

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

МОДЕЛИРОВАНИЕ СИСТЕМ

Методические указания
к выполнению лабораторных работ
по дисциплине
для студентов, обучающихся по направлениям
09.03.02 – «Информационные системы и технологии»,
09.03.03 – «Прикладная информатика»
по учебным планам подготовки бакалавров
дневной и заочной форм обучения

Севастополь
СевГУ
2021

Р е ц е н з е н т:**Е.А. Кожяев** - канд. техн. наук, доцент кафедры ИТиКС*Составители* : А.Е. Безуглая, С.А. Кузнецов

М74 Моделирование систем : методические указания к выполнению лабораторных работ по дисциплине для студентов, обучающихся по направлениям 09.03.02 – «Информационные системы и технологии», 09.03.03 – «Прикладная информатика» по учебным планам подготовки бакалавров дневной и заочной форм обучения / Сост. А.Е. Безуглая, С.А. Кузнецов ; Министерство образования и науки РФ, Севастопольский государственный университет. – Севастополь: СевГУ, 2021. – 56 с. – Текст : электронный.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Моделирование систем». Целью методических указаний является углубление и закрепление полученных теоретических знаний в области математического моделирования; формирование умений и навыков работы со специализированными программными пакетами моделирования; применение полученных знаний для исследования характеристик непрерывных и дискретных систем.

УДК 004.94

Методические указания рассмотрены и утверждены на заседании кафедры «Информационные системы», протокол № 9 от 29 апреля 2020 года, и методическом семинаре института информационных технологий и управления в технических системах, протокол № 5 от 17 июня 2020 г.

В авторской редакции

Изд. № 320/2020. Объем 3,5 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»

© ФГАОУ ВО «Севастопольский
государственный университет», 2021

СОДЕЖАНИЕ

Лабораторная работа № 1. «Исследование способов моделирования систем в рамках непрерывно-детерминированного подхода»	4
Лабораторная работа № 2. «Исследование способов моделирования непрерывно-стохастических систем».....	14
Лабораторная работа № 3. «Исследование технологии дискретно-событийного имитационного моделирования»	21
Лабораторная работа № 4. «Исследование подходов к моделированию вычислительных систем»	30
Лабораторная работа № 5. «Исследование сетевого подхода к моделированию».....	36
Приложение А.....	41
Приложение Б	53

ЛАБОРАТОРНАЯ РАБОТА № 1 . ИССЛЕДОВАНИЕ СПОСОБОВ МОДЕЛИРОВАНИЯ СИСТЕМ В РАМКАХ НЕПРЕРЫВНО-ДЕТЕРМИНИРОВАННОГО ПОДХОДА

1.1 Цель работы

Цели лабораторной работы. Исследование способов построения простейших моделей непрерывных систем с помощью методов аналитического и имитационного моделирования. Изучение технологии системно-динамического имитационного моделирования в среде AnyLogic.

Трудоемкость лабораторной работы: 9 ч (6 ч – аудиторных, 3 ч – самостоятельная работа студента).

1.2 Краткие теоретические сведения

Построение модели простого объекта

Пример 1

Рассмотрим простейший объект - грузик, расположенный на горизонтальной плоскости и прикрепленный пружиной к вертикальной стенке. Между горизонтальной плоскостью и грузиком находится жидкая (или не очень) смазка (рис. 1).

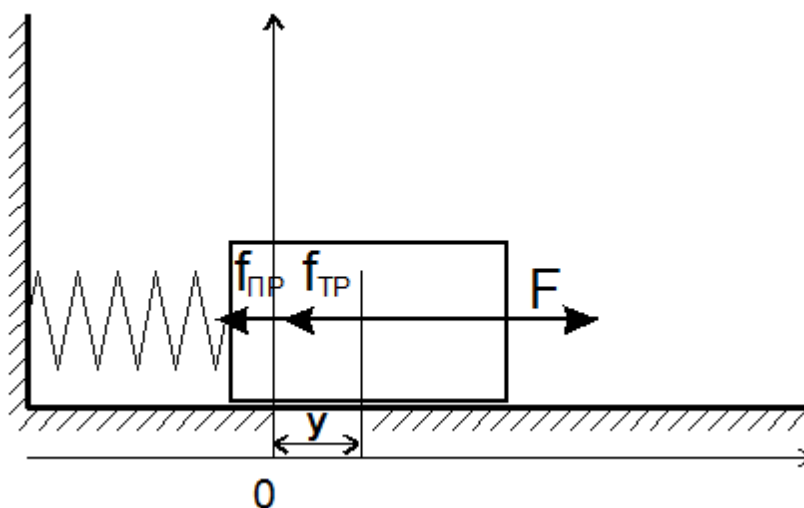


Рисунок 1 – Грузик на пружине

Грузик первоначально находился в положении равновесия (его центр масс совпадал с координатой 0 по оси X). Под действием внешней силы F грузик переместился на величину x . При этом пружина растянулась на эту же величину. В результате на грузик стала действовать возвратная сила пружины $f_{тр}$, направленная в сторону, противоположную растяжению пружины. Кроме того, при движении грузика на него действует сила трения, направленная против направления его движения, и пропорциональная скорости движения. Грузик, как и любое физическое тело подчиняется второму закону Ньютона:

$$ma = \xi \quad (1)$$

где m - масса грузика,
 a - ускорение движения грузика,
 ξ - результирующая сила, действующая на грузик.

Сила, обусловленная пружиной (предполагается, что она работает в пределах упругой деформации), пропорциональна жесткости пружины и деформации, и направлена в сторону, обратную деформации:

$$f_{np} = -cy \quad (2)$$

Сила жидкого трения, действующая на грузик, пропорциональна скорости его движения v и направлена против движения:

$$f_{mp} = -kv \quad (3)$$

Здесь k – коэффициент трения.

Результирующая сила, действующая на грузик, определяется алгебраической суммой всех сил – трения f_{mp} , пружины f_{np} и внешней F .

$$\xi = F - f_{mp} - f_{np} \quad (4)$$

Подставляя в (4) выражения (1) – (3), получаем:

$$ma = F - kv - cy \quad (5)$$

Если учесть, что скорость и ускорение являются соответственно первой и второй производной от смещения

$$v = \dot{y} \equiv \frac{dy}{dt}, \quad a = \ddot{y} \equiv \frac{d^2y}{dt^2}, \quad (6)$$

получаем окончательный вид уравнения движения грузика:

$$m\ddot{y} = F - k\dot{y} - cy \quad (7)$$

или

$$m\ddot{y} + k\dot{y} + cy = F, \quad (8)$$

$$\ddot{y} + \frac{k}{m}\dot{y} + \frac{c}{m}y = \frac{F}{m}. \quad (9)$$

Результат – обыкновенное линейное дифференциальное уравнение второго порядка. Обыкновенное дифференциальное уравнение получилось из-за того, что мы рассматривали абсолютно жесткий объект, который может быть заменен точкой массой m , а в пружине не учитывали конечной скорости распространения волны движения. Линейным уравнение получилось из-за того, что мы рассматриваем поведение пружины в пределах диапазона упругости. При нарушении указанных условий, дифференциальное уравнение может стать нелинейным, могут появиться частные производные и т. д.

Таким образом, мы получили математическую модель объекта, изображенного на рисунке 1, в виде дифференциального уравнения второго порядка. Получив решение этого уравнения, можно получить описание движения этого объекта, зависящее от начальных условий и от внешних воздействий.

При решении дифференциальных уравнений удобнее иметь дело не с уравнениями высокого порядка (выше первого), а с системой уравнений первого порядка. Известно, что дифференциальное уравнение порядка n может быть преобразовано в систему из n уравнений первого порядка. Для преобразования получившегося уравнения второго порядка (9) заменим наблюдаемую переменную y на переменную состояния x_1 :

$$y = x, \dot{y} = V \quad (10)$$

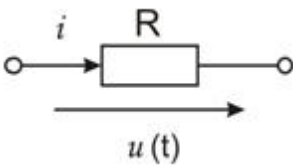
Тогда уравнение (9) принимает следующий вид системы уравнений 1-го порядка:

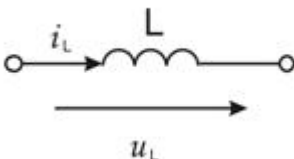
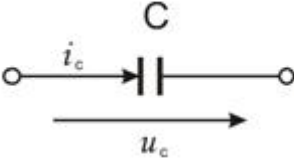
$$\begin{cases} \dot{V} = -\frac{c}{m}x - \frac{k}{m}V \\ \dot{x} = V \end{cases} \quad (11)$$

Пример 2

Рассмотрим следующий объект – четырехполюсник, схема которого приведена на рисунке 2. Для составления дифференциального уравнения этого участка цепи следует воспользоваться законами линейных электрических цепей, в частности уравнениями элементов для мгновенных значений тока и напряжения, приведенных в таблице 1.

Таблица 1 – Уравнениями элементов для мгновенных значений тока и напряжения

Элемент	Уравнение
	$u(t) = R \cdot i$

Элемент	Уравнение
	$u_L = L \frac{di}{dt}$
	$i_c = C \frac{du_c}{dt}$

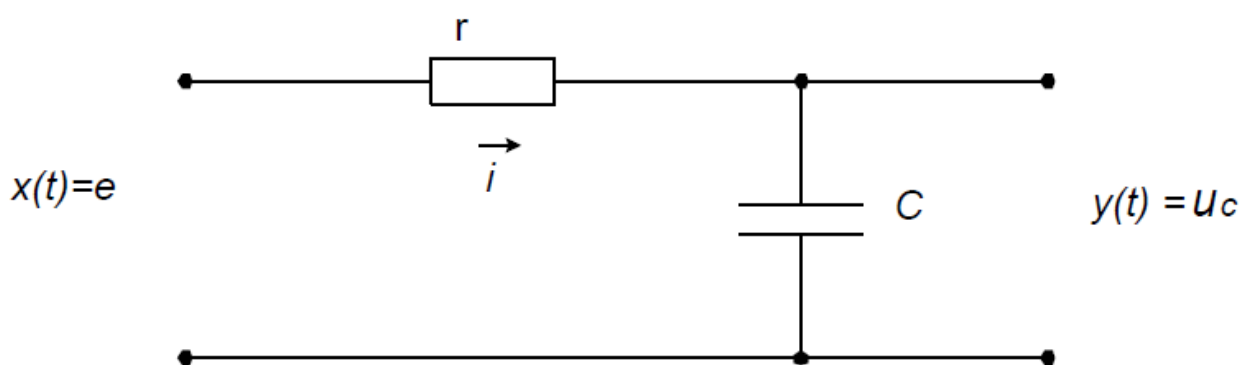


Рисунок 2 – Схема элемента электрической цепи

В соответствии с законами линейных электрических цепей записываем следующие уравнения:

$$ri + u_c = e \quad (12)$$

$$i = c \frac{du_c}{dt} \quad (13)$$

Подставляя значение тока i из выражения (13) в уравнение (12) получаем дифференциальное уравнение:

$$rc \frac{du_c}{dt} + u_c = e$$

или:

$$\dot{y} = \frac{1}{rc} x - \frac{1}{rc} y.$$

1.3 Программа и методика выполнения работы

1. Для выданного преподавателем простого динамического объекта или участка электрической цепи составить аналитическую модель в виде дифференциального уравнения.

2. С помощью любого языка программирования или пакета математического программирования произвести численное моделирование заданного объекта.

3. Провести имитационное моделирование заданного объекта с помощью средств системной динамики среды AnyLogic.

4. Найти в открытых источниках (библиотека, сеть Интернет) описание аналитической модели непрерывного процесса или объекта более сложной формы (например, математическую модель полета самолета, квадрокоптера, движения автомобиля). Изучить процесс получения модели, выяснить на каких законах строится вывод уравнений движения. Выяснить, какие силы учитываются при построении модели, а какими авторы пренебрегают и почему.

5. Оформить отчет по работе.

1.4 Описание лабораторной установки

Для выполнения имитационного моделирования (п.3 программы лабораторной работы) используются средства системной динамики пакета AnyLogic. Инструмент имитационного моделирования AnyLogic обладает рядом преимуществ, главное из которых – возможность реализации всех направлений имитационного моделирования в одной модели (многоподходовое моделирование, включающее в себя элементы дискретно-событийного, системно-динамического и агентного подходов к моделированию).

Использование AnyLogic предоставляет уникальную возможность войти в мир моделирования, имея лишь базовую подготовку в области информационных технологий.

Это современная среда разработки моделей на языке Java с русскоязычным графическим интерфейсом и тщательно продуманной контекстной справочной системой. AnyLogic содержит большую библиотеку визуальных компонентов. Разработчик может также создавать и добавлять в среду собственные компоненты. Модели сохраняются как Java-апплеты. В профессиональной версии работает отладчик и можно создавать автономные JAR-файлы. AnyLogic-модели обладают хорошими средствами 2D–3D симуляции, интерактивности и развитыми возможностями проведения экспериментов (в том числе оптимизационных).

Модели с сосредоточенными параметрами строятся на основе дифференциальных уравнений в «обыкновенных» производных по одной переменной (времени или координате). Параметры в математических моделях таких систем в каждый конкретный момент времени имеют единственное значение.

В качестве примера построим имитационную модель пружинного маятника. Математическая модель, записанная в виде системы уравнений, имеет вид:

$$\begin{cases} \dot{V} = -\frac{c}{m}x - \frac{k}{m}V \\ \dot{x} = V \end{cases}$$

где m – масса груза; x – координата; V – скорость; c – жесткость пружины; k – коэффициент трения. Начальные условия: $x(t = 0) = x_0$; $V(t = 0) = 0$.

Для построения данной модели разработаем следующий алгоритм:

1. Создадим новую модель Swing.
2. В окне графического редактора поместим элементы системы дифференциальных уравнений – математическую модель. Перетащим элемент Накопитель из палитры Системная динамика (рисунок 3).

Переименуем элемент в x , установим Свойства: Режим задания уравнения: Произвольный. Введем формулу: $d(x)/dt = V$. Установим начальное значение (отклонение) (рисунок 4).

3. Повторим действия предыдущего шага для элемента V . Вид уравнения для него (для сокращения числа параметров примем в модели «условную» массу, равную единице):

$$d(V)/dt = -x*c - k*V.$$

4. Из группы Системная динамика введем в модель два параметра: жесткость пружины – c и коэффициент трения – k . Значения по умолчанию (равные нулю) изменим: для c – на 1, для k – на 0.1. На этом этапе логика модели показана на рисунок 5.

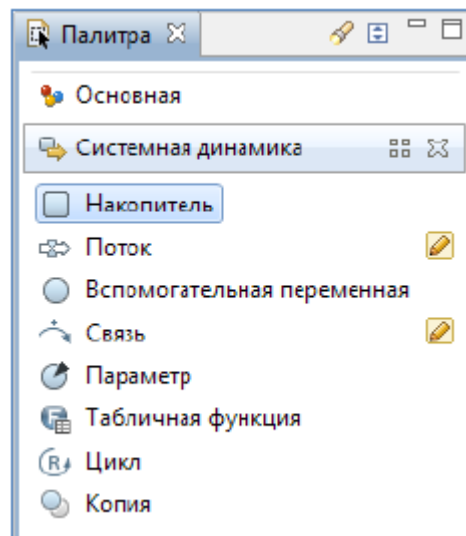


Рисунок 3 – Компоненты моделирования системной динамики

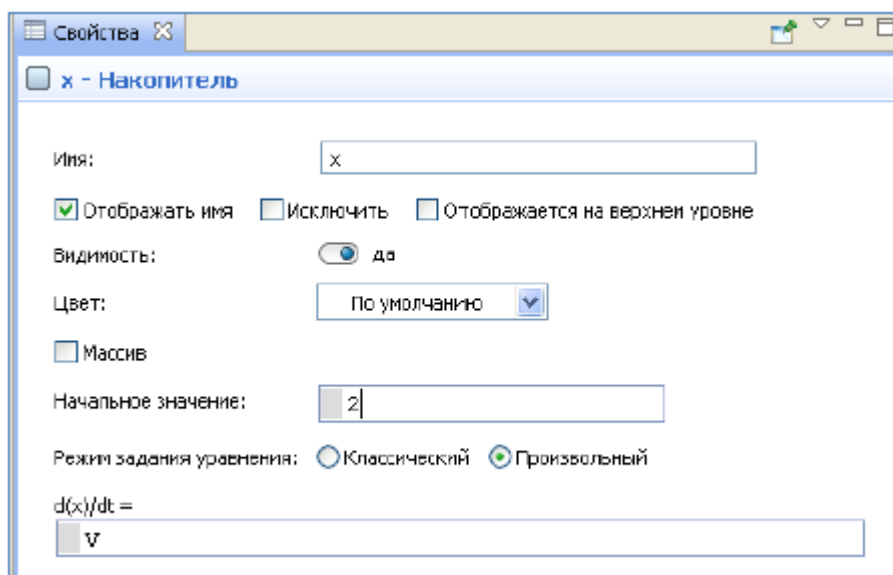


Рисунок 4 – Панель свойств координаты

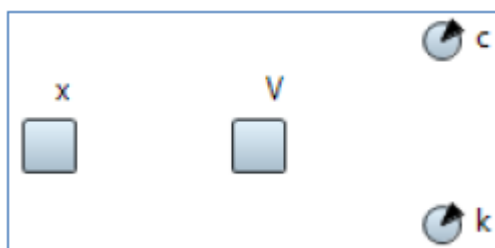


Рисунок 5 – Фрагмент окна графического редактора с элементами математической модели

5. Попробуем запустить модель. В панели ошибок появляются два сообщения (рисунок 6).

