

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Санкт-Петербургский политехнический университет Петра Великого»  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

## **КУРСОВОЙ ПРОЕКТ**

### **МОДЕЛИРОВАНИЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ**

по дисциплине «Архитектура программных систем»

Выполнил  
студент гр. 3530904/80102

Андреева Е.Д.

Руководитель  
старший преподаватель

Гончаров А.В.

«11» декабря 2020г.

## Оглавление

Оглавление .....	2
Введение .....	3
Постановка задачи.....	4
Sequence-диаграмма .....	5
Законы распределения .....	6
Ограничения и требуемые характеристики .....	6
Диаграмма классов.....	7
Модульная структура.....	8
Описание работы программы .....	9
Результаты работы .....	12
Анализ результатов: .....	13
Вывод .....	15
Исходный код .....	16

## Введение

В данной курсовой работе необходимо создать модель вычислительной системы (ВС) или ее компонентов на некотором уровне детализации, описывающей и имитирующей ее структуру и функциональность.

Каждый реальный объект ВС обладает высокой сложностью, определяемой множеством состояний, внутренних и внешних связей, анализируемых характеристик. 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.

Имитационная модель — программная модель системы, имитирующая ее поведение во времени, когда наблюдение за поведением обеспечивается наблюдением за изменением состояний системы или ее компонент, инициируемом соответствующими множествами входных и выходных сигналов или сообщений, которыми обмениваются компоненты, система и окружение

Одним из подходов к построению имитационной модели является построение ее в виде системы массового обслуживания (СМО)), с характерной для СМО терминологией:

- источник (И);
- буфер (БП);
- прибор (П);
- диспетчер (Д);
- заявка.

## Постановка задачи

1.	ИБ	ИЗ1	ПЗ2	Д10З1	Д10О1	Д2П1	Д2Б1	ОР2	ОД1	
----	----	-----	-----	-------	-------	------	------	-----	-----	--

### Источники:

ИБ — бесконечный источник;

ИЗ1 — пуассоновский закон распределения заявок;

### Приборы:

ПЗ2 — равномерный закон распределения времени обслуживания;

### Описание дисциплин постановки и выбора:

Буферизация:

Д10З1 — по кольцу;

Дисциплина отказа:

Д10О1 — под указателем;

Дисциплина постановки на обслуживание:

