

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский политехнический университет Петра Великого»

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА
МОДЕЛИРОВАНИЕ СИСТЕМЫ, ФОРМАЛИЗОВАННОЙ КАК СИСТЕМА
МАССОВОГО ОБСЛУЖИВАНИЯ
по дисциплине «Архитектура программных систем»

Выполнил студент группы: 3530904/00104:

Почернин В. С.

Преподаватель:

Смирнов Н. Г.

20 ноября 2022 г.

Санкт-Петербург
2022

Содержание

1	Введение	2
2	Первый этап	2
2.1	Бланк задания с заполненными исходными данными к работе	2
2.1.1	Вариант задания	2
2.1.2	Источники	2
2.1.3	Приборы	2
2.1.4	Описание дисциплин постановки и выбора	2
2.1.5	Дисциплины постановки на обслуживание	2
2.1.6	Виды отображения результатов работы программной модели	2
2.2	Формализованная схема ВС	3
2.3	Временная диаграмма своего варианта	3
2.4	Ответы на контрольные вопросы	5
3	Второй этап	6
3.1	Описание	6
3.1.1	Отражение работы программной модели	6
3.2	Законы распределения	8
3.2.1	Равномерный закон распределения	8
3.2.2	Экспоненциальный закон распределения	9
3.3	Ответы на контрольные вопросы	9
4	Третий этап	10
4.1	Описание	10
4.1.1	Определение эффективности ВС	11
4.1.2	Подход к проектированию модели	11
4.2	Пример ВС, соответствующей модели	11
4.2.1	Описание ВС	11
4.2.2	Соответствие модели с ВС	11
4.2.3	Ограничения и требуемые характеристики	13
4.2.4	Ответы на контрольные вопросы	13
5	Четвертый этап	14
5.1	Описание	14
5.2	Определение оптимального числа заявок	14
5.2.1	Пристрелка	15
5.3	Теория для тестирования	15
5.4	Тестирование	16
6	Описание программы	17
6.1	Модульная структура	17
6.2	Демонстрация работы программы	18
7	Вывод	20

1 Введение

Целью настоящей курсовой работы является построение модели системы массового обслуживания, описывающей процессы так, как они проходили бы в действительности. Модель должна описывать компоненты ВС на некотором уровне детализации, который будет имитировать её структуру и функциональность.

Модель должна давать приближенное описание объекта исследования с целью получения требуемых результатов с определенной точностью и достоверностью, принимая во внимание каждый реальный объект системы, обладающий большой сложностью и множеством состояний, связей и анализируемых характеристик.

Существуют различные типы моделей, однако в настоящей работе будет использоваться имитационная модель - логическо-математическое описание объекта, которое может быть использовано для экспериментирования на компьютере в целях проектирования, анализа и оценки функционирования объекта.

По результатам работы будет представлена оптимальная конфигурация исследуемой ВС, полученная на основе анализа экспериментов, поставленных над моделью.

2 Первый этап

2.1 Бланк задания с заполненными исходными данными к работе

2.1.1 Вариант задания

ИБ-ИЗ2-ПЗ1-Д10З1-Д10ОЗ-Д2П2-Д2Б2-ОР1-ОД1

2.1.2 Источники

- ИБ - бесконечный источник.
- ИЗ2 - равномерный закон распределения.

2.1.3 Приборы

- ПЗ1 - экспоненциальный закон распределения времени обслуживания.

2.1.4 Описание дисциплин постановки и выбора

- Д10З1 - относительный приоритет на обслуживание - запись в буфер, если есть место по кольцу.
- Д10ОЗ - дисциплина отказа - самая старая в буфере.

2.1.5 Дисциплины постановки на обслуживание

- Д2Б2 - принцип выбора заявки из буфера LIFO.
- Д2П2 - выбор прибора по кольцу.

2.1.6 Виды отображения результатов работы программной модели

- ОД1 - отображение динамики функционирования модели: календарь событий, буфер и текущее состояние.
- ОР1 - отображение результатов: сводная таблица результатов.

2.2 Формализованная схема ВС

Для освоения функционирования СМО, рассмотрим формализованную схему её работы:

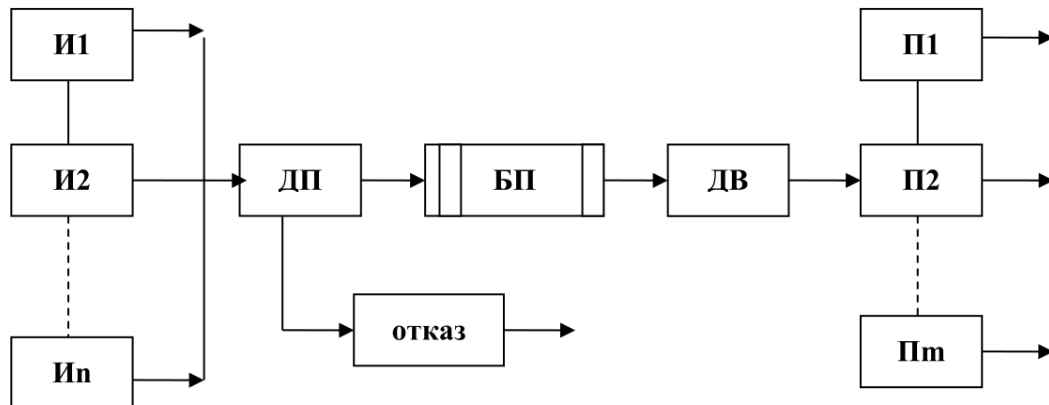


Рис. 1: Формализованная схема ВС

- $И_i$ ($i = 1..n$) представляют собой источники заявок, которые генерируют заявки. Все n источников вместе образуют входной поток заявок.
- Заявки, сгенерированные в источниках попадают на ДП - диспетчер постановки заявок в очередь. Он организует отказ или выбивание заявки из БП, если в буфере не осталось свободных мест, либо же отправляет заявку на обслуживание или в буферную память в случае отсутствия свободных приборов.
- БП - буферная память (место для хранения очереди заявок), в которой хранятся заявки от различных источников.
- ДВ - диспетчер выбора заявок из очереди отправляет заявку из буфера на свободный прибор.
- П - приборы, которые обслуживают заявки и создают тем самым выходной поток заявок после обслуживания.

