

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ  
ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ

## КУРСОВАЯ РАБОТА

по дисциплине «Архитектура программных систем»

**ВАРИАНТ 15**

Выполнил  
студент гр. 3530904/80101

Пылаев Я. С.

Руководитель

Дробинцев Д. Ф.

Санкт-Петербург  
2020

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
1.1	Подход Дельта-Т . . . . .	2
1.2	Подход особых событий . . . . .	2
<b>2</b>	<b>Формализованная схема</b>	<b>3</b>
<b>3</b>	<b>Подготовительная работа</b>	<b>4</b>
3.1	Расшифровка формулы 15-го варианта задания . . . . .	4
3.1.1	Параметры элементов модели . . . . .	4
3.1.2	Описание дисциплин постановки и выбора . . . . .	4
3.1.3	Виды отображения результатов работы программной модели . . . . .	4
3.2	Входные параметры . . . . .	4
3.3	Формулировка задания . . . . .	5
<b>4</b>	<b>Реализация</b>	<b>5</b>
4.1	Документация на ПО . . . . .	5
4.1.1	Стек технологий . . . . .	5
4.1.2	Описание бизнес-домена . . . . .	5
4.1.3	Sequence-диаграмма . . . . .	6
4.2	Исходный код . . . . .	7
4.3	Диаграмма классов . . . . .	7
<b>5</b>	<b>Результаты</b>	<b>8</b>
5.1	Автоматический режим . . . . .	8
5.2	Пошаговый режим . . . . .	9
5.3	Анализ результатов . . . . .	9
5.4	Рекомендации . . . . .	10
<b>6</b>	<b>Заключение</b>	<b>10</b>

# 1 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 ВС: *аналитические, аналоговые, физические* и ***имитационные***. 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). Рассмотрим два подхода к построению моделирующего алгоритма:

## 1.1 Подход Дельта-Т

Универсальный метод построения моделирующего алгоритма, когда состояние объекта проверяется через фиксированный интервал модельного времени. Суть его заключается в следующем: в каждый момент времени  $t_i = t_{i-1} + \Delta t_{i-1}$  получают приближенные значения характеристик исследуемого объекта.  $\Delta t$  можно получить детерминированным способом.

Основной критерий выбора  $\Delta t$  — он должен быть настолько мал, чтобы не пропустить событие в моделируемой системе, которое должно быть учтено при выбранной детальности моделирования. *Метод неэффективен*, т. к. при его использовании постоянно проверяется состояние объектов моделирования, не изменяющихся при этом, особенно при малых  $\Delta t$ .

## 1.2 Подход особых событий

При исследовании реальной системы интервалы, в которых состояние не меняется, не представляют интереса. Имеют значение только значимые (изменяющие состояние) переходы системы 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, т. е. момент прекращения генерации заявок источниками.

Использование принципа особых состояний для построения имитационной модели ВС наиболее эффективно. В настоящей работе предлагается использовать именно этот принцип.