Д2П1 — приоритет по номеру прибора,  
Д2Б1 — FIFO

### Виды отображения результатов работы программной модели:

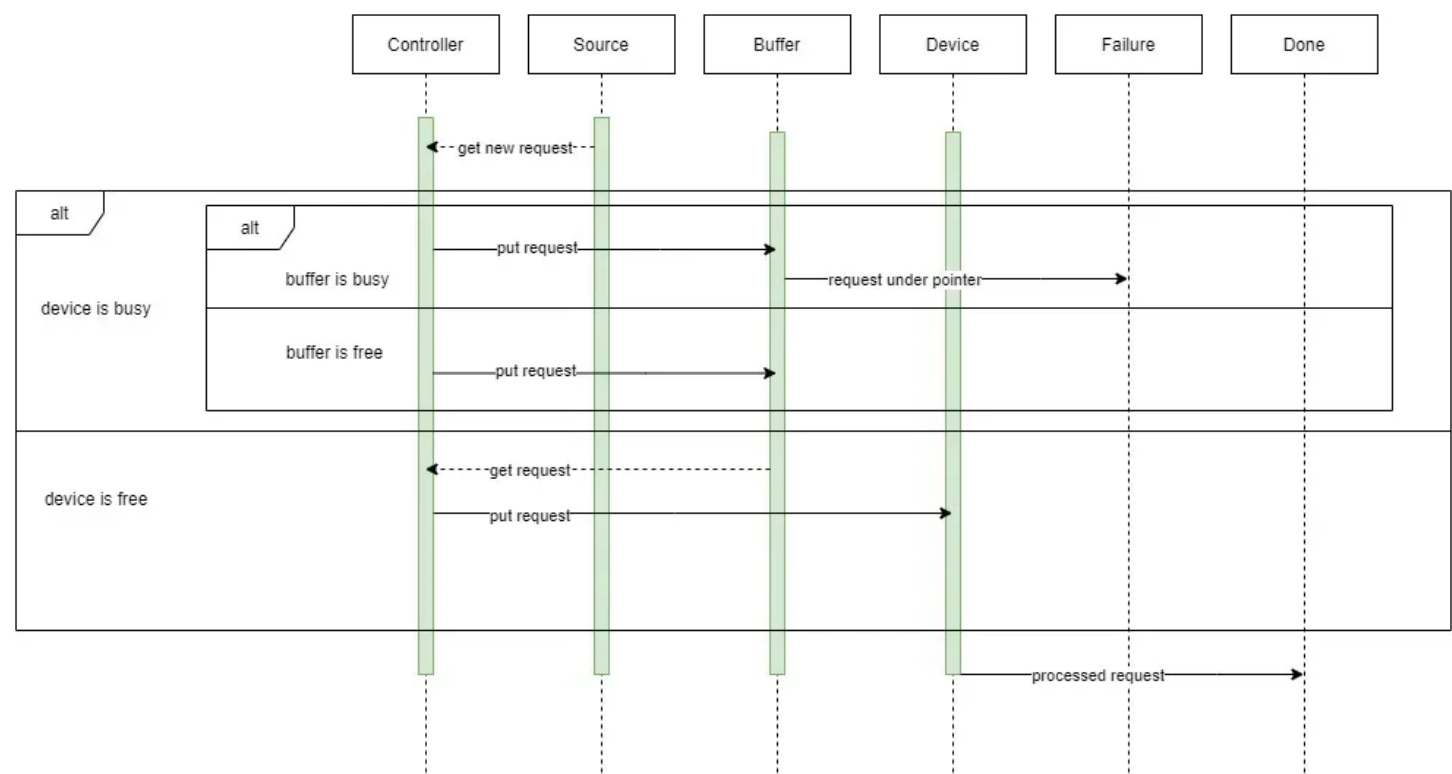
Динамическое отражение результатов:

ОД1 — календарь событий, буфер и текущее состояние;

Отражение результатов после сбора статистики:

ОР2 — сводная таблица результатов.

# Sequence-диаграмма



## Законы распределения

Равномерный закон распределения:

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

В программе:

$$\underline{((float)grand()/ (float)RAND\_MAX)*(betta - alpha) + alpha}$$

Пуассоновский закон распределения:

$$\tau = -1/\lambda \cdot \text{Ln}(r),$$

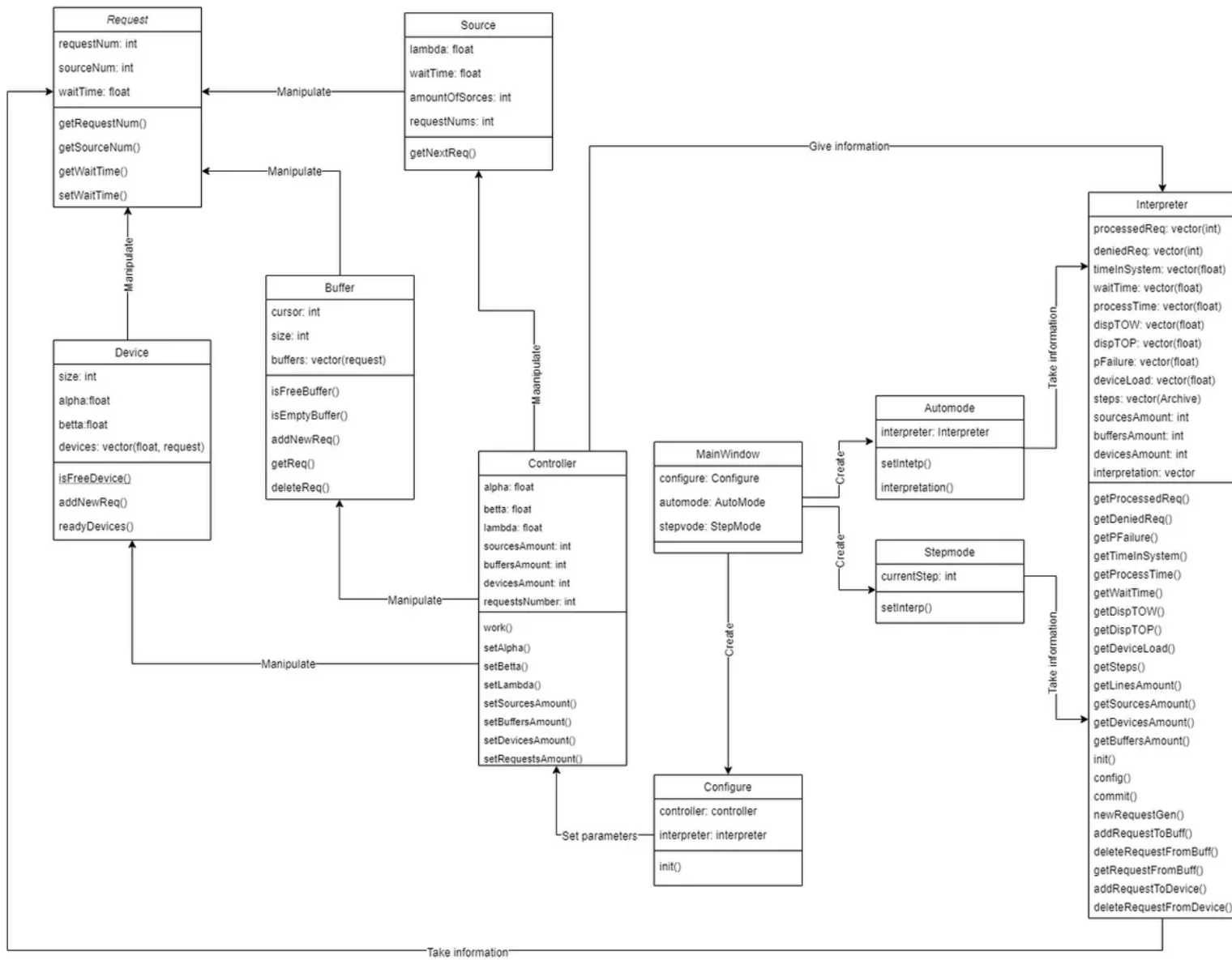
В программе:

$$\underline{(-1/lambda)*std::log(((float)grand()/ (float)RAND\_MAX))}$$

## Ограничения и требуемые характеристики

- Вероятность отказа не более 10%.
- Загрузка приборов более 90%.

