

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Кафедра «Кибербезопасность информационных систем»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине
МОДЕЛИ И МЕТОДЫ ИССЛЕДОВАНИЯ ИНФОРМАЦИОННЫХ
ПРОЦЕССОВ и СИСТЕМ**

Ростов-на-Дону
ДГТУ
2026

УДК 004.942

Составитель: Н.Ю. Батурина

Методические указания для выполнения контрольной работы по дисциплине «Модели и методы исследования информационных процессов и систем» / сост. Н.Ю. Батурина – Ростов-на-Дону : Донской государственный технический университет, 2023. – 35 с.

Содержат краткую теорию, перечень экзаменационных вопросов, заданий контрольной работы, требования по ее выполнению, список информационных ресурсов.

Предназначены для обучающихся направления подготовки 09.04.02
УДК 004.942

Ответственный за выпуск: зав. кафедрой «Кибербезопасность информационных систем»

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия: 344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный технический университет, 2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ	4
2. КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ДИСЦИПЛИНЕ «МОДЕЛИ И МЕТОДЫ ИССЛЕДОВАНИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ»	9
3. ЗАДАНИЯ КОНТРОЛЬНОЙ РАБОТЫ И ТРЕБОВАНИЯ ПО ВЫПОЛНЕНИЮ	9
Задание 1. Моделирование процесса обработки запросов сервером в Anylogic	10
Задание 2. Моделирование процесса обслуживания объектов системы в Anylogic.....	10
Задание 3. Моделирование функционирования информационной системы «файл-сервер» в Anylogic	11
ИСХОДНЫЕ ДАННЫЕ ПО ВАРИАНТАМ.....	12
4. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ	16
Задание 1. Моделирование процесса обработки сообщений сервером в Anylogic.....	16
Задание 2. Моделирование процесса обслуживания объектов системы в Anylogic.....	21
Задание 3. Моделирование функционирования информационной системы «файл-сервер»..	27
СПИСОК ИНФОРМАЦИОННЫХ РЕСУРСОВ	34

ВВЕДЕНИЕ

Цель изучения дисциплины «Модели и методы исследования информационных процессов и систем» заключается в формировании теоретических основ и практических навыков, необходимых для анализа, моделирования и оптимизации информационных процессов и систем. Основные задачи курса включают изучение методов формализации, моделирования и анализа сложных информационных систем, освоение современных подходов к исследованию динамики информационных потоков, оценку эффективности функционирования систем обработки информации и принятие обоснованных решений относительно их совершенствования. Материалы для изучения дисциплины и выполнения лабораторных работ (контрольных заданий) представлены в курсе на портале do.scif [1].

В результате освоения дисциплины «Модели и методы исследования информационных процессов и систем» у обучающегося должны сформироваться определенные учебным планом компетенции, и магистрант должен:

Знать

- методы анализа и синтеза информационных систем;
- формальные модели систем;
- модели предметных областей информационных систем;
- математические модели информационных процессов.

Уметь

- разрабатывать модели предметных областей;
- применять на практике методы и средства проектирования информационных систем;
- проводить исследования характеристик компонентов и информационных систем в целом.

Иметь навыки

- применения методов анализа и синтеза информационных систем;
- разработки математических моделей информационных систем;
- автоматизированного проектирования информационных систем;
- построения математически моделей и имитационных моделей в Matlab, Anylogic

Для достижения поставленных целей изучения дисциплины учебным планом предусмотрены лекции, лабораторные занятия, контрольная работа (заочная форма обучения), самостоятельная работа и оценка результатов обучения по дисциплине.

Данное пособие поможет овладеть навыками создания имитационных моделей информационных процессов и систем в системе моделирования Anylogic. Для самостоятельного углубленного изучения методов моделирования в этой системе рекомендуется использовать информационные ресурсы [1-5]. Знание принципов и возможностей имитационного моделирования, умение создавать и применять модели являются в настоящее время необходимыми требованиями к инженеру, менеджеру, бизнес-аналитику.

1. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Имитационное моделирование

Особым видом математических моделей являются имитационные модели. Имитационная модель – это компьютерная программа, которая описывает структуру и воспроизводит поведение реальной системы во времени. Можно сказать, что имитационная модель – это набор правил, согласно которым система переходит из одного состояния в другое. Правила могут задаваться самыми различными способами, например, дифференциальными уравнениями, диаграммами состояний, диаграммами процессов, расписаниями. Выходные данные модели всегда можно проанализировать прямо по ходу моделирования.

Имитационные модели разрабатываются с помощью специализированного программного обеспечения, в котором используются различные языки моделирования.

Можно выделить основные преимущества имитационного моделирования по сравнению с аналитическим.

- Имитационные модели позволяют анализировать системы и находить решения в тех случаях, когда такие методы, как аналитические вычисления не справляются с задачей.
- Разрабатывать имитационную модель гораздо проще, чем аналитическую, поскольку процесс создания модели будет инкрементальным (пошаговый, усложняющийся постепенно) и модульным.
- Структура имитационной модели естественным образом отображает структуру моделируемой системы.
- Имитационная модель позволяет отслеживать все объекты системы, учтенные в выбранном уровне абстракции, добавлять метрики и проводить статистический анализ.
- Имитационная модель дает возможность проигрывать модель во времени и анимировать ее поведение. Анимация будет неоспоримым преимуществом при демонстрации модели и может оказаться полезной для верификации (тестирование на предмет соответствия проекта техническому заданию – замыслу проектировщика).

В имитационном моделировании под методом понимается некая основа, которую мы используем, чтобы «перевести» систему из реального мира в мир моделей. Метод предполагает определенный язык, положения и условия для разработки модели. На данный момент, существует три метода (концепции) моделирования:

- системная динамика и динамические системы;
- дискретно-событийное моделирование;
- агентное моделирование.

Каждый метод применяется в некотором диапазоне уровней абстракции. Системная динамика предполагает очень высокий уровень абстракции и, как правило, используется для стратегического моделирования. Дискретно-событийное моделирование поддерживает средний и низкий уровни абстракции. Между ними находятся агентные модели, которые могут быть как очень детализированными, когда агенты представляют физические объекты, так и предельно абстрактными, когда с помощью агентов моделируются конкурирующие компании или правительства государств.

Общие сведения о среде AnyLogic

Среда имитационного моделирования AnyLogic основана на Java и базируется на платформе Eclipse [1] – современном стандарте для бизнес-приложений. Благодаря Eclipse AnyLogic работает на всех распространенных операционных системах (Windows, Mac, Linux и т.д.).

Для выполнения практических и лабораторных заданий по моделированию в системе AnyLogic нужно скачать и установить на компьютере свободно распространяемую версию Personal Learning Edition (время ее использования не ограничено) с официального сайта разработчика [2] <https://www.anylogic.ru/downloads/>.

При первоначальном запуске программы Anylogic отображается Начальная страница, на которой пользователю предлагается ознакомиться с многочисленными примерами моделей, разработанных в этой системе. Пока не создана своя модель, можно открыть любой из этих примеров, посмотреть структуру, код, запустить на выполнение. Попасть на эту страницу в ходе дальнейшей работы можно из меню Справка. Нужно сказать, что даже в бесплатной версии AnyLogic разработчики предоставляют огромное количество материалов для обучения.

На Рисунок 1.1 показан интерфейс среды имитационного моделирования AnyLogic. Сверху располагаются Меню, панели инструментов, по центру – рабочая область (графический редактор или область структурной диаграммы), справа – окно свойств объектов, слева – Окно панелей Проекты и Палитра, эти панели переключаются щелчком по заголовку. В нижней части размещаются окно ошибок и др. Отображение тех или иных окон можно настроить с помощью кнопки Вид в главном меню AnyLogic.

При настройке свойств объектов модели часто используются законы распределения, например, показательный, треугольный, нормальный и др. Эти законы можно выбирать из списка в окне, если на



панели инструментов Anylogic нажать кнопку - Выберите распределение вероятностей. Кроме того, в окне распределений приводится описание параметров распределения.

Окно панели Проекты (слева), навигатор системы, отображает имена проектов, открываемых в последнее время. При ненадобности их можно закрыть и окно станет пустым. Каждый проект организован иерархически в виде дерева: сам проект образует верхний уровень, эксперименты, классы активных объектов и Java классы образуют следующий уровень. Элементы, входящие в состав активных объектов, вложены в соответствующую подветвь дерева класса активного объекта и т. д. В меню AnyLogic доступна Справка (F1) по AnyLogic, ее библиотекам и учебным пособиям.

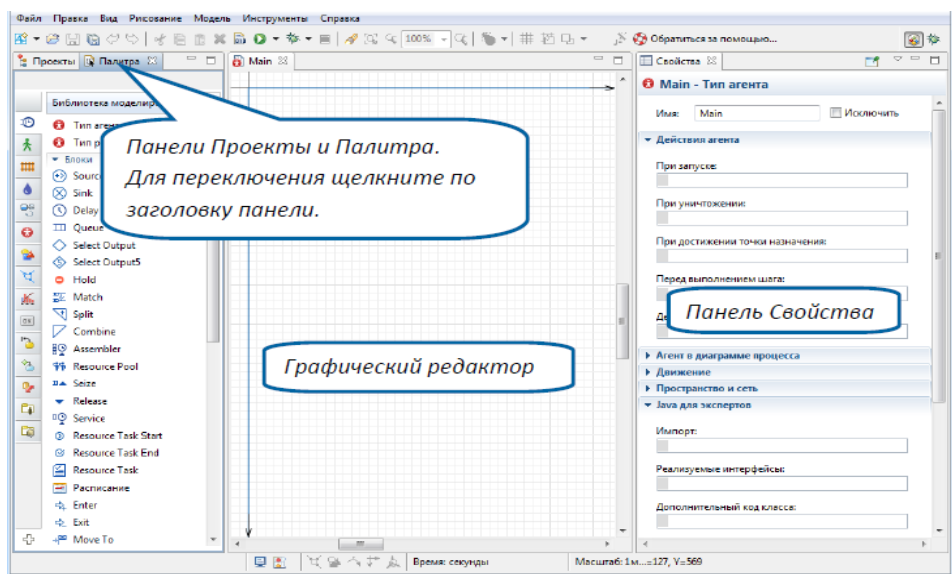


Рисунок 1.1 - Интерфейс среды имитационного моделирования AnyLogic

Окно панели Палитра (слева) включает элементы, которые могут быть добавлены на структурную диаграмму активного объекта. Элементы разбиты по группам, отображаемым на разных вкладках Палитры. Количество отображаемых вкладок можно увеличить (уменьшить), используя «+» в нижнем левом углу окна Палитра. Вкладки переключаются по щелчку. Объекты палитры добавляются на диаграмму модели путем перетаскивания их в окно графического редактора.

Для создания модели выберите в главном меню AnyLogic: **Файл > Создать > Модель**. Откроется диалоговое окно Новая модель (Рисунок 1.2), в котором нужно задать имя модели, указать путь сохранения модели или принять тот путь, который создается по умолчанию. Имя Java пакета можно поменять, но оно обязательно *начинается со строчной буквы!* Созданный проект не забывайте периодически сохранять по ходу наполнения данными.

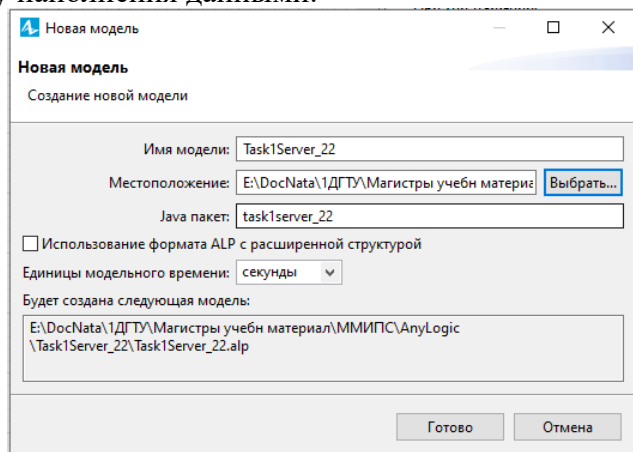


Рисунок 1.2

В результате создания модели по сути первоначально создается структура проекта (рисунок 1.3) в виде иерархического дерева, которую можно увидеть, открыв окно панели Проекты. По умолчанию в каждой модели создается один тип агента – **Main** (имя можно поменять) и один простой эксперимент **Simulation: Main**, содержащий настройки запуска этой модели, встроенную **Базу данных** и **Презентацию**. Для создания в модели новых типа агента, эксперимента, базы данных, Java-

класса и др. нужно в окне Проекты щелкнуть по имени модели правой кнопкой, далее в меню Создать выбрать из раскрывающегося списка создаваемый класс.