## 2 Формализованная схема

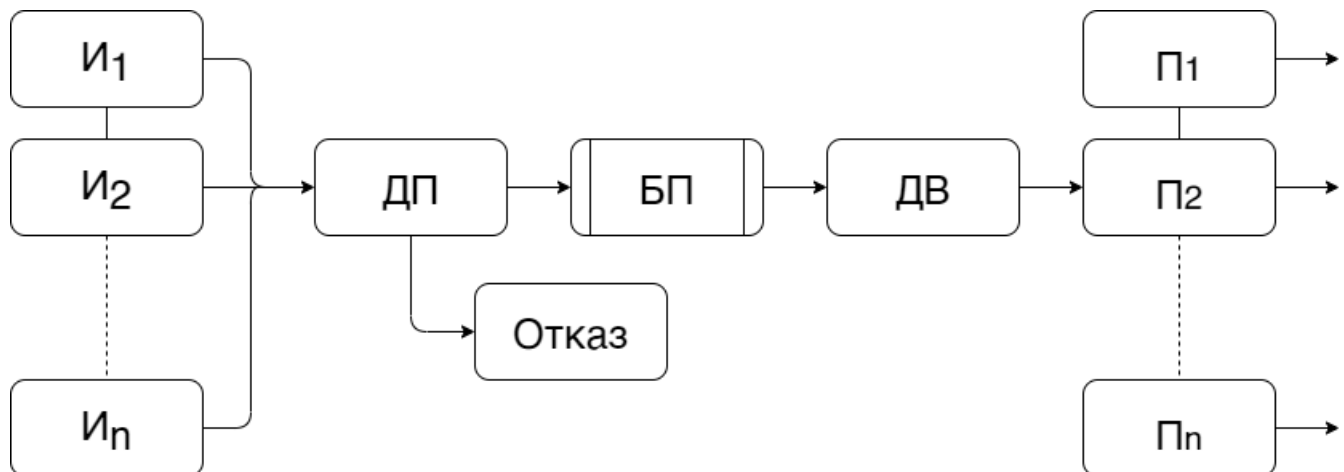


Figure 1: Формализованная схема ВС

- $I_i (i = 1...n)$  — источник заявок, который генерирует заявки, а все вместе  $n$  источников создают входной поток заявок в систему. Источники могут быть двух типов — конечные и бесконечные. Различие лишь в способе генерации заявок.
- $P_i (i = 1...n)$  — приборы, которые обслуживают заявки и создают выходной поток заявок после обслуживания.
- БП — буферная память (место для хранения очереди заявок). Буферная память может быть двух типов: общая и зонная.
- ДП — диспетчер постановки заявок в очередь. Выполняет две задачи:
  1. отправляет заявку на обслуживание или в буферную память в случае отсутствия свободных приборов;
  2. организует отказ или выбивание заявки из БП, если в буфере не осталось свободных мест.
- ДВ — диспетчер выбора заявок из очереди. Функционирование ВС моделируется через прохождение заявки, сгенерированной источником, по системе. Также выполняет две задачи:
  1. выбирает прибор, на котором обслуживает заявку;
  2. выбирает заявку из БП, если она там есть.

Проследим 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 источники только генерируют заявки, приборы их только обслуживают, а логика прохождения заявок по системе определяется ДП и ДВ.

Рассмотренный принцип функционирования ВС основан на имитации единого многомерного процесса обработки заявок. Процессы моделируются через последовательность состояний системы во времени.

Моделирующий алгоритм позволяет имитировать поведение, т. е. изменение во времени состояний объектов и событий, составляющих исследуемый процесс.

### 3 Подготовка работа

#### 3.1 Расшифровка формулы 15-го варианта задания

ИБ	ИЗ1	ПЗ2	Д1ОЗ1	Д1ОО2	Д2П1	Д2Б5	ОР1	ОД3
----	-----	-----	-------	-------	------	------	-----	-----

##### 3.1.1 Параметры элементов модели

- ИБ — тип источников — бесконечный;
- ИЗ1 — закон генерации заявок источников — пуассоновский закон распределенияпуассоновский закон распределения;
- ПЗ2 — закон распределения времени работы приборов — равномерный.

##### 3.1.2 Описание дисциплин постановки и выбора

- Д1ОЗ1 — дисциплина буферизации — заполнение буферной памяти по кольцу;
- Д1ОО2 — дисциплина отказа — приоритет по номеру источника;
- Д2Б5 — дисциплина постановки на обслуживание — приоритет по номеру источника, заявки в пакете;
- Д2П1 — выбор прибора — приоритет по номеру прибора.

##### 3.1.3 Виды отображения результатов работы программной модели

- ОД3 — отображение динамики функционирования модели — временные диаграммы, текущее состояние;
- ОР1 — отображение результатов — сводная таблица результатов.

#### 3.2 Входные параметры

- $n_1$  — количество источников;
- $n_2$  — количество приборов;
- $t$  — время обслуживания;
- $\lambda$  — интенсивность.

### 3.3 Формулировка задания

В работе требуется построить моделирующий алгоритм ВС, реализовать его на языке программирования, отладить модель и представить в отчете результаты, полученные с относительной точностью 10% и доверительной вероятностью 0.9, сопровождаемые анализом, графиками и выводами.

Таким образом, необходимо получить и проанализировать следующие характеристики ВС в зависимости от общей загрузки системы  $\rho$  :

1. Количество сгенерированных каждым источником требований.
2. Вероятность отказа в обслуживании заявок каждого источника  $P_{otk}$ .
3. Среднее время пребывания заявок каждого источника в системе  $T_{system}$ .
4. Среднее время ожидания заявок каждого источника в системе  $T_{waiting}$ .
5. Среднее время обслуживания заявок каждого источника  $T_{service}$ .
6. Дисперсии двух последних характеристик.
7. Коэффициент использования устройств  $K_{use}$ .

На основе этого исследования, ориентируясь на предъявляемые требования к результатам работы системы, т.е. к ее выходным характеристикам, необходимо найти нужную архитектуру системы, состоящую из ее элементов и работающую в соответствии с заданными требованиями.

