

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

## КУРСОВАЯ РАБОТА

Моделирование системы, формализованной как  
система массового обслуживания.

По дисциплине «Архитектура программных систем»

Выполнила

студентка гр. 5130904/10104

Ролецкая С. Н.

Руководитель

Гончаров А. В.

«1» декабря 2023г.

## Оглавление

Введение.....	3
Постановка варианта. ....	4
Расшифровка варианта: .....	4
Общая постановка задачи.....	7
Описание бизнес-домена. ....	7
Мэппинг бизнес-домена на архитектуру СМО. ....	7
Артефакты.....	9
Sequence диаграмма. ....	9
Диаграмма классов.....	10
Описание программной модели. ....	11
Описание СМО. ....	14
Ограничения и требуемые характеристики.....	14
Экономическое обоснование. ....	15
Результаты анализа системы на предмет экономической выгоды. ....	15
Вывод. ....	18

## Введение.

Целью практической курсовой является создание модели ВС или ее компоненты на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность. Модель дает приближенное описание объекта с целью получения требуемых результатов с определенной точностью и достоверностью.

Степень приближения модели к описываемому объекту может быть различной и зависит от требований задачи. Существуют различные типы моделей ВС: аналитические, аналоговые, физические и имитационные. В данной работе будет использоваться имитационная модель ВС. Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО).

Результатом курсовой работы является экономическое обоснование получившейся системы с описанием маппинга бизнес-домена на архитектуру СМО.

## Постановка варианта.

Вариант 12:

ИБ ИЗ2 ПЗ1 Д10ЗЗ Д10О4 Д2П2 Д2БЗ ОР1 ОДЗ

Расшифровка варианта:

### 1. Параметры элементов модели.

#### 1.1. Источники:

**ИБ** – бесконечный;

**ИЗ2** – равномерный закон распределения;

#### 1.2. Приборы.

**ПЗ1** — экспоненциальный закон распределения времени обслуживания;

### 2. Описание дисциплин постановки и выбора.

#### 2.1. Дисциплины буферизации.

**Д1** — постановки в буфер;

**Д1О** — относительные приоритеты на обслуживание;

**Д1ОЗ** — записи в буфер, если есть место;

**Д1ОЗЗ** — на свободное место;

Заявка встанет в очередь на первое от начала свободное место, если такое найдется. Сдвига очереди в этом случае не происходит.

#### 2.2. Дисциплины отказа.

**Д1ОО4** — последняя поступившая в буфер;

Самая последняя заявка из поступивших в буфер, т. е. заявка, меньше других простоявшая в очереди, выбивается из БП, и на её место встаёт пришедшая заявка.

#### 2.3. Дисциплины постановки на обслуживание.

**Д2** — постановки на обслуживание;

**Д2Б** — выбор заявки из буфера;

**Д2БЗ** — по кольцу;

**Д2П** — выбор прибора;

**Д2П2** — по кольцу;

Освобождение прибора или его простой означает, что прибор готов взять заявку на обслуживание. Если в буфере есть очередь, то заявка поступает на прибор в момент его освобождения. Какую заявку поставить на обслуживание на освободившийся прибор определяют дисциплины выбора заявок.

#### 2.3.1. Дисциплины выбора заявок на обслуживание.

**Д2БЗ** — по кольцу;

Эта дисциплина выбора работает аналогично дисциплине постановки в БП «по кольцу». Здесь также вводится понятие «Указателя», который принимает значение номера места, с которого начинается поиск заявок в БП. Как только будет найдено место, занятое заявкой, эта заявка ставится на освободившийся прибор, а указатель передвигается на место, следующее за тем, откуда была выбрана на обслуживание заявка. Поиск следующего занятого места буфера будет производиться также, начиная с указателя.

#### 2.3.2. Дисциплины выбора прибора.

**Д2П2** — выбор прибора по кольцу;

Эта дисциплина производит выбор свободного прибора таким же способом, как и аналогичная дисциплины выбора заявок из буфера по кольцу, т. е. поиск свободных приборов каждый раз начинается с указателя, и заявка встает на обслуживание на первый из найденных приборов.

