



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 2

по курсу «Системы массового обслуживания»

ВАРИАНТ 44

Тема: _____ **«Одноканальные системы массового обслуживания
с бесконечной очередью»** _____

Выполнил:
Студент 4-го курса
Мусатов Д. Ю

Группа: КМБО-03-18

МОСКВА – 2021

Содержание

1	Задание	3
2	Краткие теоретические сведения	6
2.1	СМО с бесконечной очередью	6
2.2	Описание рассматриваемых СМО	8
2.3	Средства языка программирования	8
3	Результаты расчётов	9
3.1	Система массового обслуживания (D M 1)	9
3.2	Система массового обслуживания (M D 1).	14
3.3	Система массового обслуживания (M M 1).	19
4	Анализ результатов	24
4.1	Система массового обслуживания (D M 1)	24
4.2	Система массового обслуживания (M D 1)	25
4.3	Система массового обслуживания (M M 1)	26
5	Список литературы	27
6	Приложение	28

1 Задание

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени t характеризуется числом заявок, находящихся в СМО.

События могут быть двух типов: 1— появление в СМО новой заявки, 2— завершение обслуживания заявки прибором (при этом прибор освобождается, и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в прибор). Если при появлении в СМО новой заявки прибор свободен, то она сразу же принимается на обслуживание прибором, в противном случае заявка становится в очередь типа FIFO.

1. Система массового обслуживания (D|M|1). Дано:

- время между приходом заявок ΔT_3 (заданная постоянная величина);
- параметр μ показательного распределения времени обслуживания заявки прибором.

В момент поступления каждой заявки на обслуживание в прибор определяется время её обслуживания $t_{\text{обсл}}$ в соответствии с показательным законом распределения с заданным параметром μ .

Предполагается, что в начальный момент времени $t = 0$ в СМО нет заявок, т.е. состояние системы 0, и через заданное время ΔT_3 в СМО поступит первая заявка (произойдет событие с номером 1). Момент наступления первого события (типа 1) равен $t_{\text{собр}}(1) = \Delta T_3$, в этот момент определяется время обслуживания $t_{\text{обсл}}(1)$ заявки 1 в соответствии с показательным законом распределения с параметром μ . После события 1 система находится в состоянии 1.

2. Система массового обслуживания (M|D|1).

Дано:

- среднее число заявок λ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром λ);
- время обслуживания заявки прибором $T_{\text{об}}$ (заданная постоянная величина).

Предполагается, что в начальный момент времени $t = 0$ система находится в состоянии 0 и в этот момент определяется время поступления в систему первой заявки $t_3(1)$ в соответствии с показательным законом распределения с параметром λ .

3. Система массового обслуживания (M|M|1).

Дано:

- среднее число заявок λ , поступающих за единицу времени (время между приходом заявок имеет показательное распределение с параметром λ);

- параметр μ показательного распределения времени обслуживания заявки прибором.

Предполагается, что в начальный момент времени $t = 0$ система находится в состоянии 0 и в этот момент определяется время поступления в систему первой заявки $t_z(1)$ в соответствии с показательным законом распределения с параметром λ , а в момент поступления каждой заявки на обслуживание в прибор определяется время её обслуживания $t_{\text{обсл}}(1)$ в соответствии с показательным законом распределения с параметром μ .

Требуется:

1. Провести моделирование первых 100 событий в развитии каждой системы.
2. Составить таблицу 1 с данными о событиях:
 - номер события l ;
 - момент наступления события $t_{\text{соб}}(l)$;
 - тип события $Type(l)$;
 - состояние СМО $C(l)$ после события l ;
 - оставшееся время $t_{\text{ост}}(l)$ обслуживания прибором заявки после события l (если после события прибор свободен, то $t_{\text{ост}}(l) = -1$);
 - время ожидания $t_{\text{ожз}}(l)$, через которое после события l в СМО появится новая заявка;
 - номер заявки $j(l)$, участвующей в событии l .
3. Составить таблицу 2 с данными о всех поступивших заявках:
 - номер заявки j ;
 - момент $t_z(j)$ появления заявки j в СМО;
 - номер места в очереди $q(j)$, на которое попала заявка j (если заявка сразу начала обслуживаться, то номер места в очереди $q(j) = 0$);
 - время пребывания заявки в очереди $t_{\text{оч}}(j)$;
 - момент начала обслуживания заявки $t_{\text{ноб}}(j)$;
 - время обслуживания заявки $t_{\text{обсл}}(j)$;
 - момент $t_{\text{коб}}(j)$ окончания обслуживания заявки j и выхода её из СМО.
4. Найти:
 - число заявок $J(100)$, поступивших в СМО на интервале $[0, t_{\text{соб}}(100)]$;
 - число $JF(100)$ полностью обслуженных заявок на интервале $[0, t_{\text{соб}}(100)]$;

- среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\bar{z}(100) = \frac{1}{100} \sum_{l=1}^{100} z(l)$, где $z(l)$ – число заявок в СМО после события l ;
- среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $t_{\text{оч}}^-(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j)$;
- среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$, которое находится по формуле $\bar{t}_{\text{СМО}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_3(j)]$;
- коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ (отношение времени простоя прибора на интервале $[0, t_{\text{cob}}(100)]$ к $t_{\text{cob}}(100)$);
- относительные частоты пребывания СМО в состояниях

i	$\nu_i(100)$
0	$\nu_0(100)$
1	$\nu_1(100)$
\dots	\dots

где i – состояние СМО $\nu_i(100)$ – отношение числа попаданий СМО в состояние i за 100 событий к 100.

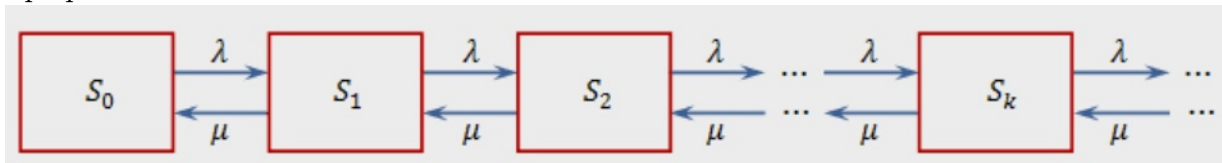
2 Краткие теоретические сведения

2.1 СМО с бесконечной очередью

Имеется одноканальная СМО с очередью, на которую не наложены никакие ограничения (ни по длине очереди, ни по времени ожидания). Поток заявок, поступающих в СМО, имеет интенсивность λ , а поток обслуживания - интенсивность μ .

Система может находиться в одном из состояний $S_0, S_1, S_2, \dots, S_k, \dots$, по числу заявок, находящихся в СМО: S_0 — канал свободен; S_1 — канал занят (обслуживает заявку), очереди нет; S_2 — канал занят, одна заявка стоит в очереди; ... S_k — канал занят, $(k - 1)$ заявок стоят в очереди и т.д.

Граф состояний СМО:



Это процесс гибели и размножения, но с бесконечным числом состояний, в котором интенсивность потока заявок равна λ , а интенсивность потока обслуживания μ .

Прежде чем записать формулы предельных вероятностей, необходимо быть уверенным в их существовании (когда время $t \rightarrow \infty$, очередь может неограниченно возрасть).

— если $\rho < 1$, т.е. среднее число приходящих заявок меньше среднего числа обслуженных заявок (в единицу времени), то предельные вероятности существуют.

— если $\rho \geq 1$, очередь растет до бесконечности.

Для определения предельных вероятностей состояний необходимо воспользоваться формулами:

$$\begin{aligned}
 p_0 &= \left(1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} + \dots + \frac{\lambda_{n-1,n}\dots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\dots\lambda_{21}\lambda_{10}} \right)^{-1} \\
 p_1 &= \frac{\lambda_{01}}{\lambda_{10}} p_0 \\
 p_2 &= \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} p_0 \\
 &\dots \\
 p_n &= \frac{\lambda_{n-1,n}\dots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\dots\lambda_{21}\lambda_{10}} p_0
 \end{aligned} \tag{1}$$

для процесса гибели и размножения (формулы для конечного числа состояний системы).

Тогда:

$$p_0 = \left[1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu} \right)^2 + \dots + \left(\frac{\lambda}{\mu} \right)^k + \dots \right]^{-1} = (1 + \rho + \rho^2 + \dots + \rho^k + \dots)^{-1} \tag{2}$$

Так как предельные вероятности существуют лишь при $\rho < 1$, то геометрический ряд со знаменателем $\rho < 1$, записанный в скобках в формуле (2), сходится к сумме, равной $\frac{1}{1-\rho}$. Поэтому **предельные вероятности состояний**:

$$\begin{aligned}
 p_0 &= 1 - \rho \\
 p_1 &= \rho(1 - \rho) \\
 p_2 &= \rho^2(1 - \rho) \\
 &\dots \\
 p_k &= \rho^k(1 - \rho), \quad \dots
 \end{aligned}
 \tag{3}$$

Среднее число заявок в системе L_{sist} определяется по формуле математического ожидания, которая с учетом (3) примет вид

$$L_{sist.} = \frac{\rho}{1 - \rho}. \tag{4}$$

Среднее число заявок в очереди $L_{och.}$

$$L_{och} = L_{sist.} - L_{ob.} \tag{5}$$

где $L_{ob.}$ - среднее число заявок, находящихся под обслуживанием.