## 4 Реализация

### 4.1 Документация на ПО

#### 4.1.1 Стек технологий

Проект представляет собой консольное приложение, написанное на объектно-ориентированном языке программирования Java 8 в среде разработки IntelliJ Idea.

#### 4.1.2 Описание бизнес-домена

ООН приехали в Пакистан для проведения медицинской работы по вакцинированию жителей местных деревень от коронавируса. Однако многие поселения расположены в крайне удаленных местах среди гор, поэтому доставить большое количество врачей не предоставляется возможности. Нужно рассчитать для деревень с разной численностью населения оптимальное количество членов медицинского штаба.

Обслуживание: в первую очередь нужно поставить прививки пожилым людям и детям, соответственно это приоритеты 4 и 3. Далее работники рудных предприятий и священники - 2. Оставшиеся жители имеют приоритет 1. Однако как быть с уже больными людьми? Они имеют наивысший приоритет 5, поэтому обслуживаются практически без очереди.

#### 4.1.3 Sequence-диаграмма

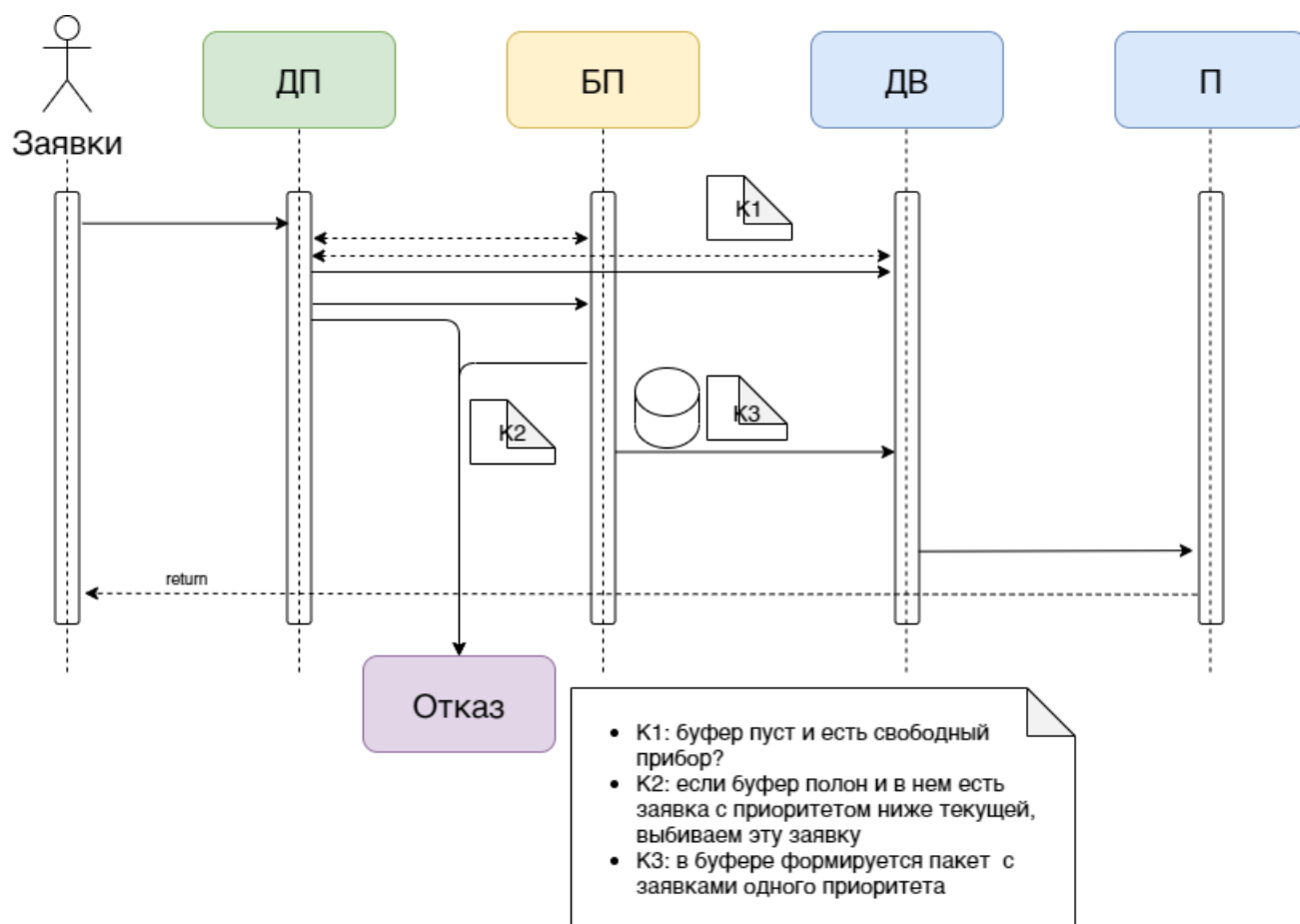


Figure 2: Sequence-диаграмма

## 4.2 Исходный код

<https://github.com/Yang-Pi/QueueingTheory>

## 4.3 Диаграмма классов

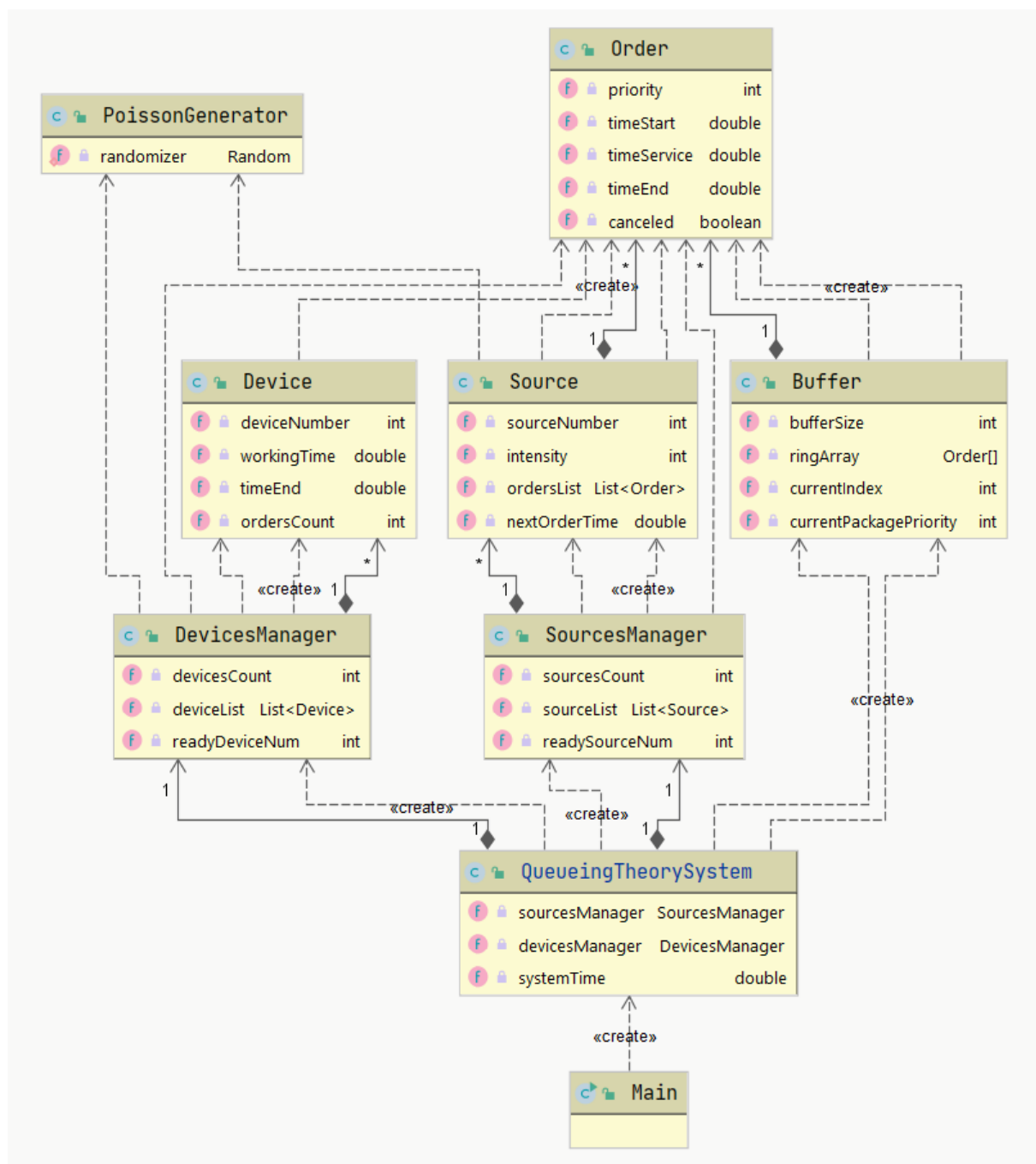


Figure 3: Диаграмма классов



## 5 Результаты

### 5.1 Автоматический режим

```
##### RESULTS
----- ORDERS COUNT:
Source #0 created all 672 orders which canceled 41
Probability of cancel: 0.06101190476190476
Source #1 created all 707 orders which canceled 1
Probability of cancel: 0.0014144271570014145
Source #2 created all 459 orders which canceled 0
Probability of cancel: 0.0
Source #3 created all 437 orders which canceled 0
Probability of cancel: 0.0
Source #4 created all 225 orders which canceled 0
Probability of cancel: 0.0
----- AVG TIMES:
--- Source #0
avg time in system 0.8053398273563461
