Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА
«Методы имитационного моделирования»
по дисциплине «Архитектура программных систем»

Выполнил Бутняков С.А.
студент гр. 3530904/00105

Руководитель Гончаров А.В.

Санкт-Петербург
2023

ОГЛАВЛЕНИЕ

Постановка задачи *3*
Принцип Δt *3*
Принцип особых состояний *3*
Диаграммы *4*
Формализованная схема и описание СМО *5*
Вариант СМО, рассматриваемой в данной работе *5*
Вывод законов распределения *6*
Пуассоновский закон распределения *6*
Равномерный закон распределения *6*
Пример технической системы (ВС или части ВС), удовлетворяющей формализованномуописанию *7*
Ограничения и требуемые характеристики *7*
Стоимость компонентов системы *7*
ПО для моделирования системы *8*
Блок-схема *8*
Структура *8*
Пользовательский интерфейс *9*
Результаты работы имитационной модели *12*
Анализ результатов, выводы и рекомендации по выбору конфигурации системы *12*
Выводы *16*

Постановка задачи

Целью курсовой работы является создание модели вычислительной системы (ВС) или ее части на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность. Каждый реальный объект (реальная ВС) обладает бесконечной сложностью, множеством характеристик, внутренних и внешних связей. Модель есть приближенное описание объекта с целью получения требуемых результатов с определенной точностью и достоверностью.

При необходимости исследования поведенческих характеристик ВС в процессе исследования выгодно использовать не сам объект, а его модель. Степень приближения модели к описываемому объекту может быть различной и зависит от требований задачи.

Существуют различные типы моделей:

1. Аналитические (математические) модели
2. Аналоговые модели
3. Физические модели
4. Имитационные модели

Последний тип моделей является предметом нашего изучения. Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО), с характерной для СМО терминологией: источник, буфер, прибор, диспетчер, заявка (требование). Существуют два подхода к построению моделирующего алгоритма:

Принцип Δt

Принцип Δt – универсальный метод построения моделирующего алгоритма, когда состояние объекта проверяется через фиксированный интервал модельного времени. Суть его заключается в следующем: в каждый момент времени t получают приближенные значения характеристик исследуемого объекта. Δt можно получить детерминированным способом. Основной критерий выбора Δt – он должен быть настолько мал, чтобы не пропустить событие в моделируемой системе, которое должно быть учтено при выбранной детальности моделирования. Метод неэффективен, т. к. постоянно проверяет состояние объектов моделирования, не изменяющихся при этом, особенно при малых Δt.