Среднее число заявок под обслуживанием определяется по формуле математического ожидания числа заявок под обслуживанием, принимающего значения 0 (если канал свободен) либо 1 (если канал занят):

$$L_{ob.} = 0 \cdot p_0 + 1 \cdot (1 - p_0) = 1 - p_0 = \rho \tag{6}$$

Теперь по формуле (5) с учетом (4) и (6)

$$L_{och} = \frac{\rho^2}{1 - \rho}. \tag{7}$$

При любом характере потока заявок, при любом распределении времени обслуживания, при любой дисциплине обслуживания:

Среднее время пребывания заявки в системе равна среднему числу заявок в системе, деленному на интенсивность потока заявок, т.е.

$$T_{sist.} = \frac{1}{\lambda} \cdot L_{sist.} = \frac{\rho}{\lambda(1 - \rho)} \tag{8}$$

Среднее время пребывания заявки в очереди равна среднему числу заявок в очереди, деленному на интенсивность потока заявок, т.е.

$$T_{och.} = \frac{1}{\lambda} \cdot L_{och.} = \frac{\rho^2}{\lambda(1 - \rho)}. \tag{9}$$

2.2 Описание рассматриваемых СМО

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени t характеризуется числом заявок, находящихся в СМО.

События могут быть двух типов: 1 - появление в СМО новой заявки, 2 - завершение обслуживания заявки прибором (при этом прибор освобождается и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в прибор). Если при появлении в СМО новой заявки прибор свободен, то она сразу же принимается на обслуживание прибором, в противном случае заявка становится в очередь типа FIFO.

2.3 Средства языка программирования

В программе расчёта был использован язык программирования Python. Работа осуществлялась в среде Jupyter Notebook.

Функции, использованные в ходе работы:

- `numpy.random.exponential (1/x)`— генерация случайной величины или случайной выборки из показательного распределения;
- `round(x,n)` — округление числа x до n -ого знака

3 Результаты расчётов

3.1 Система массового обслуживания (D|M|1)

Вариант №44.

Начальные данные: $\Delta T_3 = 0.829$, $\mu = 1.208$

Таблица 1.

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
1	0.829	1	1	0.27001	0.829	1
2	1.09901	2	0	-1	0.55899	1
3	1.658	1	1	1.25862	0.829	2
4	2.487	1	2	0.42962	0.829	3
5	2.91662	2	1	0.26547	0.39938	2
6	3.18209	2	0	-1	0.13391	3
7	3.316	1	1	0.59464	0.829	4
8	3.91064	2	0	-1	0.23436	4
9	4.145	1	1	0.07857	0.829	5
10	4.22357	2	0	-1	0.75043	5
11	4.974	1	1	0.53612	0.829	6
12	5.51012	2	0	-1	0.29288	6
13	5.803	1	1	0.19089	0.829	7
14	5.99389	2	0	-1	0.63811	7
15	6.632	1	1	0.26502	0.829	8
16	6.89702	2	0	-1	0.56398	8
17	7.461	1	1	0.27785	0.829	9
18	7.73885	2	0	-1	0.55115	9
19	8.29	1	1	0.17871	0.829	10
20	8.46871	2	0	-1	0.65029	10
21	9.119	1	1	0.71349	0.829	11
22	9.83249	2	0	-1	0.11551	11
23	9.948	1	1	1.60324	0.829	12
24	10.777	1	2	0.77424	0.829	13
25	11.55124	2	1	0.39149	0.05476	12
26	11.606	1	2	0.33673	0.829	14

