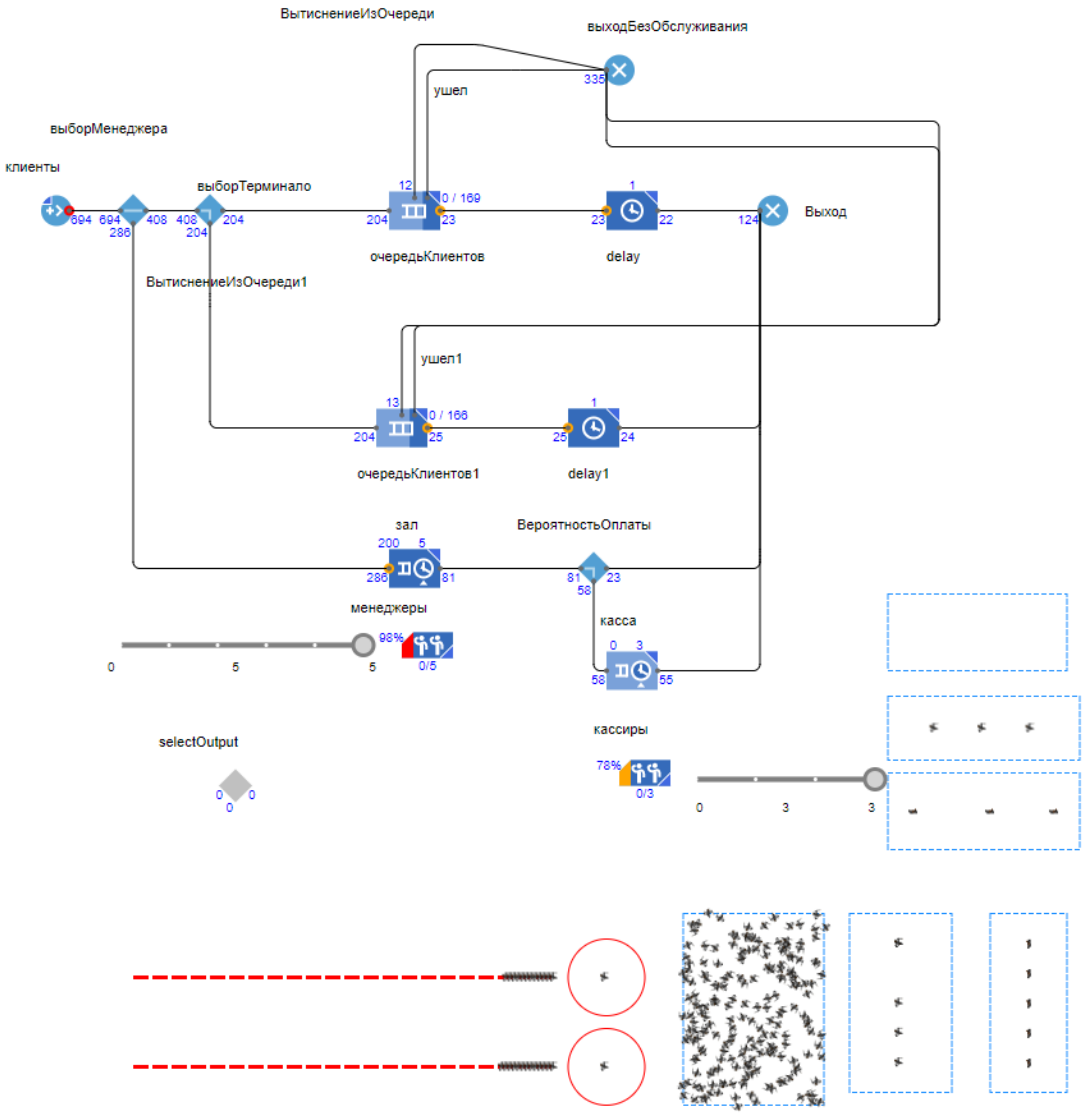
7. Нажмите элемент **resourcePool** Resource Pool **кассиры** перейдите в его свойства и установите соответствующие параметры**:**



8. Добавьте элемент **бегунок** на рабочую область и установите в свойствах данного элемента
   8.1 связь с **кассирами**

8.2 максимальное значение **3**

9. **запустите схему и проанализируйте результат, выводы и свои наблюдения запишите в отчет иллюстрируя скринами.**
10. *При запуске у вас будет ошибка на каком-то моменте работы, ошибка связана будет с ограниченностью вместимости очереди, измените вместимость элемента Менеджеры и Кассиры на 500 и 200.*
11. *Запустите и продолжите анализ схемы.*
результат должен выглядеть таким образом



Контрольные вопросы:

1. Чем отличается тип ресурса от тип агента?
2. Что такое многоугольный узел для чего его применяют?
3. Для чего нужен элемент resource Pool?
4. Для чего нужен элемент Service?
5. Как указать элементу resource Pool 5 attractor ?
6. Назовите основные свойства элемента resource Pool, Service, многоугольный узел
7. Для чего нужен бегунок? На что он указывает в системе?

Содержание отчета:

1. Тема, цель практической работы
2. Поэтапное описание выполнения практической работы
3. Скриншоты или результат практической
4. Краткие ответы на контрольные вопросы