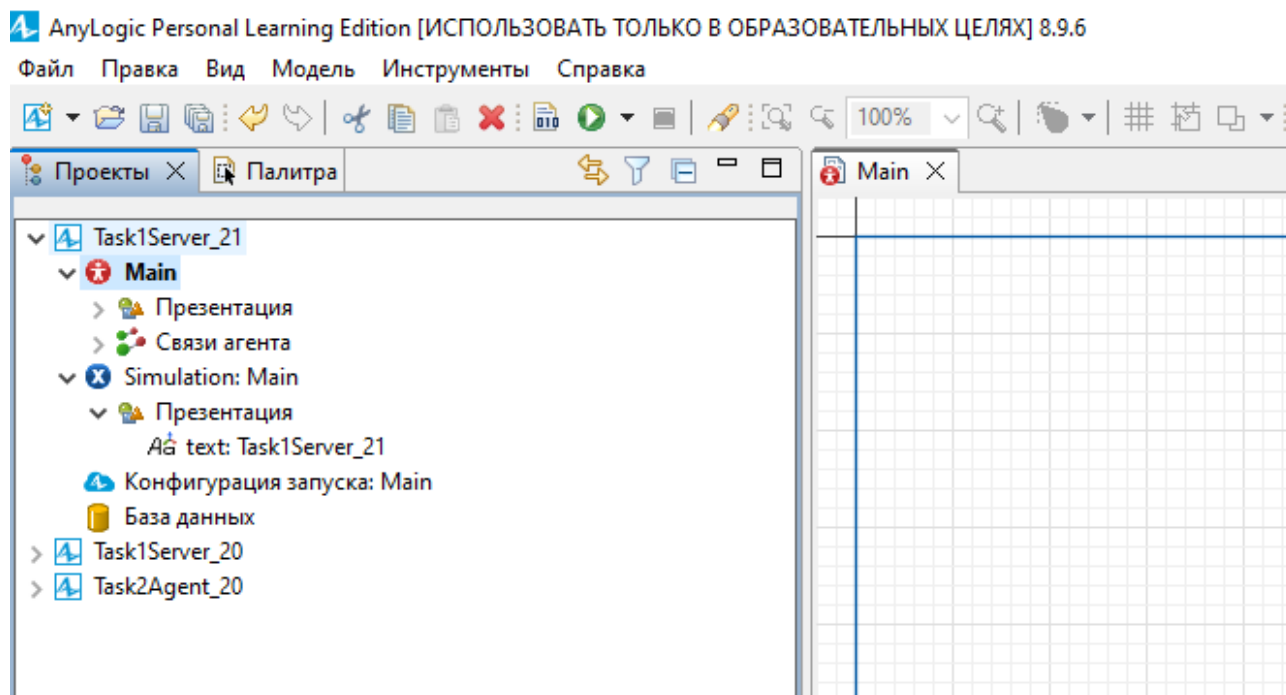


Рисунок 1.3 – Структура проекта Nask1Server_21

Если дважды щелкнуть по иконке  **Main** в окне Проекты, то справа или по центру, если открыто окно свойств, открывается графический редактор диаграммы агента верхнего уровня **Main** (Рисунок 1.). Это главное графическое окно проекта, в нем создается структурная схема, размещается анимационная картинка, которая отображается при проигрывании модели. В принципе, имя Main можно заменить на другое, но тогда так же нужно изменить имя простого эксперимента  **Simulation: Main**.

Следует заметить, что термин «Агент» в AnyLogic используется в широком смысле и применяется к различным по смыслу и действиям объектам. Здесь в качестве агентов могут быть транзакты и ресурсы, но, кроме этого, узлы иерархического дерева, проекты, идеи и т.д.

Запуск эксперимента

Прежде чем запускать модель, нужно настроить окно свойств Простого Эксперимента, которое появляется по щелчку по иконке  **Simulation: Main** в окне Проекты (Рисунок 1.4). Обязательно нужно выбрать Агент верхнего уровня Main, так как диаграмма процесса создавалась в окне графического редактора именно этого агента. В настройке, показанной на рисунке, предполагается, что эксперимент длится в течение 3600 сек. модельного времени. Если не указать этого, то по умолчанию время прогона будет не ограничено.

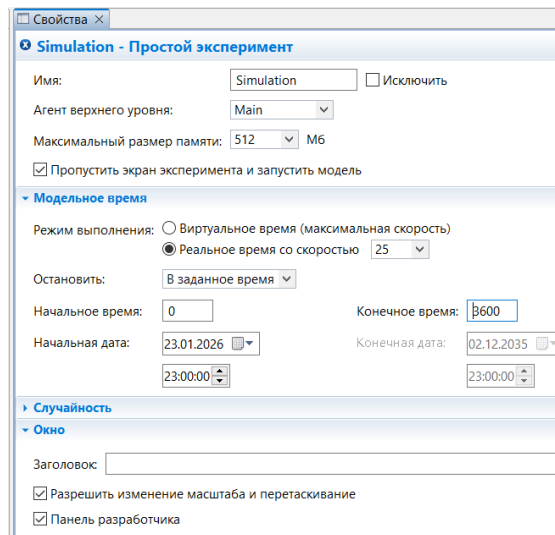


Рисунок 1.4

Если единицы модельного времени должны быть другими: минуты, часы, то нужно в окне свойств **Simulation: Main** кликнуть кнопкой мыши по имени проекта и установить нужные единицы модельного времени (рисунок 1.5).

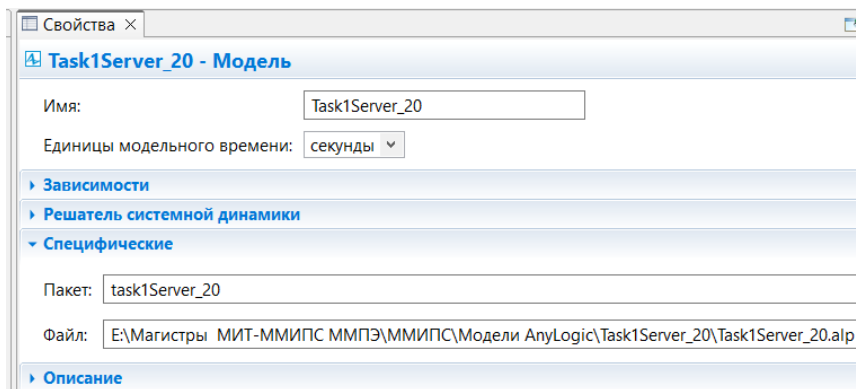


Рисунок 1.5

Для запуска модели нужно выбрать имя эксперимента в списке, отображаемом при щелчке по черной стрелке  на панели инструментов. По умолчанию по зеленой стрелке запускается эксперимент, который запускался предыдущим. После щелчка по стрелке появляется окно презентации эксперимента, в котором после запуска эксперимента отображается ход эксперимента (рисунок 1.6).

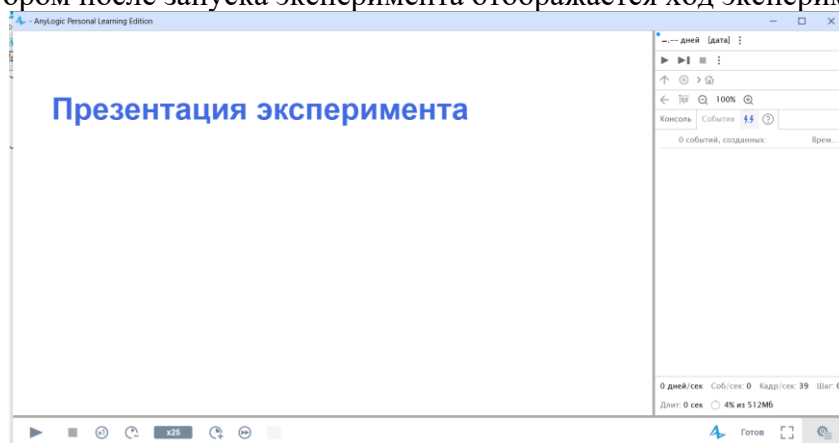


Рисунок 1.6

Справа находится окно разработчика, в котором можно выбирать различные окна просмотра и объекты во время прогона. Нижняя панель позволяет изменять запускать и останавливать эксперимент, изменять скорость прогона, режимы виртуального/реального времени.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ДИСЦИПЛИНЕ «МОДЕЛИ И МЕТОДЫ ИССЛЕДОВАНИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ»

Понятие модели, цели моделирования. Системный и классический подходы при моделировании.

Основные этапы построения математической модели. Математическая модель системы, ее составляющие.

Классификация математических моделей.

Математические схемы моделирования.

Моделирование процессов и систем в Matlab. Интерфейс, режимы работы.

Математическое моделирование в MatLab с использованием матричных операций и графики.

Основные функции.

Процедуры, функции, аноним-функции в Matlab, работа с m-файлами.

Имитационное моделирование. Этапы его развития, языки.

Концепции в имитационном моделировании.

Современные системы имитационного моделирования. Возможности и принципы работы.

Среда имитационного моделирования Simulink. Палитры компонентов.

Среда имитационного моделирования AnyLogic. Интерфейс, панели проектов, палитры инструментов, свойств. Библиотеки, графический редактор.

Дискретно-событийное моделирование в AnyLogic. Создание структурной схемы с помощью библиотеки процессов, настройка свойств объектов.

Разработка моделей системной динамики в AnyLogic. Накопители, потоки, их свойства.

Агентное моделирование в AnyLogic. Создание популяций агентов. Использование диаграмм состояний для описания поведения агентов.

Непрерывно-детерминированные модели (D-схемы).

Моделирование динамических систем в MatLab+Simulink.

Линейные динамические системы, исследование устойчивости в Matlab+Simulink.

Нелинейные динамические системы. Модель Лотки-Вольтерра. Применение в информационных технологиях.

Непрерывно-стохастические модели (Q-схемы). Задачи систем массового обслуживания. Классификация систем массового обслуживания.

Марковские случайные процессы. Поток событий. Граф состояний системы. Уравнения Колмогорова.

Математические модели систем массового обслуживания. Многоканальная СМО с отказами.

Моделирование СМО в MatLab+Simulink.

3. ЗАДАНИЯ КОНТРОЛЬНОЙ РАБОТЫ И ТРЕБОВАНИЯ ПО ВЫПОЛНЕНИЮ

Контрольная работа (ЗФО) оформляется в печатном виде согласно требованиям к оформлению письменных работ обучающихся в ДГТУ, в соответствии со своим вариантом и включает:

- титульный лист;
- постановку задач исследования (формулировку заданий);
- по каждому заданию описание хода работы (проводимых расчетов и исследований с указанием используемых формул, алгоритмов, методов и инструментальных средств);
- результаты и выводы по каждому заданию;
- перечень используемых источников литературы и интернет-источников.

Исходные данные по вариантам представлены в таблицах 3.1-3.3 и в гугл-таблице [6] .

Контрольная работа (ЗФО) включает три задания. Номером варианта служит число N, равное порядковому номеру в электронном журнале.

Задание 1. Моделирование процесса обработки запросов сервером в Anylogic

Постановка задачи. Сервер обрабатывает запросы, поступающие с автоматизированных рабочих мест с интервалами, распределенными по показательному закону со средним значением времени между прибытиями $timeBetween$ (сек.).

Время обработки сервером одного запроса распределено по показательному закону с заданным средним значением $timeServ$ (или по нормальному закону с математическим ожиданием $timeServ$).

Сервер имеет входной буфер с заданной ёмкостью $bufferCapacity$.

Построить имитационную модель и провести модельный эксперимент в течение времени $timeMax$ (модельное время - секунды). Определить характеристики модели (математические ожидания МО случайных величин, вероятности P событий).

Сделать анимацию процесса перемещений запросов и отображение результатов на диаграммах, гистограммах. Использовать элементы управления для параметров модели. Провести несколько экспериментов с различными значениями управляемых параметров.

В выводах провести анализ полученных результатов.

Исходные данные по вариантам приведены в таблице 3.1. Недостающие значения параметров задать самостоятельно.

Задание 2. Моделирование процесса обслуживания объектов системы в Anylogic

Постановка задачи.

Система состоит из n однотипных объектов, каждый из которых может находиться в трех состояниях:

S1- соответствует штатной работе объекта,

S2 -тестирование агрегатов объекта,

S3 - сервисное обслуживание, когда объект не работает штатно.

В начальный момент все объекты системы находятся в состоянии S1. Переход из состояния S1 в S2 определяется параметром λ_{12} . Известно, что в результате тестирования некоторые объекты оказываются исправными и возвращаются в состояние S1 (переход λ_{21}). Если объект в результате тестирования показал неисправности, то он переходит в состояние обслуживания S3 (переход λ_{23}). После устранения неисправностей объект возвращается в рабочее состояние S1(переход λ_{31}). Возможен также экстренный переход из состояния S1 в состояние S3 (переход λ_{13}).

Создать агентную имитационную модель и провести модельный эксперимент в течение времени $timeMax$ (дн). Настроить анимацию, отображающую все объекты (выбрать для объектов картинку) в заданном пространстве, в которой цвет объекта будет меняться при изменении состояния.

Построить диаграмму с накоплением, отражающую количество объектов, находящихся в состояниях S1,S2,S3. Определить среднее время нахождения в каком-либо состоянии.

В выводах провести анализ полученных результатов.

Исходные данные по вариантам приведены в таблице 3.2. Недостающие значения параметров задать самостоятельно. Для каждого перехода λ_{ij} выбрать закон распределения. Использовать различные распределения (показательный, нормальный, треугольный законы) для разных переходов.

Задание 3. Моделирование функционирования информационной системы «файл-сервер» в Anylogic

Постановка задачи.

Создать модель информационной системы сети типа «файл-сервер», отождествляя функционирование системы с системой массового обслуживания (СМО), описываемой марковским случайным процессом с непрерывным временем и n дискретными состояниями:

состояние S_0 – все пользователи в пультовой фазе, p_0 – вероятность этого состояния;

состояние S_i – i пользователей в системной фазе ($i=1,2,\dots,n$), $(n-i)$ – в пультовой фазе, p_i – вероятность этого состояния.