### 3. Виды отображения результатов работы программной модели.

#### 3.1. Динамическое отражение результатов (пошаговый режим).

**ОД** — отображение динамики функционирования модели;

**ОДЗ** — временные диаграммы, текущее состояние.

Шаг в этом случае — интервал модельного времени от одного особого события до другого ближайшего по времени особого события.

### 3.2. Отражение результатов после сбора статистики ОР1-ОР2 (автоматический режим).

ОР — отображение результатов;

**ОР1** — сводная таблица результатов;

## Общая постановка задачи.

### Описание бизнес-домена.

Бизнес-домен представляет собой систему автоматизации проектирования деталей для круизных лайнеров. Процесс включает в себя обработку заявки, конструирование и создание чертежей по заданным требованиям, перевод созданных чертежей в 3D-модель и примерку 3D-модели на корабле.

### Маппинг бизнес-домена на архитектуру СМО.

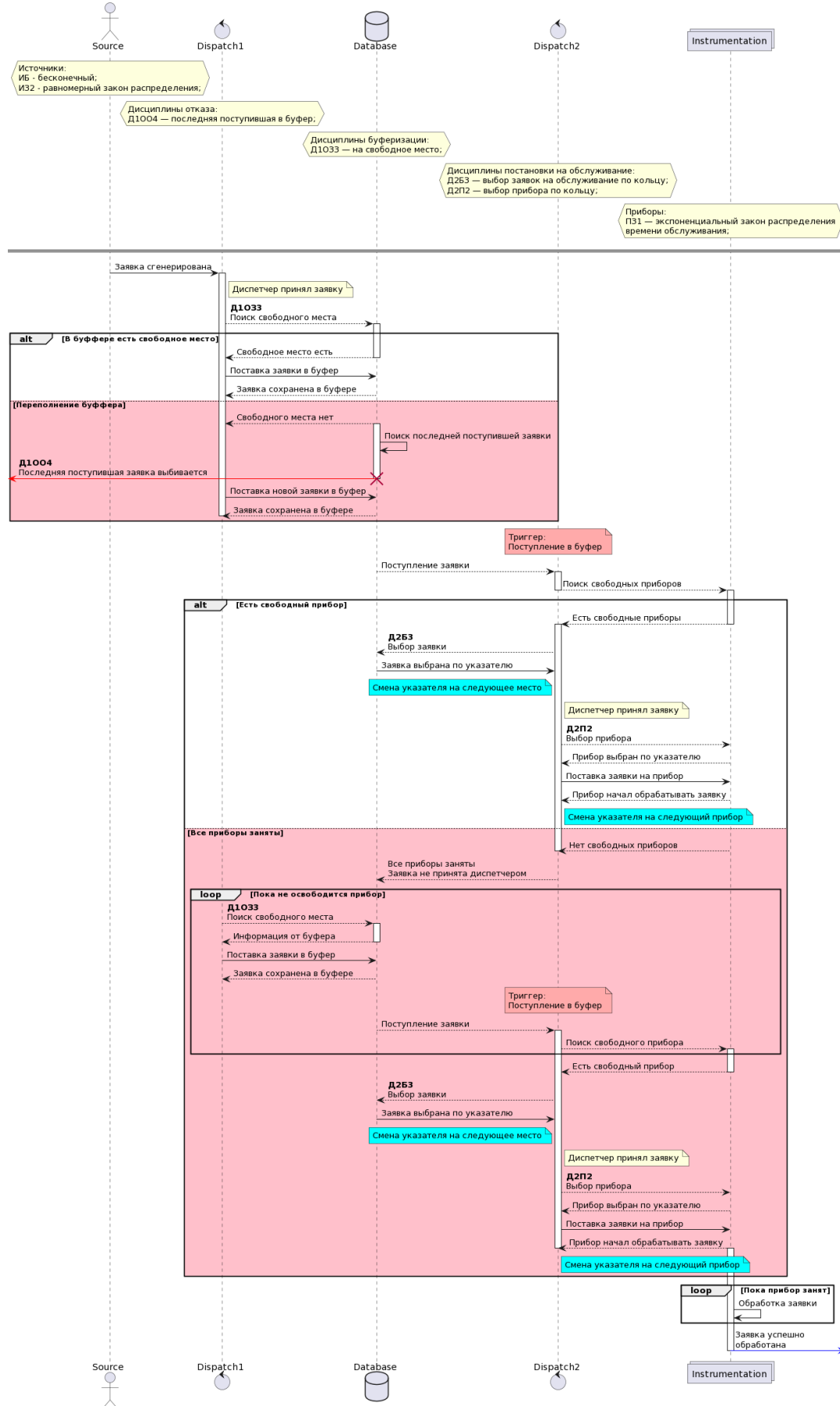
Предметная область	Система автоматизации проектирования деталей для круизных лайнеров
Источники	Источником является заявка от компании на проектирование детали, в которой отражены требования для ее проектирования, ее размеры и характеристики.
Приборы	Прибором является ЭВМ с инженером, которые получают информацию из заявки и реализуют модель детали по соответствующим требованиям. Обработанные заявки записываются в базу данных для «поднятия» информации по ним в случае непредвиденных ситуаций.
Буфер	Буфером является база данных заявок в облачном хранилище компании.
Дисциплина постановки в буфер	В порядке поступления, ни одна из заявок не имеет приоритета.
Дисциплина отказа	Самая последняя заявка из поступивших в буфер, т. е. заявка, меньше других