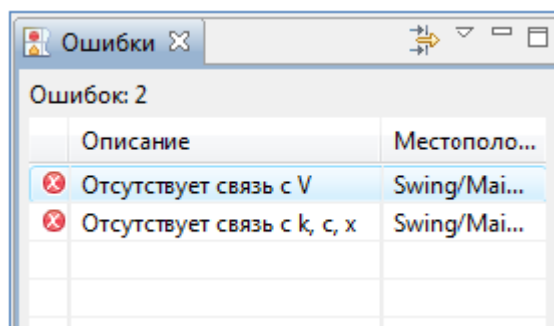


Рисунок 6 – Панель ошибок

Ситуация объясняется тем, что в нашей системе уравнений «уравнения есть, а системы нет». Элементы необходимо связать. Можно сделать это вручную с помощью элементов Связь (см. рисунок 3). Однако проще щелкнуть мышью по описанию ошибки, и AnyLogic сам направит нас в панель Свойства элемента, с которым связана данная проблема. Далее будем просто последовательно подтверждать предлагаемые средой действия (рисунок 7). Связи будут установлены с учетом «правильных» направлений (влияний) (рисунок 8).

Затем снова запустим модель. В этот раз она работает без ошибок. Поставим выполнение на паузу. Щелкнем по одному из накопителей. Откроется «инспект» состояния элемента (рисунок 9).

С помощью мыши его можно перемещать и растягивать (изменять размеры). Кнопка позволяет корректировать текущее числовое значение данной величины. Щелчок по кнопке приводит к показу графика ее динамики во времени (рисунок 10).

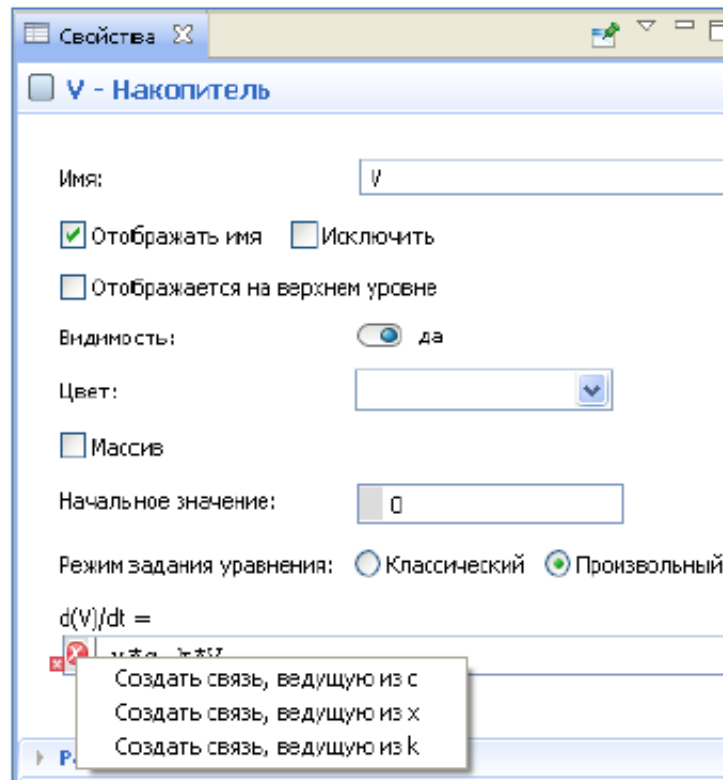


Рисунок 7 – Диалог исправления ошибок

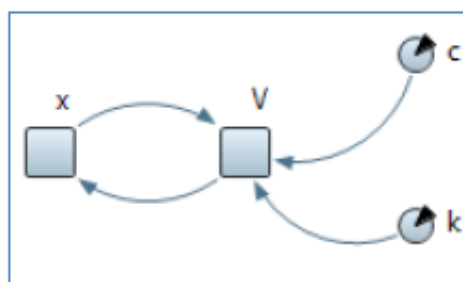


Рисунок 8 – Математическая модель с установленными связями

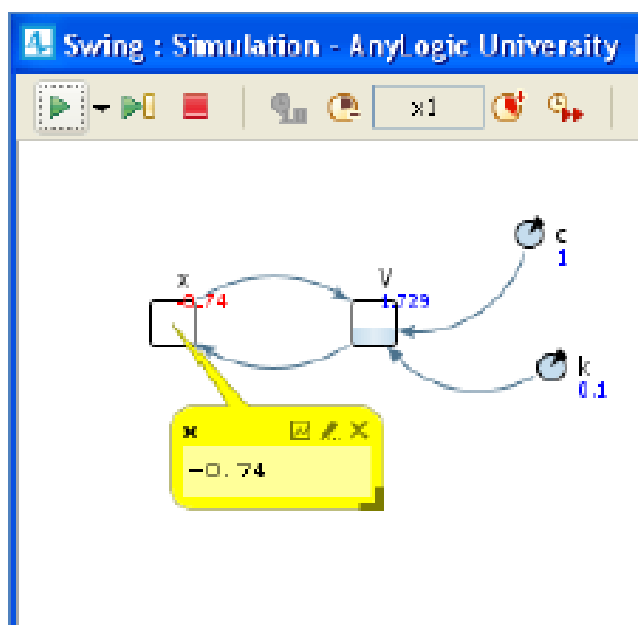


Рисунок 9 – Фрагмент презентации модели с открытым «инспектом»

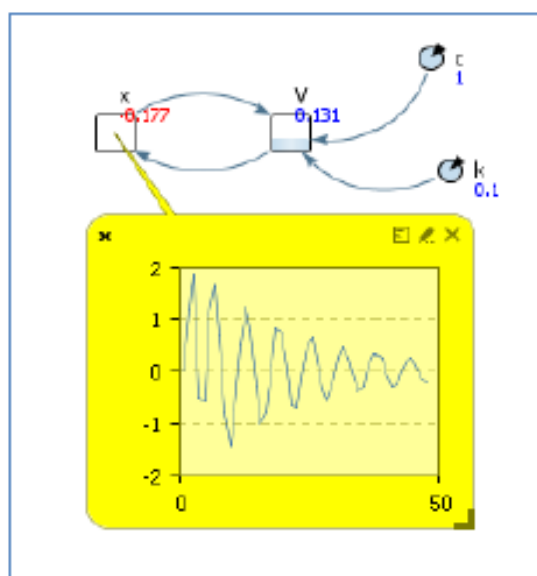


Рисунок 10 – Временной график координаты тела

Как видим, процесс протекает корректно – с затуханием. Изменение (пока вручную) числовых значений жесткости пружины c и коэффициента трения k показывает эффекты, соответствующие реальным системам.

Таким образом, получили адекватную модель в соответствии с задачей данного этапа.

1.5 Результаты экспериментальных исследований

Результаты численного и имитационного моделирования оформить в виде графиков. Сравнить полученные результаты: графики должны совпадать.

1.6 Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- вывод аналитической модели заданного объекта;
- текст программы численного моделирования и результат ее работы в виде графика;
- структуру имитационной модели и результат прогона модели в виде графика;
- описание найденной аналитической модели непрерывного процесса или объекта;
- выводы по работе.

1.7 Контрольные вопросы

1. Что такое моделирование?
2. Классификация моделей.
3. Какие методы применяются для аналитического моделирования непрерывных систем?
4. С помощью каких математических объектов могут быть описаны непрерывные процессы, происходящие в детерминированных системах?
5. Что подразумевается под численным моделированием?
6. Чем имитационное моделирование отличается от аналитического?
7. Что такое структурная диаграмма?
8. Для чего применяются динамические значения параметров в окне презентации AnyLogic?
9. Как запустить модель AnyLogic на выполнение?
10. Как переключиться из режима виртуального времени в реальное?
11. Как изменить скорость выполнения модели?
12. Как показать график изменения переменной модели?

1.8 Библиографический список рекомендуемой литературы

1. Самарский А.А. Математическое моделирование [Электронный ресурс]: идеи. Методы. Примеры/ Самарский А.А., Михайлов А.П.— Электрон. текстовые данные.— М.: ФИЗМАТЛИТ, 2005.— 320 с.— Режим доступа: <http://www.iprbookshop.ru/24708>.— ЭБС «IPRbooks», по паролю.
2. Дьяконов В.П. MATLAB. Полный самоучитель [Электронный ресурс]/ Дьяконов В.П.— Электрон. текстовые данные.— М.: ДМК Пресс, 2014.— 768 с.— Режим доступа: <http://www.iprbookshop.ru/7911>.— ЭБС «IPRbooks», по паролю.
3. Справочная система AnyLogic. <http://www.anylogic.ru/anylogic/help/>

ЛАБОРАТОРНАЯ РАБОТА № 2. ИССЛЕДОВАНИЕ СПОСОБОВ МОДЕЛИРОВАНИЯ НЕПРЕРЫВНО-СТОХАСТИЧЕСКИХ СИСТЕМ

2.1 Цель работы

Исследование характеристик одноканальной системы массового обслуживания, используя аналитический и имитационный методы моделирования. Изучение особенностей работы и получение практических навыков постановки, отладки и получения результатов с помощью пакета моделирования Anylogic.

Трудоемкость лабораторной работы: 9 ч (6 ч – аудиторных, 3 ч – самостоятельная работа студента).

2.2 Краткие теоретические сведения

Рассмотрим модель системы, состоящей из одного канала обслуживания запросов, поступающих на вход системы (одноканальная СМО), используя аналитический и имитационный подход.

Аналитическая модель одноканальной СМО

При ряде упрощающих предположений (допущение об отсутствии последствия) для оценки вероятностно-временных характеристик процесса обслуживания можно использовать дискретно-событийный подход к моделированию. Самая известная модель – это СМО типа М/М/1, где М – процессы поступления и обслуживания заявок, интервалы времени которых распределены по экспоненциальному закону, 1 – число обслуживающих устройств. При этом предполагается, что поток входных заявок является простейшим.

Простейший поток обладает следующими свойствами:

стационарностью – вероятность характеристик потока не зависит от времени;

отсутствием последствия – заявки поступают не зависимо друг от друга, длина интервала времени до момента поступления следующей заявки не зависит от того, поступила в начальный момент заявка или нет;

ординарностью – в каждый момент времени в систему может поступить не более одной заявки.

Для простейшего потока число заявок, поступающих в систему за промежуток времени t , описывается распределением Пуассона.

$$P(t) = \frac{(\lambda t)^n}{n!} \cdot e^{-\lambda t}, \quad (14)$$

вероятность того, что за время t в систему поступит точно n заявок; λ – интенсивность потока заявок.

Для того, чтобы при моделировании задать пуассоновский поток заявок в систему, достаточно задать экспоненциальное распределение интервалов времени поступления для соседних заявок.

Длительность обслуживания экспоненциально распределена с плотностью

$$f(t) = \mu \cdot e^{-\mu t}, \quad (15)$$

μ – интенсивность обслуживания – количество заявок, которое может быть обслужено в единицу времени.

Соответственно

$a = 1/\lambda$ – средний интервал времени между поступлением заявок;

$b = 1/\mu$ – среднее время обслуживания заявок.

Для стационарного режима ($\lambda \ll \mu$) справедливо рекуррентное соотношение для расчета вероятностей

$$(1 + \rho)P_n = P_{n+1} + \rho \cdot P_{n-1}, \quad n > 1, \quad P_1 = \rho \cdot P_0, \quad (16)$$

где $\rho = \frac{\lambda}{\mu}$ – загрузка системы (характеристика качества функционирования системы),

Так как $P_0 = 1 - \rho$ – вероятность того, что заявки в системе отсутствуют, то

$$P_n = \rho^n \cdot (1 - \rho). \quad (17)$$

вероятность того, что в системе находится n – заявок.

Среднее и дисперсия числа заявок в системе определяются как:

$$m = \sum_{n=0}^{\infty} n \cdot P_n = (1 - \rho) \cdot \sum_{n=0}^{\infty} n \cdot \rho^n = \frac{\rho}{1 - \rho} \quad (18)$$

$$\sigma_m^2 = \sum_{n=0}^{\infty} (n - m)^2 \cdot P_n = \frac{\rho}{(1 - \rho)^2}. \quad (19)$$

Среднее и дисперсия числа заявок, находящихся в очереди к прибору, соответственно равны:

$$l = \sum_{n=0}^{\infty} n \cdot P_n = (1 - \rho) \cdot \sum_{n=1}^{\infty} n \cdot \rho^n = \frac{\rho^2}{1 - \rho} \quad (20)$$

$$\sigma_l^2 = \sum_{n=1}^{\infty} (n - m) \cdot P_n - m^2 = \frac{\rho^2 \cdot (1 + \rho - \rho^2)}{(1 - \rho)^2}. \quad (21)$$

Вероятность того, что время пребывания заявки в системе T не превосходит t при условии, что длина очереди $l = n$ определяется из условия

$$P(T < t | l = n) = \int_0^t \frac{\mu^{n+1} \cdot \tau^n \cdot e^{-\mu \tau}}{\Gamma(n+1)} d\tau, \quad (22)$$

где $\Gamma(n)$ – гамма-функция.

По формуле полной вероятности можно определить вероятность того, что время пребывания заявки в системе T не превосходит t

$$Q(t) = P(T \leq t) = \sum_{n=0}^{\infty} P(T \leq t | l = n) \cdot \rho \cdot (1 - \rho) = 1 - e^{-\mu(1-\rho)t}. \quad (23)$$

Среднее и дисперсия времени пребывания заявки в системе рассчитывается по формулам

$$u = \int_0^{\infty} t dQ(t) = \frac{1}{\mu \cdot (1-\rho)}, \quad (24)$$

$$\sigma_u^2 = \int_0^{\infty} (t - u)^2 dQ(t) = \frac{1}{(\mu \cdot (1-\rho))^2}. \quad (25)$$

Возможности оценки характеристик стохастических систем при помощи аналитических моделей ограничены по сравнению с имитационными моделями, созданными в специализированных пакетах моделирования (GPSS World, Anylogic, Matlab Simulink, и др.) или использующими специально разработанные языки моделирования (GPSS, UML). Эти ограничения вызваны:

- различием в обслуживании заявок разных типов, например, связанных с наличием приоритетов;
- невозможностью представления в аналитических моделях обслуживания заявок одновременно несколькими приборами;
- трудностями получения зависимостей в явной аналитической форме при большом числе систем в сети; сети, содержащие более трех систем, уже не поддаются аналитическому исследованию [4,5].

Минимальное время моделирования, можно определить из следующей зависимости

$$\Delta t = (a + b). \quad (26)$$

Имитационная модель одноканальной СМО

Имитационная модель представляет собой систему, отображающую структуру и функционирование исходного объекта в виде алгоритма, связывающего входные и выходные переменные, принятые в качестве характеристик исследуемого объекта. При ее реализации на ЭВМ производится накопление статистических данных по тем атрибутам модели, характеристики которых являются предметом исследований. По окончании моделирования накопленная статистика обрабатывается, и результаты моделирования получаются в виде выборочных распределений исследуемых величин [1,3,5].