Работа этой системы описывается системой дифференциальных уравнений Эрланга-Колмогорова относительно вероятностей состояний

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} p_j(t) \lambda_{ji} - p_i(t) \sum_{j \neq i} \lambda_{ij}$$

где λ_{ij} – интенсивности переходов.

Процесс взаимодействия пользователей и системы складывается из повторяющихся системной и пультовой фаз. Продолжительности пультовой и системной фаз являются случайными величинами, распределенными по показательному закону с математическими ожиданиями $tpAvg$, $tsAvg$. Количество пользователей – n . Интенсивности переходов λ_{ij} выражаются через $tpAvg$, $tsAvg$.

Создать имитационную модель как модель системной динамики, провести модельный эксперимент в течение времени $timeMax$ (мин). Использовать элементы управления для параметров.

Определить вероятности различных состояний системы (отобразить их на графиках), показатели эффективности работы СМО (среднее количество запросов в системной фазе и среднее время нахождения запроса в системной фазе и др.).

Создать базу данных для хранения значений показателей эффективности и вероятностей состояний.

Исходные данные по вариантам приведены в таблице 3.3. Недостающие значения параметров задать самостоятельно.

В выводах провести анализ полученных результатов.

ИСХОДНЫЕ ДАННЫЕ ПО ВАРИАНТАМ

Таблица 3.1

Номер варианта N	timeBetween (с заданной интен- сивностью, по- казательный за- кон, значение и управление)	timeServ (значение и управление)	bufferCapacity (значение и управление)	Длительность эксперимента timeMax	Характе- ристики	Анима- ция (разные формы линии, обозначения сервера, кар- тинки для за- проса)	Диаграммы	Гистограммы плотности рас- пределения
1	бегунок			36000	Р обработки за- проса	Движение за- проса по линии к серверу	Средний коэф- фициент использования сервера	Р обработки за- проса
2		бегунок		72000	МО времени обработки за- проса	Движение за- проса по линии к серверу	Средняя длина очереди	МО времени обработки за- проса
3			бегунок	36000	Р необработки запроса	Движение за- проса по линии к серверу	Средний коэф- фициент использования сервера	Р необработки запроса
4	бегунок			72000	МО времени нахождения в очереди	Движение за- проса по линии к серверу	Средняя длина очереди	МО времени нахождения в очереди
5		бегунок		36000	Р обработки за- проса	Движение за- проса по линии к серверу	Средний коэф- фициент использования сервера	Р обработки за- проса
6			бегунок	72000	МО времени обработки за- проса	Движение за- проса по линии к серверу	Средняя длина очереди	МО времени обработки за- проса
7	бегунок			36000	Р необработки запроса	Движение за- проса по линии к серверу	Средний коэф- фициент использования сервера	Р необработки запроса

8		бегунок		72000	МО времени нахождения в очереди	Движение запроса по линии к серверу	Средняя длина очереди	МО времени нахождения в очереди
9			бегунок	36000	Р обработки запроса	Движение запроса по линии к серверу	Средний коэффициент использования сервера	Р обработки запроса
10	бегунок			72000	МО времени обработки запроса	Движение запроса по линии к серверу	Средняя длина очереди	МО времени обработки запроса

Таблица 3.2

Номер варианта N	Количество объектов системы n	λ_{12}	λ_{21}	λ_{23}	λ_{31}	λ_{13}	Характеристики	Длительность эксперимента timeMax
1	100	По таймауту	При выполнении условия	С заданной интенсивностью	По таймауту	С заданной интенсивностью	Среднее время нахождения в сост S3	200
2	120	С заданной интенсивностью	По таймауту	По таймауту	С заданной интенсивностью	При выполнении условия	Среднее время нахождения в сост S1	150
3	150	По таймауту	С заданной интенсивностью	С заданной интенсивностью	По таймауту	При выполнении условия	Среднее время нахождения в сост S2	200
4	100	С заданной интенсивностью	При выполнении условия	По таймауту	С заданной интенсивностью	С заданной интенсивностью	Среднее время нахождения в сост S3	150
5	120	По таймауту	При выполнении условия	С заданной интенсивностью	По таймауту	По таймауту	Среднее время нахождения в сост S1	200
6	150	С заданной интенсивностью	По таймауту	По таймауту	С заданной интенсивностью	При выполнении условия	Среднее время нахождения в сост S2	150
7	100	По таймауту	С заданной интенсивностью	С заданной интенсивностью	По таймауту	При выполнении условия	Среднее время нахождения в сост S3	200

8	120	С заданной интенсивностью	По таймауту	По таймауту	С заданной интенсивностью	При выполнении условия	Среднее время нахождения в сост S1	150
9	150	По таймауту	С заданной интенсивностью	С заданной интенсивностью	По таймауту	При выполнении условия	Среднее время нахождения в сост S2	200
10	100	С заданной интенсивностью	При выполнении условия	По таймауту	С заданной интенсивностью	С заданной интенсивностью	Среднее время нахождения в сост S1	150

Таблица 3.3

Номер варианта N	Число пользователей n	Продолжительность пультовой фазы tpAvg (показательный закон, значение и управление)	Продолжительность системной фазы tsAvg (показательный закон, значение и управление)	Длительность эксперимента TimeMax	Показатели эффективности работы СМО	Временные графики вероятностей состояний, показателей эффективности работы СМО	Столбцы базы данных, условие записи значений в базу
1	4		бегунок	60	Среднее время нахождения запроса в системной фазе	Все состояния. Среднее время нахождения запроса в системной фазе	Каждые 2 дня, столбцы: текущий день, r3,r4, среднее время пребывания запроса в системной фазе
2	5	бегунок		40	Среднее число пользователей в пультовой фазе	Все состояния. Среднее число пользователей в пультовой фазе	Каждые 3 дня, столбцы: текущий день, r4,r5, Среднее число пользователей в пультовой фазе
3	4		бегунок	80	Среднее число пользователей в системной фазе	Все состояния. Среднее число пользователей в системной фазе	Каждые 2 дня, столбцы: текущий день, r3,r4, Среднее число пользователей в системной фазе

4	5		бегунок	60	Среднее время нахождения запроса в системной фазе	Все состояния. Среднее время нахождения запроса в системной фазе	Каждые 3 дня, столбцы: текущий день, р4,р5, Среднее время нахождения запроса в системной фазе
5	4	бегунок		40	Среднее число пользователей в пультовой фазе	Все состояния. Среднее число пользователей в пультовой фазе	Каждые 2 дня, столбцы: текущий день, р3,р4, Среднее число пользователей в пультовой фазе
6	5		бегунок	80	Среднее число пользователей в системной фазе	Все состояния. Среднее число пользователей в системной фазе	Каждые 3 дня, столбцы: текущий день, р4,р5, Среднее число пользователей в системной фазе
7	4		бегунок	60	Среднее время нахождения запроса в системной фазе	Все состояния. Среднее время нахождения запроса в системной фазе	Каждые 2 дня, столбцы: текущий день, р3,р4, Среднее число пользователей в системной фазе
8	5	бегунок		40	Среднее число пользователей в пультовой фазе	Все состояния. Среднее число пользователей в пультовой фазе	Средняя длина очереди
9	4		бегунок	80	Среднее число пользователей в системной фазе	Все состояния. Среднее число пользователей в системной фазе	Средний коэффициент использования сервера
10	5		бегунок	60	Среднее время нахождения запроса в системной фазе	Все состояния. Среднее время нахождения запроса в системной фазе	Средняя длина очереди

4. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Задание 1. Моделирование процесса обработки сообщений сервером в Anylogic

Постановка задачи. Сервер обрабатывает запросы, поступающие с автоматизированных рабочих мест с интервалами, распределенными по показательному закону со средним значением времени между прибытиями timeBeween (сек.).

Время обработки сервером одного запроса распределено по показательному закону с заданным средним значением timeServ (или по нормальному закону с математическим ожиданием timeServ).

Сервер имеет входной буфер с заданной ёмкостью bufferCapacity.

Построить имитационную модель и провести модельный эксперимент в течение времени timeMax (модельное время - секунды). Определить характеристики модели (математические ожидания МО случайных величин, вероятности Р событий).

Сделать анимацию процесса перемещений запросов и отображение результатов на диаграммах, гистограммах. Использовать элементы управления для параметров модели. Провести несколько экспериментов с различными значениями управляемых параметров.

В выводах провести анализ полученных результатов.

Пример выполнения задания (Вариант 20)

Исходные данные, используемые в модели, представлены в таблице 4.1.

Таблица 4.1

timeBeween (с заданной интенсивностью, показательный закон)	timeBeween =120
timeServ (значение и управление)	timeServ =100, бегунок [50,400]
bufferCapacity(значение)	bufferCapacity =3
timeMax	timeMax =36000
Характеристики	МО времени обработки запроса и Р обработки запроса
Анимация (разные формы линии, обозначения сервера, картинки для запроса)	ломаная, прямоугольник, конверт
Диаграммы	средняя длина очереди и средний коэффициент загрузки сервера
Гистограммы плотности распределения	МО времени обработки запроса и Р обработки запроса

Создадим проект. Дадим имя модели в соответствии с заданием и вариантом Task1Server_20. Используем на палитре инструментов вкладки: Библиотека моделирования процессов, Агент, Статистика, Презентация, Элементы управления, Разметка пространства и др.

Более подробно описание создания аналогичного проекта приведено в [4].

В окне проекта (рисунок 4.1.1-4.1.2) показано дерево проекта, содержащее все компоненты модели. Модель содержит созданный пользователем тип агента Inquiry (запрос), тип агента MainAll, который создается в модели AnyLogic автоматически.

Тип агента Inquiry создается, если кликнуть правой кнопкой мыши по имени проекта Task1Server_20 и в всплывающем окне выбрать Создать Тип Агента.

В графическое окно MainAll объекты добавлены перетаскиванием из палитры инструментов. Для каждого объекта в окне свойства настроены свойства (имя, тип, значение и др.)

В графическом окне MainAll (рисунок 4.1.1) отображены:

– Диаграмма процессов, описывающая логику процесса обработки запросов сервером (блоки из Библиотеки моделирования процессов).

- Сеть, содержащую ломаную и прямоугольник (сервер) для анимации движения запросов (в палитре вкладка Разметка пространства, Презентация).
- Параметры timeBeween, timeServ, bufferCapacity (в палитре вкладка Агент).
- Бегунок, связанный с параметром timeServ (в палитре вкладка Элементы управления).
- Столбиковые диаграммы для отображения средней длины очереди и среднего коэффициента загрузки сервера (в палитре вкладка Статистика).
- Данные гистограммы: время_обработки_запроса, вероятность_обработки_запросов (в палитре вкладка Статистика).
- Гистограммы: Плотность распределения время_обработки_запроса, Плотность распределения: вероятность_обработки_запросов (в палитре вкладка Статистика).

При настройке свойств блоков диаграммы процесса следует использовать законы распределения показательный, треугольный, нормальный и др. Эти законы можно выбирать из списка в окне, если на панели инструментов Anylogic нажать кнопку  - Выберите распределение вероятностей.

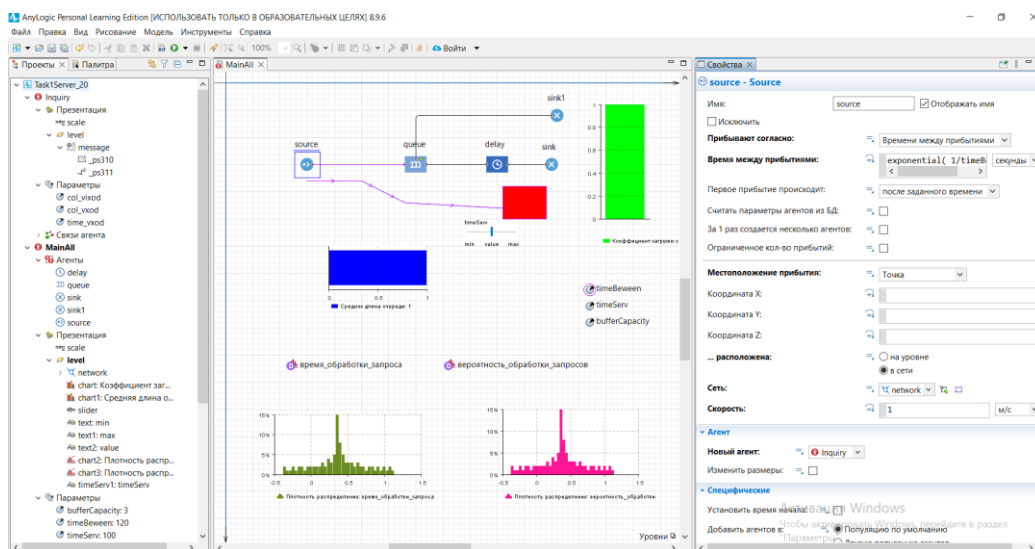


Рисунок 4.1.1 - Основные окна модели

При настройке свойств блоков диаграммы процесса нужно выбирать агента Inquiry из предложенного списка.