	простоявшая в очереди, выбивается из БП, и на её место встаёт пришедшая заявка.
Дисциплина выбора из буфера	По кольцу; Выбор осуществляется с помощью указателя, который принимает значение номера места, с которого начинается поиск заявок в БП. Как только будет найдено место, занятое заявкой, эта заявка ставится на освободившийся прибор, а указатель передвигается на место, следующее за тем, откуда была выбрана на обслуживание заявка. Поиск следующего занятого места буфера будет производиться также, начиная с указателя.
Дисциплина постановки на обслуживание	По кольцу; Поиск свободных приборов каждый раз начинается с указателя, и заявка встает на обслуживание на первый из найденных приборов.

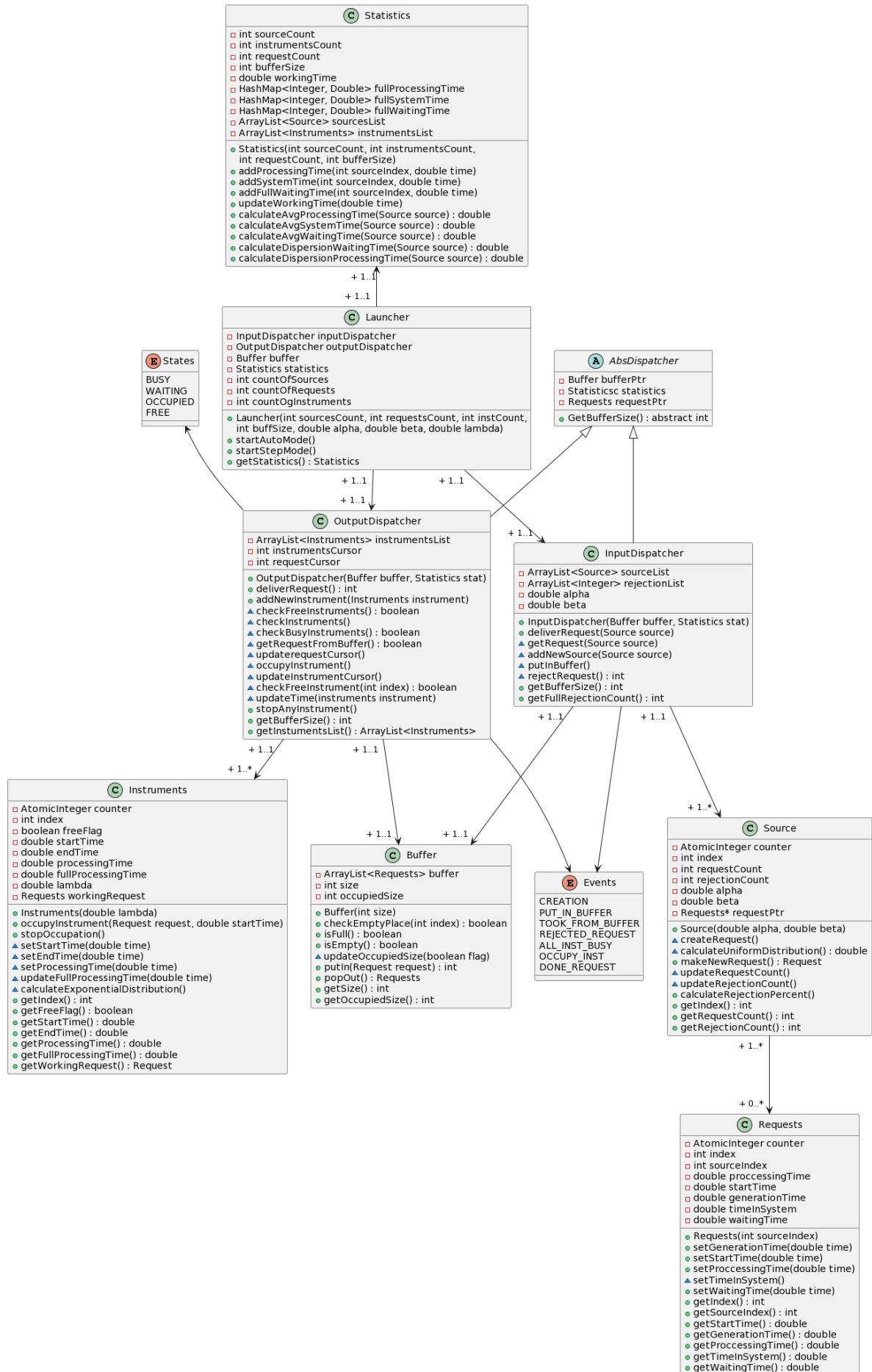


# Артефакты.

## Sequence диаграмма.



## Диаграмма классов.



## Описание программной модели.

Разработка производилась в среде IntelliJ IDEA 2023 на языке Java. Результаты работы программы выводятся при помощи графического интерфейса, написанного с помощью инструментов JavaFX.