На основе этого, мы можем проследить путь заявки, вошедшей в модель системы:

- 1) Постановка заявки в буфер.
- 2) Отказ или выбивание (удаление) заявки из переполненного буфера.
- 3) Выбор заявки из БП на обслуживание.
- 4) Поиск свободного прибора.
- 5) Обслуживание заявки прибором.
- 6) Выход заявки из СМО.

При этом, вся логика прохождения заявок по системе определяется диспетчерами ДП и ДВ, когда источники и приборы их только генерируют и обслуживают соответственно.

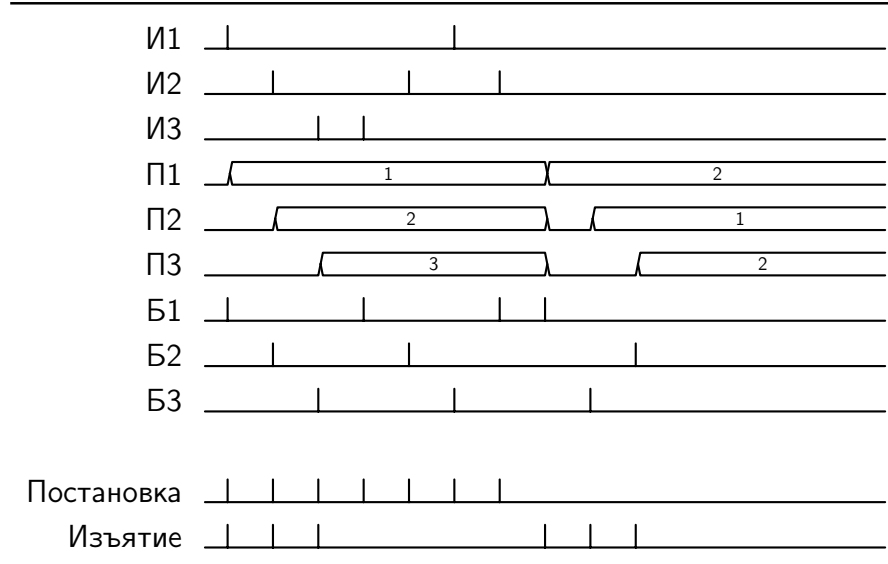
2.3 Временная диаграмма своего варианта

Рассмотрим временную диаграмму функционирования системы, на которой покажем

- Моменты постановки заявок на приборы.

- Заполнение буферной памяти по заданной дисциплине постановки заявки в буфер.
- Отказ заявке или выбивание её при отсутствии свободных мест в БП.
- Функционирование дисциплины выбора заявок из буфера и дисциплины выбора приборов.

Пусть наша система будет состоять из 3 источников (И1, И2, И3), 3 позиций в буфере (Б1, Б2, Б3) и 3 приборов (П1, П2, П3). Получим следующую диаграмму:



- Вначале мы видим, как 3 заявки поочередно создаются на источниках с первого по третий и, проходя через буфер (заходя туда по кольцу), моментально оказываются на приборах (куда они также попадают по кольцу).
- Затем, опять по правилу кольца, заявки с источников 3, 2 и 1 попадают на буферы 1, 2 и 3, в то время, как первые 3 заявки продолжают обрабатываться приборами, следовательно, у нас происходит заполнение буферной памяти.
- Внезапно, источник И2 генерирует еще одну заявку, но в буфере уже заполнены места, поэтому, приходится применять дисциплину отказа - самая старая в буфере. Заявка в буфере Б1 от источника И3, являясь самой старой, выбивается из буфера новоиспеченной заявкой от И2.
- Далее, по стечению обстоятельств, все три прибора одновременно заканчивают работу над своими заявками, благодаря чему мы можем увидеть работу дисциплин выбора заявок из буфера и выбора приборов. Так, как мы имеем выбор из буфера LIFO, сначала пойдет заявка с Б1, затем с Б3 и только потом Б2. Приборы же, продолжая выбираться по кольцу, возьмут заявки в порядке П1, П2, П3, так как последним прибором, взявшим заявку был П3.

2.4 Ответы на контрольные вопросы

- 1) Назовите типы источников, опишите принципы их работы, различия между ними.

Ответ: Источники бывают двух типов: бесконечные и конечные.

Бесконечные источники генерируют заявки, после чего определяют (по определенному закону) интервал для генерации следующей заявки. Заявка попадает в систему в момент генерации и проходит по ней свой индивидуальный путь.

В конечных источниках в определенный момент генерируется и отправляется в систему пакет заявок (из конечного числа заявок). Каждая заявка проходит свой индивидуальный путь по системе. Следующий пакет от такого источника генерируется тогда, когда последняя заявка из предыдущего пакета удаляется из системы (в результате обслуживания или отказа). Момент генерации определяется случайным интервалом между пакетами и новый пакет состоит из того же количества заявок.

- 2) Можно ли сказать, что бесконечный источник есть частный случай конечного?

Ответ: Как мне кажется, мы не можем назвать бесконечный источник частным случаем конечного. Действительно, мы можем считать одну заявку пакетом размера 1, однако случае конечного источника мы имеем четкое условие: "Момент генерации следующего пакета определяется событием, когда последняя заявка этого пакета удаляется из системы". В случае же бесконечного источника следующая заявка может сгенерироваться в любой момент времени, в том числе тогда, когда предыдущая заявка еще находится в системе.

- 3) Опишите два принципа построения моделирующего алгоритма, их преимущества и недостатки.

Ответ: Существуют два подхода к построению моделирующего алгоритма ВС: подход Дельта-Т и подход особых событий.