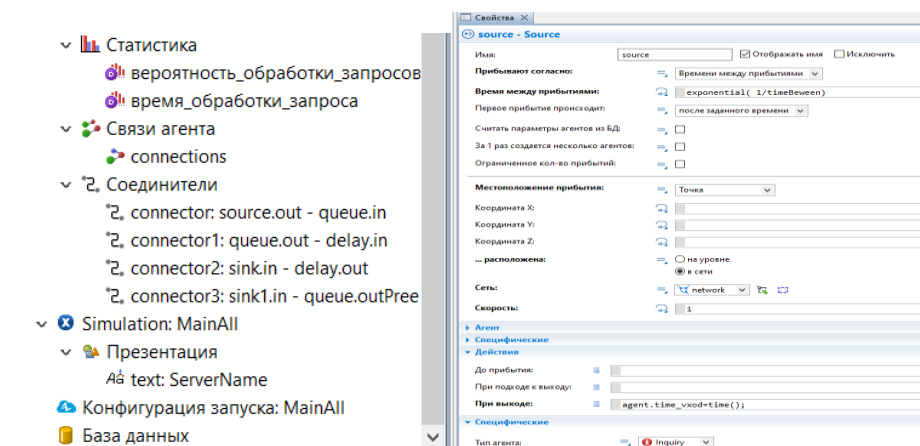


Рисунок 4.1.2 - Продолжение дерева проекта и окно свойств блока source

При задании времени между прибытиями в формуле показательного закона в качестве аргумента следует указывать интенсивность прибытия, которая является обратной величиной к времени между прибытиями, т.е. $\text{exponential}(1/\text{timeBetween})$

Логiku работы буфера и сервера моделируют блоки queue, delay.

The image contains two screenshots of software property windows. The top window is titled 'queue - Queue' and contains the following fields: 'Имя:' (Name) with value 'queue' and a checked 'Отображать' (Show) checkbox; 'Вместимость:' (Capacity) with a value of 'bufferCapacity'; 'Максимальная вместимость:' (Maximum capacity) with an empty field; 'Место агентов:' (Agent location) with a value of 'path' and a dropdown menu; 'Тип агента:' (Agent type) with a value of 'Inquiry' and a dropdown menu; and radio buttons for 'Одиночный агент' (Single agent) and 'Популяция агентов' (Agent population). The bottom window is titled 'delay - Delay' and contains the following fields: 'Имя:' (Name) with value 'delay' and a checked 'Отображать имя' (Show name) checkbox; 'Тип задержки:' (Delay type) with radio buttons for 'Определенное время' (Fixed time) and 'До вызова функции stopDelay()' (Until stopDelay() function call); 'Время задержки:' (Delay time) with a value of 'timeServ' and a unit dropdown set to 'сек' (sec); 'Вместимость:' (Capacity) with a value of '1'; 'Максимальная вместимость:' (Maximum capacity) with an empty field; 'Место агентов:' (Agent location) with a value of 'node' and a dropdown menu; and radio buttons for 'Одиночный агент' (Single agent) and 'Популяция агентов' (Agent population).

Рисунок 4.1.3 - Окна свойств блоков queue, delay

The image shows a screenshot of a software property window titled 'slider - Бегунок'. It contains the following fields: 'Имя:' (Name) with value 'slider' and an unchecked 'Исключить' (Exclude) checkbox; 'Отображается на верхнем агенте' (Shown on top agent) with a checked checkbox and 'Блокировать' (Block) with an unchecked checkbox; 'Ориентация:' (Orientation) with radio buttons for 'Вертикальная' (Vertical) and 'Горизонтальная' (Horizontal), with 'Горизонтальная' selected; 'Связать с:' (Link to:) with a checked checkbox and a value of 'timeServ' and a dropdown menu; 'Минимальное значение:' (Minimum value) with a value of '50'; 'Максимальное значение:' (Maximum value) with a value of '400'; 'Шаг:' (Step) with an empty field; and 'Доступность:' (Availability) with a checked radio button for 'да' (yes).

Рисунок 4.1.4. Окно свойств бегунка

У каждого блока Queue, Delay есть встроенные наборы данных statsSize, Utilization. Функция mean() возвращает среднее из всех измеренных этим набором данных значений. С помощью этой функции создаются столбиковые диаграммы.

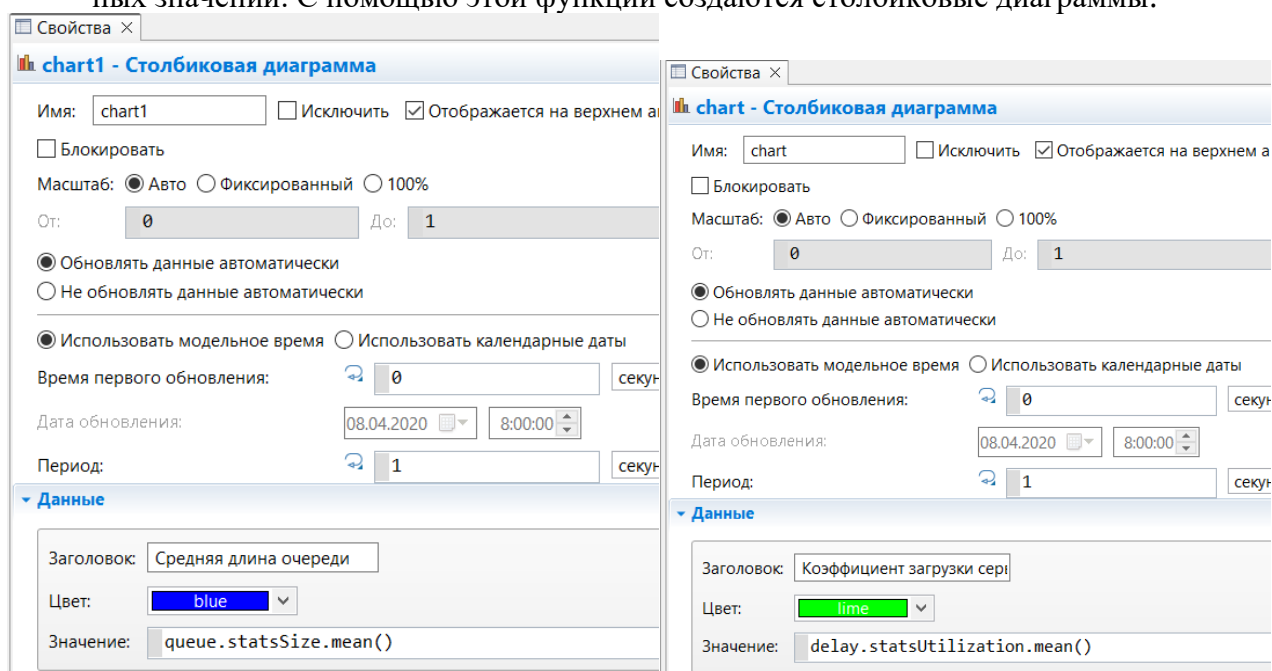


Рисунок 4.1.5 - Окно свойств столбиковых диаграмм для средней длины очереди и коэффициента загрузки сервера.

Агенту Inquiry соответствует, графическое окно, показанное на рис. 4.1.6, в котором содержится прорисованное изображение запроса в виде конверта (объекты из палитры Презентация, Картинки), параметры агента time_vxod, col_vxod, col_vixod. Эти параметры доступны в MainAll как поля agent и используются в коде, прописанном в разделе Действия в блоках источника source и стока sink (рисунки 4.1.7-4.1.8).

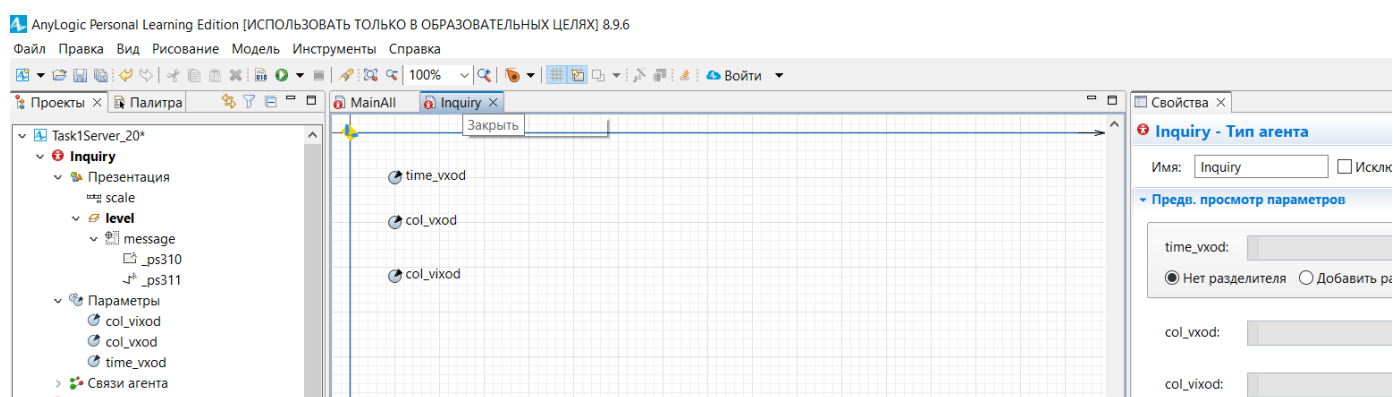


Рисунок 4.1.6 - Графическое окно для Агента Inquiry

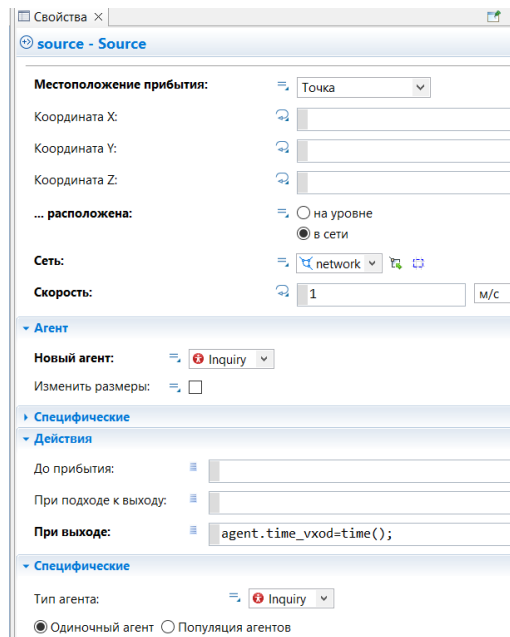


Рисунок 4.1.7. Окно свойств source

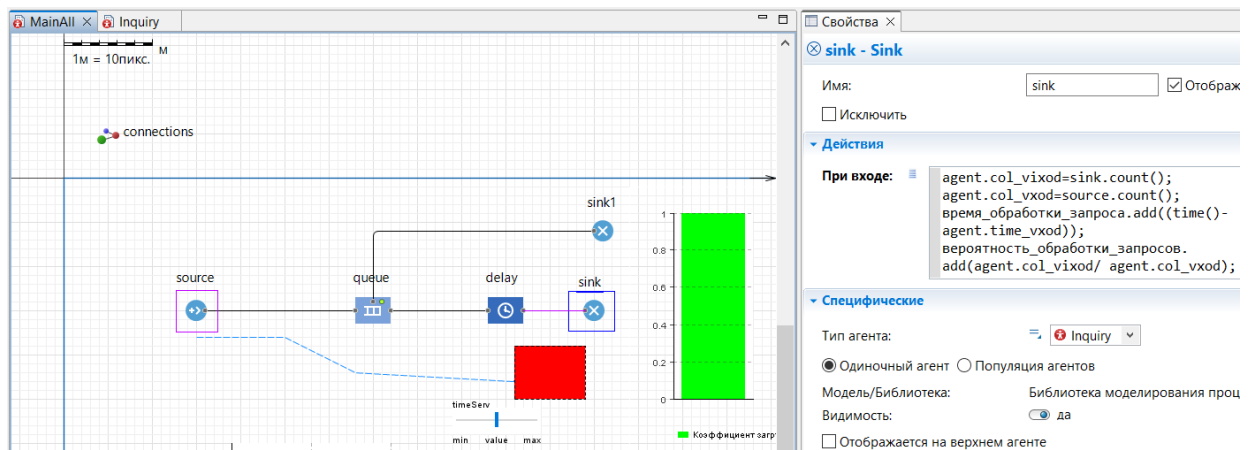


Рисунок 4.1.8. Окно свойств sink

Для проведения модельного эксперимента настроим свойства Простого эксперимента Simulation (рис. 4.1.9).

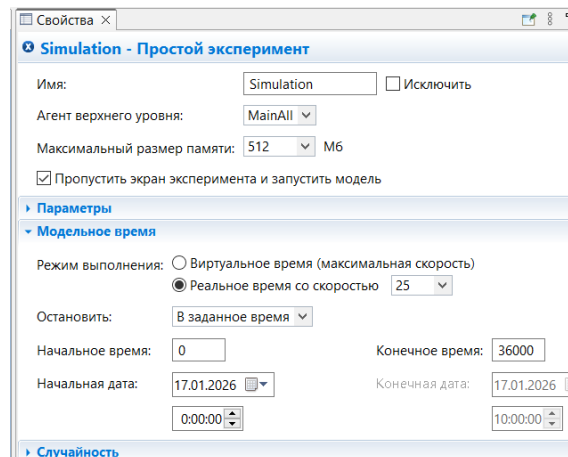


Рисунок 4.1.9

В результате прогона получены следующие результаты (рис. 4.1.10).