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	11.94273	2	1	1.07789	0.49227	13
28	12.435	1	2	0.58562	0.829	15
29	13.02062	2	1	0.35339	0.24338	14
30	13.264	1	2	0.11001	0.829	16
31	13.37401	2	1	1.27383	0.71899	15
32	14.093	1	2	0.55484	0.829	17
33	14.64784	2	1	0.30153	0.27416	16
34	14.922	1	2	0.02737	0.829	18
35	14.94937	2	1	0.71013	0.80163	17
36	15.6595	2	0	-1	0.0915	18
37	15.751	1	1	0.28507	0.829	19
38	16.03607	2	0	-1	0.54393	19
39	16.58	1	1	0.56245	0.829	20
40	17.14245	2	0	-1	0.26655	20
41	17.409	1	1	0.6314	0.829	21
42	18.0404	2	0	-1	0.1976	21
43	18.238	1	1	0.48688	0.829	22
44	18.72488	2	0	-1	0.34212	22
45	19.067	1	1	0.28475	0.829	23
46	19.35175	2	0	-1	0.54425	23
47	19.896	1	1	0.21626	0.829	24
48	20.11226	2	0	-1	0.61274	24
49	20.725	1	1	0.50909	0.829	25
50	21.23409	2	0	-1	0.31991	25
51	21.554	1	1	2.41683	0.829	26
52	22.383	1	2	1.58783	0.829	27
53	23.212	1	3	0.75883	0.829	28
54	23.97083	2	2	1.21117	0.07017	26
55	24.041	1	3	1.141	0.829	29
56	24.87	1	4	0.312	0.829	30
57	25.182	2	3	0.10332	0.517	27
58	25.28532	2	2	0.71498	0.41368	28
59	25.699	1	3	0.3013	0.829	31
60	26.0003	2	2	2.50218	0.5277	29
61	26.528	1	3	1.97448	0.829	32
62	27.357	1	4	1.14548	0.829	33
63	28.186	1	5	0.31648	0.829	34

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	28.50248	2	4	0.8775	0.51252	30
65	29.015	1	5	0.36498	0.829	35
66	29.37998	2	4	1.74737	0.46402	31
67	29.844	1	5	1.28335	0.829	36
68	30.673	1	6	0.45435	0.829	37
69	31.12735	2	5	2.26231	0.37465	32
70	31.502	1	6	1.88766	0.829	38
71	32.331	1	7	1.05866	0.829	39
72	33.16	1	8	0.22966	0.829	40
73	33.38966	2	7	0.70134	0.59934	33
74	33.989	1	8	0.102	0.829	41
75	34.091	2	7	1.32142	0.727	34
76	34.818	1	8	0.59442	0.829	42
77	35.41242	2	7	0.90424	0.23458	35
78	35.647	1	8	0.66966	0.829	43
79	36.31666	2	7	1.62862	0.15934	36
80	36.476	1	8	1.46928	0.829	44
81	37.305	1	9	0.64028	0.829	45
82	37.94528	2	8	2.88623	0.18872	37
83	38.134	1	9	2.69751	0.829	46
84	38.963	1	10	1.86851	0.829	47
85	39.792	1	11	1.03951	0.829	48
86	40.621	1	12	0.21051	0.829	49
87	40.83151	2	11	0.24419	0.61849	38
88	41.0757	2	10	0.99167	0.3743	39
89	41.45	1	11	0.61737	0.829	50
90	42.06737	2	10	1.24753	0.21163	40
91	42.279	1	11	1.0359	0.829	51
92	43.108	1	12	0.2069	0.829	52
93	43.3149	2	11	0.21035	0.6221	41
94	43.52525	2	10	1.42463	0.41175	42
95	43.937	1	11	1.01288	0.829	53
96	44.766	1	12	0.18388	0.829	54
97	44.94988	2	11	0.10273	0.64512	43
98	45.05261	2	10	1.92007	0.54239	44
99	45.595	1	11	1.37768	0.829	55
100	46.424	1	12	0.54868	0.829	56

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.829	0	0	0.829	0.27001	1.09901
2	1.658	0	0	1.658	1.25862	2.91662
3	2.487	1	0.42962	2.91662	0.26547	3.18209
4	3.316	0	0	3.316	0.59464	3.91064
5	4.145	0	0	4.145	0.07857	4.22357
6	4.974	0	0	4.974	0.53612	5.51012
7	5.803	0	0	5.803	0.19089	5.99389
8	6.632	0	0	6.632	0.26502	6.89702
9	7.461	0	0	7.461	0.27785	7.73885
10	8.29	0	0	8.29	0.17871	8.46871
11	9.119	0	0	9.119	0.71349	9.83249
12	9.948	0	0	9.948	1.60324	11.55124
13	10.777	1	0.77424	11.55124	0.39149	11.94273
14	11.606	1	0.33673	11.94273	1.07789	13.02062
15	12.435	1	0.58562	13.02062	0.35339	13.37401
16	13.264	1	0.11001	13.37401	1.27383	14.64784
17	14.093	1	0.55484	14.64784	0.30153	14.94937
18	14.922	1	0.02737	14.94937	0.71013	15.6595
19	15.751	0	0	15.751	0.28507	16.03607
20	16.58	0	0	16.58	0.56245	17.14245
21	17.409	0	0	17.409	0.6314	18.0404
22	18.238	0	0	18.238	0.48688	18.72488
23	19.067	0	0	19.067	0.28475	19.35175
24	19.896	0	0	19.896	0.21626	20.11226
25	20.725	0	0	20.725	0.50909	21.23409
26	21.554	0	0	21.554	2.41683	23.97083
27	22.383	1	1.58783	23.97083	1.21117	25.182
28	23.212	2	1.97	25.182	0.10332	25.28532