Библиотека Моделирования Процессов AnyLogic поддерживает дискретно-событийный ("процессный") подход моделирования. С помощью объектов Библиотеки Моделирования Процессов вы можете моделировать системы реального мира, динамика которых представляется как последовательность операций (прибытие, задержка, захват ресурса, разделение, ...) над агентами, представляющими клиентов, документы, звонки, пакеты данных, транспортные средства и т.п. Эти агенты могут обладать определёнными атрибутами, влияющими на

процесс их обработки (например, тип звонка, сложность работы) или накапливающими статистику (общее время ожидания, стоимость).

Процессы задаются в форме потоковых диаграмм (блок-схем) - графическом представлении, принятом во многих областях: производстве, бизнес-процессах, центрах обработки звонков, логистике, здравоохранении и т.д. Потоковые диаграммы AnyLogic иерархичны, масштабируемы, расширяемы и объектно-ориентированы, что позволяет пользователю моделировать сложные системы любого уровня детальности.

Пример процесса создания дискретно-событийной модели и сбора статистики в программе AnyLogic приведен в Приложении А.

Одноканальная экспоненциальная СМО содержит

- блок *Source* (источник заявок), в параметрах которого следует задать экспоненциальный закон распределения интервала поступления заявок;
- блок *Queue* (очередь), в параметрах которого следует установить вместимость накопителя;
- блок *Delay* (задержка), моделирующий работу одного прибора, в его параметрах следует также задать экспоненциальный закон распределения времени обработки заявок;
- блок *Sink* (уничтожает обработанные заявки).

Полная последовательность блоков описания модели приведена в Таблице 2.

Таблица 2 – Блоки описания модели с учетом блоков сбора статистики

№ блока	Имя блока	Параметры блока	Комментарии
1	Source	интенсивность (или интервал) поступления заявок, закон распределения интервалов поступления заявок	интенсивность λ обратнопропорциональна интервалу a
2	Queue	вместимость очереди	поставить заявку в очередь к прибору
3	Delay	интенсивность (или время задержки) обслуживания, закон распределения интервалов обслуживания заявок	интенсивность μ обратнопропорциональна времени задержки b
4	Sink		

2.3 Программа и методика выполнения работы

1. Оценить аналитическими методами вероятность нахождения в системе n заявок P_n (3) для $n = 0, 1, 2, \dots, 10$, среднее число и дисперсию числа заявок в системе и в очереди (5,6,7,8).

2. Построить графики функции распределения времени пребывания заявки в системе $Q(t)$ (9), (10) для $t = 0, \Delta t, 2 * \Delta t, \dots, 10 * \Delta t$.

3. Оценить среднее и дисперсию времени пребывания заявки в системе (11), (12).

4. Запрограммировать модель одноканальной СМО, в соответствии с требованиями программы моделирования (Приложение А). Подставить в нее исходные данные (для источника и обслуживающего прибора) согласно варианту задания. Вывести всю необходимую статистику и сохранить ее для дальнейшего анализа.

5. Повторить п.4, введя в программу снятие статистики об ожидании в очереди при обслуживании устройством. Сопоставить полученные файлы результатов. Определить среднее время пребывания заявки в системе u .

6. Повторить п.4 для значений $t = \Delta t, 5 * \Delta t, \dots, 50 * \Delta t$. Определить u . Построить график зависимости u и коэффициента использования прибора (загрузки системы ρ).

7. Сравнить результаты моделирования с расчетами по аналитическим зависимостям. Сделать выводы.

8. Оформить отчет по работе.

Таблица 2 – Варианты задания

№	1	2	3	4	5	6	7	8	9	10
λ, c^{-1}	3,5	2,4	1,5	0,8	3,0	4,5	3,0	9,0	20	30
μ, c^{-1}	7,0	4,0	2,0	1,0	5,0	5,0	4,0	10	25	40
№	11	12	13	14	15	16	17	18	19	20
λ, c^{-1}	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
b, c	9,5	3,0	2,5	2,0	1,0	1,5	0,5	1,0	1,0	0,8
№	21	22	23	24	25	26	27	28	29	30
a, c	2,0	5,0	3,5	2,5	2,0	1,6	1,25	1,0	0,6	0,5
b, c	0,1	2,5	3,0	2,0	1,5	1,0	1,0	0,7	0,2	0,1

2.4 Описание лабораторной установки

При выполнении лабораторной работы используется компьютер с установленным программным пакетом Anylogic. Пример создания дискретно-событийной модели приведен в приложении А.

2.5 Результаты экспериментальных исследований

После прогона модели будет получена статистика об использовании устройства и состоянии очереди на входе устройства. Оценкой среднего времени пребывания заявки в системе по имитационной модели может служить величина u , рассчитываемая по формуле (24).

Сопоставляя результаты моделирования с расчетами по аналитическим зависимостям:

- среднего время пребывания заявки в системе,
- значений загрузки системы,

– наличия очереди и ее характеристик,
можно на примере простейшей СМО качественно оценить пригодность использования аппарата имитационного моделирования.

2.6 Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- результаты расчетов (таблицы, графики) по аналитическим зависимостям;
- структурная схема модели;
- графики зависимости длины очереди, загрузки и среднего времени пребывания заявки в системе от времени моделирования Δt ;
- результаты моделирования в среде Anylogic;
- выводы по работе.

2.7 Контрольные вопросы

1. Какие математические модели применяются для аналитического моделирования стохастических систем? Чем отличаются эти модели?
2. Какими основными характеристиками пользуются для аналитического моделирования СМО и СеМО?
3. Что такое поток событий?
4. Свойства простейшего потока событий.
5. К каким классам задач, применим аппарат имитационного моделирования?
6. Структурная схема трехканальной СМО с одной очередью. Дисциплины буферизации.
7. Структурная схема одноканальной СМО с тремя очередями. Дисциплины обслуживания.
8. Для чего применяется программа Anylogic?
9. Каково назначение блоков delay, source, sink? Какие параметры в них можно задавать?
10. Как описывают устройства обслуживания и очереди в Anylogic?

2.8 Библиографический список рекомендуемой литературы

1. Салмина Н.Ю. Имитационное моделирование [Электронный ресурс]: учебное пособие/ Салмина Н.Ю.— Электрон. текстовые данные.— Томск: Эль

Контент, Томский государственный университет систем управления и радио-электроники, 2012.— 90 с.— Режим доступа: <http://www.iprbookshop.ru/13930>.— ЭБС «IPRbooks», по паролю.

2. Замятина О.М. Моделирование сетей [Электронный ресурс]: учебное пособие/ Замятина О.М.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2012.— 160 с.— Режим доступа: <http://www.iprbookshop.ru/34683>.— ЭБС «IPRbooks», по паролю.

3. Шелухин О.И. Моделирование информационных систем [Электронный ресурс]: учебное пособие/ Шелухин О.И.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 536 с.— Режим доступа: <http://www.iprbookshop.ru/12002>.— ЭБС «IPRbooks», по паролю.

ЛАБОРАТОРНАЯ РАБОТА № 3 . ИССЛЕДОВАНИЕ ТЕХНОЛОГИИ ДИСКРЕТНО-СОБЫТИЙНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

3.1 Цель работы

Исследование технологии дискретно-событийного имитационного моделирования. Изучение базовых блоков программы моделирования Anylogic и получение практических навыков программирования имитационных моделей.

Трудоемкость лабораторной работы: 9 ч (6 ч – аудиторных, 3 ч – самостоятельная работа студента).

3.2 Краткие теоретические сведения

Модельное время

Для обеспечения правильной временной последовательности событий в модели, организованы часы, хранящие значения текущего момента времени в модели. Время в языках моделирования организовано так, что часы меняют свое значение только для того, чтобы указать время наступления ближайшего события. Например, текущее значение часов модели равно 2, а очередное событие должно наступить в момент времени 7 (например, генерация новой заявки), следовательно, значение часов увеличивается сразу на 5 единиц. Основной условной единицей времени в модели можно выбрать любую единицу (секунда, минута, час, год), которая позволит получить необходимую точность моделирования.

В программе Anylogic есть также возможность переключиться в режим реального времени с коэффициентом масштабирования (время может быть замедлено или ускорено относительно реального).

Заявки (транзакты, сообщения)

Заявки, или сообщения являются абстрактными подвижными элементами, которые моделируют объекты реального мира: клиентов, покупателей, сообщения, программы, сбои. Перемещаясь между блоками модели, заявки вызывают (и испытывают) различные действия (задержка в некоторых точках модели, изменение маршрутов и направления движения, расщепление заявок на несколько копий).

Типовые конструкции программ моделирования

Рассмотрим алгоритмы функционирования типовых узлов моделей систем:

Задача 1. Очередь к прибору с ограниченным количеством мест в очереди.

Схема функционирования данной задачи приведена на Рисунке 11.

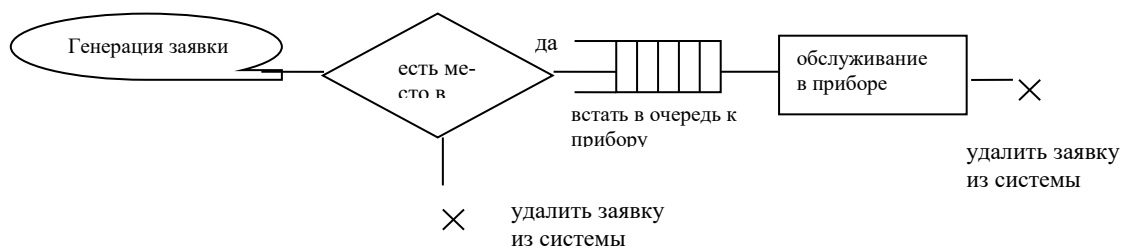


Рисунок 11 – Схема функционирования модели системы с потерями

Очередь с ограниченным числом мест в Anylogic реализуется с помощью блока Queue: размер накопителя задается параметром Capacity, а дисциплина буферизации – во вкладке Advanced.

Задача 2. Параллельная обработка заявки двумя приборами.
Схема функционирования данной задачи приведена на Рисунке 12.



Рисунок 12 – Схема функционирования модели системы с параллельной обработкой заявки двумя приборами

Из приведенной схемы видно для того, чтобы проимитировать параллельную обработку заявки двумя приборами необходимо:

- создать копию заявки, поступающей на обслуживание;
- заявка направляется на первый прибор, где обслуживается заданное время;
- указанная копия направляется на второй прибор, где обслуживается заданное время;
- после обработки объединить заявку со своей копией.

В Anylogic копию заявки можно создать с помощью блока Split, а объединить две заявки в одну – с помощью блока Combine.

Задача 3. Два потока заявок, поступающих к одному прибору. Схема функционирования данной задачи приведена на Рисунке 13.

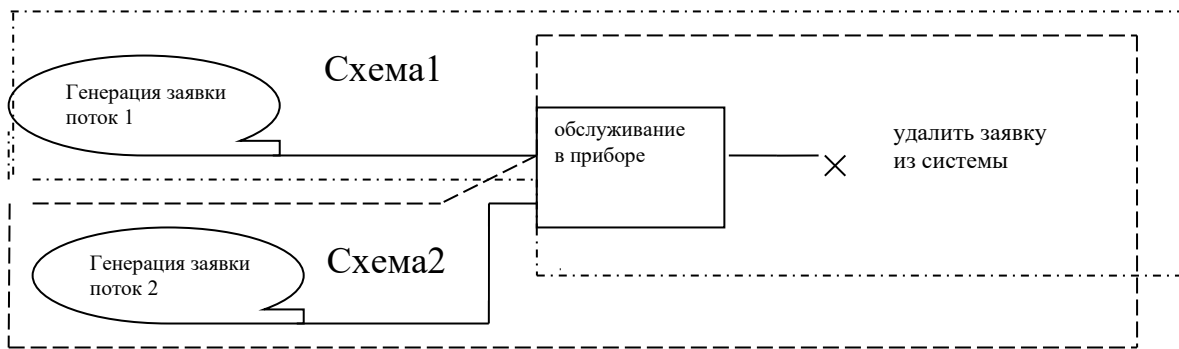


Рисунок 13 – Схема функционирования модели системы с двумя потоками к одному обслуживающему прибору

Имитацию системы с двумя потоками можно разложить на две независимые схемы:

Схема 1 – Поток 1 поступает к прибору и обслуживается в приборе.

Схема 2 – Поток 2 поступает к прибору и обслуживается в приборе.

Соответственно программа может быть представлена двумя независимыми наборами блоков, в каждый из которых входят блоки, описывающие работу устройства обслуживания. Захват прибора заявками из первого или из второго потока будет зависеть от времени появления заявки в системе. Если второй поток прерывает первый, то соответственно второй набор будет включать группу блоков, обеспечивающую прерывание работы прибора и возврат его в исходное состояние.

В Anylogic такой функционал обеспечивается с помощью механизма захвата ресурсов. Для этого надо создать две параллельно работающие схемы с элементами Source, Delay, Sink, задать им соответствующие параметры поступления и обработки каждого потока. Для захвата одного прибора нужен блок ResourcePool емкостью 1, т.е. 1 ресурс (например, это может быть 1 человек, исполняющий операцию, или 1 прибор, с помощью которого выполняется обработка обоих поступающих потоков). Затем в каждую из двух схем до блока Delay надо добавить блок захвата ресурсов Seize. А после блока Delay – блок освобождения ресурсов Release. Параметры этих блоков задаются в соответствии с задачей, в поле Resource sets блока Seize следует добавить созданный ранее ресурс.

Задача 4. Поток заявок, поступающих к прибору, проходит несколько циклов обслуживания.

Для того, чтобы обеспечить полное обслуживание заявки, она должна пройти несколько циклов обработки, т.е. многократно занимать и покидать устройство. Так, например, заявка №1 должна пройти четыре цикла обработки, после второго цикла заявка №1 в очередной раз покидает устройство, но в очередь к устройству уже поставлена заявка №2, следовательно, устройство будет занято заявкой №2, которая пойдет на свой первый цикл обслуживания в устройстве, после первого цикла обслуживания заявка №2 покидает устройство и устройство занимает заявка №1.

Схема функционирования данной задачи приведена на рисунке 14, а пояснение приведенного примера – на диаграмме (рисунок 15).

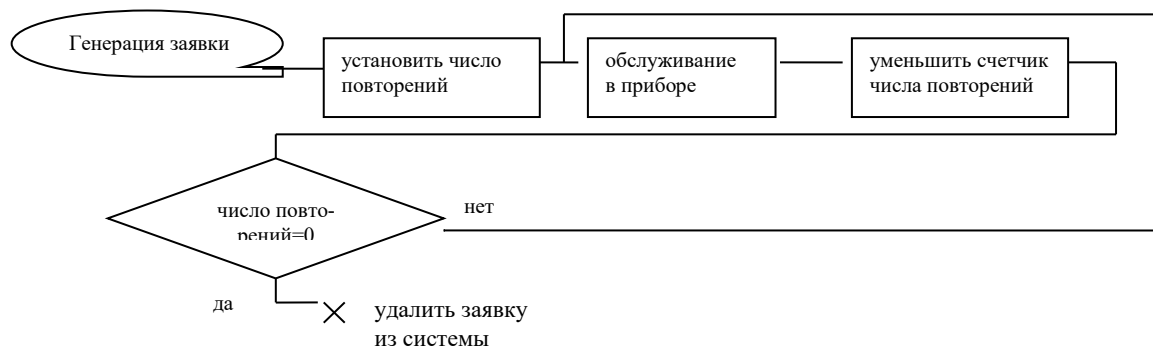


Рисунок 14 – Схема функционирования модели системы с заданным количеством циклов обработки заявки в устройстве



Рисунок 15 – Диаграмма функционирования модели системы с заданным количеством циклов обработки заявки в устройстве

Проверка счетчика числа повторений (циклов) для каждой заявки производится блоком SelectOutput. Для организации этого счетчика можно создать, например, соответствующее свойство агента.