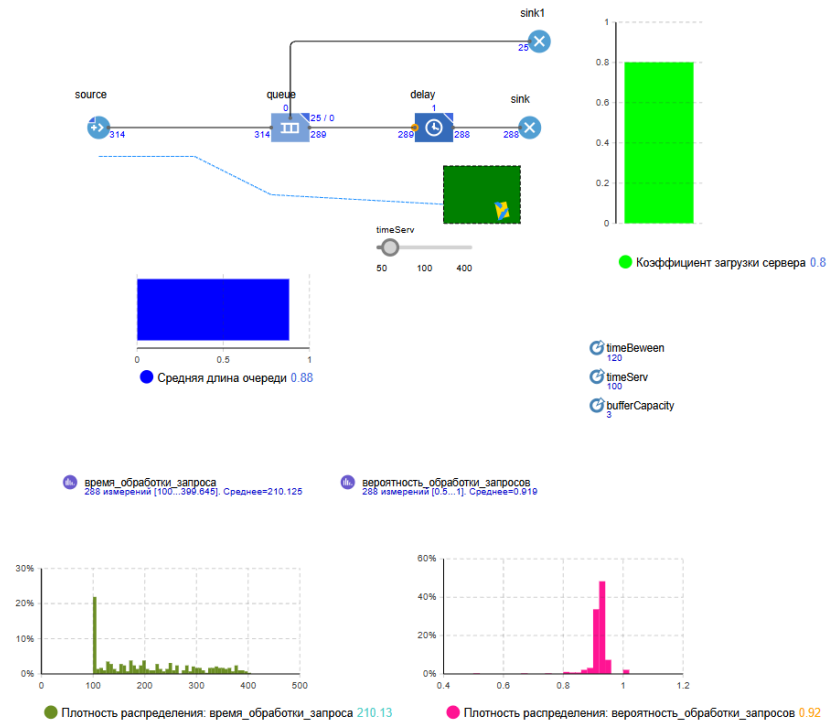


Рисунок 4.1.10

ВЫВОДЫ (анализ полученных результатов)

Задание 2. Моделирование процесса обслуживания объектов системы в Anylogic

Постановка задачи.

Система состоит из n однотипных объектов, каждый из которых может находиться в трех состояниях:

- S1- соответствует штатной работе объекта,
- S2 -тестирование агрегатов объекта,
- S3 - сервисное обслуживание, когда объект не работает штатно.

В начальный момент все объекты системы находятся в состоянии S1. Переход из состояния S1 в S2 определяется параметром λ_{12} . Известно, что в результате тестирования некоторые объекты оказываются исправными и возвращаются в состояние S1 (переход λ_{21}). Если объект в результате тестирования показал неисправности, то он переходит в состояние обслуживания S3 (переход λ_{23}). После устранения неисправностей объект возвращается в рабочее состояние S1 (переход λ_{31}). Возможен также экстренный переход из состояния S1 в состояние S3 (переход λ_{13}).

Создать агентную имитационную модель и провести модельный эксперимент в течение времени timeMax (дн). Настроить анимацию, отображающую все объекты (выбрать для объектов картинку) в заданном пространстве, в которой цвет объекта будет меняться при изменении состояния.

Построить диаграмму с накоплением, отражающую количество объектов, находящихся в состояниях S1, S2, S3. Определить среднее время нахождения в каком-либо состоянии.

В выводах провести анализ полученных результатов.

Исходные данные по вариантам приведены в таблице 3.2. Недостающие значения параметров задать самостоятельно. Для каждого перехода λ_{ij} самостоятельно выбрать закон

распределения. Использовать различные распределения (показательный, нормальный, треугольный законы) для разных переходов.

Пример выполнения задания (Вариант 20)

Исходные данные и обозначения параметров представлены в таблице 4.2.

Таблица 4.2

Количество объектов системы n	100
λ_{12} (с заданной интенсивностью, показательный закон) $\lambda_{12} = 1 / \text{timeBeforeTest}$	timeBeforeTest=10
λ_{21} (по таймауту, треугольный закон и при выполнении условия)	timeTest=0,3 randomTrue(P), P=0,1
λ_{23} (по таймауту, треугольный закон)	timeTest=0,2
λ_{31} (по таймауту, нормальный закон)	intensityToWork=2
λ_{13} (при получении сообщения, с заданной интенсивностью, показательный закон)	intensityToExtrServ=0,05 "extrServ"
Характеристики	Среднее время нахождения в состоянии тестирования S2
Анимация (картинка объекта, изменение цвета при смене состояния)	factory, цвета лавандовый, желтый, зеленый
Время прогона	timeMax=100

Создадим проект (дадим имя модели в соответствии с заданием и вариантом Task2Server_20), используя на палитре инструментов вкладки: Диаграмма состояний, Агент, Статистика, Презентация и др.

Более подробно описание создания аналогичного проекта приведено в [3].

В окне проекта (рисунок 4.2.1) показано дерево проекта, содержащее все компоненты модели. Модель содержит созданный пользователем тип агента SistemObject и популяцию этих агентов objects. Популяция objects создается путем перетаскивания из палитры объекта Agent в графическую область Main, далее появляется окно Мастера для настройки свойств популяции или агента.

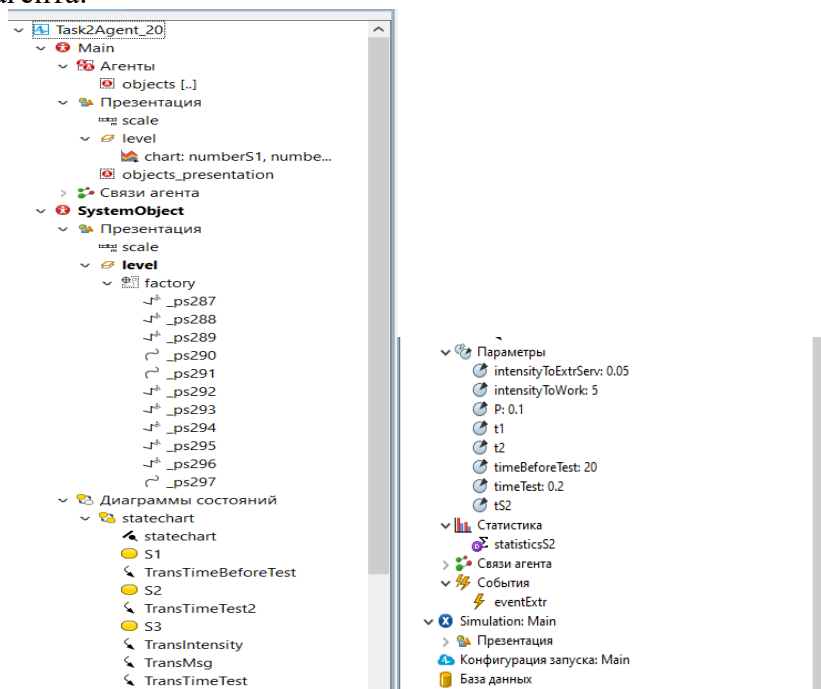


Рисунок 4.2.1

Графическое окно Main (рисунок 4.2.2) содержит популяцию objects и временную диаграмму с накоплением для отображения числа объектов, находящихся в состояниях S1,S2,S3.

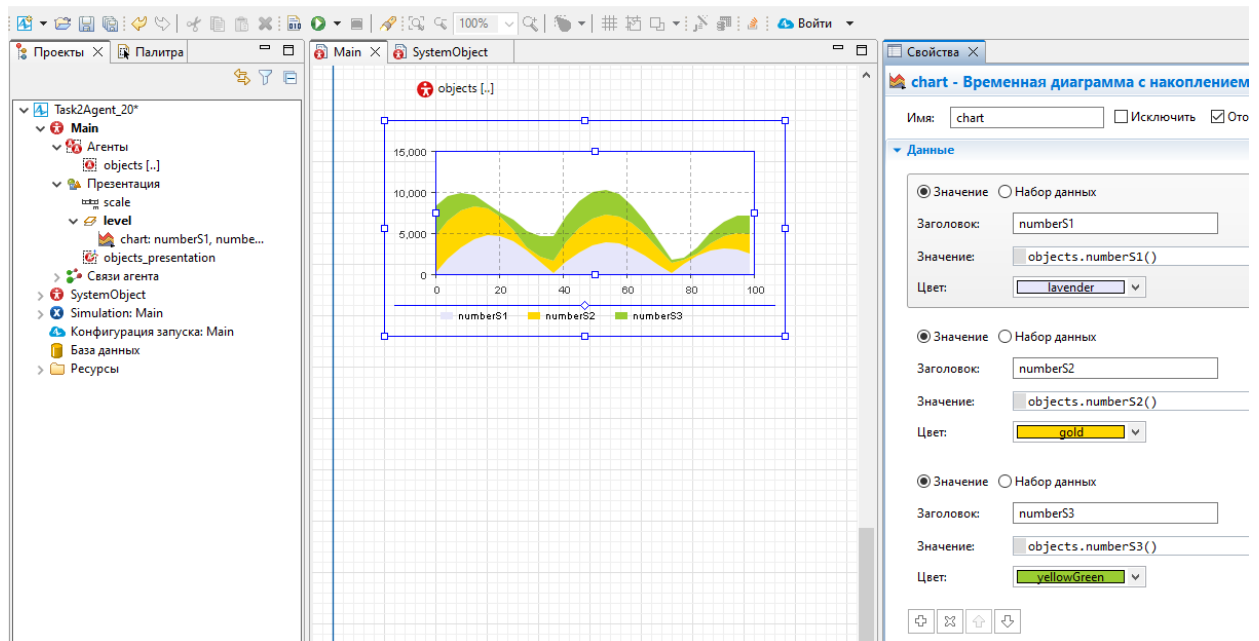


Рисунок 4.2.2

The screenshot shows the 'objects - SystemObject' properties window. It contains several parameters and sections. The 'Статистика' section shows three data series: numberS2, numberS3, and numberS1, each with a type (Кол-во, Сумма, Среднее, Мин., Макс.) and a condition (agent.inState(SystemObject.S2), agent.inState(SystemObject.S3), agent.inState(SystemObject.S1)).

Рисунок 4.2.3

В свойствах популяции (рисунок 4.2.3) есть встроенная Статистика. Для подсчета количества объектов, находящихся в состоянии S1, в Статистике с именем numberS1 вводится условие:

`agent.inState(SystemObject.S1).`

Аналогично по другим состояниям. Эти данные используются в поле Значение окна свойств временной диаграммы (рисунок 4.2.2).

Графическое окно SistemObject (рисунок 4.2.4) содержит:

- диаграмму состояний,
- параметры агента,
- событие eventExtr для рассылки сообщений (имитация экстренного обслуживания),
- статистику statisticsS2 для определения среднего времени нахождения в состоянии S2.

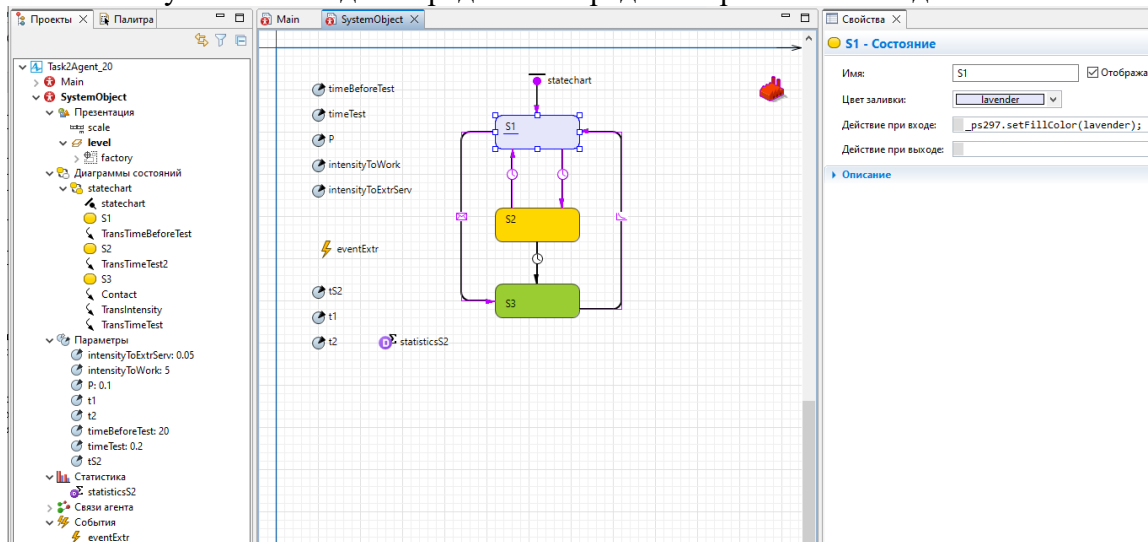
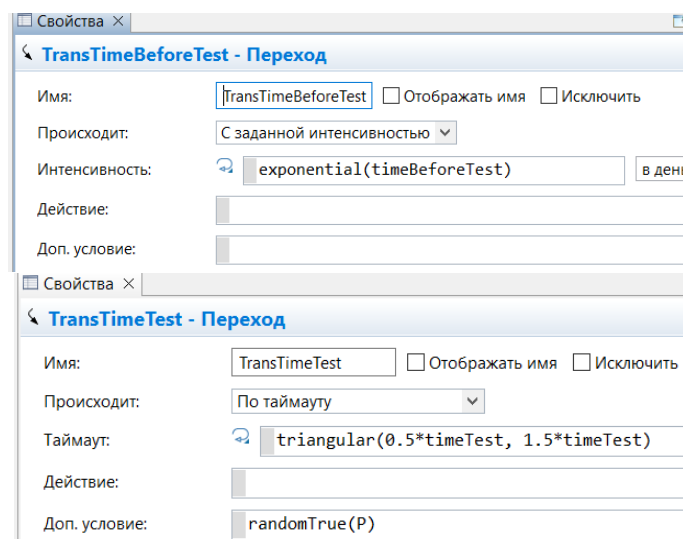


Рисунок 4.2.4

На рисунке 4.2.5 показаны окна свойств переходов TransTimeBeforeTest и др., которые настроены в соответствии с постановкой задачи и принятыми распределениями. В переходах используется размерность – дни. Заметим, что распределение `exponential(timeBeforeTest)` возвращает случайное значение $1 / \text{timeBeforeTest}$, являющиеся интенсивностью перехода.