Подход Дельта-Т является универсальным методом построения моделирующего алгоритма, в котором состояние объекта проверяется через фиксированный интервал времени. Каждый момент времени $t_i = t_{i-1} + \Delta t_{i-1}$ мы получаем приближенные значения характеристик исследуемого объекта. Сам промежуток Δt должен быть настолько мал, чтобы не пропустить событие в моделирующей системе, которое должно быть учтено при выбранной детальности моделирования. Метод удобен тем, что он является универсальным, однако, есть и недостаток: при его использовании постоянно проверяется состояние объектов моделирования, не изменяющихся при особо малых Δt , что делает метод неэффективным.

Подход особых событий мы руководствуемся тем, что интервалы времени, в которых состояние не меняется не представляют для нас интереса. Только значимые (изменяющие состояние) переходы системы имеют для нас значение. Они определяются особыми состояниями или событиями, например:

- Поступление заявки в СМО (момент генерации заявки источником).
- Освобождение прибора (готовность прибора взять заявку на обслуживание).
- Окончание процесса моделирования, т. е. момент прекращения генерации заявок источниками.

Достоинством является эффективность данного принципа, из-за чего именно он используется в настоящей работе. К недостаткам же можно отнести сопутствующую сложность отслеживания вышеприведенных событий.

- 4) Опишите дисциплины буферизации и постановки заявки на обслуживание, заданные в вашем варианте.

Ответ: в моем варианте дисциплинами буферизации являются Д1031 - запись в буфер, если есть место по кольцу и Д1003 - дисциплина отказа - самая старая в буфере. Дисциплинами постановки заявки на обслуживание являются Д2Б2 - принцип выбора заявки из буфера LIFO и Д2П2 - выбор прибора по кольцу. Рассмотрим каждую из них.

- Д1031 - запись в буфер, если есть место по кольцу. При такой дисциплине поиск свободного места в буфере осуществляется, начиная с номера места, следующего за последним занятым. В случае, если указатель, который ищет свободное место его не найдет - начнет действовать дисциплина, организующая отказ или выбивание заявки из БП.
- Д1003 - дисциплина отказа - самая старая в буфере. Эта дисциплина рассматривает только время прихода заявок в систему (момент генерации заявок источником). Заявка, раньше других вставшая в буфер получает отказ, уходит из системы и на её место встает пришедшая заявка.
- Д2Б2 - принцип выбора заявки из буфера LIFO (последним пришел - первым обслужен). В этом случае раньше других будет выбрана из буфера та заявка, которая пришла последней.
- Д2П2 - выбор прибора по кольцу. Поиск свободных приборов здесь каждый раз начинается с указателя и заявка встает на обслуживание в первый из найденных приборов.

5) Назовите некоторые варианты (комбинации) значений входных параметров, при которых на представленной временной диаграмме могут появиться отказы из БП и будут хорошо проиллюстрированы дисциплины выбора приборов и выбора заявок.

Ответ: отказы из БП могут появиться тогда и только тогда, когда любой из источников генерирует заявку в то время, как буфер находится в заполненном состоянии. Например, мы получили 6 заявок от источников, 3 отправились на приборы, 3 в буфер (который после этого заполнился). Генерируется 7-я заявка, которой нет места в буфере, из-за чего применяется дисциплина отказа.

Выбор заявок можно явно увидеть, если, например, заполнить по очереди весь буфер и заметить, что забирается заявка будет в обратном порядке. Это покажет нам, что буфер работает как стек, то есть по правилу LIFO.

Что касается приборов, мы можем заметить их выбор по кольцу в самом начале, когда они начнут по очереди заполняться заявками. Если же затем приборы одновременно закончат обрабатывать заявки, то новыми заявками они начнут заполняться в том же порядке, поскольку указатель кольца начнет с первого прибора (так как последним прибором был последний прибор в кольце).

3 Второй этап

3.1 Описание

В рамках второго этапа необходимо разработать и отладить программу, которая смоделирует поведение текущей ВС.

3.1.1 Отражение работы программной модели

В текущем задании предусмотрены два вида отражения работы программной модели:

- Отображение **динамики** функционирования модели в **пошаговом** режиме.
- Отображение **результатов** работы программной модели в **автоматическом** режиме.

Динамика в пошаговом режиме Здесь нужно отразить изменение состояние ВС при каждом наступлении особого события.

Фиксация осуществляется пошагово, где **шаг** - расстояние по времени от одного особого события до ближайшего другого.

В текущем задании необходимо отразить работу программной модели в пошаговом режиме в виде **календаря событий** (календарь событий, буфер, текущее состояние).

Данная функциональность будет реализована посредством четырех частей:

- Кнопки перехода вперед/назад/на конкретный шаг.
- Текстовое описание текущего шага.
- Таблица календарь событий/текущее состояние:

Календарь событий		Число заявок	Число отказов
Событие	Время		
И1			
...			
Иm			
П1			
...			
Пk			
Конец мод.			

- Событие - наименование описываемого элемента.
- Время - абсолютное время очередного события (например, время, когда источник сгенерирует заявку или когда прибор обработает заявку).
- Число заявок - число созданных (в случае источника) или обслуженных (в случае прибора) заявок на данный момент.
- Число отказов - количество отказов на данный момент.

- Таблица буфер:

Позиция	Время постановки	Источник	Заявка
1			
...			
N			

Результаты в автоматическом режиме Для представления результатов моделирования в автоматическом режиме, в процессе работы программной модели для каждой заявки происходит сбор **статистической информации** для расчета следующих характеристик системы:

- Количество заявок сгенерированных каждым источником.
- Вероятность отказа в обслуживании заявок каждого источника:

$$p = \frac{m}{n}, \quad \text{где}$$

- n - общее количество заявок, сгенерированных источником.
- m - количество заявок этого источника, получивших отказ.