Задача 5. Поток заявок распределяется на два прибора, четные заявки обрабатываются на первом приборе, нечетные на втором.

Схема функционирования данной задачи приведена на Рисунке 16.

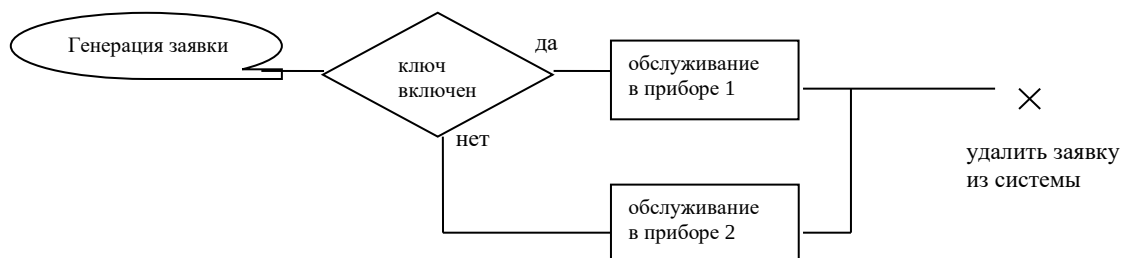


Рисунок 16 – Схема функционирования модели системы с перенаправлением потока на два прибора

Задача реализуется с помощью блока SelectOutput.

3.3 Программа и методика выполнения работы

1. Запрограммировать имитационные модели в Anylogic с помощью библиотеки моделирования процессов согласно варианту (Таблица 3).
2. Организовать в моделях вывод статистики: вывести гистограммы среднего времени пребывания заявки в системе, загрузки системы, средней длины очереди и среднего времени ожидания заявки.
3. Для каждой задачи разработать граф состояний системы.
4. Запрограммировать имитационные модели в Anylogic с помощью библиотеки диаграмм состояний.
5. Предъявить преподавателю действующие программы (по две имитационные модели на каждую задачу).
6. Оформить отчет по работе.

Таблица 3 – Варианты заданий

Вариант	Номера задач	Значения параметров
1	1 4 9	X1=100, X2=50, X3=70, X4=20 X23=40, X24=15, C=10
2	10 8	X5=200, X6=50, X7=100, X8=30, X9=50, X10=10 S1=15 X17=300, X18=50, X19=70, X20=20, X21=30, X22=15
3	1 5 11	X1=200, X2=60, X3=100, X4=45 X13=70, X14=25, X15=45, X16=20
4	1 7 9	X1=100, X2=50, X3=70, X4=20 S1=10 X23=33, X24=13, C=10
5	1 2 5	X1=120, X2=45, X3=40, X4=15 X5=170, X6=30, X7=120, X8=50, X9=80, X10=10, X11=100, X12=10
6	1 4 8	X1=12, X2=5, X3=4, X4=1 X17=10, X18=5, X19=7, X20=2, X21=3, X22=2
7	2 5 6	X5=20, X6=5, X7=10, X8=3, X9=5, X10=1, X11=20, X12=7 P1=0,4, P2=0,6
8	1 10 9	X1=220, X2=30, X3=44, X4=22 X5=220, X6=30, X7=80, X8=30, X9=70, X10=10 S1=5 X23=34, X24=22, C=2

Вариант	Номера задач	Значения параметров
9	3	X5=440, X6=66, X7=88, X8=23, X9=50, X10=20, X11=15, X12=5
	8	X17=220, X18=50, X19=50, X20=20, X21=60, X22=15
10	1	X1=400, X2=50, X3=120, X4=20
	10	X5=50, X6=20, X7=40, X8=20, X9=25, X10=5, S1=10
	6	P1=0,2, P2=0,8
11	1	X1=100, X2=50, X3=70, X4=20
	11	
	9	X23=33, X24=13, C=10
12	1	X1=100, X2=30, X3=30, X4=20
	4	
	5	
13	1	X1=15, X2=8, X3=8, X4=4
	7	S1=7
	8	X17=15, X18=5, X19=8, X20=6, X21=4, X22=1
14	3	X5=15, X6=4, X7=15, X8=2, X9=4, X10=2, X11=15, X12=5
	6	
	8	X17=20, X18=5, X19=5, X20=2, X21=4, X22=2

Согласно варианту, из таблицы 4 необходимо выбрать номера задач, имитационные модели которых необходимо запрограммировать.

Таблица 4 – Словесное описание типовых конструкций программ

Номер задачи	Описание типовых конструкций программы
1	Процесс прохождения заявок, поступление которых подчиняется равномерному закону с интервалом $X1 \pm X2$ единицы времени, а обработка – равномерному закону со средним временем $X3 \pm X4$ единицы.
2	На обработку поступает два потока заявок: первый – с интервалом $X5 \pm X6$ единиц, второй $X7 \pm X8$ единицы. Время обработки заявок первого потока $X9 \pm X10$ единицы, второго $X11 \pm X12$ единицы. а) потоки имеют равные приоритеты обработки; б) приоритет 1-го потока выше приоритета 2-го потока
3	На обработку поступает два потока заявок: первый – с интервалом $X5 \pm X6$ единиц, второй $X7 \pm X8$ единицы. Второй поток прерывает обработку заявок первого. Время обработки заявок первого потока $X9 \pm X10$ единицы, второго $X11 \pm X12$ единицы.

Номер задачи	Описание типовых конструкций программы
4	В условиях задачи 1 необходимо произвести параллельную обработку заявки двумя приборами.
5	В условия задачи 1 добавляется второй прибор, скорость которого в 2 раза выше первого. При этом предпочтительнее обработка на первом приборе.
6	В условиях задачи 5 заявки поступают к первому прибору с вероятностью P_1 и ко второму прибору с вероятностью P_2 .
7	В условиях задачи 1 заявки поступают на обработку с ограниченным числом мест в очереди, равным S_1 . Если очередь заполнена, то заявки покидают систему.
8	На обработку поступает поток заявок с интервалом $X_{17} \pm X_{18}$ единиц времени. Нечетные заявки обрабатываются на первом приборе со временем $X_{19} \pm X_{20}$ единицы, четные заявки – на втором приборе со временем $X_{21} \pm X_{22}$ единицы.
9	В условиях задачи 1 каждая заявка проходит S -циклов обработки на приборе со временем $X_{23} \pm X_{24}$
10	На обработку поступает два потока заявок: первый – с интервалом $X_5 \pm X_6$ единиц, второй $X_7 \pm X_8$ единицы. Оба потока поступают на обработку с ограниченным числом мест в очереди, равным S_1 . Если очередь заполнена, то заявки покидают систему. Время обработки заявок для каждого потока $X_9 \pm X_{10}$. Оценить потери для каждого потока.
11	В условиях задачи 1 необходимо произвести параллельную обработку заявки тремя приборами.

3.4 Описание лабораторной установки

При выполнении лабораторной работы используется компьютер с установленным программой имитационного моделирования Anylogic.

3.5 Результаты экспериментальных исследований

Зависимости длины очереди, $Util$ и среднего времени пребывания заявки в системе от времени моделирования оформить в виде графиков.

3.6 Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- имитационные модели и результаты их выполнения с описанием;
- графики среднего времени пребывания заявки в системе, загрузки системы, средней длины очереди и среднего времени ожидания заявки;
- выводы по работе.

3.7 Контрольные вопросы

1. Состав и назначение библиотеки моделирования процессов Anylogic.
2. Состав и назначение библиотеки диаграмм состояний Anylogic.
3. Состав и назначение библиотеки статистики Anylogic.
4. Понятие транзакта (заявки). Сколько транзактов может находиться в модели одновременно? Сколько транзактов может двигаться в модели в один и тот же момент времени? В каких случаях прекращается движение транзакта в модели? Каким образом осуществляется продвижение времени в имитационных моделях?
5. Структура пакета Anylogic, 3 уровня моделирования в программе.
6. Назначение блоков Source, Queue, Delay, Seize, Release, Sink, ResourcePool, SelectOutput. Split, Combine, Service.
7. Каким образом осуществляется сбор статистики в Anylogic?
8. Схема функционирования модели системы с потерями.
9. Схема функционирования модели системы с параллельной обработкой заявки двумя приборами.
10. Схема функционирования модели системы с двумя потоками к одному обслуживающему прибору.
11. Схема функционирования модели системы с заданным количеством циклов обработки заявки в устройстве.
12. Схема функционирования модели системы с перенаправлением потока на два прибора.

3.8 Библиографический список рекомендуемой литературы

1. Салмина Н.Ю. Имитационное моделирование [Электронный ресурс]: учебное пособие/ Салмина Н.Ю.— Электрон. текстовые данные.— Томск: Эль Контент, Томский государственный университет систем управления и радиоэлектроники, 2012.— 90 с.— Режим доступа: <http://www.iprbookshop.ru/13930>.— ЭБС «IPRbooks», по паролю.
2. Замятина О.М. Моделирование сетей [Электронный ресурс]: учебное пособие/ Замятина О.М.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2012.— 160 с.— Режим доступа: <http://www.iprbookshop.ru/34683>.— ЭБС «IPRbooks», по паролю.
3. Шелухин О.И. Моделирование информационных систем [Электронный ресурс]: учебное пособие/ Шелухин О.И.— Электрон. текстовые данные.—

М.: Горячая линия - Телеком, 2012.— 536 с.— Режим доступа:
<http://www.iprbookshop.ru/12002>.— ЭБС «IPRbooks», по паролю.

ЛАБОРАТОРНАЯ РАБОТА № 4.

ИССЛЕДОВАНИЕ ПОДХОДОВ К МОДЕЛИРОВАНИЮ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

4.1 Цель работы

Исследование технологий моделирования вычислительных систем на примере имитационного моделирования алгоритмов.

Трудоемкость лабораторной работы: 9 ч (6 ч – аудиторных, 3 ч – самостоятельная работа студента).

4.2 Краткие теоретические сведения

Дискретно-событийное моделирование вычислительных систем

Ресурсы вычислительной системы – это различные устройства (процессор, дисплей, клавиатура, накопители на магнитных дисках и т.д.), а также память, файлы, программные модули. В случае занятости нужного ресурса, который требует программа, к нему создается очередь. Таким образом, если программы пользователей, которые выполняются компьютером используют одинаковые ресурсы в некоторые промежутки времени, то выполнение таких программ будет существенно задерживаться.

Компьютерная система обычно работает в мультипрограммном или многозадачном режиме. В любом режиме работы (мультипрограммном или многозадачном) компьютер с одним процессором в каждый момент времени выполняет только одну программу (одну команду). Если используется мультипроцессорная система с несколькими одинаковыми процессорами, то становится возможным выполнение нескольких программ на разных процессорах. Большей частью такая система моделируется как многоканальная СМО в отличие от СМО с одним устройством обслуживания, которая моделирует работу однопроцессорной системы.

При выполнении операций ввода-вывода для записи или считывания данных с магнитных носителей информации, эти операции большей частью осуществляются через буфер ввода-вывода. Обмен данными между памятью и внешними устройствами осуществляется через контроллеры ввода-вывода, но скорость их работы на порядок больше скорости электромеханических устройств. При моделировании можно считать, что эти операции выполняют сами устройства. Поиск информации на магнитных дисках связан с перемещением головок на нужный сектор и с непосредственным выполнением операции записи или считывания данных, которые зависят от скорости вращения диска. Поэтому для моделирования времени поиска информации используют равномерное распределение, для которого минимальное и максимальное время берут из технических характеристик устройств.

Таким образом, работу вычислительной системы можно представить как работу сети СМО.

При моделировании обычно исследуется время пребывания программы пользователя в системе или число программ, выполненных системой за единицу времени. Типичная программа пользователя может описываться: временем поступления в систему, емкостью, занимаемой памяти, количеством запросов к устройствам ввода–вывода и распределением их по времени, количеству данных, которые вводятся и выводятся на тех или иных устройствах.

Динамическое моделирование вычислительных систем

Методология IDEF2 реализует динамическое моделирование системы. В данной методологии модель разбивается на четыре подмодели:

- подмодель возможностей, которая описывает их инициаторов;
- подмодель потока сущностей, которая определяет их трансформацию;
- подмодель распределения ресурсов, необходимых для осуществления переходов между состояниями;
- подмодель системы, которая описывает внешние взаимодействия.

Методология предполагает, что набор подмоделей может быть переведен в динамическую модель, построенную на базе сетей Петри. Классические сети Петри ввел Карл Адам Петри в 60-х гг. XX в. С тех пор их использовали для моделирования и анализа самых разных систем с приложениями от протоколов, аппаратных средств и внедренных систем до гибких производственных систем, пользовательского взаимодействия и бизнес-процессов.

Сети Петри предназначены для моделирования систем, которые состоят из множества взаимодействующих друг с другом компонент. При этом компонента сама может быть системой. Действиям различных компонент системы присущ параллелизм. Примерами таких систем могут служить вычислительные системы, в том числе и параллельные, компьютерные сети, программные системы, обеспечивающие их функционирование, а также экономические системы, системы управления дорожным движением, химические системы, и т. д.

Представление системы сетью Петри основано на двух основополагающих понятиях: событиях и условиях. Возникновением событий управляет состояние системы, которое может быть описано множеством условий. Условие может принимать либо значение «истина», либо значение «ложь».

Возникновение события в системе возможно, если выполняются определённые условия – предусловия события. Возникновение события может привести к выполнению других условий – постусловий события. В качестве примера рассмотрим следующую ниже задачу моделирования.

Важная особенность сетей Петри — это их асинхронная природа. В сетях Петри отсутствует измерение времени. В них учитывается лишь важнейшее свойство времени – частичное упорядочение событий.

Выполнение сети Петри (или поведение моделируемой системы) рассматривается здесь как последовательность дискретных событий, которая является одной из возможных. Если в какой-то момент времени разрешено более одного перехода, то любой из них может стать «следующим» запускаемым.

Переходы в сети Петри, моделирующей некоторую систему, представляют её примитивные события (длительность которых считается равной 0), и в один момент времени может быть запущен только один разрешённый переход.

Моделирование одновременного (параллельного) возникновения независимых событий системы в сети Петри демонстрируется на рисунке 17.

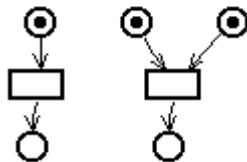


Рисунок 17 – Моделирование параллельных событий с помощью сети Петри

В этой ситуации два перехода являются разрешенными и не влияют друг на друга в том смысле, что могут быть запущены один вслед за другим в любом порядке.

Другая ситуация приведена на рисунке 18.

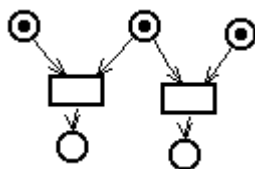


Рисунок 18 – Моделирование конфликта с помощью сети Петри

Эти два разрешённые перехода находятся в конфликте, т. е. запуск одного из них удаляет фишку из общей входной позиции и тем самым запрещает запуск другого. Таким образом, моделируются взаимоисключающие события системы.

Вырожденным случаем параллельной системы процессов является система с одним процессом. Стандартный способ представления структуры программы и потока управления в ней — это схема алгоритма, которая в свою очередь может быть представлена сетью Петри. Схема алгоритма и соответствующая ей сеть Петри представлена на рисунке 19.

Параллельная система может строиться несколькими способами. Один из способов состоит в простом объединении процессов, без взаимодействия во время их одновременного выполнения. Так, например, если система строится этим способом из двух процессов, каждый из которых может быть представлен сетью Петри, то сеть Петри моделирующая одновременное выполнение двух процессов, является простым объединением сетей Петри для каждого из двух процессов. Начальная маркировка составной сети Петри имеет две фишки, по одной в каждой сети, представляя первоначальный счетчик команд процесса.

Такой способ введения параллелизма имеет низкое практическое значение. Далее будем рассматривать параллельные системы процессов, допускающие взаимодействие процессов во время их параллельного выполнения.

Существуют различные виды взаимодействия (синхронизации) процессов, в том числе: взаимодействие посредством общей памяти; - посредством передачи сообщения различных видов.