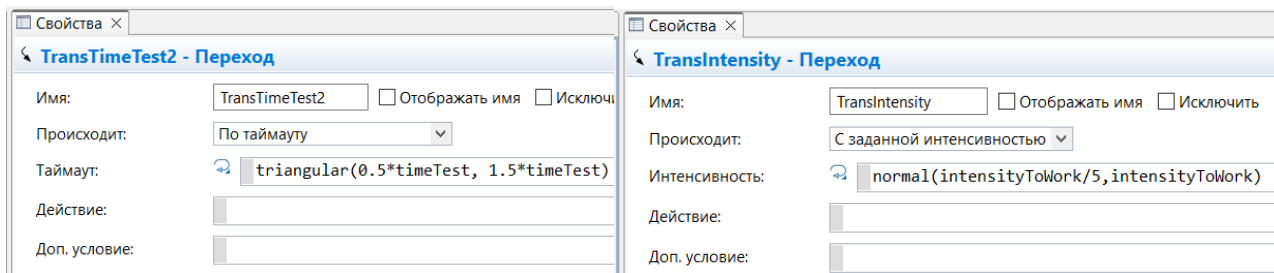


Рисунок 4.2.5

Чтобы имитировать экстренные поломки, используется событие eventExtr, которое с заданной интенсивностью (показательный закон распределения) случайно посылает сообщения. Переход TransMsg срабатывает при получении сообщения. Окна свойств этих объектов показаны на рисунке 4.2.6.

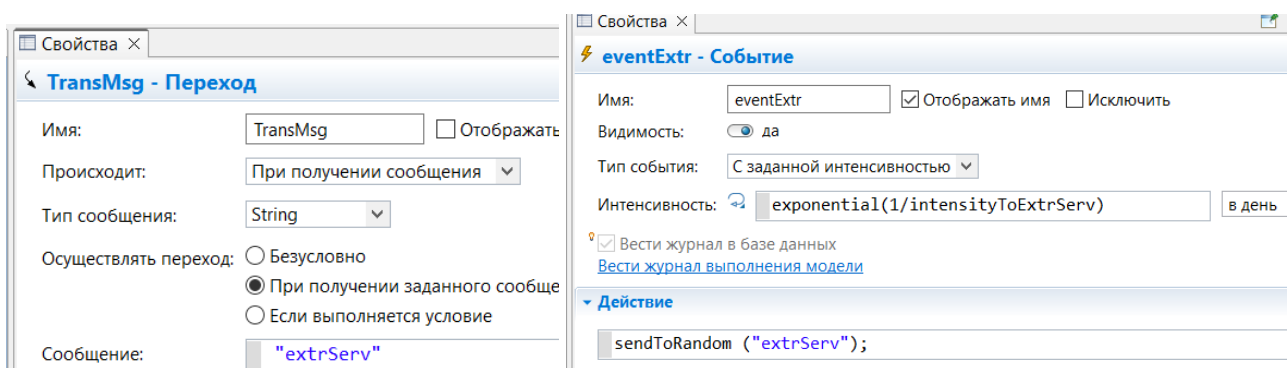


Рисунок 4.2.6

Для нахождения среднего времени пребывания в состоянии S2 собирается статистика statisticsS2 для переменной tS2. Значения этой переменной вычисляются в поле Действия при входе и выходе состояния S2 (Рисунок 4.2.7).

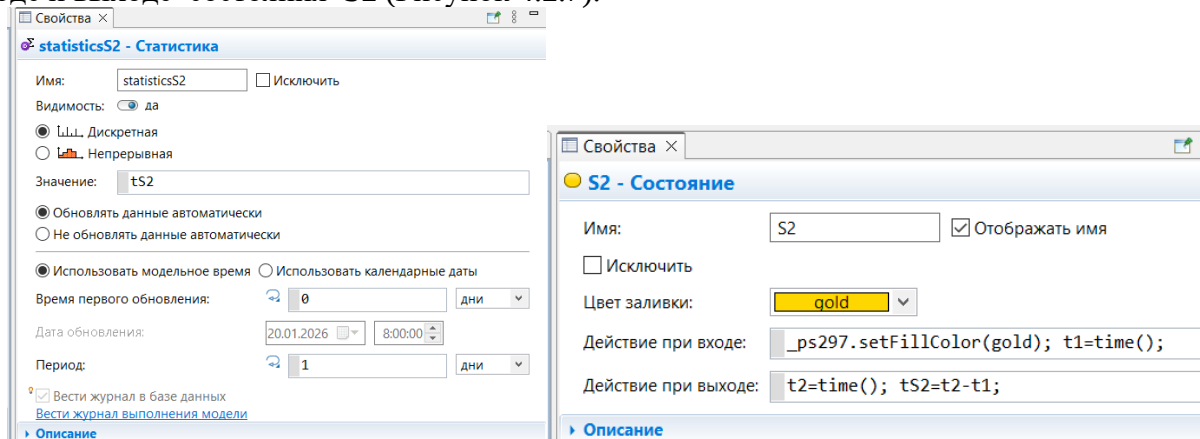


Рисунок 4.2.7

В коде для состояния S2 присутствует оператор `_ps297.setFillColor(gold)`. Аналогичные операторы следует записать в поле Действия при входе в свойствах состояний S1, S3

```
_ps297.setFillColor(lavender);
_ps297.setFillColor(yellowGreen);
```

Поясним их назначение. Для настройки анимации добавлена картинка завод (factory) для отображения объектов. Эта картинка получается путем группировки линий. Линия, которая обрисовывает объект имеет имя `_ps297`, которое можно увидеть в дереве проекта



(рисунок 4.2.1) . В результате действий указанных операторов при входе в состояние цвет меняется на лавандовый, желтый или зеленый.

Настройка окна простого эксперимента представлена на рисунке 4.2.8.

Рисунок 4.2.8

В результате прогона получены следующие результаты (рисунок 4.2.9).

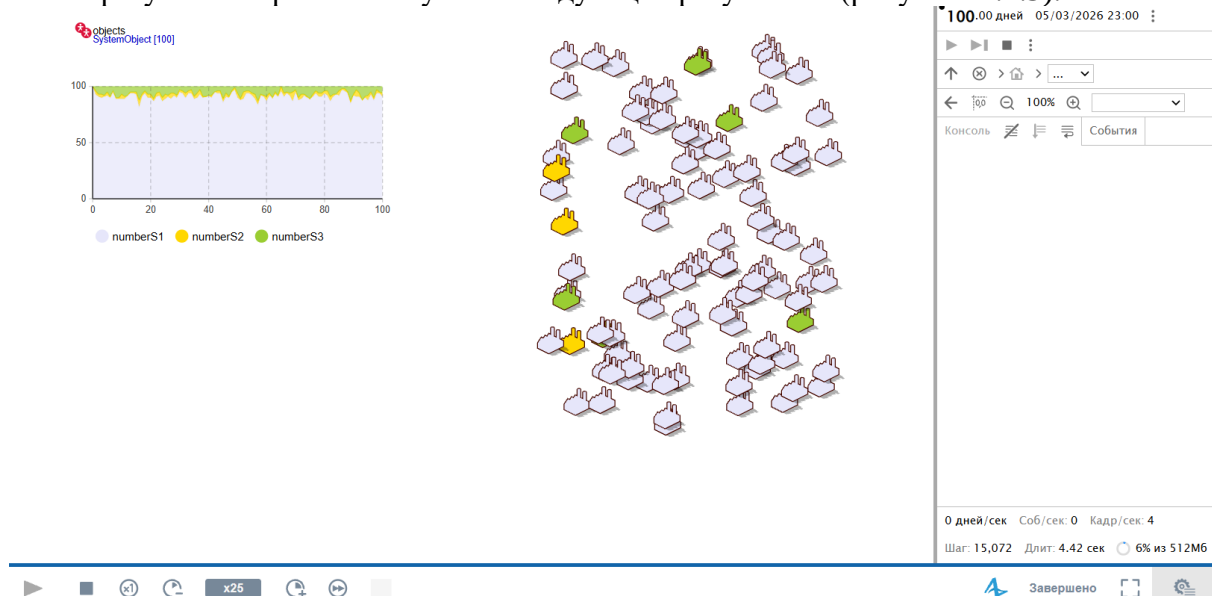


Рисунок 4.2.9

Чтобы увидеть результаты сбора статистики нужно открыть панель разработчика и переключиться в окно objects (рисунок 4.2.10). Среднее время нахождения в S2 равно 0,2

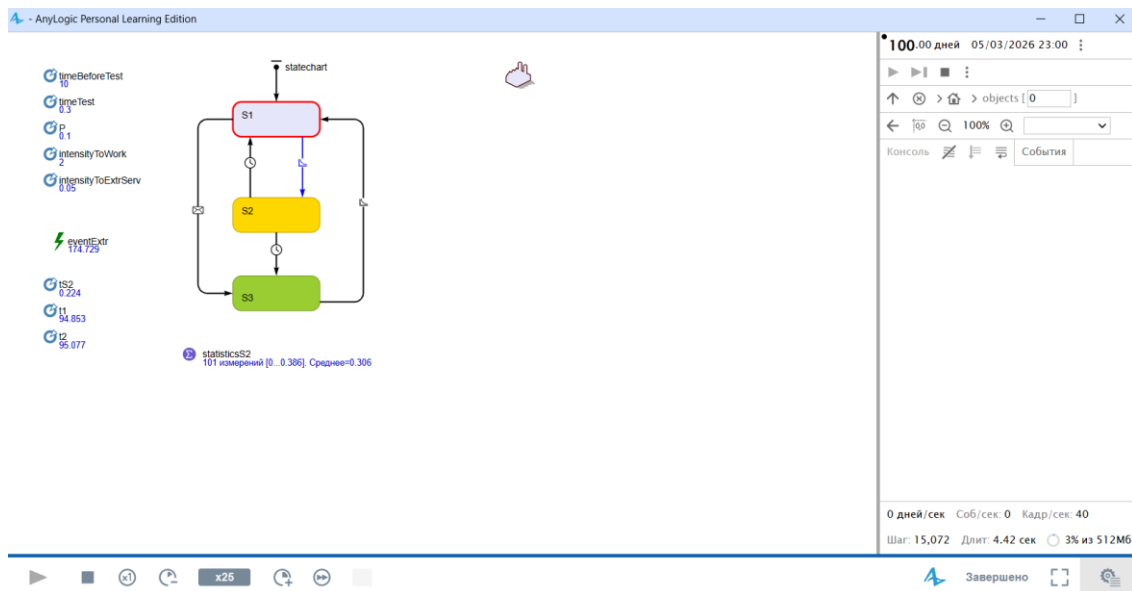


Рисунок 4.2.10

ВЫВОДЫ (анализ полученных результатов)

Задание 3. Моделирование функционирования информационной системы «файл-сервер»

Постановка задачи.

Создать модель информационной системы сети типа «файл-сервер», отождествляя функционирование системы как системы массового обслуживания (СМО), т.е. марковского процесса с непрерывным временем и n дискретными состояниями:

состояние S_0 – все пользователи в пультовой фазе, p_0 – вероятность этого состояния;

состояние S_i – i пользователей в системной фазе ($i=1,2,\dots,n$), $(n-i)$ – в пультовой фазе, p_i – вероятность этого состояния.

Работа этой системы описывается системой дифференциальных уравнений Эрланга-Колмогорова относительно вероятностей состояний

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} p_j(t) \lambda_{ji} - p_i(t) \sum_{j \neq i} \lambda_{ij}$$

Процесс взаимодействия пользователей и системы складывается из повторяющихся системной и пультовой фаз. Продолжительности пультовой и системной фаз являются случайными величинами, распределенными по показательному закону с математическими ожиданиями $trAvg$, $tsAvg$. Количество пользователей – n . Интенсивности переходов λ_{ij} выражаются через $trAvg$, $tsAvg$.

Создать имитационную модель как модель системной динамики, провести модельный эксперимент в течение времени $timeMax$ (мин). Использовать элементы управления для параметров.

Определить вероятности различных состояний системы (отобразить их на графиках).

Показатели эффективности работы СМО (среднее количество запросов в системной фазе и среднее время нахождения запроса в системной фазе и др.).

Создать базу данных для хранения значений показателей эффективности и вероятностей состояний.

Исходные данные по вариантам приведены в таблице 3.3. Недостающие значения параметров задать самостоятельно.

Пример выполнения задания (Вариант 20)

Исходные данные и обозначения параметров представлены в таблице 4.3.