- Среднее время пребывания заявки каждого источника в системе:

$$T_{\text{преб}} = T_{\text{обсл}} + T_{\text{БП}}, \quad \text{где}$$

- $T_{\text{преб}}$ - среднее время пребывания заявки в системе (время ответа на запрос).

- $T_{\text{обсл}}$ - среднее время обслуживания заявки данного источника.
- $T_{\text{БП}}$ - среднее время пребывания заявки в БП или среднее время ожидания заявки каждого источника.
- Дисперсии $T_{\text{обсл}}$ и $T_{\text{БП}}$.
- Коэффициенты использования приборов ($\frac{\text{Суммарное время занятости каждого прибора}}{\text{Общее время реализации}}$).

Соответственно, после завершения процесса моделирования должны быть получены две таблицы результатов:

Характеристики источников ВС							
№ источника	Количество заявок	$p_{\text{отк}}$	$T_{\text{преб}}$	$T_{\text{БП}}$	$T_{\text{обсл}}$	$D_{\text{БП}}$	$D_{\text{обсл}}$
И1							
И1							
...							
Иm							

Характеристики приборов ВС	
№ прибора	Коэффициент использования
П1	
П2	
...	
Пk	

Отдельно стоит отметить, что **окончание процесса генерации заявок** (конец моделирования) происходит в момент **генерации последней заявки**, однако в СМО могут остаться заявки как на приборах, так и в буферной памяти, поэтому процесс **обслуживания заявок** продолжается до момента **выхода из системы последней заявки**.

Общим временем реализации называется время, которым заканчивается обслуживание последней заявки (именно его значение используется при расчете коэффициентов использования приборов).

3.2 Законы распределения

При моделировании ВС подразумевается использование двух законов распределения:

- Равномерный закон распределения времени генерации для бесконечных источников.
- Экспоненциальный закон распределения времени обслуживания для приборов.

Рассмотрим каждый из них.

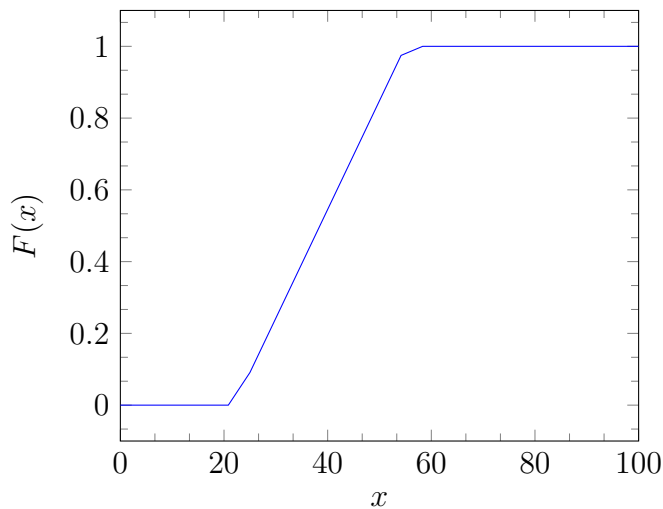
3.2.1 Равномерный закон распределения

Равномерный закон распределения характеризуется функцией распределения вида:

$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1 & x > b \end{cases}$$

Построим график распределения при $a = 22$ и $b = 55$:

Равномерный закон распределения



Соответственно:

$$x = F(x)(b - a) + a \quad (1)$$

Где $F(x)$ будет случайным числом в диапазоне от 0 до 1.

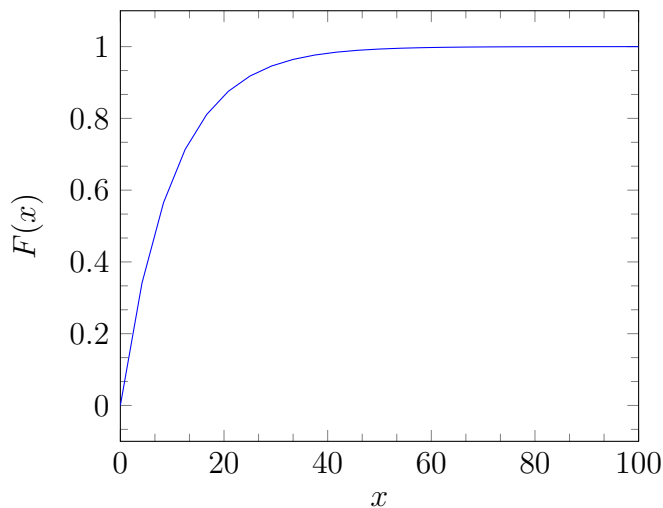
3.2.2 Экспоненциальный закон распределения

Экспоненциальный закон распределения характеризуется функцией распределения вида:

$$F(x) = 1 - e^{-\lambda x}$$

Построим график распределения при $\lambda = 3$:

Экспоненциальный закон распределения



Соответственно:

$$x = -\frac{1}{\lambda} \log_e(1 - F(x)) \quad (2)$$

Где $F(x)$ будет случайным числом в диапазоне от 0 до 1.

3.3 Ответы на контрольные вопросы

- 1) Какие изменения состояние ВС должны фиксировать программа для текущего задания в пошаговом режиме?

Ответ: Мы работаем с методом особых состояний, а значит, что для нас имеют значение только значимые (изменяющие состояние) переходы системы в некоторые моменты времени. Они будут определяться следующими состояниями или событиями:

- Поступление заявки в СМО (момент генерации заявки источником).
- Поступление заявки в БП (либо же срабатывание отказа).
- Переход заявки из буфера в свободный прибор.
- Освобождение прибора (и взятие им заявки из буфера, при наличии таковых).
- Окончание процесса моделирования (момент прекращения генерации заявок источниками).
- Выход из системы последней заявки.

2) Что должно являться шагом при работе программной модели в режиме динамического отражения результатов?

Ответ: При работе программной модели в режиме динамического отражения результатов **шаг** - это интервал модельного времени от одного особого события до другого ближайшего по времени особого события.