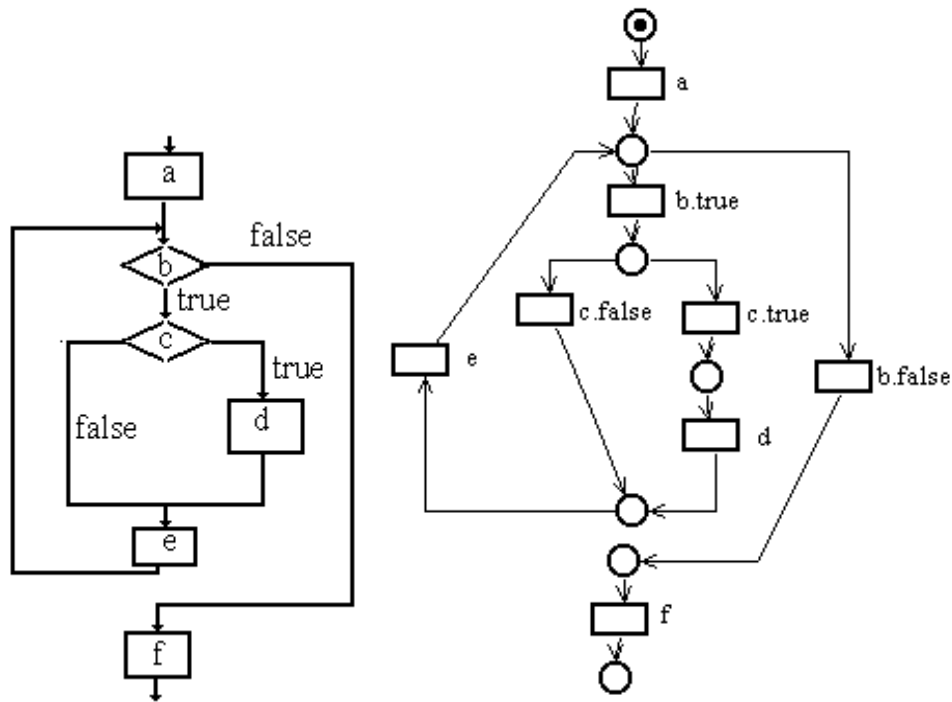


Рисунок 19 – Моделирование работы алгоритма с помощью сети Петри

Таким образом, для моделирования сетями Петри параллельных систем процессов, помимо последовательных процессов, необходимо уметь моделировать различные механизмы взаимодействия (синхронизации) процессов.

Имитационное моделирование вычислительных систем

Пример реализации сети Петри приведен в справочном руководстве Anylogic: Справка>Примеры моделей>Petri Nets.

Пример процесса моделирования алгоритма поиска элемента в бинарном дереве с помощью диаграммы действий приведен в приложении Б.

4.3 Программа и методика выполнения работы

1. Создать аналитическую модель заданного по варианту процесса в форме сети Петри.
2. Создать имитационную модель заданного по варианту процесса в среде Anylogic на основе сети Петри.
3. Создать имитационную модель заданного по варианту процесса в среде Anylogic с использованием диаграммы действий на основе схемы алгоритма. Организовать визуализацию работы алгоритма.

Варианты заданий

Примерный перечень возможных для моделирования алгоритмов приведен в таблице 5. По желанию студента и согласованию с преподавателем может быть выбран любой другой алгоритм.

Таблица 5 – Варианты заданий

№ варианта	Описание задания
1	Алгоритм бинарного поиска
2	Алгоритм линейного поиска
3	Алгоритм поиска методом золотого сечения
4	Алгоритм Дейкстры (кратчайший путь на графе)
5	Алгоритм поиска на графе в ширину
6	Алгоритм поиска на графе в глубину
7	Алгоритм поиска подстроки в строке
8	Алгоритм сортировки выбором
9	Алгоритм сортировки вставками
10	Алгоритм сортировки слиянием
11	Алгоритм сортировки пузырьком
12	Алгоритм быстрой сортировки
13	Рекурсивный алгоритм: возведение в степень
14	Рекурсивный алгоритм: вычисление факториала
15	Рекурсивный алгоритм: Ханойская башня
16	Алгоритм тасования Фишера – Йетса (Кнута)
17	Алгоритм построения быстрой оболочки
18	Алгоритм Грэхема (построение оболочки)
19	Алгоритм Джарвиса (построение оболочки)
20	Алгоритм Чана (построение оболочки)

4.4 Описание лабораторной установки

При выполнении лабораторной работы используется компьютер с установленным программным пакетом Anylogic.

4.5 Результаты экспериментальных исследований

При анализе результатов работы системы необходимо проанализировать полное время обслуживания заявки. Оценивается как гистограмма распределения времени обслуживания заявок, так и усредненное значение времени обслуживания для каждого прогона модели.

4.6 Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- постановку задачи и описание заданного процесса;

- граф полученной сети Петри с описанием;
- имитационную модель на основе сети Петри;
- имитационную модель в виде диаграммы действий с описанием;
- результаты выполнения моделей;
- выводы по работе.

4.7 Контрольные вопросы

1. Какими достоинствами и недостатками обладает имитационное моделирование по сравнению с другими методами моделирования?
2. Что такое сеть Петри и для моделирования каких систем она предназначена?
3. Элементы сети Петри. Графическое описание.
4. Элементы сети Петри. Аналитическое описание.
5. Какими элементами сети Петри моделируются в задачах условия, действия, события, состояния?
6. Каким образом сеть Петри может быть смоделирована в Anylogic?
7. Состав и назначение библиотеки диаграмм действий Anylogic.
8. Каким образом запускается диаграмма действий, как визуализировать ее работу?
9. Опишите классы, входящие в построенную модель.
10. Какие события определены в модели, какие функции с ними связаны?

4.8 Библиографический список рекомендуемой литературы

1. Салмина Н.Ю. Имитационное моделирование [Электронный ресурс]: учебное пособие/ Салмина Н.Ю.— Электрон. текстовые данные.— Томск: Эль Контент, Томский государственный университет систем управления и радиоэлектроники, 2012.— 90 с.— Режим доступа: <http://www.iprbookshop.ru/13930>.— ЭБС «IPRbooks», по паролю.
2. Замятина О.М. Моделирование сетей [Электронный ресурс]: учебное пособие/ Замятина О.М.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2012.— 160 с.— Режим доступа: <http://www.iprbookshop.ru/34683>.— ЭБС «IPRbooks», по паролю.
3. Шелухин О.И. Моделирование информационных систем [Электронный ресурс]: учебное пособие/ Шелухин О.И.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 536 с.— Режим доступа: <http://www.iprbookshop.ru/12002>.— ЭБС «IPRbooks», по паролю.

ЛАБОРАТОРНАЯ РАБОТА № 5. ИССЛЕДОВАНИЕ СЕТЕВОГО ПОДХОДА К МОДЕЛИРОВАНИЮ

5.1 Цель работы

Исследование сетевого подхода к моделированию систем, а также технологии многоподходового моделирования в среде AnyLogic.

Трудоемкость лабораторной работы: 9 ч (6 ч – аудиторных, 3 ч – самостоятельная работа студента).

5.2 Краткие теоретические сведения

СеМО с однородным потоком заявок

Для описания линейных разомкнутых и замкнутых однородных экспоненциальных СеМО необходимо задать следующие **параметры**:

- число узлов в сети n ;
- число обслуживающих приборов в узлах сети $K_1, \dots, K_n$;
- матрицу вероятностей передач $P = [p_{ij} \mid i, j = 0, 1, \dots, n]$;
- интенсивность λ_0 источника заявок, поступающих в РСеМО, или число заявок M , циркулирующих в ЗСеМО;
- средние длительности обслуживания заявок в узлах сети $b_1, \dots, b_n$.

Условие отсутствия перегрузок в разомкнутой СеМО предполагает отсутствие перегрузок в каждом из узлов сети. В замкнутой СеМО перегрузки не возникают.

Характеристики СеМО делятся на узловые и сетевые.

Состав узловых характеристик СеМО, работающей в стационарном режиме, такой же, как и для СМО.

На основе узловых характеристик рассчитываются средние значения сетевых характеристик СеМО:

- суммарная нагрузка Y и загрузка R ;
- среднее суммарное число заявок L во всех очередях сети;
- среднее суммарное число заявок M в разомкнутой сети (во всех узлах);
- среднее время ожидания W и пребывания заявок U в сети;
- $\alpha_j = \lambda_j / \lambda_0$ – коэффициент передачи для узла j , показывающий среднее число попаданий заявки в узел j за время ее нахождения в сети;
- производительность λ_0 замкнутой СеМО.

Для неоднородной СеМО перечисленные характеристики определяются как для каждого класса в отдельности, так и для объединенного (суммарного) потока заявок.

Линейные разомкнутые однородные экспоненциальные СеМО

Условие отсутствия перегрузок в РСемо:

$$\lambda_0 < \min \left(\frac{K_1}{\alpha_1 b_1}, \frac{K_2}{\alpha_2 b_2}, \dots, \frac{K_n}{\alpha_n b_n} \right)$$

Расчет характеристик базируется на эквивалентном преобразовании сети, позволяющем представить разомкнутую экспоненциальную СеМО в виде совокупности независимых экспоненциальных СМО, и проводится в три этапа:

- 1) расчет интенсивностей потоков заявок в узлах РСемо;
- 2) расчет узловых характеристик:
 - загрузка узла ρ_j ;
 - нагрузка узла u_j ;
 - коэффициент простоя узла η_j ;
 - время ожидания заявок в узле: w_j ;
 - время пребывания заявок в узле u_j ;
 - длина очереди заявок l_j ;
 - число заявок в узле (в очереди и на обслуживании в приборе): m_j .

Линейные замкнутые однородные экспоненциальные СеМО

В замкнутых СеМО всегда существует установившийся режим.

Расчет характеристик функционирования замкнутых СеМО с одноканальными узлами проводится с использованием метода средних значений в три этапа:

- 1) расчет коэффициентов передач в узлах замкнутой;
- 2) расчет характеристик ЗСеМО с использованием рекуррентных соотношений для значений M ;
- 3) расчет остальных узловых и сетевых характеристик.

Имитационная модель СеМО в Anylogic

Разместите источники агентов сети массового обслуживания Source (например, в случае моделирования гипермаркета с множеством магазинов вход один, значит, и источник в модели один) и требуемое количество результирующих выходов Sink.

Для моделирования узлов СеМО следует воспользоваться библиотекой моделирования процессов. Каждый узел СеМО представляет собой СМО, а значит, очередь (блок Queue) и задержку (блок Delay). Для упрощения схемы СеМО элементы целесообразно объединить в блок. Для этого в меню ПКМ на выбранных блоках (моделирующих отдельную СМО) надо выбрать «Создать блок диаграммы процесса».

Для моделирования переходов заявки из узла в узел (например, клиента гипермаркета – из магазина в магазин) созданные СМО соединяются через блок SelectOutput (если 2 варианта перехода) или SelectOutput5 (если 5 вариантов переходов). При ином количестве вариантов переходов применяются комбинации

этих элементов. По умолчанию переход по какой-либо из ветвей блока SelectOutput вероятностный.

Следует также учесть, что из узла, моделирующего СМО, заявка может перейти как в другой узел, так и на выход из сети. Также надо помнить, что различные узлы сети могут содержать разное количество приборов, очередей, иметь разные параметры. Например, при моделировании гипермаркета, могут быть узлы без очередей (только с одним блоком Delay), имитирующие лестницы, эскалаторы и т.п.). Важно отследить, чтобы все агенты «нашли выход».

Пример возможной схемы СеМО и ее структурных элементов в Anylogic приведены на рисунках 22, 23.

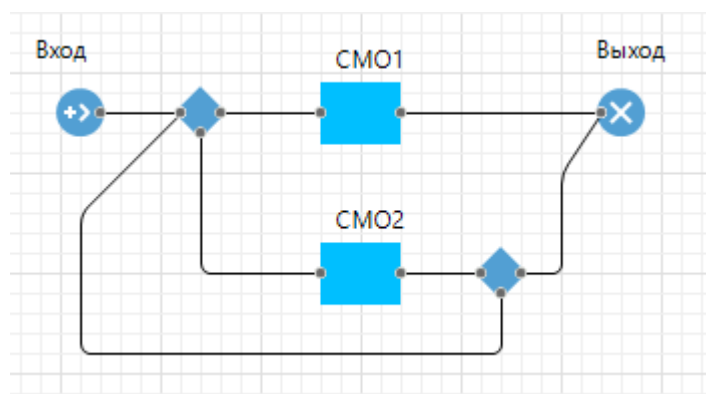


Рисунок 22 – Структурная схема СеМО в Anylogic

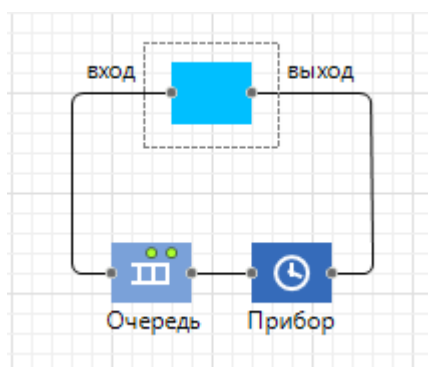


Рисунок 23 – Структурная схема узла СеМО в Anylogic

5.3 Программа и методика выполнения работы

1. Нарисовать граф сетевой модели вычислительной системы, заданной по варианту (Таблицы 6, 7) известны следующие параметры:

- 1) матрица P вероятностей передач (Таблица 7);
- 2) интенсивность поступления заявок в сеть (Таблица 6);
- 3) число обслуживающих приборов в узлах 1, 2, 3 (Таблица 6);
- 4) средние длительности обслуживания заявок в узлах 1, 2, 3 (Таблица 6).

2. Для заданной модели проверить аналитически, существует ли перегрузка сети. Если сеть перегружена, определить максимально допустимое значение интенсивности потока заявок в сеть, при котором в сети будут отсутствовать

перегрузки. Определить, в каком из узлов происходит перегрузка и выдать рекомендации об изменении количества приборов в этом узле при условии заданной по варианту интенсивности потока заявок в сеть.

3. Построить имитационную модель сети. В модели организовать сбор статистики для подтверждения аналитических расчетов из п.2. Организовать эксперимент с подбором параметров (количества приборов в СМО, интенсивности входящего потока заявок).

Таблица 6 – Варианты заданий

Вариант	P	λ_0, c^{-1}	K_1	K_2	K_3	b_1, c	b_2, c	b_3, c
1	P_1	0,1	1	2	3	2	6	4
2	P_2	0,2	2	4	5	4	5	3
3	P_3	0,4	3	1	2	3	1	2
4	P_4	0,2	4	3	4	1	9	5
5	P_5	0,3	5	5	1	3	2	4
6	P_6	0,1	1	4	3	2	4	6
7	P_7	0,4	2	2	4	1	2,5	3
8	P_8	0,2	3	5	2	3	6	4
9	P_9	0,1	4	3	1	4	3	5
10	P_{10}	0,5	5	1	5	0,5	6	4
11	P_{11}	0,3	1	5	3	1	5	3
12	P_{12}	0,2	2	3	1	5	2	0,6
13	P_{13}	0,1	3	1	5	3	2	5
14	P_{14}	0,3	4	4	2	2	7	4
15	P_{15}	0,5	5	2	4	5	0,6	2

Таблица 7 – Матрицы вероятностей передач

	P1				P2				P3			
Узел	0	1	2	3	0	1	2	3	0	1	2	3
0	0	0.5	0.5	0	0	0.8	0.2	0	0	0.5	0.5	0
1	0	0.95	0	0.05	0	0.8	0	0.2	0	0.5	0	0.5
2	0	0	0	1	0	0	0	1	0	0	0	1
3	1	0	0	0	0.5	0	0.5	0	0.2	0	0.8	0

5.4 Описание лабораторной установки

При выполнении лабораторной работы используется компьютер с установленным программным пакетом AnyLogic. Структура программного пакета и реализуемые им функции приведены на официальном сайте программного продукта <http://www.anylogic.ru/anylogic/help/>.

5.5 Результаты экспериментальных исследований

Привести аналитические расчеты и статистические данные из имитационной модели.

5.6 Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- аналитические расчеты в соответствии с программой выполнения работы;
- результаты экспериментальных исследований на имитационной модели;
- выводы по работе.