Таблица 4.3

Число пользователей	N=5
Продолжительность пультовой фазы tpAvg (показательный закон, значение и управление)	tpAvg =5, бегунок [0,10]
Продолжительность системной фазы tsAvg (показательный закон, значение)	tsAvg =2
Длительность эксперимента TimeMax	TimeMax =60
Показатели эффективности работы СМО	Среднее время нахождения запроса в системной фазе
Временные графики вероятностей состояний, показателей эффективности работы СМО	Все состояния, среднее время нахождения запроса в системной фазе
Столбцы базы данных, условие записи значений в базу	Каждый день, столбцы: текущий день, p5, среднее время пребывания запроса в системной фазе

Создадим проект (дадим имя модели в соответствии с заданием и вариантом Task3Server_20), используя на палитре инструментов вкладки: Системная динамика, Агент, Статистика, Презентация и др.

Более подробно описание создания аналогичного проекта приведено в [3].

В окне проекта (рисунок 4.3.1) показано дерево проекта, содержащее все компоненты модели.

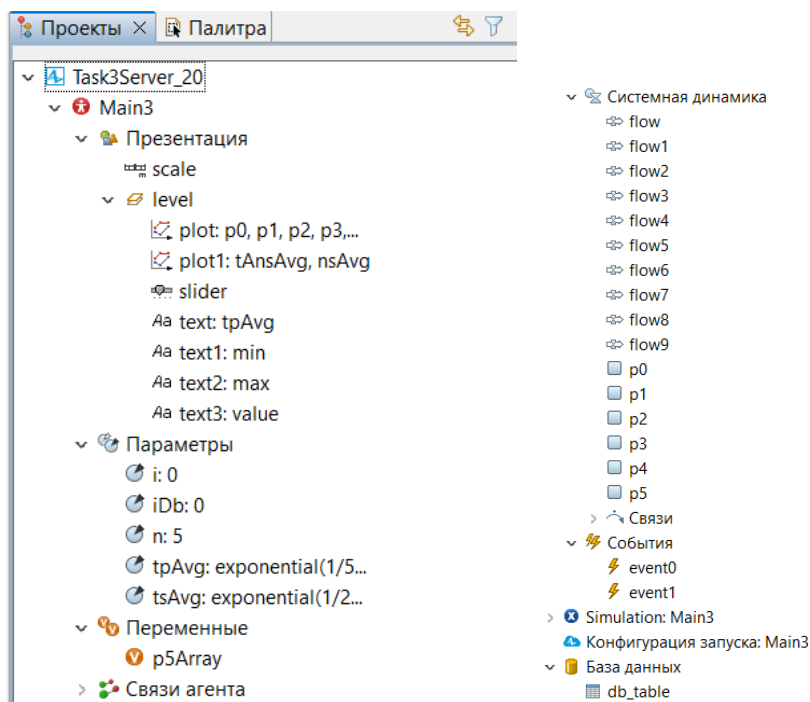


Рисунок 4.3.1

Графическое окно Main3 (рисунок 4.3.2) содержит диаграмму потоков и накопителей (блоки перетаскиваются в графическую область из вкладки Системная динамика), временные графики, переменные, параметры, события модели.

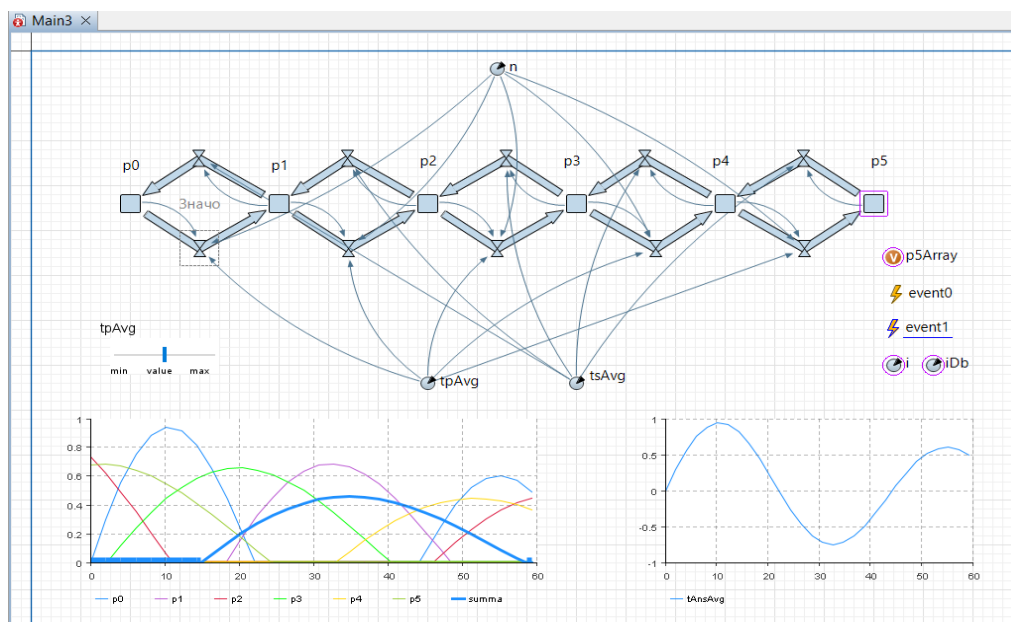


Рисунок 4.3.2

Потоковая диаграмма соответствует решению системы дифференциальных уравнения Эрланга-Колмлгорова относительно вероятностей и содержит

- 6 накопителей (квадраты), соответствующих состояниям системы, значения которых - соответствующие вероятности;
- потоки(стрелки), соответствующие скоростям изменения вероятностей.

Дифференциальные уравнения к классической (нормальной) форме создаются для накопителя автоматически в зависимости от входных и выходных потоков. На рисунке 4.3.3 показаны окна свойств накопителей p0, p1. Начальное значение равно 1 только для p0, для остальных 0. Это связано с тем, что в начальный момент вероятность отсутствия пользователей системной фазе равна 1, а вероятность нахождения хоть одного пользователя в системной фазе равна 0. Сумма всех вероятностей должна быть в любой момент эксперимента равна 1.

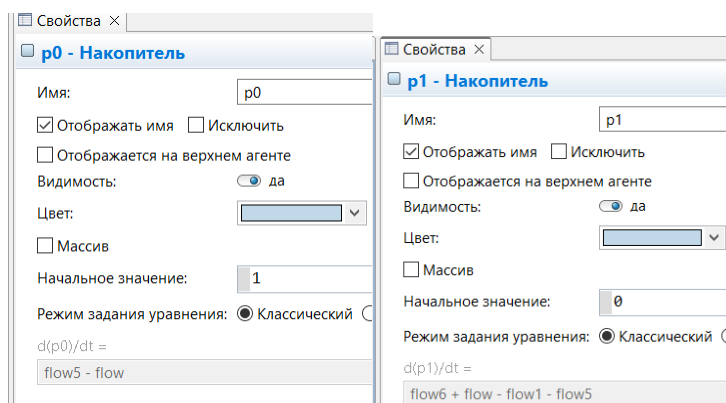


Рисунок 4.3.3

Формулы потоков определяются правилом составления уравнений Эрланга-Колмогорова.

Правило составления уравнений. В каждом уравнении системы в правой его части стоит **со знаком минус** вероятность данного состояния p_i , умноженная на суммарную интенсивность всех потоков, **ведущих из данного состояния**, а **со знаком плюс** стоит сумма произведений интенсивностей всех потоков, **входящих в i -е состояние**, на вероятности тех состояний, из которых эти потоки исходят.

Интенсивности входных потоков в диаграмме слева направо уменьшаются, так как уменьшается число пользователей, переходящих из пултовой в системную фазу.

Интенсивности входных потоков в диаграмме справа налево остаются постоянными, так как при работе сервера предполагается, что нет параллельного решения нескольких задач, время сервера распределяется равномерно между всеми задачами, поэтому время решение отдельной задачи увеличивается пропорционально числу пользователей, находящихся в системной фазе.

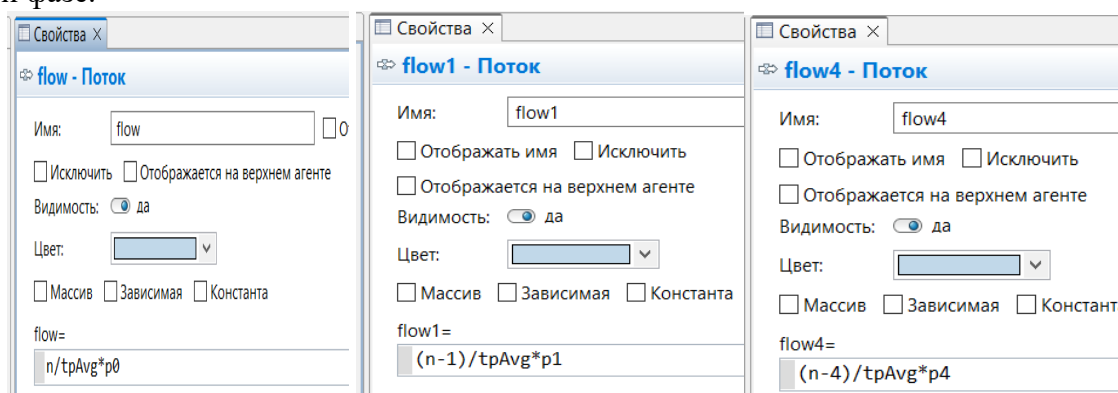


Рисунок 4.3.4 - Потоки $S_0 \rightarrow S_1$, $S_1 \rightarrow S_2$, $S_5 \rightarrow S_5$

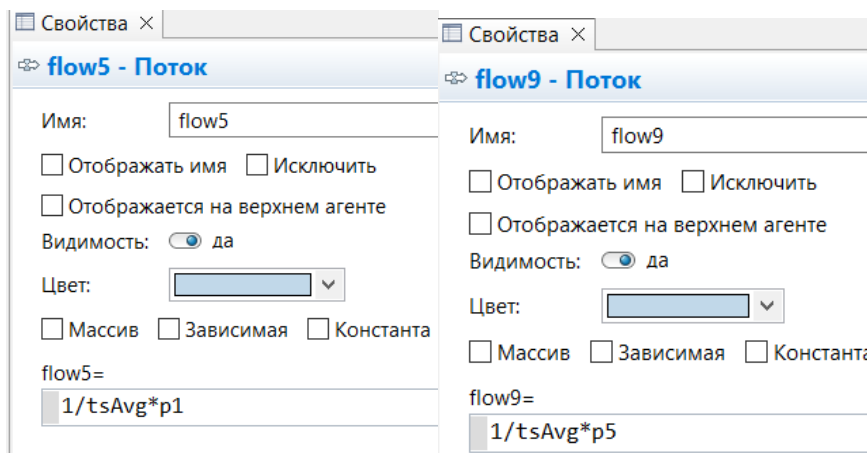


Рисунок 4.3.5 - Потоки $S_1 \rightarrow S_0$, $S_5 \rightarrow S_4$

Все параметры, переменные используемые в потоках, становятся доступными, если явно указаны связи (на рисунке 4.3.2 это тонкие линии со стрелками). Все необходимые связи добавляются, если выделить с помощью левой кнопки мыши всю потоковую диаграмму (она оказывается в пунктирном прямоугольнике), затем правой кнопкой выбрать Исправить ошибки в связях.

Для создания временных графиков настроим их свойства (Рисунок 4.3.6). Для контроля равенства 1 суммы вероятностей добавлен график суммы.

Чтобы рассчитать среднее время нахождения запроса в системной фазе используем математическое ожидание случайной величины времени нахождения запроса в системной фазе. Значения этой случайной величины зависят от состояний и могут быть 0, ts_{Avg} , $2*ts_{Avg}$, $3*ts_{Avg}$, $4*ts_{Avg}$, $5*ts_{Avg}$ соответственно в состояниях $s_0, S_1, \dots, S_5$. Математическое ожидание получается путем умножения возможных значений на вероятности, т.е. $0*p_0 + ts_{Avg}*p_1 + 2*ts_{Avg}*p_2 + 3*ts_{Avg}*p_3 + 4*ts_{Avg}*p_4 + 5*ts_{Avg}*p_5$.