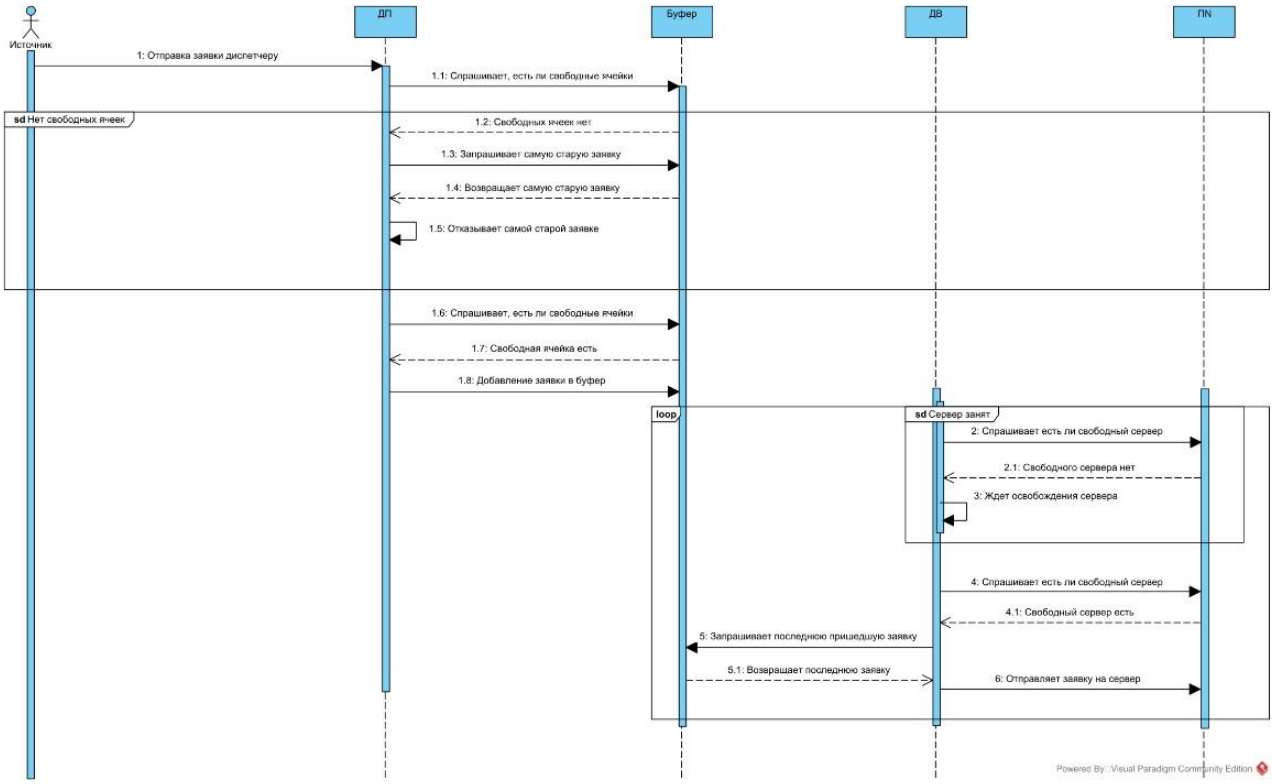
Принцип особых состояний

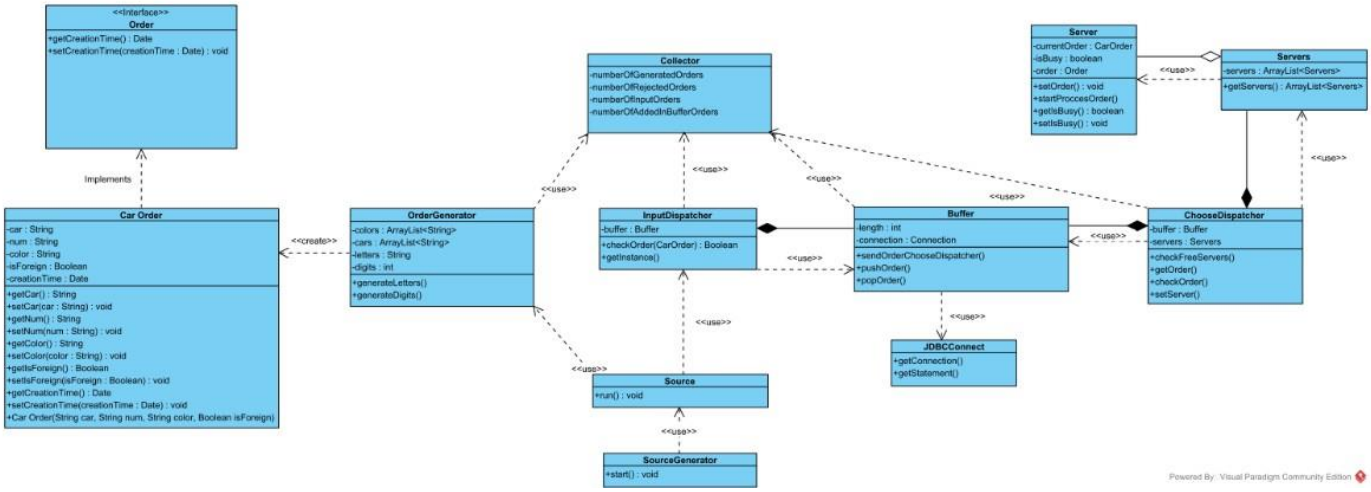
При исследовании реальной системы интервалы, в которых состояние ее не меняется, не представляют интереса. Имеют значение только переходы системы из одного состояния в другое в некоторые моменты времени. Эти переходы определяются особыми состояниями или событиями. Рассмотрим некоторые типы особых событий, которые изменяют состояние системы:

1. Поступление заявки в СМО (момент генерации заявки источником);
2. Освобождение прибора (готовность прибора взять заявку на обслуживание);
3. Окончание процесса моделирования.

Использование принципа особых событий для построения имитационной модели наиболее эффективно. В настоящей курсовой работе предлагается использовать именно этом принцип.