5.7 Контрольные вопросы

1. Что такое сеть массового обслуживания, из каких компонентов она состоит?
2. Параметры и характеристики СеМО.
3. В чем смысл эксперимента в программе AnyLogic? Какие типы экспериментов поддерживаются программой AnyLogic?
4. Как запустить компиляцию модели в программный код на языке Java?
5. Из каких элементов состоит агентная модель? Что является агентом в выполненной работе? Каким образом задается индивидуальное поведение агента?
6. Как установить синхронизацию действий агентов?
7. Опишите классы, входящие в построенную модель.
8. Какие события определены в модели, какие функции с ними связаны?
9. Какие типы сбора статистики поддерживаются программой?
10. Состав и назначение библиотеки статистики Anylogic.

5.8 Библиографический список рекомендуемой литературы

1. Шелухин О.И. Моделирование информационных систем [Электронный ресурс]: учебное пособие/ Шелухин О.И.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 536 с.— Режим доступа: <http://www.iprbookshop.ru/12002>.— ЭБС «IPRbooks», по паролю.
2. Справочное руководство пакета AnyLogic
<http://www.anylogic.ru/anylogic/help/>
3. Куприяшкин, А.Г. Основы моделирования систем [Текст]: учеб. пособие / А.Г. Куприяшкин; Норильский индустр. ин-т. – Норильск: НИИ, 2015. – 135 с. http://www.anylogic.ru/upload/pdf/osnovi_modelirovania_sistem.pdf

ПРИЛОЖЕНИЕ А

(обязательное)

Процесс создания дискретно-событийной модели в пакете моделирования Anylogic

Наиболее полная информация по библиотеке моделирования процессов находится в справочной системе help.anylogic.ru, раздел *Справочные руководства по библиотекам > Библиотека моделирования процессов*, а также *Справка AnyLogic > Основные принципы > Методы моделирования > Дискретно-событийное моделирование*.

Для создания новой модели щелкните мышью по кнопке «Создать проект». Появится диалоговое окно, в котором вы должны будете дать имя файлу вашей модели и выбрать каталог, где он будет храниться.

Рассмотрим рабочее окно AnyLogic. В левой части рабочей области находится панель «Проект». Панель «Проект» обеспечивает легкую навигацию по элементам моделей, открытых в текущий момент времени. Поскольку модель организована иерархически, то она отображается в виде дерева: сама модель образует верхний уровень дерева; 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 правой рабочей области отображается панель «Палитра», а внизу - панель «Свойства». Панель «Палитра» содержит разделенные по категориям элементы, которые могут быть добавлены на диаграмму класса активного объекта или эксперимента. Панель «Свойства» используется для просмотра и изменения свойств выбранного в данный момент элемента (или элементов) модели.

В центре рабочей области AnyLogic открывается графический редактор диаграммы класса активного объекта Main.

Чтобы добавить объект на блок-схему модели, щелкните по объекту в окне палитры Enterprise Library и перетащите его мышью на структурную диаграмму. При этом его свойства будут отображены на панели «Свойства». В этом окне вы можете изменять свойства элемента в соответствии с требованиями вашей модели. Позднее для изменения свойств элемента нужно будет сначала щелчком мыши выделить его на диаграмме или в дереве проекта.

Объекты должны взаимодействовать между собой, поэтому вы должны будете соединять их друг с другом. Можно соединять объекты с помощью мыши, перетаскиванием порта одного объекта на порт другого или с помощью специального средства «Соединитель». Чтобы соединить порты объектов, щелкните мышью по кнопке панели инструментов Соединитель, а затем щелкните мышью поочередно по обоим портам. Для добавления точки изгиба щелкните мышью по кнопке панели инструментов «Редактировать точки».

Модель выполняется в соответствии с набором конфигурационных установок, называемым экспериментом. Вы можете создать несколько эксперимен-

тов и изменять рабочую конфигурацию модели, просто меняя текущий эксперимент модели. Один эксперимент, названный Simulation, создается автоматически. Выберите его щелчком мыши по элементу дерева и измените настройки модели в окне Свойства (рисунок А.1). Окно Свойства имеет вкладки: основные, дополнительные, модельное время, презентация, окно, параметры, описание.

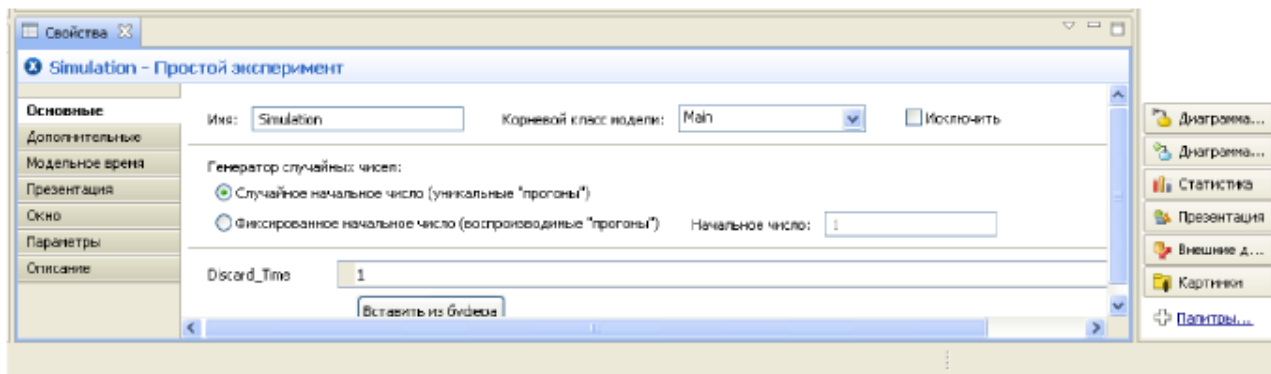


Рисунок А.1 – Настройки модели

На вкладке Основные можно выбрать класс, который будет запущен при запуске модели. По умолчанию в качестве корневого объекта выбран объект класса Main, автоматически создаваемого в каждой модели. Вы можете переименовывать классы модели. Для этого нужно выделить класс щелчком мыши по значку класса в дереве модели и затем изменить его имя в окне Свойства.

На вкладке Модельное время можно:

- задать режим моделирования. В режиме реального времени задается связь модельного времени с физическим, т.е. задается количество единиц модельного времени, выполняемых в одну секунду. Режим реального времени лучше всего подходит для показа анимации. В режиме виртуального времени модель выполняется без привязки к физическому времени - она выполняется так быстро, как это возможно. Данный режим лучше всего подходит, когда требуется моделировать работу системы в течение достаточно длительного периода времени;

- запустить модель так, чтобы она работала бесконечно, но можно и остановить ее в заданный момент времени. Вы можете остановить модель по достижении переменной заданного значения или по выполнении какого-нибудь определенного условия.

Дополнительные свойства эксперимента (вкладка Дополнительные) позволяют управлять выполнением модели. Можно задать действие перед и после запуска модели, а также задать численные методы для прогона и точность получаемых значений.

На вкладке Презентация можно определить вид и скорость выполнения прогона.

Построим с помощью элементов библиотеки Enterprise Library модель простой системы массового обслуживания – модель банковского отделения.

В банковском отделении находятся банкомат и стойки банковских кассиров, которые предназначены для быстрого и эффективного обслуживания посетителей банка. Операции с наличностью клиенты банка производят с помощью

банкомата, а более сложные операции, такие как оплата счетов, - с помощью кассиров.

Необходимо произвести оценку затрат операций и определить, сколько денег тратится на обслуживание одного клиента и какую часть этой суммы составляют накладные расходы на оплату работы персонала банка, а какую на обслуживание посетителей.

1. Создание нового проекта.

Создайте новую модель. Переименуйте класс Main в Model. В свойствах эксперимента Simulation задайте выполнение модели в режиме реального времени с выполнением одной единицы модельного времени в одну секунду. В этой модели под единицей модельного времени мы будем понимать одну минуту работы банковского отделения.

2. Создание блок-схемы.

Создайте блок-схему модели, которая пока будет состоять только из банкомата. Для этого перетащите в окно структуры элементы библиотеки Enterprise Library и соедините их так, как показано на рисунке A.2.

Объект source генерирует заявки (entities) определенного типа через заданный временной интервал. Заявки представляют собой объекты, которые производятся, обрабатываются, обслуживаются или еще каким-нибудь образом подвергаются воздействию моделируемого процесса: это могут быть клиенты в системе обслуживания, детали в модели производства, документы в модели документооборота и др. В нашем примере заявками будут посетители банка, а объект source будет моделировать их приход в банковское отделение.

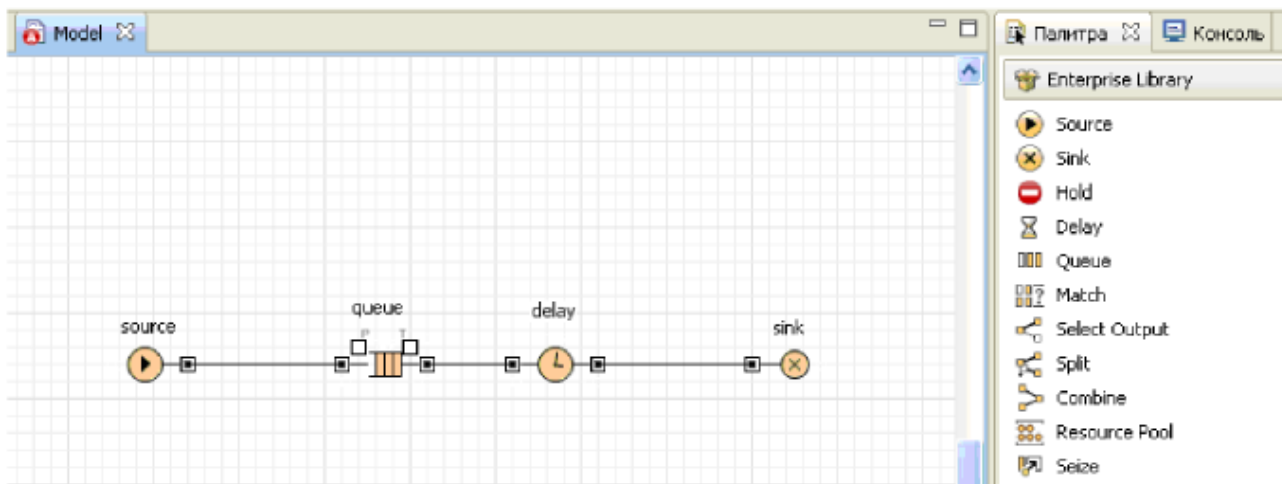


Рисунок A.2 – Структурная диаграмма модели банковского отделения

Объект queue моделирует очередь клиентов, ожидающих обслуживания. Объект delay моделирует задержку. В нашем примере он будет имитировать банкомат, тратящий определенное время на обслуживание клиента.

Объект sink обозначает конец блок-схемы.

3. Запуск модели.

Для каждой модели, созданной в Enterprise Library, автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой вы можете изучить текущее состояние модели, например, длину очереди, количество

обслуженных человек и т.д.

Для запуска модели (рисунок А.3) 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, нарисованную вами для главного класса активного объекта этого эксперимента (Main).

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.

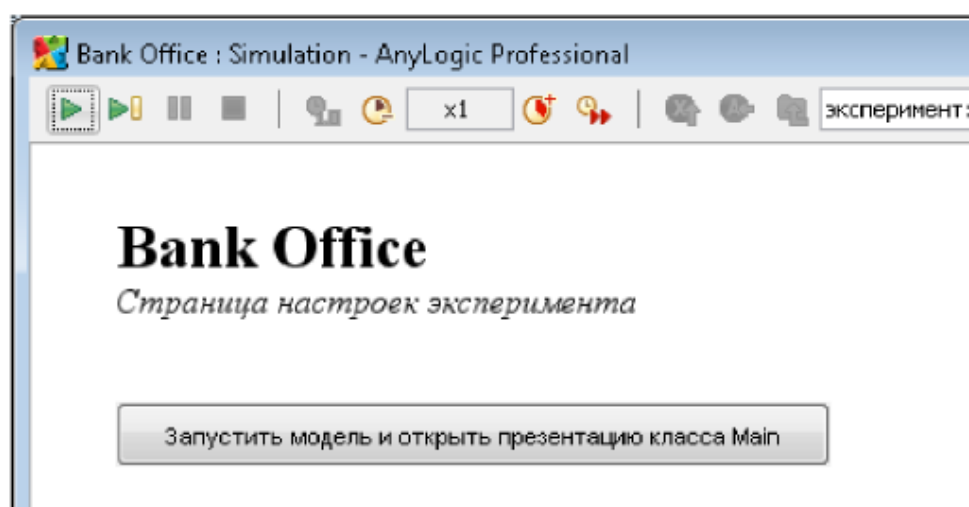


Рисунок А.3 – Запуск модели

На рис. 4 видно, что 4 человека стоят в очереди, а 23 человека покинули очередь (блок *queue*), из них 22 обслужили (блок *sink*), а один еще обслуживается у банкомата (блок *delay*).

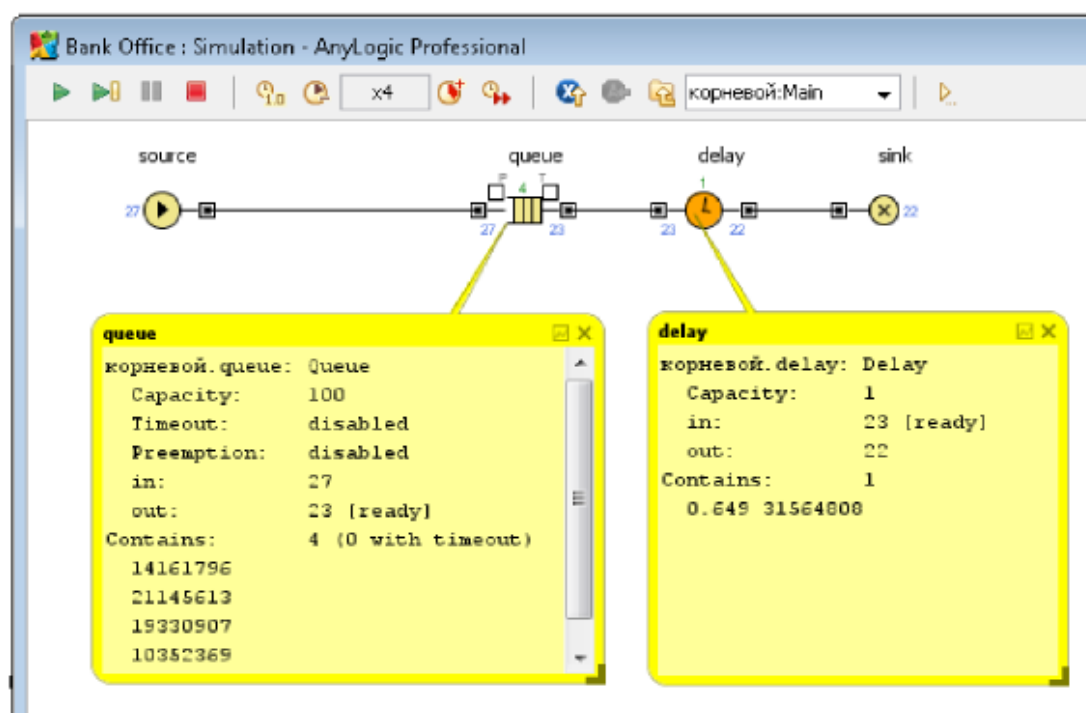


Рисунок А.4 – Процесс выполнения модели

С помощью кнопок панели инструментов «Замедлить» и «Ускорить» можно изменить скорость выполнения модели. Во время выполнения модели можно следить за состоянием любого блока диаграммы процесса с помощью окна инспекта этого объекта. Чтобы открыть окно инспекта, щелкните мышью по значку блока. В окне инспекта будет отображена базовая информация по выделенному блоку, например, для блока Queue будет отображена вместимость очереди, количество заявок, прошедшее через каждый порт объекта, и т.д.

4. Изменение данных модели.

Задайте данные модели, изменяя свойства созданных объектов (рисунок А.5).

В свойстве `interarrivalTime` объекта `source` укажите, как часто в отделение приходят клиенты - `exponential(0.67)`.