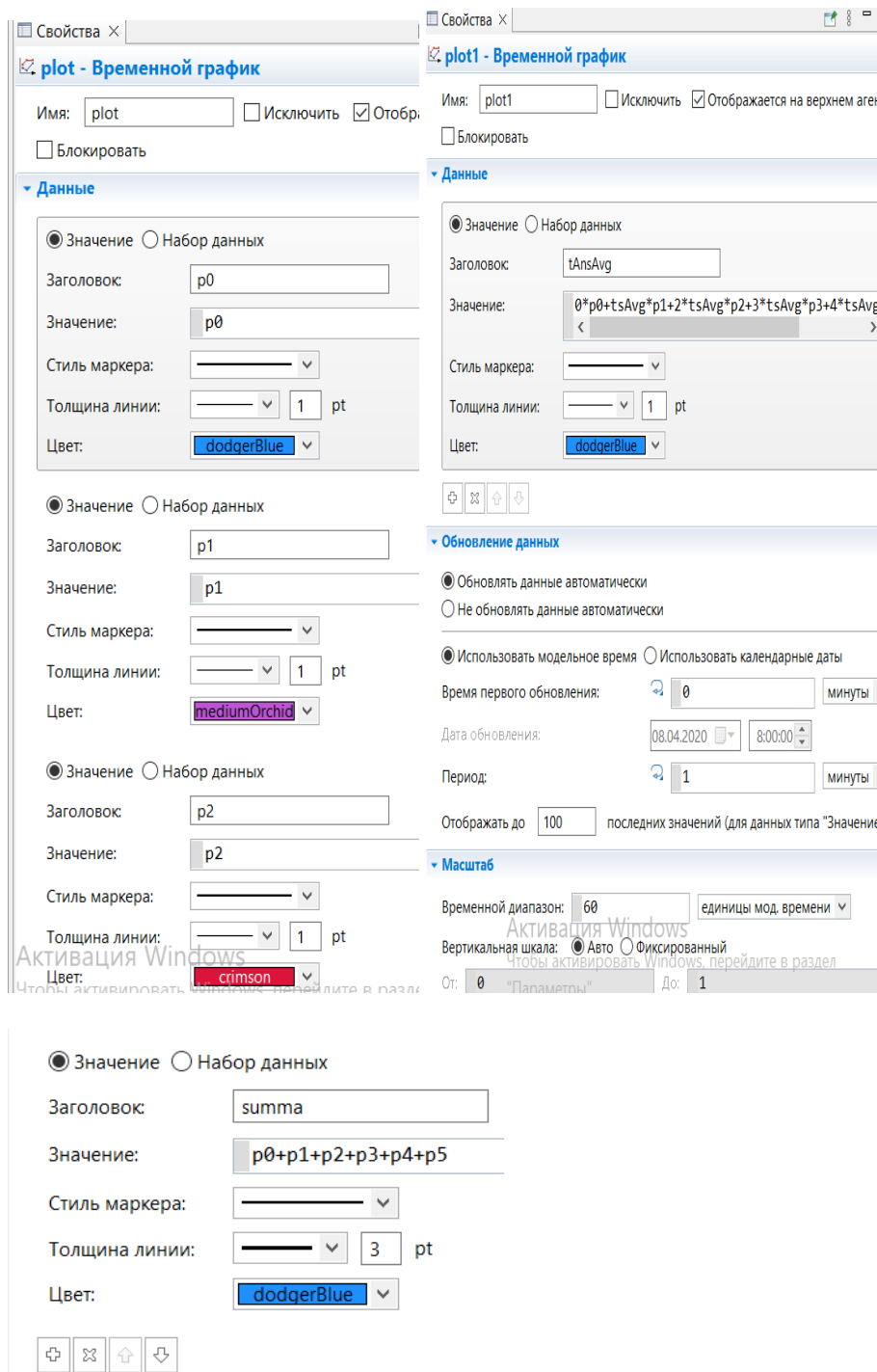


Рисунок 4.3.6

Для создания таблицы базы данных используется события event0 (очистка базы, созданной в предыдущем эксперименте и event1 для записи данных в таблицу по ходу эксперимента (период записи -1 день). Поскольку для марковского случайного процесса вероятности быстро стабилизируются, становятся установившимися, то нет смысла записывать одинаковые значения во время всего эксперимента. Водится дополнительное условие остановки записи данных в базу, если разность двух последующих значений отличается на малую величину, запись прекращается. Для формирования условия вводится переменная -массив p5Array.

На рисунке 4.3.7 показана окна свойств p5Array и таблицы. Значение по умолчанию не заполняется, так как заполнение таблицы будет происходить в коде, прописанном в event1. На рисунке 4.3.7 показанная заполненная таблица, открытая после прогона.

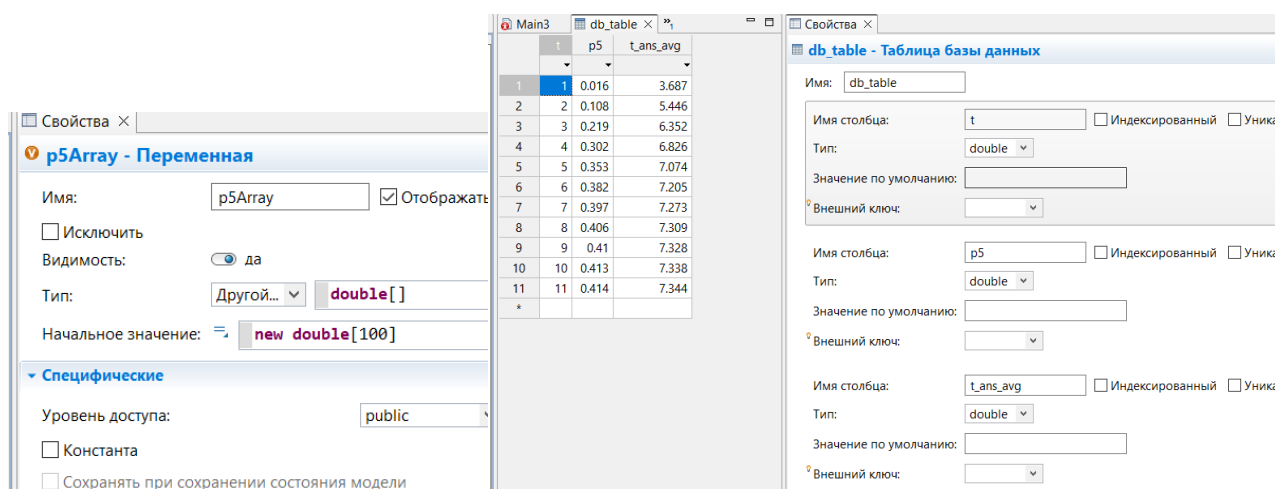


Рисунок 4.3.7

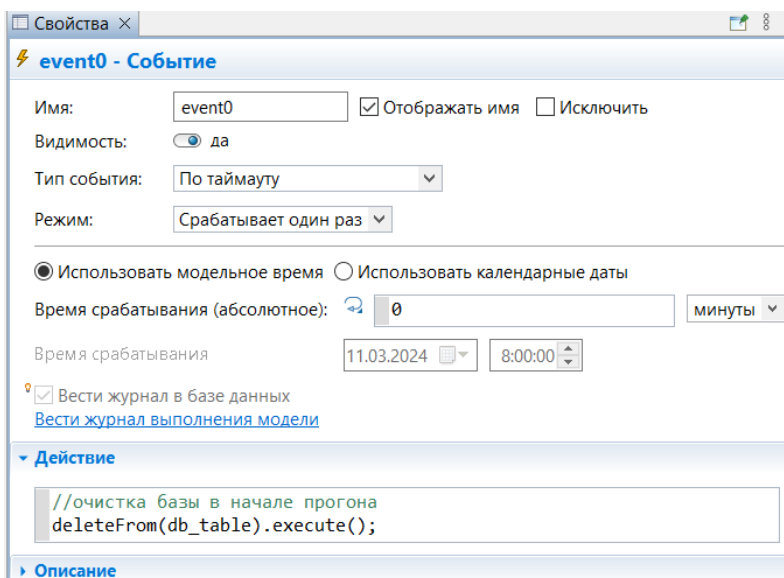


Рисунок 4.3.8

Окно свойств event0 показано на рисунке 4.3.8. Окно свойств event1 показано на рисунке 4.3.9. В коде предусмотрен подсчет записываемых в базу строк (параметр iDb) и

проверка условия, что бы разность двух последующих записей в колонке p5 не была меньше 0,001.

event1 - Событие

Имя: event1 ☒ Отображать имя ☐ Исключить

Видимость: ☒ да

Тип события: По таймауту

Режим: Циклический

☒ Использовать модельное время ☐ Использовать календарные даты

Время первого срабатывания (абс.): 1 минуты

Время срабатывания: 11.03.2024 8:00:00

Период: 1 минуты

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

Действие

```
//код заполнения таблицы базы данных
i++;
p5Array[i]=p5;

if (i==1)
{insertInto(db_table).columns(db_table.t,db_table.p5,db_table.t_ans_avg).
values(time(),p5,0*p0+tsAvg*p1+2*tsAvg*p2+3*tsAvg*p3+4*tsAvg*p4+5*tsAvg*p5).execute(); iDb=i;}

if (i>1 && abs(p5Array[i]-p5Array[i-1])>0.001)
{insertInto(db_table).columns(db_table.t,db_table.p5,db_table.t_ans_avg).
values(time(),p5,0*p0+tsAvg*p1+2*tsAvg*p2+3*tsAvg*p3+4*tsAvg*p4+5*tsAvg*p5).execute(); iDb=i;}
```

Рисунок 4.3.9

Окно настройки свойств простого эксперимента представлено на рисунке 4.3.10. Если изменить генератор случайных чисел, то значения случайных величин tsAvg, tpAvg будут различными в разных прогонах

Simulation - Простой эксперимент

Имя: Simulation ☐ Исключить

Агент верхнего уровня: Main3

Максимальный размер памяти: 512 M6

☒ Пропустить экран эксперимента и запустить модель

Параметры

Модельное время

Режим выполнения: ☐ Виртуальное время (максимальная скорость)
☒ Реальное время со скоростью 5

Остановить: В заданное время

Начальное время: 0 Конечное время: 60

Начальная дата: 25.01.2026 Конечная дата: 25.01.2026

0:00:00 1:00:00

Случайность

Генератор случайных чисел:

☐ Случайное начальное число (уникальные "прогоны")

☒ Фиксированное начальное число (воспроизводимые "прогоны") Начальное число: 1

☐ Нестандартный генератор (подкласс класса Random): new Random()

Окно

Заголовок: Task3Server_20 : Simulation

☒ Разрешить изменение масштаба и перетаскивание

☒ Панель разработчика

Рисунок 4.3.9

Результаты одного прогона представлены на рисунке 4.3.11.

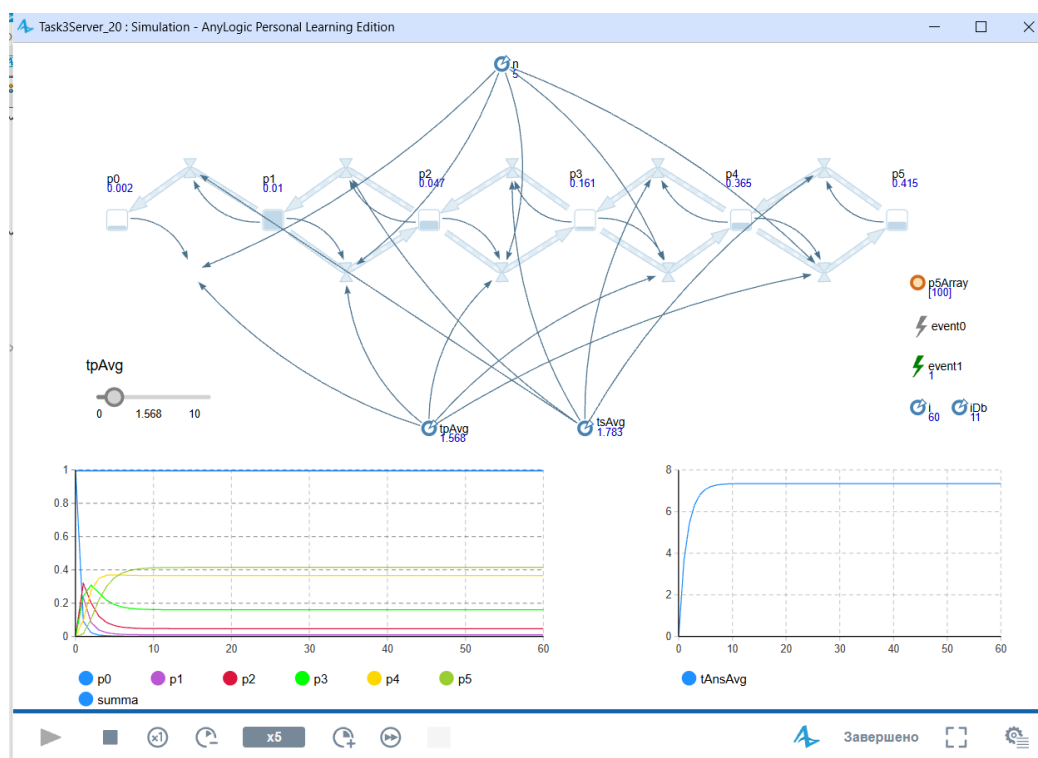


Рисунок 4.3.11

ВЫВОДЫ (анализ полученных результатов)

СПИСОК ИНФОРМАЦИОННЫХ РЕСУРСОВ

1. Онлайн-курс по дисциплине «Модели и методы исследования информационных процессов и систем» [Электронный ресурс].
– URL: <https://do.skif.donstu.ru/course/view.php?id=780>
2. Официальный сайт разработчика AnyLogic [Электронный ресурс].
– URL: <https://www.anylogic.ru/downloads/>
3. Григорьев И. AnyLogic за три дня. Практическое пособие по имитационному моделированию [Электронный ресурс].
– URL: <https://www.anylogic.ru/resources/books/free-simulation-book-and-modeling-tutorials>
4. Боев, В.Д. Компьютерное моделирование в среде AnyLogic [Электронный ресурс] : учебное пособие / В.Д. Боев ; В. Д. Боев. - Москва : Юрайт, 2023. - 298 с.
5. Имитационное моделирование. Разработка имитационных моделей средствами iWebsim и AnyLogic [Электронный ресурс] : учебное пособие для вузов / А.Г. Палей, Г.А. Поллак ; Палей А. Г., Поллак Г. А.; Палей А. Г. - 3-е изд., стер. - Санкт-Петербург : Лань, 2023. - 208 с.
6. Гугл-таблицы с исходными данными [Электронный ресурс]. – URL : <https://docs.google.com/spreadsheets/d/1LrRlk8INFcyM075qJdN2KzFkFVSJj-CAJl8-Z3cLALw/edit?gid=0#gid=0>