Диаграммы


Источник
ДП
Буфер
ДВ
ПN
1: Отправка заявки диспетчеру
1.1: Спрашивает, есть ли свободные ячейки
sd Нет свободных ячеек
1.2: Свободных ячеек нет
1.3: Запрашивает самую старую заявку
1.4: Возвращает самую старую заявку
1.5: Отказывает самой старой заявке
1.6: Спрашивает, есть ли свободные ячейки
1.7: Свободная ячейка есть
1.8: Добавление заявки в буфер
loop
sd Сервер занят
2: Спрашивает есть ли свободный сервер
2.1: Свободного сервера нет
3: Ждет освобождения сервера
4: Спрашивает есть ли свободный сервер
4.1: Свободный сервер есть
5: Запрашивает последнюю пришедшую заявку
5.1: Возвращает последнюю заявку
6: Отправляет заявку на сервер
Powered By: Visual Paradigm Community Edition


<<Interface>>
Order
+getCreationTime() : Date
+setCreationTime(creationTime : Date) : void
Implements
Car Order
-car : String
-num : String
-color : String
-isForeign : Boolean
-creationTime : Date
+getCar() : String
+setCar(car : String) : void
+getNum() : String
+setNum(num : String) : void
+getColor() : String
+setColor(color : String) : void
+getIsForeign() : Boolean
+setIsForeign(isForeign : Boolean) : void
+getCreationTime() : Date
+setCreationTime(creationTime : Date) : void
+Car Order(String car, String num, String color, Boolean isForeign)
<<create>>
OrderGenerator
-colors : ArrayList<String>
-cars : ArrayList<String>
-letters : String
-digits : int
+generateLetters()
+generateDigits()
<<use>>
Collector
-numberOfGeneratedOrders
-numberOfRejectedOrders
-numberOfInputOrders
-numberOfAddedInBufferOrders
InputDispatcher
-buffer : Buffer
+checkOrder(CarOrder) : Boolean
+getInstance()
Buffer
-length : int
-connection : Connection
+sendOrderChooseDispatcher()
+pushOrder()
+popOrder()
JDBCConnect
+getConnection()
+getStatement()
Source
+run() : void
SourceGenerator
+start() : void
Server
-currentOrder : CarOrder
-isBusy : boolean
-order : Order
+setOrder() : void
+startProcessOrder()
+getIsBusy() : boolean
+setIsBusy() : void
Servers
-servers : ArrayList<Servers>
+getServers() : ArrayList<Servers>
ChooseDispatcher
-buffer : Buffer
-servers : Servers
+checkFreeServers()
+getOrder()
+checkOrder()
+setServer()
Powered By: Visual Paradigm Community Edition

Формализованная схема и описание СМО

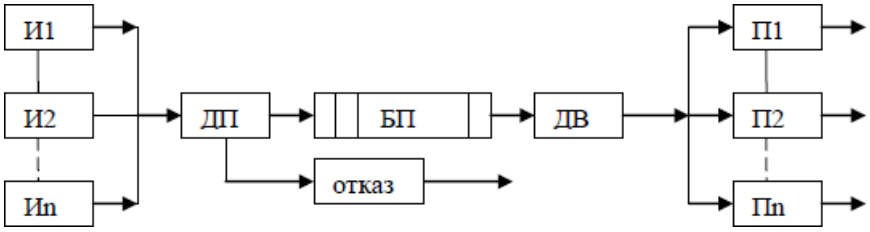


Рис 1. Формализованная схема СМО

Здесь $И_i$ (i = 1..n) – источник заявок, который генерирует заявки, а все вместе n источников создают входной поток заявок в систему. Каждая заявка приходит в СМО со своими характеристиками. Это входное Т – время генерации заявки (время поступления ее в СМО) и номер заявки, составленный из номера источника, сгенерировавшего заявку, и порядкового номера заявки от этого источника. Например, (2.3) – третья заявка от второго источника.

П – приборы, которые обслуживают заявки и создают выходной поток заявок после обслуживания.

БП – буферная память (место для хранения очереди заявок). В общей памяти хранятся заявки от различных источников. Порядок их записи в БП определяется только дисциплиной буферизации.

ДП – диспетчер постановки заявок.

ДВ – диспетчер выбора заявок.