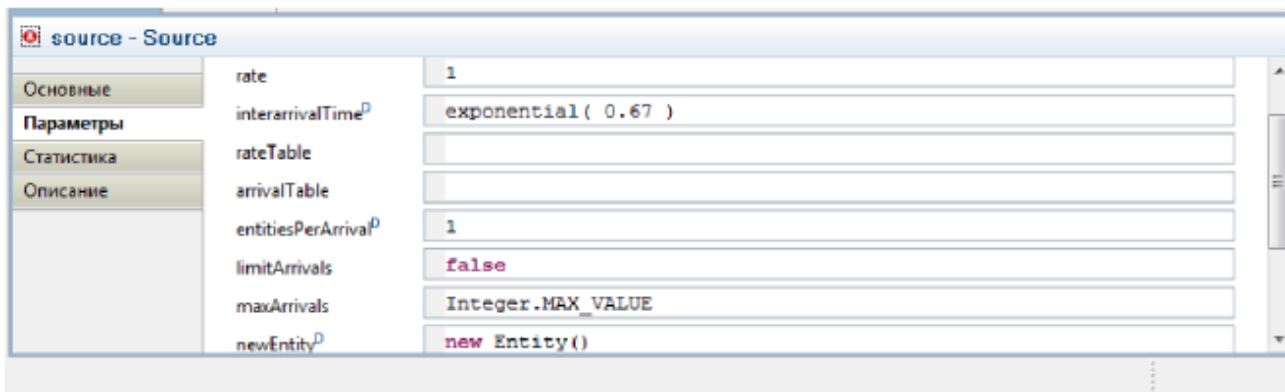


Рисунок А.5 – Свойства объекта `source`

Интервал между приходом клиентов распределен экспоненциально со средним значением, равным 1.5 единицы модельного времени. Заметьте, что аргумент функции `exponential()` равен 0.67, потому что в качестве аргумента задается интенсивность прихода клиентов.

Функция `exponential()` является стандартной функцией генератора случайных чисел AnyLogic. AnyLogic предоставляет функции и других случайных распределений, таких как нормальное, равномерное, треугольное и т.д. За детальным описанием функций и их параметров обращайтесь к руководству пользователя или справочнику классов (см. методы класса `Func`). Для вызова руководства пользователя, справочника классов AnyLogic выберите соответствующие пункты меню Справка.

В свойстве `capacity` объекта `queue` (рисунок А.6) задайте максимальную длину очереди - 15.

В свойстве *delay Time* объекта *delay* (рисунок А.7) задайте время задержки (время обслуживания) – `triangular(0.8, 1, 1.3)`.

Обслуживание одного клиента занимает примерно 1 минуту. Здесь время обслуживания распределено по треугольному закону со средним значением, равным 1 минуте, минимальным - 0.8 и максимальным - 1.3 минуты.

Запустите модель и проанализируйте ее работу.

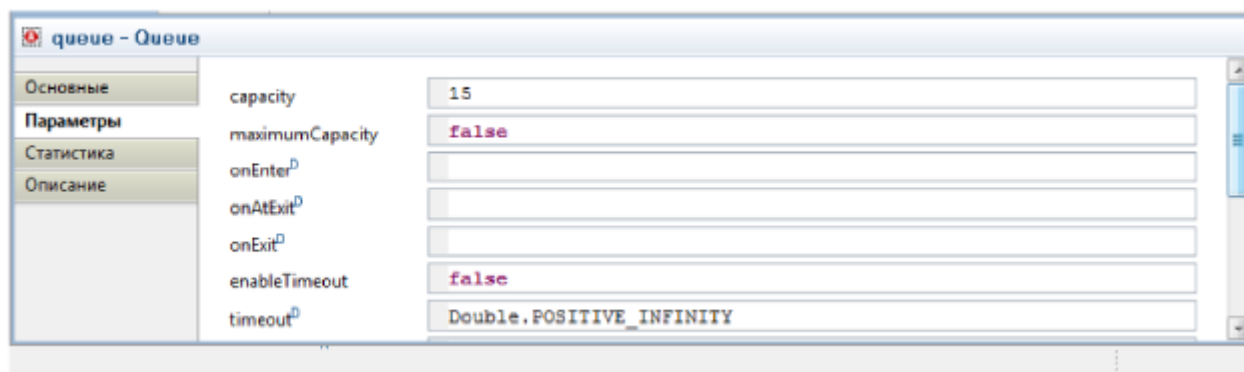


Рисунок А.6 – Свойства объекта queue

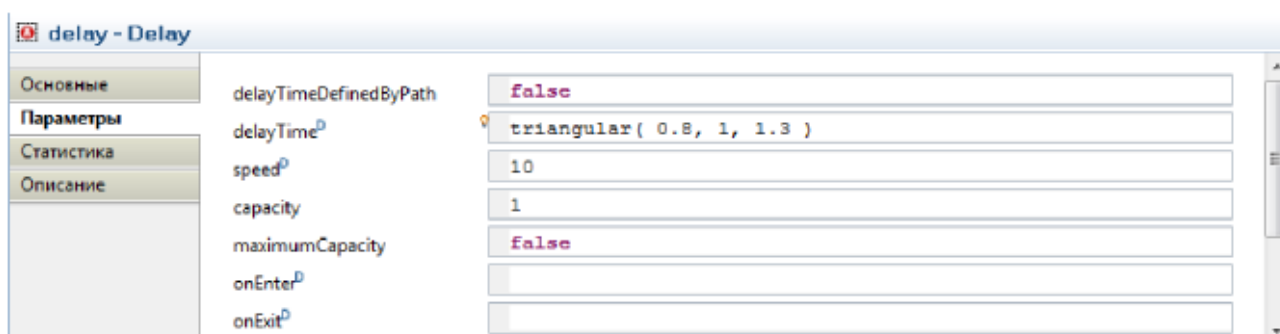


Рисунок 7 – Свойства объекта delay

5. Сбор статистики.

AnyLogic позволяет производить сбор сложной статистики. Для этого нужно лишь включить у объекта режим сбора статистики, поскольку по умолчанию он отключен для повышения скорости выполнения модели.

В системе собирается статистика по длине очереди для блока queue (length) и статистика по коэффициенту использования для блока delay (utilization). Чтобы включить сбор статистики для объекта, установите переключатель «Включить сбор статистики» на вкладке Основные свойств объекта.

Запустите модель и просмотрите в окне инспекта статистику для блоков queue и delay. Можно также просмотреть собранную статистику с помощью диаграмм и графиков или путем вывода числовых значений на анимацию.

6. Моделирование банковских кассиров.

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 $\text{uniform()} < 0.5$, 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;

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 (рисунок А.8).

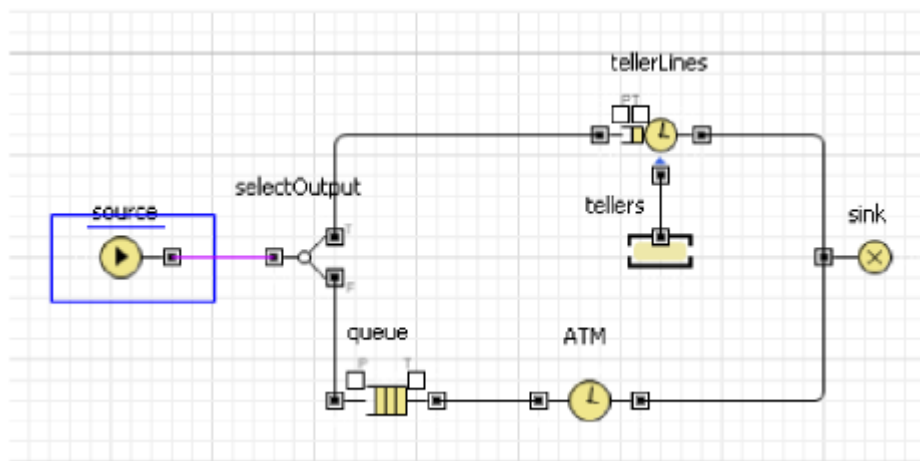


Рисунок А.8 – Структурная диаграмма банковского отделения

Запустите модель и изучите ее поведение.

7. 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 (рисунок А.9);

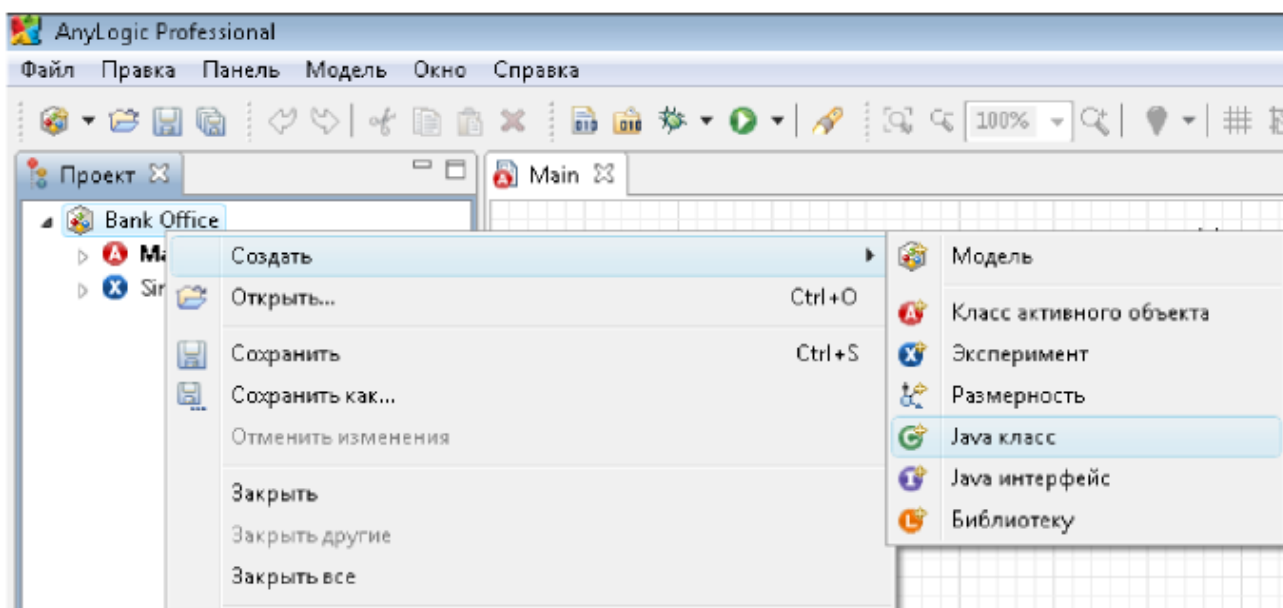


Рисунок А.9 – Создание собственного Java класса

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 (рисунок А.10): выберите из выпадающего списка Базовый класс полное имя данного класса: com.xj.anylogic.libraries.enterprise.Entity;

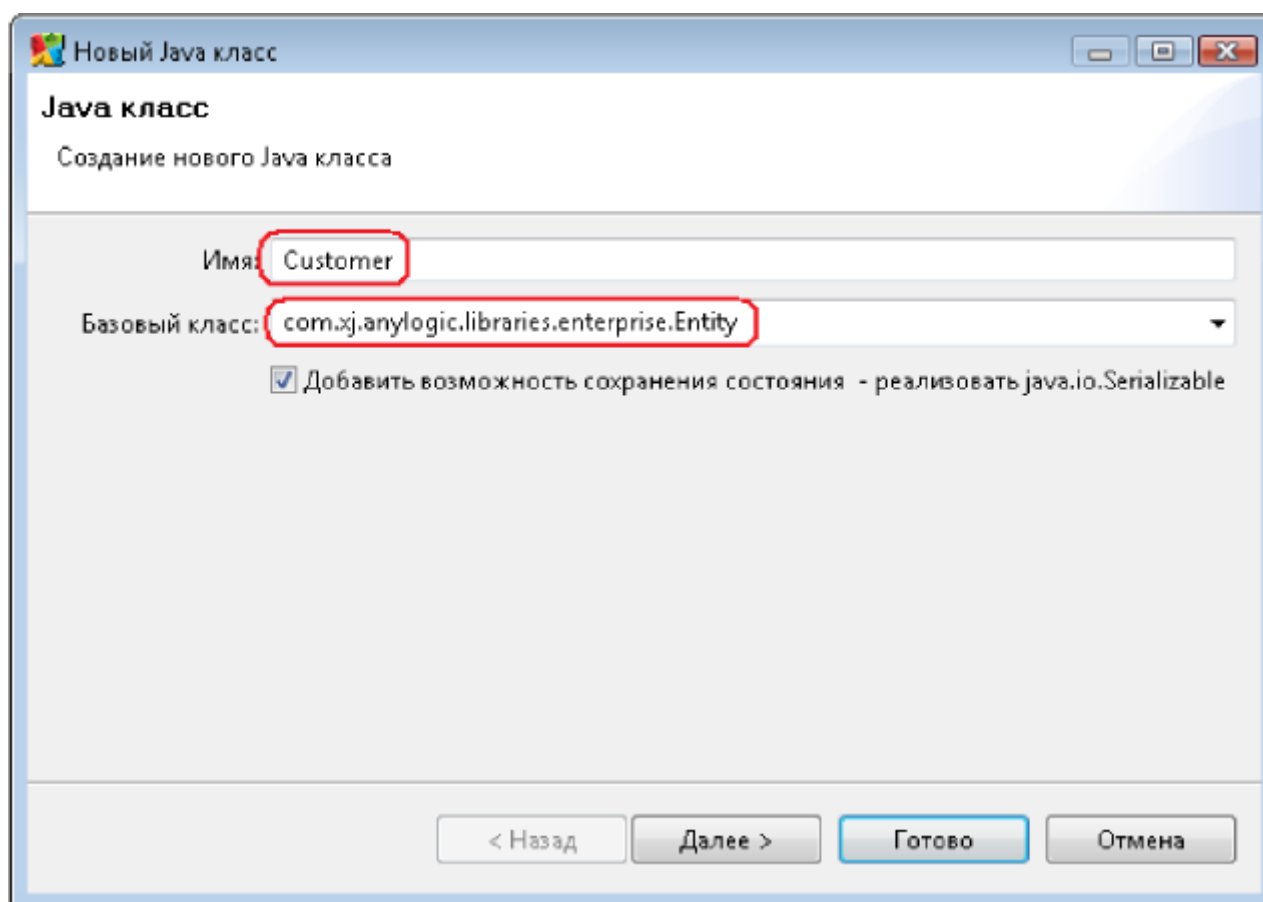


Рисунок А.10 – Создание собственного Java класса

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;

5) 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.

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, среднеквадратичное 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 (рисунок А.11). Измените Имя на waitTimeDistr.

Измените Заголовок на Waiting time distribution.

Сделайте Кол-во интервалов равным 50. Задайте Начальный размер интервала: 0.01;

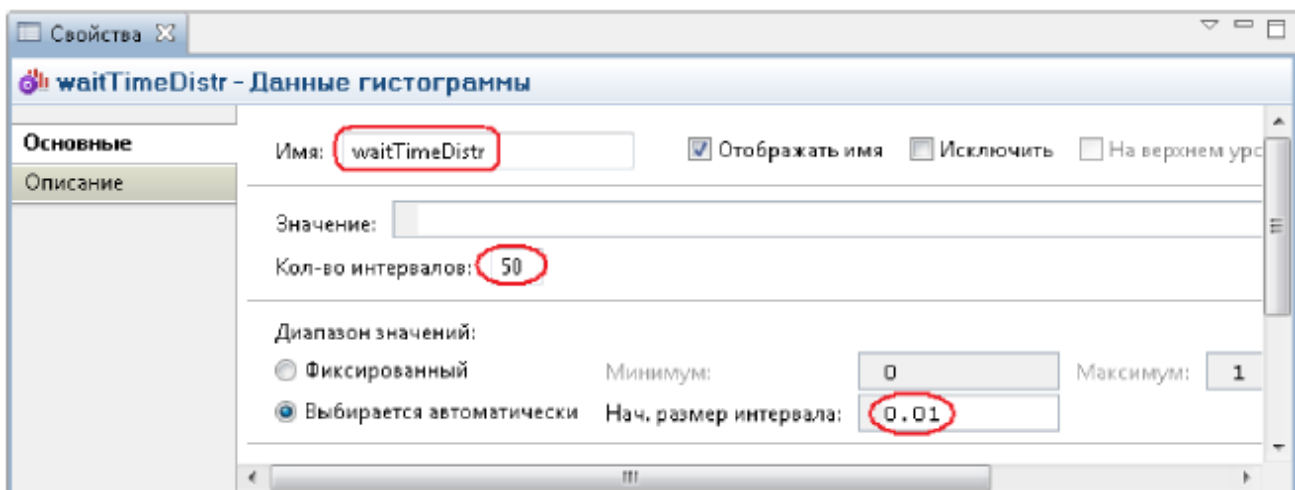


Рисунок А.11 – Свойства элемента сбора данных гистограммы

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 при выходе. 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 `АТМ` (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 графического редактора, 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.