Начальный экран программы – это экран конфигурации. Пользователю необходимо ввести данные, чтобы сконфигурировать модель: количество источников, количество заявок, количество приборов, размер буфера,  $\alpha$  и  $\beta$  для расчета времени генерации заявок,  $\gamma$  для расчета времени обслуживания заявки на приборе.

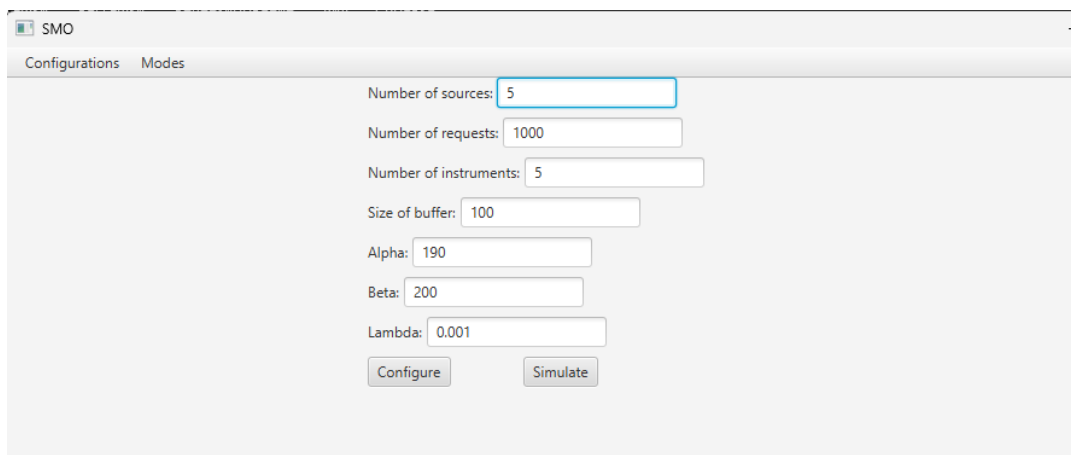
Время генерации высчитывается по равномерному закону распределения:

$$F(x) = \begin{cases} 0, & x < \alpha \\ \frac{x - \alpha}{\beta - \alpha}, & \alpha \leq x < \beta \\ 1, & x \geq \beta \end{cases} \rightarrow x = F(x)(\beta - \alpha) + \alpha$$

где  $F(x)$  – функция генерации случайного числа в диапазоне от 0 до 1

Время обработки высчитывается по экспоненциальному закону распределения:

$$F(x) = 1 - e^{-\gamma x} \rightarrow \log(1 - F(x)) = -\gamma x \\ \rightarrow x = -\frac{1}{\gamma} \log(1 - F(x))$$



После ввода всех данных необходимо нажать кнопку «Конфигурировать». Чтобы просмотреть результаты модели нужно нажать на кнопку «Симулировать».