Вариант (25) СМО, рассматриваемой в данной работе

- ИБ – бесконечный источник.
- ИЗ1 – пуассоновский закон распределения.
- ПЗ2 – равномерный закон распределения времени обслуживания.
- Д1ОЗ3 – дисциплина буферизации; запись в буфер, если есть место – на свободное место.
  Заявка встанет в очередь на первое от начала свободное место, если такое найдется. Сдвига очереди в этом случае не происходит.
- Д1ООЗ – 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 ее место встает пришедшая заявка.
- Д2П1 – дисциплина постановки на обслуживание; выбор прибора; приоритет по номеру прибора.
  Приоритеты приборов, так же, как и приоритеты источников, определяются номерами приборов. Поэтому поиск свободного прибора ведется последовательным перебором, каждый раз начиная с самого приоритетного.
- Д2Б2 – дисциплина постановки на обслуживание; выбор заявки из буфера; LIFO.
  В этом случае раньше других будет выбрана из буфера на обслуживание та заявка, которая пришла последней.
- ОР1 – отображение результатов в автоматическом режиме; сводная таблица результатов.
  Шаг в этом случае — интервал модельного времени от одного особого события до другого ближайшего по времени особого события.
- ОД1 – отображение результатов в пошаговом режиме; календарь событий, текущее состояние.

Вывод законов распределения

Пуассоновский закон распределения

$$F_k = \frac{e^{-\gamma} * \gamma^k}{k!}$$

где γ – заданное значение.

$$x = \frac{-1}{\gamma} * \ln(F_k)$$

В программе данное выражение записано следующим образом:

```
(-1 / lambda) * Math.log(Math.random());
```

Равномерный закон распределения

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

где a и b – заданные значения.

$$x = F(x)(b - a) + a$$

В программе выражение записано следующим образом:

```
Math.random() * (b - a) + a;
```

Пример технической системы (ВС или части ВС), удовлетворяющей формализованному описанию

Система	Автомастерская
Источники	Источником является сайт с авиабилетами, который отсылает данные на обработку в виде пакета размером 16 Кбайт.
Приборы	Приборами являются сервера, которые обрабатывают полученную заявку и отправляют результат авиакомпаниям.
Буфер	Буфером является буфер коммутатора, который имеет объем 16 Кбайт (т. е. может хранить 1 заявку) и может быть наращен до 128 кб для хранения 8 заявок.
Дисциплина постановки в буфер	Заявка встает на ближайшее от начала свободное место в буфере. Сдвига в таком случае не происходит.
Дисциплина выборки из буфера	Выбирается последняя добавленная в буфер заявка.
Дисциплина отказа	Самая старая заявка в буфере.
Дисциплина постановки на обслуживание	Приоритет по номеру прибора.

Ограничения и требуемые характеристики

Вероятность отказа должна составлять не более 10%.

Загрузка приборов более 90%.

Время пребывания заявки в системе не более 5 мс. Рассматриваемый диапазон характеристик системы и доступные типы процессоров и характеристики программно-аппаратного комплекса, построенного на данном типе процессора приведены ниже в таблицах компонентов системы.

Количество сайтов-источников	4 штуки
Вес заявки	16 Кб
Объем буфера	От 16Кб до 128 Кб
Количество приборов	От 1 до 7

Стоимость компонентов системы

Условимся, что стоимость одного прибора для обработки составляет – 10000 рублей.

Стоимость расширения буфера на 1 слот – 500 рублей.