# Диаграмма классов



## Модульная структура

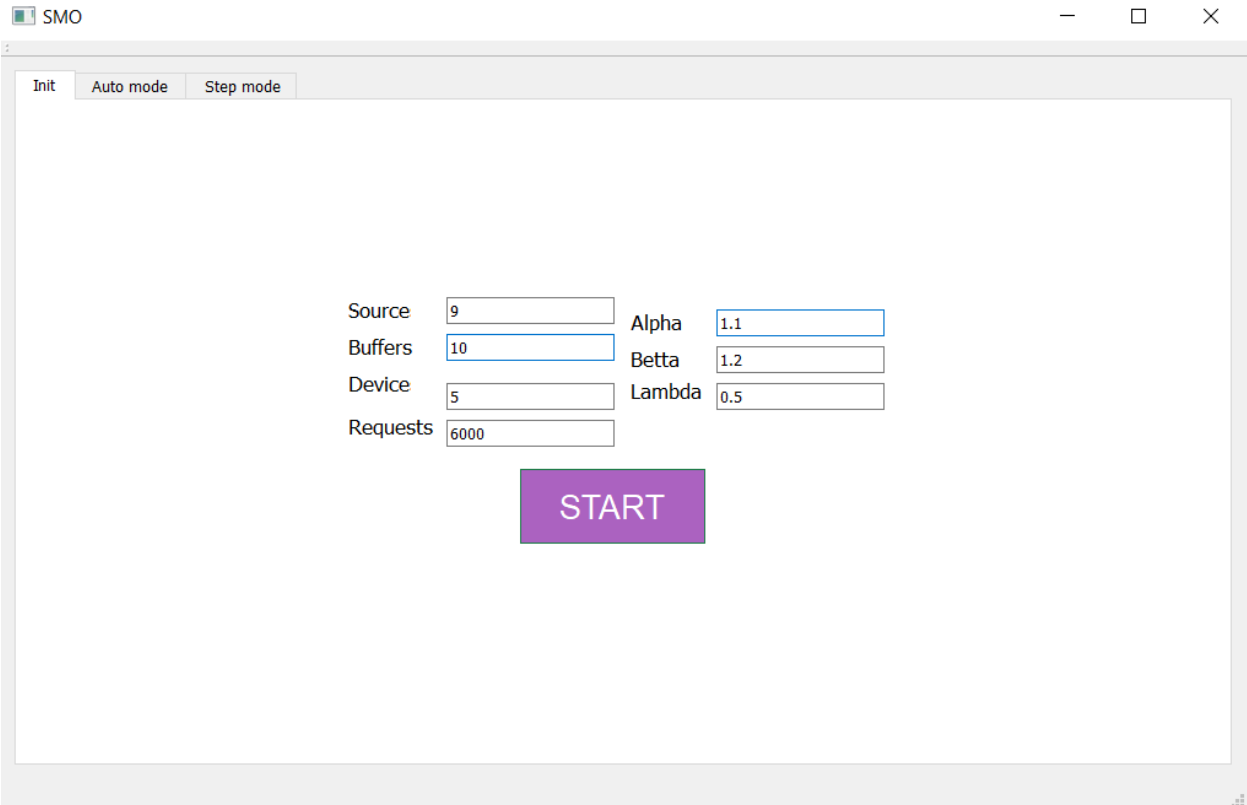
Работа выполнена в среде Qt Creator 4.12.0 на языке программирования C++, интерфейс с помощью графической библиотеки Qt. Приложение – объектно-ориентированное, содержит следующие классы:

- Request — описание заявки, методы для их получения;
- Source — методы создания заявки и генерации времени заявки;
- Buffer — методы проверки свободного места в буфере, добавления заявки в буфер, выбора заявки из буфера;
- Device — методы проверки свободных приборов, загрузки заявки на прибор, удаления заявки из прибора;
- Interpreter — запись событий и подсчеты характеристик для пошагового и автоматического отображения;
- Controller — реализует основной цикл работы системы;
- Configure — задает значения основных элементов;
- StepMode — пошаговый режим, состояние системы в разные моменты времени;
- AutoMode – автоматический режим, состояние системы после симуляции;
- MainWindow — запуск графического интерфейса, главная форма.



## Описание работы программы

При запуске программы появляется окно:



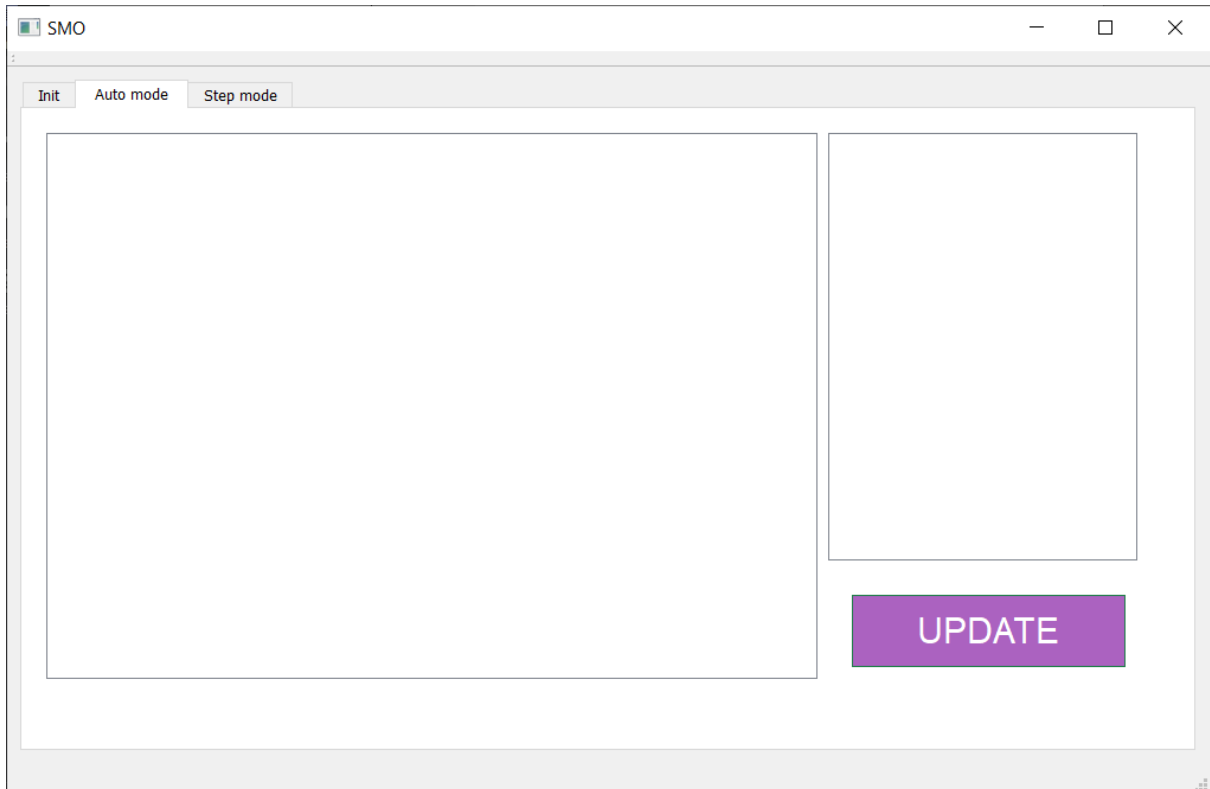
The screenshot shows a window titled "SMO" with three tabs: "Init", "Auto mode", and "Step mode". The "Init" tab is active. It contains a form with the following parameters and values:

Parameter	Value
Source	9
Buffers	10
Device	5
Requests	6000
Alpha	1.1
Betta	1.2
Lambda	0.5