3) Из чего складывается время пребывания заявки в системе и как рассчитывать среднее время пребывания заявки в системе?.

Ответ: Время пребывания заявки в системе рассчитывается по следующей формуле:

$$T_{\text{преб}} = T_{\text{обсл}} + T_{\text{БП}}, \quad \text{где}$$

- $T_{\text{преб}}$ - среднее время пребывания заявки в системе (время ответа на запрос).
- $T_{\text{обсл}}$ - среднее время обслуживания заявки данного источника.
- $T_{\text{БП}}$ - среднее время пребывания заявки в БП или среднее время ожидания заявки каждого источника.

Чтобы рассчитать именно **среднее** время - будем для каждого источника суммировать времена заявок и их количество, а затем возьмем среднее арифметическое.

4) Как рассчитать коэффициент использования приборов $K_{\text{исп}}$?

Ответ: Коэффициент использования приборов = $\frac{\text{Суммарное время занятости каждого прибора}}{\text{Общее время реализации}}$.

5) В чем разница между окончанием моделирования и общим временем реализации?

Ответ:

- **Окончание процесса моделирования** (конец моделирования, окончание процесса генерации заявок) происходит в момент **генерации последней заявки**, но в этот момент в СМО еще могут оставаться заявки как на приборах, так и в буферной памяти.
- **Общее время реализации** - это время, которым заканчивается обслуживание последней заявки. То есть момент выхода из системы последней заявки.

6) Какие сведения должны быть выведены на экран по истечении времени реализации?.

Ответ: После истечения времени реализации, на экран должны быть выведены две таблицы результатов:

- Характеристики источников ВС.
- Характеристики приборов ВС.

4 Третий этап

4.1 Описание

На третьем этапе нужно привести пример потенциально-существующей ВС (реальной или гипотетической), которая бы полностью соответствовал бы дисциплинам и законам варианта задания.

Также нужно иметь достаточно четкое представление о моделируемой системе, чтобы написать количественные значения входных параметров.

В итоге, мы хотим получить выходные характеристики и сравнить их со сформулированными требованиями к результатам работы.

4.1.1 Определение эффективности ВС

Определять эффективность нашей ВС будем по следующим выходным характеристикам:

- $p_{\text{отк}}$ - вероятность отказа в обслуживании заявок;
- $T_{\text{БП}}$ - среднее время пребывания заявок в буфере;
- $K_{\text{исп}}$ - коэффициент использования приборов;

Задавая желаемые количественные значения этим трём выходным характеристикам, мы формируем требования к работе создаваемой ВС.

4.1.2 Подход к проектированию модели

Говоря грубо, мы должны определить, что нужно подать на вход нашей модели, чтобы получить подходящие нам выходные характеристики.

Однако, комбинаций входных значений может оказаться бесконечное множество, поэтому на множество входных значений следует наложить ограничения, исходя из особенности конкретно нашей ВС.

Таким образом, получится спроектирована модель архитектуры системы, которая будет работать по заданным требованиям.

4.2 Пример ВС, соответствующей модели

Приведем **пример потенциально-существующей ВС**, которая бы соответствовала всем законам и дисциплинам нашей модели:

4.2.1 Описание ВС

- Мы - отечественная сыродельная компания, которая занимается производством молодых сыров из собственного сырья.
- В нашем распоряжении есть коровы (источники заявок), которые поставляют нам молоко. Это молоко собирается в резервуары и отправляется на заготовку сыра в ёмкости для сырной массы (приборы).
- Если в настоящий момент времени у нас нет свободных емкостей для сыра, то мы отправляем резервуары с молоком на склад (буфер).
- Наша цель - добиться лучшего результата - соотношения потраченных денег (на коров и закваску) к количеству производимого сыра.

4.2.2 Соответствие модели с ВС

Часть системы	Описание закона/дисциплины	Соответствие с реальной ВС
Источники	Бесконечный источник с равномерным законом распределения времени генерации заявок.	Источники в нашей системе - это коровы , которые дают нам молоко. Производство молока у коровы зависит от множества факторов: её самочувствия, настроения, погоды и породы. Поэтому коровы будут давать нам молоко бесконечно с равномерным законом распределения времени между сбором. При этом, мы можем выбрать между тремя породами коров, которые имеют разное время "генерации молока".
Приборы	Экспоненциальный закон распределения времени обслуживания заявок.	Приборами у нас будут емкости для сырных масс , в которых будут производиться наши молодые сыры. Хотя мы и специализируемся на молодых сырах - здесь тоже есть некоторый разброс по времени (очень молодой сыр, умеренно молодой сыр, итд). При этом, чаще всего мы будем производить очень молодой сыр, но иногда будем делать линейку сыров, готовящихся подольше. Этим будет объясняться экспоненциальный закон распределения времени. При производстве сыра, мы можем также выбрать один из трех вариантов: не добавлять закваску вовсе, добавить дешевую закваску или добавить дорогую закваску, что будет менять время изготовления нашего сыра.
Буфер		Буфером у нас будет склад с резервуарами , где будет храниться выдоенное молоко, готовящееся к превращению в сыр.
Постановка в буфер	Запись в буфер, если есть место по кольцу.	Каждый резервуар, где хранится молоко у нас не отличается от другого, и использоваться они должны равномерно . Здесь идеально подойдет распределение мест по кольцу.
Отказ в буфере	Если при постановке в буфер в нем нет мест - из него выбивается самая старая заявка.	Если выбирать - приготовить сыр из более старого или из более нового молока - мы всегда будем выбирать новое , поскольку сыр в таком случае будет получаться крайне нежным. Поэтому, когда у нас появляется свежесвыдоенное молоко, а места для него нет - оно будет занимать место самого старого на данный момент молока.
Выбор из буфера	Выбор заявки по LIFO(стек).	Опять же, выбирая между старым и новым молоком, лучше выбрать новое, потому что это принесет нам больше пользы. Так как мы хотим держать планку качества, то будем пытаться всегда выбирать самое молодое молоко со склада.
Выбор прибора	Выбор прибора по кольцу.	Как и резервуары для молока, емкости для сыра также не отличаются друг от друга и лучше всего использовать их равномерно , в чем нам поможет выбор прибора по кольцу.