avg waiting time 0.3881109493276692
dispersion 0.3050269616062526
avg service time 0.4172288780286762
dispersion 0.06763974381785207
--- Source #1
avg time in system 0.7492384208189812
avg waiting time 0.3061868751167984
dispersion 0.12811559765980113
avg service time 0.44305154570218436
dispersion 0.0681343330741527

--- Source #2
avg time in system 0.7435438526351402
avg waiting time 0.2737230022111422
dispersion 0.09600811824060553
avg service time 0.4698208504239972
dispersion 0.07984222523393864
--- Source #3
avg time in system 0.6985629685148681
avg waiting time 0.2181994510903356
dispersion 0.05017119899713954
avg service time 0.48036351742453115
dispersion 0.08773446260501967
--- Source #4
avg time in system 0.5730312555715424
avg waiting time 0.15807975048153075
dispersion 0.02633439857489145
avg service time 0.4149515050900116
dispersion 0.03945668385509081
----- DEVICE USING
Device #0 0.8108277458155096
Device #1 0.9148890225095532
Device #2 0.712234831734521
Device #3 0.8512761321729765
Device #4 0.7281403181814099
Device #5 0.7249907984878783
```

Figure 4: Результаты в автоматическом режиме

## 5.2 Пошаговый режим

```
CREATE ORDER: 1 0.2176376001107348; next order 0.9514240781238725
DEVICE 5 set order 1 0.7696505322331444
Next (n) or Auto (a)
n
----- BUFFER: 1 package 1 index
null ; null ; null ; null ; null ; null ; null ; null ; null ; null ;
-----
----- SOURCES CLOSEST EVENTS:
Source #0 future order time 0.6019785803619118
Source #1 future order time 0.9514240781238725
Source #2 future order time 0.6302971516722741
Source #3 future order time 0.5143299136339636
Source #4 future order time 0.46573260320206206
-----
----- DEVICE CLOSEST EVENTS:
Device #0 end work at 0.0
Device #1 end work at 0.0
Device #2 end work at 0.0
Device #3 end work at 0.0