ПО для моделирования системы

Блок-схема

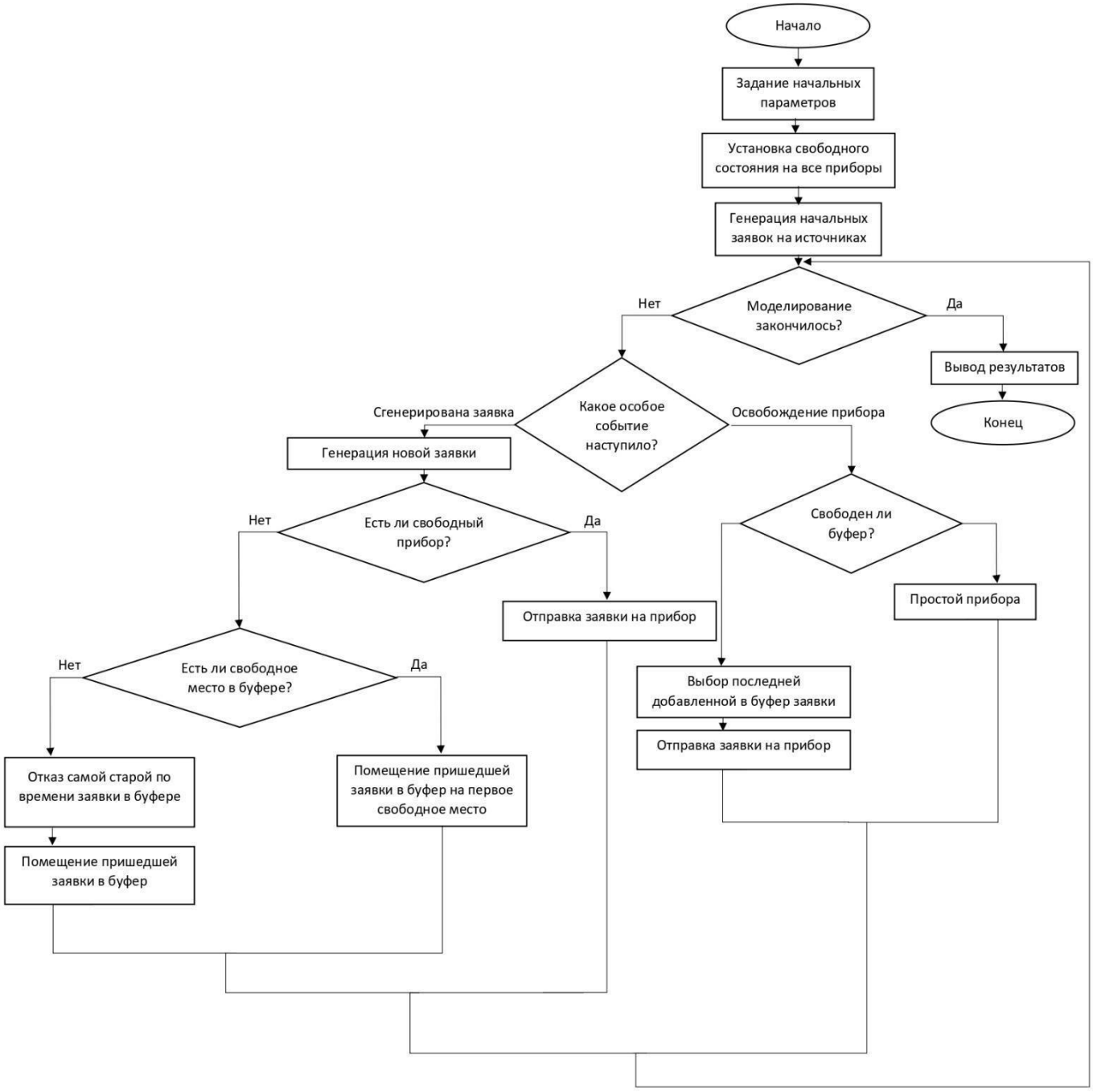


Рис 3. Блок-схема ПО для моделирования системы

Структура

Для моделирования системы была написана программа на языке Java с использованием платформы для создания приложений с графическим пользовательским интерфейсом Java FX. Приложение использует принцип ООП и содержит следующие классы:

- Ticket — содержит описание заявки, реализует методы для получения этих описаний;
- Controller — реализует функционирование системы в автоматическом и пошаговом режимах;
- Source — содержит описание источника: его номер, параметры закона распределения, текущую заявку;
- SourceMetrics – реализует сбор статистики по каждому источнику: количество сгенерированных, обработанных и выбитых заявок одного источника, а также время нахождения в системе, время ожидания обслуживания и время обработки заявок данного источника;
- SourceController – осуществляет управление источниками системы: хранит все источники,

регулирует отправку заявок в буфер по кольцу;

- Device — содержит описание прибора: его номер, коэффициент закона распределения, время окончания обработки предыдущей заявки, текущую заявку;
- DeviceMetrics – реализует сбор статистики по каждому прибору: время работы прибора и количество обработанных заявок;
- DeviceController – осуществляет управление приборами системы: хранит все приборы, регулирует получение заявок из буфера и их обработку;
- Buffer - содержит описание буфера: его номер и текущую заявку;
- BufferController — реализует методы управления буфером: выбирает какой из буферов освободить и какой занять;
- MetricsController – формирует общую статистику по всей системе, объединяя информацию, полученную от SourceMetrics и DeviceMetrics;
- AutoMode — класс графического интерфейса, отображающий автоматический режим модели;
- StepByStepMode — класс графического интерфейса, отображающий пошаговый режим модели;
- MainMenu - класс графического интерфейса, отображающий общее меню программы для настройки параметров

Пользовательский интерфейс

При запуске приложения первоначально открывается окно настроек:

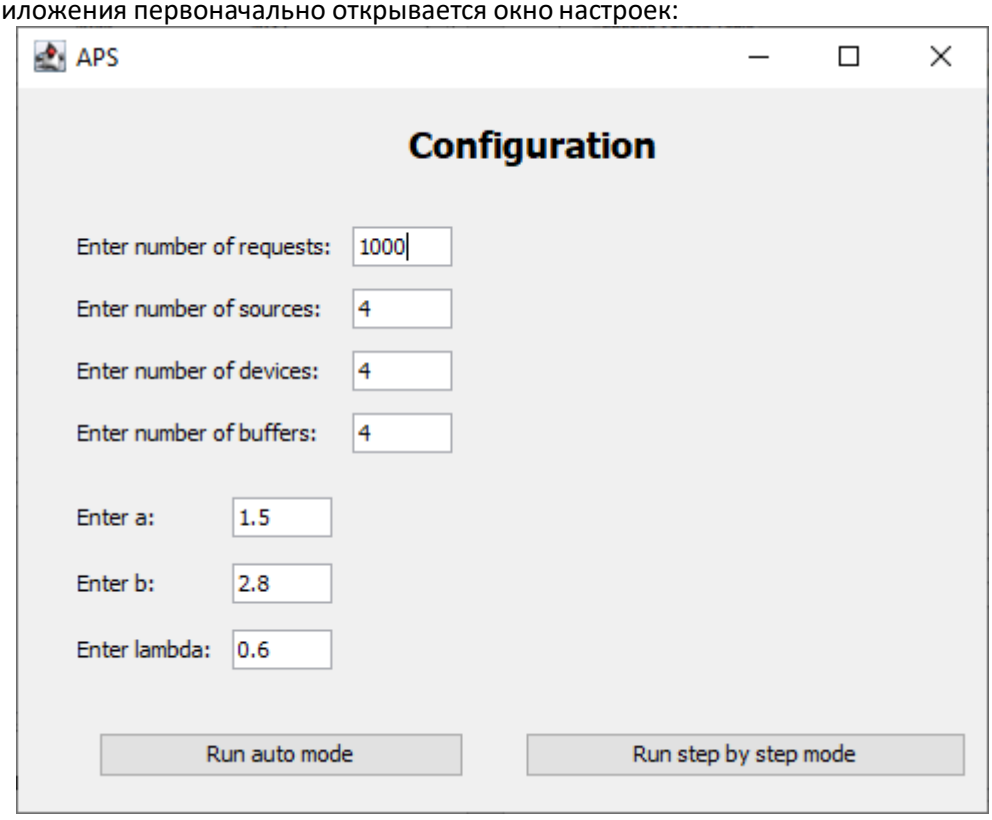


Рис 4. Окно настроек приложения

Приложение ожидает от пользователя ввода количества источников, количества устройств буфера, количества приборов, интенсивности Пуссановского потока λ, границы равномерного потока, а также количества заявок, которые будут сгенерированы в автоматическом режиме.

Здесь пользователь может выбрать использование автоматического или пошагового режима.