4.2.3 Ограничения и требуемые характеристики

Согласно нашей системе, можно выдвинуть следующие ограничения:

- **Вероятность отказа** должна составлять **не более 15%** - мы не хотим, производимое нами молоко уходило впустую
- **Коэффициент использования приборов** должен быть **более 80%** - Чем больше работают приборы - тем больше денег мы сможем заработать. Нельзя, чтобы был простой и мы бы тратились на емкости для сыра впустую.
- **Среднее время пребывания заявок в буфере** Среднее время пребывания заявки (резервуара с молоком) в буфере должно быть не более 5 суток, иначе качество продукции может упасть.

Количество источников (коров)	от 1 до 3
Размер буфера (количество резервуаров)	от 30 до 50
Количество приборов (емкостей для сыра)	от 40 до 60
Скорость работы источников (коров)	$a = 10, b = 12$ - коровы 1го класса $a = 12, b = 15$ - коровы 2го класса $a = 15, b = 24$ - коровы 3го класса
Скорость работы приборов (емкостей с сыром)	$\lambda = 0.001$ - без использования закваски $\lambda = 0.003$ - закваска 2го класса $\lambda = 0.005$ - закваска 1го класса

Примечание

- Чем быстрее корова дает молоко, тем больше её стоимость.
- Чем лучше сыворотка, тем больше её стоимость.
- Единицы времени - часы.
- $\lambda = 0.001$ значит, что в среднем сыр будет созревать за 40 суток.
- $\lambda = 0.003$ значит, что в среднем сыр будет созревать за 15 суток.
- $\lambda = 0.005$ значит, что в среднем сыр будет созревать за 8 суток.

4.2.4 Ответы на контрольные вопросы

1) Опишите элементы СМО для примера своего варианта ВС по следующей схеме:

- Источник - тип и характеристики.
- Прибор - количество и характеристики.
- Буфер - тип и характеристики.
- Заявка - транзакция конкретного размера.

Ответ: указано в таблицах.

2) Соответствуют ли случайные законы и дисциплины постановки и выбора вашей ВС законами и дисциплинами, указанными в строчке задания вашего варианта?

Ответ: да, соответствуют, указано в таблицах.

3) Значениями каких выходных характеристик определяется качество работы системы?

Ответ: Качество работы системы определяется следующими выходными характеристиками:

- $p_{\text{отк}}$ - вероятность отказа в обслуживании заявок;

- $T_{\text{преб}}$ - среднее время пребывания заявок в СМО;
- $T_{\text{БП}}$ - среднее время пребывания заявок в буфере;
- $K_{\text{исп}}$ - коэффициент использования приборов;

4) Составьте требования к работе ВС (значения, которые вы хотите получить для выходных характеристик)

Ответ: составлено выше.

5) Напишите ограничения (область изменения) для всех пяти входных параметров исследуемой ВС.

Ответ: составлено выше в таблице.

5 Четвертый этап

5.1 Описание

Последний этап посвящен исследованию работы модели ВС и выбору её оптимальной конфигурации.

Нам необходимо получить выходные характеристики системы для всевозможных конфигураций входных параметров и отсеять только те конфигурации, выходные характеристики которой будут соответствовать заданным в третьем этапе ограничениям.

В результате мы получим некоторое множество решений исследуемой системы, т.е. некоторое количество конфигураций входных параметров, которые будут соответствовать сформулированным требованиям.

Здесь же необходимо определить количество заявок, которое должно пройти по системе, чтобы выходные характеристики были получены с заданной точностью и достоверной вероятностью. Наконец, из полученного множества решений (вариантов архитектуры ВС) нужно будет выбрать на основе анализа результатов самый оптимальный вариант системы.

5.2 Определение оптимального числа заявок

Для того, чтобы иметь адекватные результаты - нам необходимо выбрать оптимальное количество заявок, которое мы будем использовать в наших конфигурациях.

В рамках задания необходимо обеспечить:

- **Относительную точность:** 10%.
- **Доверительную вероятность:** 0.9.

Для этого нужно определить необходимое и достаточное количество заявок, которое должно быть сгенерировано источником (чем больше заявок - тем больше точностью получения выходных характеристик).

Существует *связь* между:

- Количеством заявок, прошедших через ВС.
- Относительной точностью.
- Доверительной вероятностью.
- Случайной величиной $p(A)$ (вероятностью события A).

$$N = \frac{t_{\alpha}^2(1-p)}{p\delta^2}, \text{ где} \quad (3)$$

- N - количество заявок, пропущенных через систему.
- p - вероятность отказа заявкам в обслуживании.

- $t_\alpha = 1.643$ для $\alpha = 0.9$.
- $\delta = 0.1$ - относительная точность.

У нас имеется две неизвестные величины (N и p). Мы хотим найти N - значит нужно иметь представление о p . Для этого произведем **пристрелку**.

5.2.1 Пристрелка

Алгоритм пристрелки будет следующим:

- 1) Назначим некоторое $N_0 = 100$ и проведем с ним процесс моделирования, получив характеристики (включая p_0).
- 2) Подставим полученные характеристики в (3), вычислив некоторое N_1 .
- 3) Проведем с ним аналогичный процесс моделирования, получим p_1 .
- 4) Сравним p_1 с p_0 :
 - (а) Если $|p_0 - p_1| < 0.1p_0$, то $N = 100$ удовлетворяет заданной точности результатов.
 - (б) Если $|p_0 - p_1| \geq 0.1p_0$, то мы продолжаем процесс моделирования с новым N_2 и так далее до достижения заданной точности результатов.