В программе реализованы два варианта отображения результатов:

- Пошаговый режим с календарем событий
- Автоматический режим с выводами о результатах

В пошаговом режиме отражается состояние системы: на каждом шаге выводятся состояния буфера, источников и приборов, указываются перемещения указателей для выбора заявки и выбора прибора. Каждый шаг представляет собой одно из событий: заявка сгенерирована, заявка отправлена в буфер, выбивание заявки из буфера, выбор заявки на обработку, отправка заявки на обработку, обработка заявки. По событиям можно перемещаться с помощью кнопок «Шаг назад» и «Шаг вперед» или непосредственно введя нужный шаг.

SMO

Configurations Modes

Event: Occupied instrument 2 with request 904

Time: 197832.123

Step: 4769

All steps: 5098

Enter the step:  Go

Name	Request count	Reject count
Source 1	188	19
Source 2	191	16
Source 3	208	24
Source 4	208	25
Source 5	205	18

Pointer	Place	State	Request ID	Source ID
	24	Free		
	25	Free		
>	26	Occupied	905	3
	27	Occupied	906	2
	28	Occupied	907	1
	29	Occupied	908	1
	30	Occupied	909	3
	31	Occupied	910	4
	32	Occupied	911	1
	33	Occupied	912	1
	34	Occupied	914	2

Pointer	Name	State	Request ID	Source ID
	Instrument 1	Busy	902	1
	Instrument 2	Busy	904	4
>	Instrument 3	Busy	901	4
	Instrument 4	Busy	903	1
	Instrument 5	Busy	900	1

Previous step Next step

В автоматическом режиме выводятся две таблицы с выводами по источникам и по приборам. В таблице с источниками для каждого источника высчитывается общее количество заявок, количество отказанных заявок, среднее время пребывания заявки в системе, среднее время обработки заявки на приборе, среднее время ожидания обработки заявки, дисперсия времени ожидания, дисперсия времени обработки заявок и вероятность отказа заявки.

Дисперсии высчитываются по формуле:

$$D_w = \frac{\sum (T_i - T_w)^2}{N_{req}}$$

Для таблицы с приборами высчитывается коэффициент загрузки для каждого прибора.

SMO								
Configurations Modes								
Name	Request count	Reject count	Time in system	Processing time	Waiting time	Disp. of waiting time	Disp of processing time	Reject percent, %
Source 1	188	19	16465.617	897.247	15373.23	9.5081261201E7	1148306.028	10.0
Source 2	191	16	16522.621	794.493	15533.489	8.3883217112E7	698775.548	8.0
Source 3	208	24	16082.176	879.49	15007.678	9.3058533461E7	754505.122	12.0
Source 4	208	25	16227.754	855.512	15176.805	9.8457467254E7	766682.589	12.0
Source 5	205	18	17647.888	931.984	16521.163	9.1947199601E7	861664.592	9.0
Name	Coefficient							
Instrument 1	0.801							
Instrument 2	0.811							
Instrument 3	0.806							
Instrument 4	0.801							
Instrument 5	0.792							

## Описание СМО.

### Ограничения и требуемые характеристики.

- Вероятность отказа не должна превышать 15%, чтобы компания не упала в убытки.
- Загрузка приборов не должна составлять более 90%, чтобы обеспечить адекватную загруженность работников.
- Время пребывания заявки в системе не ограничено, так как в зависимости от сложности проектируемых деталей заявки могут обрабатываться достаточно длительное время.
- Заявка от компании-источника генерируется некоторое время по равномерному закону, так как часто нужно уточнение многих деталей по проектируемому механизму.