Auto Mode

Auto Mode

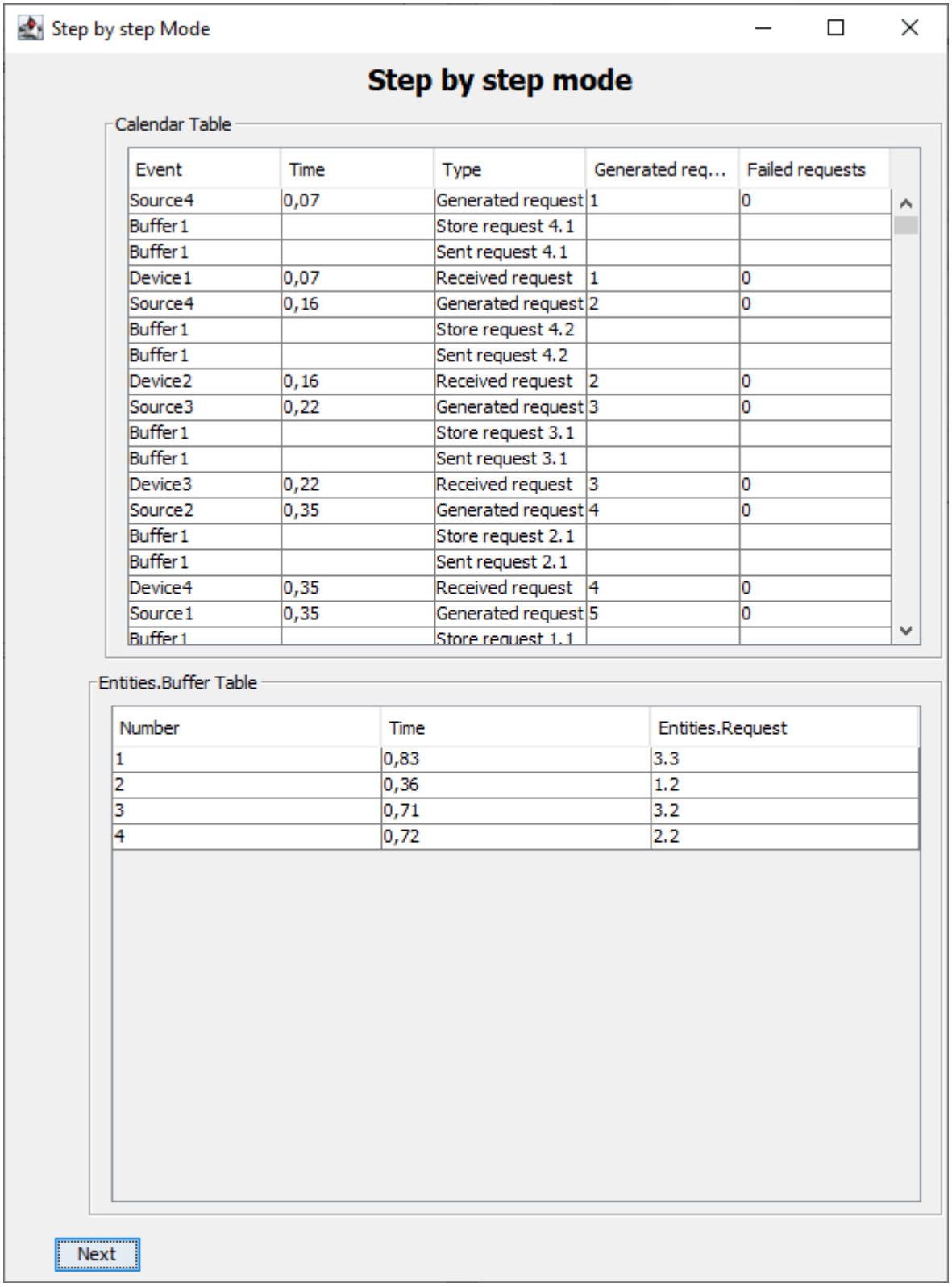
Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	256,00	0,89	3,55	0,10	3,46	0,01	0,08
2	246,00	0,90	3,61	0,06	3,55	0,00	0,09
3	251,00	0,92	3,56	0,13	3,43	0,02	0,06
4	247,00	0,89	3,53	0,08	3,46	0,00	0,07

Entities.Device Table

Number	Usage rate
1	0,98
2	0,97
3	0,98
4	0,96

Рис 5. Автоматический режим приложения



Event	Time	Type	Generated req...	Failed requests
Source4	0,07	Generated request	1	0
Buffer1		Store request 4.1		
Buffer1		Sent request 4.1		
Device1	0,07	Received request	1	0
Source4	0,16	Generated request	2	0
Buffer1		Store request 4.2		
Buffer1		Sent request 4.2		
Device2	0,16	Received request	2	0
Source3	0,22	Generated request	3	0
Buffer1		Store request 3.1		
Buffer1		Sent request 3.1		
Device3	0,22	Received request	3	0
Source2	0,35	Generated request	4	0
Buffer1		Store request 2.1		
Buffer1		Sent request 2.1		
Device4	0,35	Received request	4	0
Source1	0,35	Generated request	5	0
Buffer1		Store request 1.1		

Number	Time	Entities.Request
1	0,83	3.3
2	0,36	1.2
3	0,71	3.2
4	0,72	2.2

Рис 6. Автоматический режим приложения

Результаты работы имитационной модели

Количество реализаций, необходимое для получения нужной точности при заданной доверительной вероятности, можно оценивать по формуле:

$$\square = \frac{\square_{\alpha}^{2}(1-\square)}{\square\square^{2}}$$

где p – вероятность отказа заявкам в обслуживании,
t_{α} = 1.643, для α = 0.9,

σ = 0.1 - относительная точность.