Этот процесс был проведен для некоторого количества конфигураций.

По результатам работы программы, были получены следующие **результаты**:

- В случае, когда $p \approx 0.3$, нам достаточно $N \approx 600$.
- В случае, когда $p \approx 0.1$, нам достаточно $N \approx 2500$.
- В случае, когда $p < 0.05$, число заявок N может достигать 10000.

Вследствие этого было принято решение поддерживать для всех конфигураций число заявок равное:

$$N = 10000$$

5.3 Теория для тестирования

Итак, для создания нашей конфигурации, мы можем варьировать следующие величины:

- От 1 до 3 источников (коров).
- От 30 до 50 ячеек буфера (резервуаров для молока).
- От 40 до 60 приборов (емкостей для сыра).
- 3 вида коров (коровы 1го класса, коровы 2го класса, коровы 3го класса).
- 3 вида заквасок (без закваски, закваска 2го класса, закваска 1го класса).

Если перемножить все варианты, то мы получим следующее число возможных конфигураций:

$$3 * 21 * 21 * 3 * 3 = 11907$$

Основная цель нашей компании - получение прибыли, поэтому мы будем искать ту конфигурацию, которая будет наиболее выгодна для нас. Это означает, что она должна будет принести нам **наибольшую прибыль за час**.

Мы знаем количество заявок для каждой конфигурации - это $N = 10000$. Перемножив его на $(1 - p_{\text{отк}})$, мы сможем получить **количество произведенных головок сыра**:

$$N_{\text{сыра}} = N_{\text{заявок}} \times (1 - p_{\text{отказа}})$$

В зависимости от того, какую закваску мы будем использовать - будет отличаться время выращивания сыра. Чем дольше выдерживается сыр, тем он дороже, следовательно, мы определим следующую цену сыра, в зависимости от закваски:

Тип закваски	Цена головки сыра Р	Цена закваски Р
Без использования закваски	30 000р.	0р.
Закваска 2го класса	20 000р.	1000 р.
Закваска 1го класса	10 000р.	2000 р.

Все наше оборудование (коровы, резервуары для молока, емкости для сыра) требуют обслуживания. В табличке ниже можно увидеть формулы, по которой будет вычисляться эта стоимость.

Тип оборудования	Цена обслуживания (р/час) S
Коровы 1го класса	$300 * N_{\text{коров}}$
Коровы 2го класса	$200 * N_{\text{коров}}$
Коровы 3го класса	$100 * N_{\text{коров}}$
Резервуар для молока	$5 * N_{\text{резервуаров}}$
Емкость для сыра	$10 * N_{\text{емкостей}}$

Соответственно, мы сможем посчитать наш **доход** в час как:

$$\text{Доход в час} = \frac{N_{\text{сыра}} \times (P_{\text{сыра}} - P_{\text{закваски}})}{T_{\text{симуляции}}}$$

Аналогично, посчитаем **расходы** в час как:

$$\text{Расход в час} = S_{\text{коров}} + S_{\text{резервуаров}} + S_{\text{емкостей}} + S_{\text{заквасок}}$$

Наконец, мы сможем посчитать **прибыль** в час как:

$$\text{Прибыль в час} = \text{Доход в час} - \text{Расход в час}$$

Именно этот параметр мы постараемся максимизировать и выберем 10 лучших (по этому параметру конфигураций).

5.4 Тестирование

Тестирование нашей системы происходило **следующим образом**:

- 1) Для каждой конфигурации было проведено тестирование и получены выходные характеристики, данные были занесены в Excel-таблицу.
- 2) Изначально были получены выходные характеристики для всех 11907 конфигураций.
- 3) После отсеивания заявок по ограничению вероятности отказа (не более 15%) осталось 6148 конфигураций.
- 4) После отсеивания заявок по ограничению коэффициента использования приборов (не менее 80%) осталось 2097 конфигураций.
- 5) После отсеивания заявок по ограничению среднего времени пребывания в буфере (не более 5 суток) осталось 1257 заявок.
- 6) К оставшимся заявкам были применены формулы вычисления нашего критерия эффективности системы (**прибыль в час**) и был произведена сортировка по убыванию этого критерия.

Ниже можно увидеть 10 самых эффективных конфигураций:

N	N ист	N буф	N приб	N заявок	a	b	Лямбда	Доход, р/час	Расход, р/час	Прибыль, р/час
1	2	33	60	10000	10	12	0,003	3530,864	1365	2165,864
2	2	32	60	10000	10	12	0,003	3512,071	1360	2152,071
3	2	39	60	10000	10	12	0,003	3543,956	1395	2148,956
4	2	37	60	10000	10	12	0,003	3521,841	1385	2136,841
5	2	37	59	10000	10	12	0,003	3496,786	1375	2121,786
6	2	31	60	10000	10	12	0,003	3475,582	1355	2120,582
7	2	42	60	10000	10	12	0,003	3527,819	1410	2117,819
8	2	40	59	10000	10	12	0,003	3501,347	1390	2111,347
9	2	30	60	10000	10	12	0,003	3458,064	1350	2108,064
10	2	34	60	10000	10	12	0,003	3473,689	1370	2103,689

Как мы можем заметить, наилучших результатов добились конфигурации, которые использовали:

- 1й класс коров (самые лучшие).
- 2й класс закваски (средняя по скорости и по цене производимого сыра).
- Максимальное количество приборов.
- Среднее количество коров.
- Размер буфера, близкий к нижней границе.

Отдельно стоит отметить, что наихудшая прибыль (среди всех отсортированных конфигураций) составила 369 рублей в час. Это значит, что нам удалось улучшить нашу систему **почти в 6 раз** с помощью произведенной симуляции.

6 Описание программы

6.1 Модульная структура

Разработка программы моделирования производилась на языке **Java** с использованием среды разработки **IntelliJ IDEA**. Графический интерфейс был создан как веб-приложение с использованием фреймворка **Spring**.