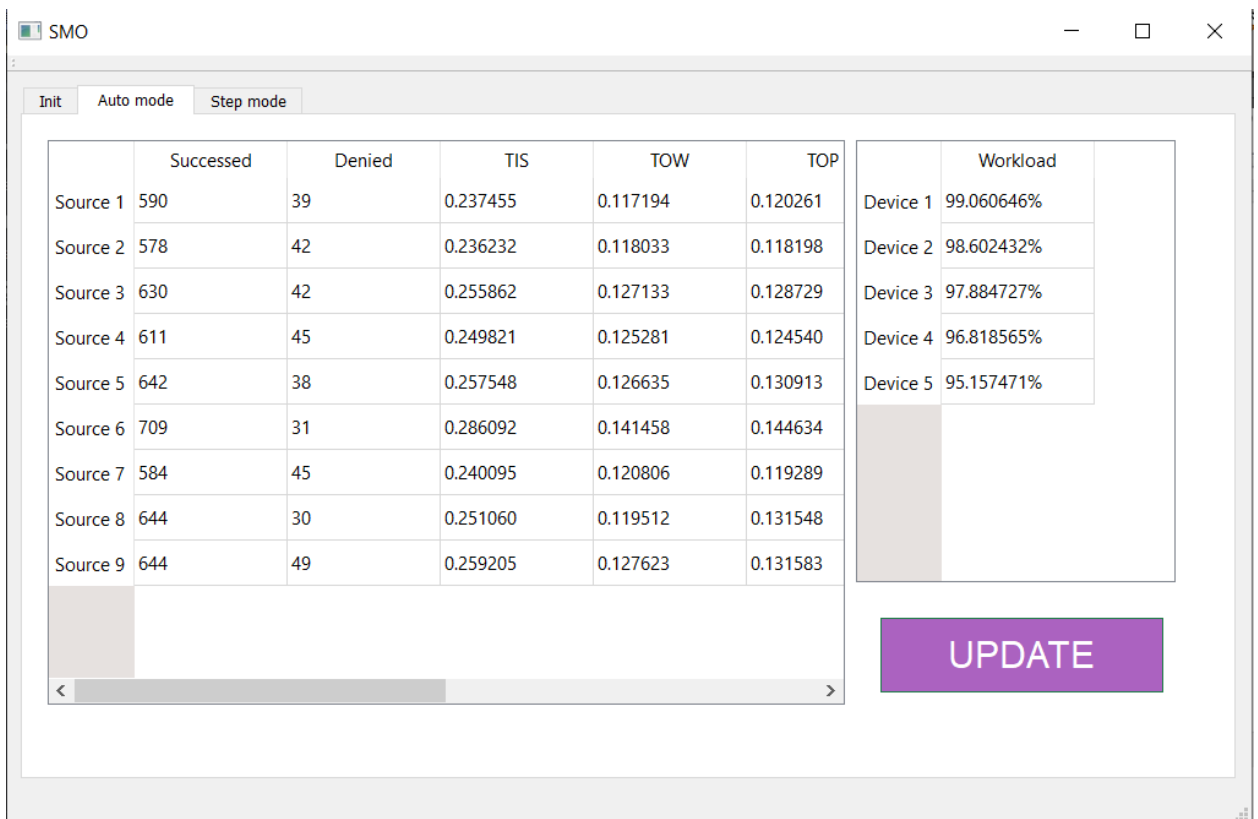
Below the input fields is a large purple button labeled "START".

Мы находимся на вкладке «Init». Здесь необходимо задать параметры для СМО и нажать кнопку «START» для запуска программы. На вкладках «Auto mode» и «Step mode» будут отображены результаты работы нашей системы в итоговых таблицах и состояниях на каждом шаге соответственно.

## Вкладка «Auto mode»:



При нажатии на кнопки «UPDATE» происходит симуляция работы и все аналитические данные появляются в таблицах.



Вкладка «Step mode»:

SMO

Init

Auto mode

Step mode

Buffer

Device

Comment:

Time: 0

Step: 0

All Steps: 0

UPDATE

BACK

NEXT

В данном окне можно переходить к предыдущему или следующему шагу выполнения в системе, чтобы увидеть состояние системы. Для появления результатов так же необходимо нажать «UPDATE».

SMO

Init

Auto mode

Step mode

Buffer

Device

	State	Request
Buffer 1	Free	-
Buffer 2	Busy	8.2
Buffer 3	Free	-
Buffer 4	Free	-
Buffer 5	Free	-
Buffer 6	Free	-
Buffer 7	Free	-
Buffer 8	Free	-
Buffer 9	Free	-

	State	Request
Device 1	Busy	4.2
Device 2	Busy	6.2
Device 3	Busy	3.1
Device 4	Free	-
Device 5	Free	-

Comment:

Create request 8.2 and load in buffer 2

Time: 3.715872

Step: 23

All Steps: 11994

UPDATE

BACK

NEXT

## Результаты работы

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

$$N = \frac{t_{\alpha}^2 (1-p)}{p\delta^2}$$

где  $p$  — вероятность отказа заявок в обслуживании,

$t_{\alpha} = 1.643$  для  $\alpha = 0.9$ ,

$\delta = 0.1$  — относительная точность.