По результатам работы программы получено, что в большинстве случаев для достиженияязаданной точности необходимо около 1000 заявок

Анализ результатов, выводы и рекомендации по выбору конфигурации системы

Рассмотрим вариант конфигурации, когда имеется всего 4 источника. Т. к. целью моделирования является выбор конфигурации системы, требующей наименьшее количество ресурсов и обрабатывающей максимальный поток информации, то начнем с проверки конфигурации с минимальным числом приборов и минимальным размером буфера.
Также ограничим время пребывания заявки в буфере – 5 мс.

Количество источников во всех опытах возьмем равным 4.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	265,00	0,85	2,77	0,36	2,40	0,14	2,48
2	234,00	0,85	2,87	0,35	2,52	0,21	2,16
3	249,00	0,84	3,25	0,31	2,93	0,13	2,16
4	252,00	0,83	2,88	0,31	2,57	0,13	1,87

Entities.Device Table

Number	Usage rate
1	0,98

В первом опыте прибор эффективно используется, время нахождения в системе допустимо, но 84% заявок уходят в отказ, что является недопустимым, поэтому добавляем ещё один прибор и расширяем буфер на 1.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	263,00	0,62	2,68	0,35	2,33	0,15	2,31
2	263,00	0,62	2,90	0,54	2,36	0,40	2,30
3	243,00	0,63	2,81	0,41	2,39	0,21	1,68
4	231,00	0,67	2,93	0,51	2,41	0,30	2,35

Entities.Device Table

Number	Usage rate
1	0,98
2	0,97

Вероятность отказа снизилась, но до сих пор не является удовлетворительной. Время нахождения в системе допустимо. Возьмем для обработки еще один прибор и еще одно место в буфере.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	247,00	0,40	2,70	0,55	2,16	0,47	0,12
2	253,00	0,42	2,74	0,53	2,21	0,43	0,13
3	246,00	0,46	2,71	0,59	2,13	0,37	0,13
4	254,00	0,39	2,60	0,52	2,08	0,35	0,14

Entities.Device Table

Number	Usage rate
1	0,99
2	0,99
3	0,98

Вероятность отказа снизилась, но до сих пор не является удовлетворительной. Время нахождения в системе допустимо. Возьмем для обработки еще один прибор и еще одно место в буфере.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	254,00	0,24	2,65	0,54	2,11	0,53	0,14
2	253,00	0,20	2,77	0,63	2,15	0,73	0,14
3	237,00	0,24	2,86	0,71	2,15	1,15	0,14
4	256,00	0,26	2,78	0,65	2,13	0,89	0,14

Entities.Device Table

Number	Usage rate
1	0,98
2	0,97
3	0,97
4	0,95

Вероятность отказа упала, но все еще выше нормы, возьмем для обработки еще один прибор и один слот для буфера.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	241,00	0,09	2,73	0,58	2,15	1,34	0,14
2	240,00	0,09	2,76	0,61	2,15	1,46	0,15
3	275,00	0,11	2,73	0,57	2,16	0,97	0,14
4	244,00	0,10	2,71	0,60	2,11	1,14	0,14

Entities.Device Table

Number	Usage rate
1	0,97
2	0,94
3	0,93
4	0,91
5	0,89

Вероятность отказа существенно упала, но при этом существуют случаи превышения нормы отказа, а также один прибор загружен менее, чем на 90 процентов. Увеличим количество слотов в буфере до 7.

Auto Mode

Auto Mode

Entities.Source Table

Number	Generated requests	Fail probability	Average TimeInSys...	Average TimeWaiting	Average TimeProce...	Dispersion TimeWai...	Dispersion TimePro...
1	247,00	0,08	2,94	0,74	2,20	2,73	0,14
2	262,00	0,09	3,21	1,06	2,16	5,32	0,13
3	264,00	0,06	2,89	0,75	2,13	2,94	0,14
4	227,00	0,04	2,82	0,65	2,16	1,98	0,14

Entities.Device Table

Number	Usage rate
1	0,95
2	0,95
3	0,94
4	0,93
5	0,9

Время нахождения в системе допустимо. Вероятность отказа снизилась до 4–9%, что нас полностью устраивает. При уменьшении количества приборов (даже при улучшении их качества) система не будет удовлетворять нашим требованиям.

Стоимость системы: 10000 * 5 + 500 * 7 = 53500 рублей.

Выводы

В ходе выполнения курсовой работы мной была написана система массового обслуживания на язык Java с использованием платформы Java FX. С помощью получившейся программы была проанализирована реальная система и подобрана максимально выгодная комплектация данной системы.