Приложение является объектно-ориентированным и содержит следующую структуру:

- 1) Пакет **controller** - хранит в себе контроллеры:
 - (a) **SimulationController** - отвечает за обработку HTTP запросов графического интерфейса.
- 2) Пакет **data** - хранит в себе дата-классы для хранения информации:
 - (a) **BufferTableRow** - дата-класс, описывающий строку таблицы буфера.
 - (b) **CalendarAndStateTableRow** - дата-класс, описывающий строку таблицы календаря событий и текущего состояния.
 - (c) **DeviceCharacteristicTableRow** - дата-класс, описывающий строку таблицы характеристик приборов.
 - (d) **InputCharacteristicTableRow** - дата-класс, описывающий строку таблицы характеристик источников.
 - (e) **SimulationResult** - дата-класс, описывающий результаты симуляции.
 - (f) **StepData** - дата-класс, описывающий шаг симуляции.
- 3) Пакет **research** - хранит в себе классы, ответственные за исследование модели:
 - (a) **OutputCharacteristics** - дата-класс, описывающий выходные характеристики моделирования.
 - (b) **Sighting** - класс, ответственный за пристрелку.
 - (c) **Tests** - класс, ответственный за тест модели (прогон всевозможных конфигураций).

- 4) Пакет **simulation** - хранит в себе классы, ответственные за симуляцию:
- (a) **Buffer** - класс, реализующий функциональность буфера.
 - (b) **BufferPlace** - класс, реализующий функциональность места в буфере.
 - (c) **Config** - класс, хранящий в себе конфигурацию системы.
 - (d) **Device** - класс, реализующий функциональность прибора.
 - (e) **DeviceKit** - класс, реализующий функциональностью набора приборов.
 - (f) **Event** - класс, реализующий функциональность события.
 - (g) **EventType** - перечисление, хранящее тип события.
 - (h) **Input** - класс, реализующий функциональностью источника.
 - (i) **InputKit** - класс, реализующий функциональностью набора источников.
 - (j) **Request** - класс, реализующий функциональностью заявки.
 - (k) **Simulation** - класс, ответственный за симуляцию.
- 5) Пакет **utils** - хранит в себе вспомогательные классы.
- (a) **Utils** - класс, хранящий вспомогательные методы (получение распределений времени).

6.2 Демонстрация работы программы

Так как у нас написано веб-приложение - программа запускается в браузере. По умолчанию это происходит по адресу *http://localhost:8080/*.

При переходе по данному адресу открывается главная страница:

Главная страница

Навигация

Результаты автоматического режима Результаты пошагового режима

Всего шагов 0
Время окончания моделирования 0.0
Общее время реализации 0.0

Средняя вероятность отказа в обслуживании заявки 0.0
Среднее время пребывания заявки в СМО 0.0
Среднее время пребывания заявки в буфере 0.0
Средний коэффициент использования прибора 0.0

Сконфигурируйте и запустите симуляцию

Количество источников

3

Размер буфера

3

Количество приборов

3

Количество запросов

4

Параметр a

15.0

Параметр b

100.0

Параметр lambda

0.001

Запустить симуляцию

Рис. 2: Главная страница

После запуска симуляции, мы можем заметить, как отобразились результаты симуляции:

Главная страница

Навигация

Результаты автоматического режима

Результаты пошагового режима

Всего шагов 9
Время окончания моделирования 99.61893028831908
Общее время реализации 1998.0356675915025

Средняя вероятность отказа в обслуживании заявки 0.0
Среднее время пребывания заявки в СМО 958.5052103882302
Среднее время пребывания заявки в буфере 108.42927934500834
Средний коэффициент использования прибора 0.4963314647857622

Сконфигурируйте и запустите симуляцию

Количество источников

3

Размер буфера

3

Количество приборов

3

Количество запросов

4

Параметр a

15.0

Параметр b

100.0

Параметр lambda

0.001

Запустить симуляцию

Рис. 3: Отображение результатов симуляции

Мы можем посмотреть результаты автоматического режима:

Результаты автоматического режима

Навигация

Главная

Результаты пошагового режима

Характеристики источников ВС

Номер источника	Количество заявок	Вероятность отказа	Tпрб	Tбп	Tобсл	Дбп	Добсл
0	1	0.0	552.2627782215075	0.0	552.2627782215075	0.0	0.0
1	1	0.0	1898.4167373031835	325.287838035025	1573.1288992681584	0.0	0.0
2	2	0.0	424.8361156399999	0.0	424.8361156399999	0.0	688.6194624810311

Характеристики приборов ВС

Номер прибора	Коэффициент использования
0	0.9868309595552454
1	0.27640286266121716
2	0.22576057214082393

Рис. 4: Результаты автоматического режима

А также увидеть, что произошло на каждом шаге:

Результаты пошагового режима

Навигация

[Главная](#)[Результаты автоматического режима](#)[Предыдущий шаг](#)[Следующий шаг](#)

Шаг 3

- Текущее время моделирования: 41.48233023016436.
- Источник 2 сгенерировал заявку (2; 1).
- Созданная заявка поступила на свободный прибор 2.

Календарь событий/текущее состояние

Событие	Время	Число заявок (Номер источника)	Число отказов (Номер заявки)
И0		1	0
И1	99.61893028831908	0	0
И2		2	0
Д0	424.90676832334407	2	0
Д1	580.2927429620069	0	0
Д2	492.560005703395	2	1

Буфер

Позиция	Время постановки	Источник	Заявка
0			
1			
2			

Рис. 5: Результаты пошагового режима

7 Вывод

По результатам курсовой работы была разработана программа для имитационного моделирования системы массового обслуживания при помощи языка **Java** и фреймворка для созданий веб-приложений **Spring**.

Посредством данной программы была проанализирована реальная система массового обслуживания, для которой были подобраны идеальные конфигурации, удовлетворяющие всем ограничениям и способные в лучшем случае повысить эффективность системы практически в **6** раз.