Количество источников – государственных организаций и частных туристических компаний	От 1 до 6
Размер заявки – пакета документов с параметрами и требованиями для модели	От 1 до 3 Мбайт
Размер буфера – базы данных в облачном хранилище компании	От 50 до 200 Гбайт
Количество приборов – сотрудников с ЭВМ	От 3 до 8
Скорость генерации заявок	По равномерному закону распределения с $\alpha = 195$ и $\beta = 200$ - простые проекты $\alpha = 299$ и $\beta = 300$ - проекты средней детальности $\alpha = 495$ и $\beta = 500$ - проекты высокой детальности
Скорость работы прибора	По экспоненциальному закону с $\gamma = 0.00001 - 0.0000$ - сотрудник с устаревшим ПО $\gamma = 0.0001 - 0.0006$ - сотрудник с обновленным ПО $\gamma = 0.001 - 0.005$ - сотрудник с новейшим ПО

### Экономическое обоснование.

- Затраты компании:

Покупка лицензии на ПО для моделирования	Новейшее ПО – 164 тыс. руб. в год за штуку
	Обновленное ПО - 61 тыс. руб. в год за штуку
	Устаревшее ПО - 21 тыс. руб. в год за штуку
Заработная плата сотрудникам	90 тыс. руб. в месяц на одного сотрудника
Облачное хранилище на основе Microsoft OneDrive	2 тыс. руб. в месяц за 10 гб

- Прибыль компании:

Средняя стоимость обслуживания одной заявки

Простой проект – ~50 тыс. руб.

Проект средней детальности – до 80 тыс. руб.

Проект высокой детальности – от 100 тыс. руб.

Результаты анализа системы на предмет экономической выгоды.

Возьмем среднее количество источников – заказчиков и оставим это количество фиксированным. Рассмотрим различные сочетания сложности проекта и качества ПО. При вычислении следующих результатов были учтены все ограничения и требуемые характеристики.

$N_{ист}$	$N_z$	$N_{пр}$	$S_{буф}$ , гб	$\alpha$	$\beta$	$\gamma$	Доходы, млн руб.	Расходы, млн руб.	Прибыль, млн руб.
5	200	6	150	195	200	0.00006	10	6,9	3
5	200	5	80	195	200	0.0005	10	5,9	4,1
5	200	4	50	195	200	0.001	10	5,1	4,9
5	200	8	150	299	300	0.00006	16	9,2	6,8
5	200	4	50	299	300	0.0006	16	4,7	11,3

5	150	5	50	299	300	0.005	12	6,3	5,7
5	100	8	80	495	500	0.00001	10	9	1
5	100	4	70	495	500	0.0005	10	4,7	5,3
5	100	4	50	495	500	0.001	10	5	4,9

Сначала рассматривались варианты с легкими проектами, затем с проектами средней детализации и с проектами высокой детализации.

Пояснение результатов:

Из таблицы видно, что, чем хуже качество ПО, тем медленнее работают сотрудники и, следовательно, чтобы обеспечить качество труда и процент отказа, нужно увеличивать штат сотрудников, а также увеличивать объем хранилища. Использование новейшего ПО улучшает эту ситуацию, но, чтобы получать достаточную прибыль, нет смысла брать простые проекты, а с увеличением сложности проекта уменьшается количество заявок, что уменьшает и прибыль.

В случае простых проектов самым оптимальным вариантом будет использование новейшего ПО, так как можно сократить штат сотрудников и объем облачного хранилища.

Обработывая заявки высокой детализации, лучше использовать обновленное программное обеспечение, чтобы сохранять баланс между затратами на сотрудников и затратами на ПО.

Самый выгодный для компании вариант – это работа с заявками средней детализации на обновленном ПО. В этом варианте компания может обрабатывать достаточно большое количество заявок, при этом не перегружая сотрудников и не увеличивая объем облачного хранилища. Худший расклад для компании – это работа с заявками высокой сложности на устаревшем ПО. В теории, этот вариант должен быть самым прибыльным, но, чтобы



соответствовать требуемым нормам, приходится увеличивать штат сотрудников и облачного хранилища при небольшом количестве заявок.

## Вывод.

В ходе курсовой работы была написана система массового обслуживания на языке Java с использованием графической библиотеки JavaFX. С помощью данной программы была проанализирована реальная система и подобрана максимально выгодная комплектация данной системы.