8. Оценка затрат операций.

Enterprise Library предоставляет инструменты для проведения оценки затрат операций. Метод оценки затрат операций (*activity-based costing*, АВС-метод) оценивает процесс и эффективность операций, определяет стоимость обслуживания/производства и указывает возможности для усовершенствования продуктивности и эффективности процесса. С помощью этого метода производится количественная оценка стоимости и производительности операций, эффективности использования ресурсов и стоимости объектов.

Проведем учет затрат операций в нашем примере, чтобы понять, во сколько в среднем обходится обслуживание одного клиента и какие накладные расходы связаны с обслуживанием клиентов, ожидающих своей очереди.

Сначала необходимо написать вспомогательную функцию для пересчета почасовой зарплаты в поминутную.

Создайте математическую функцию (перетащите элемент Функция с палитры Основные на диаграмму активного класса), назовите ее toMinute. На странице свойств задайте тип, аргументы и выражение функции. Тип возвращаемого значения должен быть double. Аргумент функции perHour типа double. Напишите выражение функции (тело функции) - `return perHour / 60;`. Функция должна быть статической, поскольку эта функция не использует значения, специфичные для конкретного экземпляра класса Model. В противном случае нам пришлось бы вызывать ее со ссылкой на конкретный экземпляр класса Model.

Добавьте необходимые параметры в классе сообщений Customer:

- serviceCost будет хранить информацию о том, во сколько компании обходится обслуживание этого клиента. Тип - double. Значение по умолчанию - 0;
- waitCost - затраты на ожидание клиента в очереди. Тип - double. Значение по умолчанию - 0. Заметим, что банк несет издержки, связанные с обслуживанием: клиентов, ожидающих своей очереди;
- existenceCostPerHour задает, во сколько обходится компании пребывание клиента в банке. Тип - double. Значение по умолчанию - `Main.toMinute(1.5)`.

Теперь необходимо добавить код в свойства блоков:

1) блок tellerLines (блок Service) - добавьте код в Действие при выходе:

```
waitTimeDistr.add(time() - entity.startWaiting);
Costwait.add(entity.existenceCostPerHour*(time() - entity.startWaiting));
Costservice.add ((time()-entity.enteredSystem)*(entity.existenceCostPerHour +
Model.busyCostRate));
```

2) блок queuee добавьте код в Действие при выходе:

```
waitTimeDistr.add(time() - entity.startWaiting); Costwait.add(
entity.existenceCostPerHour*(time()- entity.startWaiting));
Costservice.add((time()-entity.enteredSystem)* (entity.existenceCostPerHour +
Model.busyCostRate));
```

Добавьте в модель вспомогательные элементы, собирающие статистику затрат компании, для чего необходимо создать:

1) переменные, задающие заработную плату кассиров. Для этого перетащите элемент Простая переменная с палитры Основные на диаграмму активного класса:

- busyCostRate, тип double, значение по умолчанию toMinute (6.5), глобальный;
- idleCostRate, тип double, значение по умолчанию toMinute (4.0), глобальный.

Мы платим кассиру 6.5 дол. в час, если он был занят обслуживанием клиентов и 4.0 дол., если он был свободен;

2) простую переменную, задающую расходы, которые связаны с работой банкомата:

ATMCostPerUse, тип - double, значение по умолчанию - 0.3, глобальный. Одна операция банкомата обходится компании в 0.30 дол.;

3) переменную timeUpdateCosts. Создайте еще несколько переменных и назовите их так:

- tellersIdleTime;
- tellersBusyTime;
- tellersIdleCost;
- tellersBusyCost.

Эти переменные будут хранить информацию о том, сколько времени кассиры были заняты обслуживанием клиентов и сколько им требуется выплатить за работу;

5) два набора данных гистограммы (палитра Статистика) и назовите их так:

- Costwait;
- Costservice.

Данные наборы данных будут хранить статистику затрат компании на обслуживание клиентов;

6) функцию, которая будет обновлять статистику (палитра Основные). Назовите ее UpdateCosts.

```
double dt = time()-timeUpdateCosts; tellersIdleTime += tellers.idle()*dt;
tellersBusyTime+=(tellers.capacity-tellers.idle() )*dt;
tellersIdleCost += tellersIdleTime*idleCostRate;
tellersBusyCost += tellersBusyTime*busyCostRate;
timeUpdateCosts = time();
return 0 ;
```

Сосчитайте затраты на обслуживание клиента:

1) измените свойства объекта tellers (рисунок А.12). Добавьте следующий код:

```
onSeize - UpdateCosts ();
onRelease - UpdateCosts ();
```

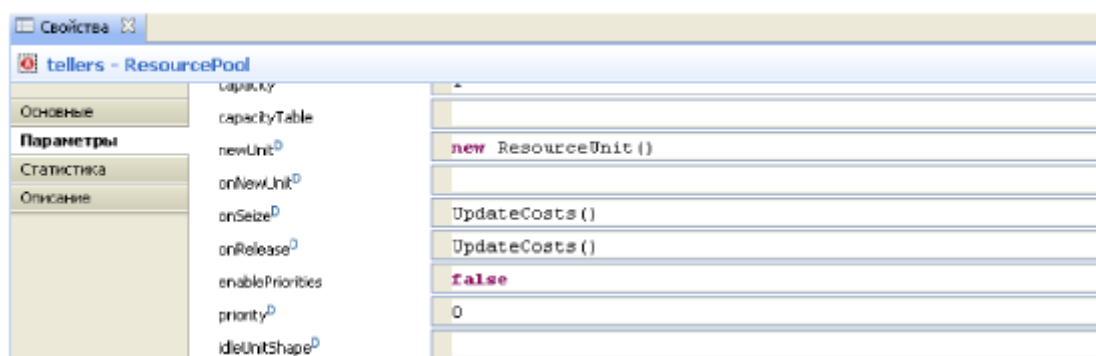


Рисунок А.12 – Свойства объекта tellers

2) измените свойства объекта АТМ . Добавьте строки, чтобы учесть затраты на обслуживание клиента:

```
Costservice.add((time()-entity.enteredSystem)* (entity.existenceCostPerHour
+ Model.ATMCostPerUse));
```

3) запустите модель. Посмотрите статистику затрат компании на обслуживание клиентов. Покажите, во сколько в среднем обходится обслуживание одного клиента и какие накладные расходы связаны с обслуживанием клиентов, ожидающих своей очереди.

ПРИЛОЖЕНИЕ Б

(обязательное)

Процесс создания диаграммы действий на основе заданного алгоритма в пакете моделирования Anylogic

Задача заключается в том, чтобы произвести моделирование **процесса поиска элемента в бинарном дереве**. Дерево хранит целые числа в диапазоне от 0 до 1020 элементы. Считается, что вновь поступающие элементы подчинены равномерному распределению.

Для моделирования данного процесса произведем следующие действия:

1. Определим переменные, которые будут необходимы для моделирования процесса. Для этого добавим десять целочисленных параметров из палитры свойств (Палитра -> Системная динамика) и устанавливаем свойствам видимость значение ложь (см. рисунок Б.1).

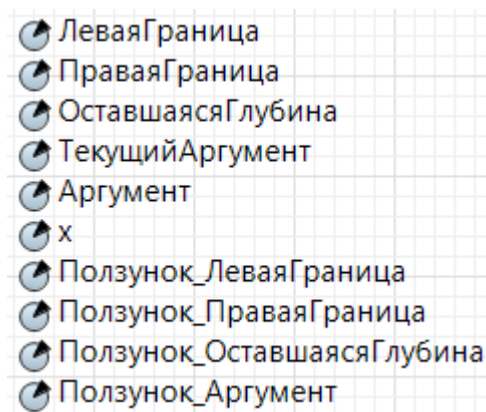


Рисунок Б.1 – Параметры модели

2. Далее добавим четыре набора данных (Палитра -> Статистика) и установим им свойства «Использовать время в качестве значения по оси X» = ЛОЖЬ, хранить до 2000 последних измерений и устанавливаем параметр «Не обновлять данные автоматически» (см. рисунок Б.2).

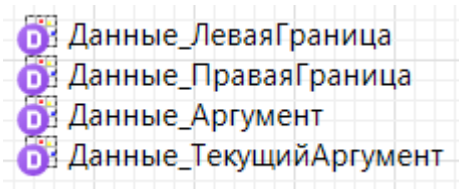


Рисунок Б.2 – Наборы данных

3. Добавим на поле элемент «График» (Палитра -> Статистика). Свяжем наши наборы данных с соответствующим графиком и устанавливаем графику свойства «Не отображать данные автоматически и отображать до 20000 эл. (см. рисунки Б.3, Б.4).

▼ Данные

☐ Значение ☒ Набор данных

Заголовок:

Набор данных:

Стиль маркера:

Толщина линии: pt

Цвет:

☐ Значение ☒ Набор данных

Заголовок:

Набор данных:

Стиль маркера:

Толщина линии: pt

Цвет:

Рисунок Б.3 – Добавление левой и правой границы на график

☐ Значение ☒ Набор данных

Заголовок:

Набор данных:

Стиль маркера:

Толщина линии: pt

Цвет:

☐ Значение ☒ Набор данных

Заголовок:

Набор данных:

Стиль маркера:

Толщина линии: pt

Цвет:

Рисунок Б.4 – Добавление аргумента и текущего аргумента на график

4. Добавим на форму четыре бегунка (Палитра -> Элементы управления) (см. рисунок Б.5) и устанавливаем им следующие свойства:

- Связать с – Бегунок_ЛеваяГраница, мин. знач. 0, макс. знач. 20;
- Связать с – Бегунок_ПраваяГраница, мин. знач. Бегунок_ЛеваяГраница + 1, макс. знач. Бегунок_ЛеваяГраница + 1000;
- Связать с – Бегунок_Аргумент, мин. знач. Бегунок_ЛеваяГраница - 40, макс. знач. Бегунок_ЛеваяГраница + 40;

- d. Связать с – Бегунок_ОставшаясяГлубина, мин. знач. 0, макс. знач. 20;



Рисунок Б.5 – Установка бегунков

5. Добавим кнопку запуска (Палитра->Элементы управления) и текст (Палитра -> Презентация) (см. рисунок Б.6) и установим им следующие свойства:

- Действие - Данные_ТекущийАргумент.reset(); Данные_ЛеваяГраница.reset(); Данные_ПраваяГраница.reset(); Данные_Аргумент.reset(); Поиск();
- Имя - Результат



Рисунок Б.6 – Элементы запуска и получения результата

6. Подготовительный процесс завершен и можно переходить к процессу создания модели. Для этого воспользуемся разделом палитры «Диаграмма действий» и путем перетаскивания элементов на диаграмму повторим модель, представленную на рисунке Б.7. На данной модели белым цветом обозначены блоки вспомогательного кода, производящего отображение данных.

7. Произведем изменение свойств объектов:

Имя – Поиск.

Код - ЛеваяГраница = Бегунок_ЛеваяГраница; ПраваяГраница = Бегунок_ПраваяГраница; ОставшаясяГлубина = Бегунок_ОставшаясяГлубина; Аргумент = Бегунок_Аргумент; ТекущийАргумент = **(int)**(rint(uniform(ЛеваяГраница,ПраваяГраница))); x = -1;

Условие - ОставшаясяГлубина > 0

Код - x = x + 1; ОставшаясяГлубина = ОставшаясяГлубина - 1; Данные_ЛеваяГраница.add(x,ЛеваяГраница);Данные_ПраваяГраница.add(x,ПраваяГраница);Данные_Аргумент.add(x,Аргумент);Данные_ТекущийАргумент.add(x,ТекущийАргумент);

Условие - Аргумент == ТекущийАргумент

Код - Результат.setText("Найден");

Условие - ТекущийАргумент < Аргумент

Код - ПраваяГраница = ТекущийАргумент; ТекущийАргумент = **(int)**rint(uniform(ЛеваяГраница, ПраваяГраница));

Код - $\text{ЛеваяГраница} = \text{ТекущийАргумент} + 1$; $\text{ТекущийАргумент} = (\text{int})\text{rint}(\text{uniform}(\text{ЛеваяГраница}, \text{ПраваяГраница}))$; $\text{ОставшаясяГлубина} = (\text{ЛеваяГраница} == \text{ПраваяГраница}) ? 0 : \text{ОставшаясяГлубина}$;
 Код - $\text{Результат.setText}(\text{"Не найден"})$;

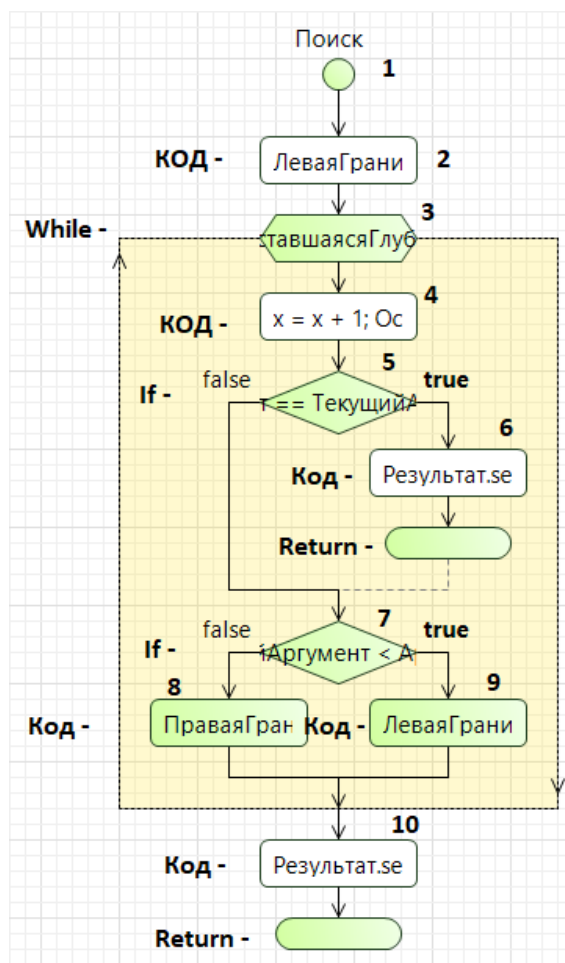


Рисунок Б.7 – Модель алгоритма поиска элемента в бинарном дереве

8. Все готово к запуску процесса моделирования: настраиваем входные данные и наблюдаем полученные результаты (см. рисунок Б.8).

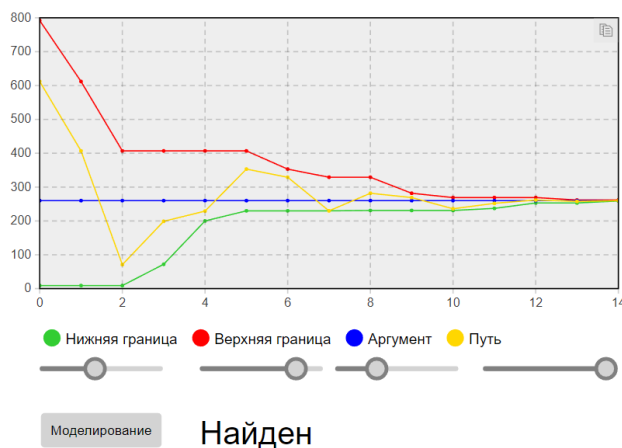


Рисунок Б.8 – Процесс моделирования алгоритма поиска элемента в бинарном дереве