По результатам работы программы получено, что в большинстве случаев для достижения заданной точности необходимо около 3000 заявок. Однако, в случаях, когда  $p$  мало ( $<0.05$ ) для достижения точности в 10% может потребоваться существенно больше заявок (6000-7000).

## Анализ результатов:

Так как целью моделирования является выбор конфигурации системы, требующей наименьшее количество ресурсов и обрабатывающей максимальный поток информации, то начнем с проверки конфигурации с максимальным числом источников, минимальным числом приборов и максимальным размером буфера.

Sources	Devices	Buffer	Alpha	Betta	Lambda	Workload	Pr. Failure
15	1	20	1.1	1.2	0.5	100%	99%
12	2	15	1.1	1.2	0.5	99%	75%
9	3	10	1.1	1.2	0.5	98%	36%

Мы получили необходимые показатели загруженности приборов, но величина вероятности отказа еще слишком высока, несмотря на уменьшение числа источников и увеличения приборов. Пусть прибор с этими показателями производительности будет второго класса. Попробуем добавить еще приборов.

Медленные приборы второго класса:

Sources	Devices	Buffer	Alpha	Betta	Lambda	Workload	Pr. Failure
9	3	10	1.1	1.2	0.5	98%	36%
9	4	10	1.1	1.2	0.5	98%	23%
9	5	10	1.1	1.2	0.5	97%	8%

Так как загруженность уменьшается, то можно попробовать:

- Увеличить количество источников;
- Заменить эти приборы на меньшее количество приборов с большей производительностью.

Увеличим производительность приборов, быстрые приборы первого класса:

Sources	Devices	Buffer	Alpha	Betta	Lambda	Workload	Pr. Failure
9	3	10	1.0	1.0	0.5	100%	35%
9	4	10	1.0	1.0	0.5	100%	12%
9	5	10	1.0	1.0	0.5	96%	2%

Доменом системы выбрана часть парка аттракционов. Прибор – тележка с сахарной ватой. Заявки – люди, которые хотят получить сахарную вату. Вместо ожидания в очереди им предлагается кататься на колесе обозрения. Таким образом буфер – колесо обозрения. Источник – калитка в заборе, который ограждает эту территорию.

Основываясь на полученных данных, можно сделать вывод, что оптимальное соотношение количества калиток, к количеству кабинок на колесе, к числу тележек с сахарной ватой составляет 9:10:5.

Рассчитаем примерную стоимость приборов.

Прибор	Производительность	Стоимость
Второй класс	Медленные	1000 руб
Первый класс	Быстрые	2000 руб

Как можно увидеть из таблиц мы получаем итоговую стоимость всех приборов 5000 руб., а при использовании приборов с более высокой производительностью получаем итоговую стоимость 10000 руб.

Увеличение скорости работы тележек приведёт к уменьшению вероятности отказов, однако при этом уменьшится загруженность. Также более эффективные тележки стоят в два раза дороже и увеличение производительности почти не сказывается на загруженности и вероятности отказа. Если увеличить число калиток, то возрастет загруженность на прибор и получится большая вероятность отказа, от которой мы старались уйти. Следовательно исходное соотношение 9:10:5 и использование использовать приборы второго класса более выгодно.

## **Вывод**

В ходе курсовой работы была написана система массового обслуживания на языке программирования C++ с использованием графической библиотеки Qt. С помощью данной программы была проанализирована реальная система и подобрана наиболее выгодная конфигурация данной системы.

## **Исходный код**

<https://github.com/eliizaveta/SMO>