Device #4 end work at 0.0
Device #5 end work at 0.7696505322331444 ACTIVE
-----
SOURCES:
All orders 0, canceled ones 0;
All orders 1, canceled ones 0;
All orders 0, canceled ones 0;
All orders 0, canceled ones 0;
All orders 0, canceled ones 0;
```

Figure 5: Пошаговый режим

## 5.3 Анализ результатов

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 заявкам в обслуживании,  $t_{\alpha} = 1.643$  для  $\alpha = 0.9$ ,  $\beta = 0.1$  – относительная точность.

В результате исследования было получено, что для достижения заданной точности при вероятности отказа не более 10% необходимо около 2500 заявок. Ориентируясь на бизнес-домен, это число можно изменять в зависимости от количества жителей населенного пункта. Соответственно, если увеличивать точность, сохраняя ограничения для вероятности отказа, то необходимо изменять и другие параметры системы, например, интенсивность заявок от каждого источника или количество приборов.

## 5.4 Рекомендации

Для улучшения точности моделирования можно использовать более надежный генератор случайных чисел, применяемый в пуассоновском законе распределения. В программе был использован класс `java.util.Random`, а можно через API воспользоваться сервисом [RANDOM.ORG](https://RANDOM.ORG), в основе алгоритма генерации случайных чисел которого лежат параметры атмосферного шума.

## 6 Заключение

В ходе курсовой работы были изучены основы функционирования систем массового обслуживания, а также архитектура соответствующих вычислительных систем. При моделировании системы массового обслуживания использовался метод особых состояний. Также были закреплены навыки разработки объектно-ориентированного программного обеспечения на языке **Java** и навыки оптимизации вычислительной системы.

<https://github.com/Yang-Pi/QueueingTheory>