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
29	24.041	2	1.24432	25.28532	0.71498	26.0003
30	24.87	3	1.1303	26.0003	2.50218	28.50248
31	25.699	2	2.80348	28.50248	0.8775	29.37998
32	26.528	2	2.85198	29.37998	1.74737	31.12735
33	27.357	3	3.77035	31.12735	2.26231	33.38966
34	28.186	4	5.20366	33.38966	0.70134	34.091
35	29.015	4	5.076	34.091	1.32142	35.41242
36	29.844	4	5.56842	35.41242	0.90424	36.31666
37	30.673	5	5.64366	36.31666	1.62862	37.94528
38	31.502	5	6.44328	37.94528	2.88623	40.83151
39	32.331	6	8.50051	40.83151	0.24419	41.0757
40	33.16	7	7.9157	41.0757	0.99167	42.06737
41	33.989	7	8.07837	42.06737	1.24753	43.3149
42	34.818	7	8.4969	43.3149	0.21035	43.52525
43	35.647	7	7.87825	43.52525	1.42463	44.94988
44	36.476	7	8.47388	44.94988	0.10273	45.05261
45	37.305	8	7.74761	45.05261	1.92007	46.97268
46	38.134	8	-1	-1	-1	-1
47	38.963	9	-1	-1	-1	-1
48	39.792	10	-1	-1	-1	-1
49	40.621	11	-1	-1	-1	-1
50	41.45	10	-1	-1	-1	-1
51	42.279	10	-1	-1	-1	-1
52	43.108	11	-1	-1	-1	-1
53	43.937	10	-1	-1	-1	-1
54	44.766	11	-1	-1	-1	-1
55	45.595	10	-1	-1	-1	-1
56	46.424	11	-1	-1	-1	-1

3.2 Система массового обслуживания (M|D|1).

Вариант №44.

Начальные данные: $\lambda = 1.082$, $T_{об} = 0.843$

Таблица 1.

l	$t_{cob}(l)$	$Type(l)$	$C(l)$	$t_{ост}$	$t_{ожз}$	$j(l)$
1	0.16365	1	1	0.843	0.16365	1
2	0.3273	1	2	0.67935	0.55431	2
3	0.88161	1	3	0.12504	0.78073	3
4	1.00665	2	2	0.843	0.65569	1
5	1.66234	1	3	0.18731	0.22756	4
6	1.84965	2	2	0.843	0.04025	2
7	1.8899	1	3	0.80275	0.27641	5
8	2.16631	1	4	0.52634	0.61522	6
9	2.69265	2	3	0.843	0.08888	3
10	2.78153	1	4	0.75412	1.27946	7
11	3.53565	2	3	0.843	0.52534	4
12	4.06099	1	4	0.31766	0.39996	8
13	4.37865	2	3	0.843	0.0823	5
14	4.46095	1	4	0.7607	0.74904	9
15	5.20999	1	5	0.01166	0.49347	10
16	5.22165	2	4	0.843	0.48181	6
17	5.70346	1	5	0.36119	0.88058	11
18	6.06465	2	4	0.843	0.51939	7
19	6.58404	1	5	0.32361	0.14042	12
20	6.72446	1	6	0.18319	0.3186	13
21	6.90765	2	5	0.843	0.13541	8
22	7.04306	1	6	0.70759	0.38027	14
23	7.42333	1	7	0.32732	0.53853	15
24	7.75065	2	6	0.843	0.21121	9
25	7.96186	1	7	0.63179	3.13887	16
26	8.59365	2	6	0.843	2.50708	10

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	9.43665	2	5	0.843	1.66408	11
28	10.27965	2	4	0.843	0.82108	12
29	11.10073	1	5	0.02192	0.10524	17
30	11.12265	2	4	0.843	0.08332	13
31	11.20597	1	5	0.75968	0.52003	18
32	11.726	1	6	0.23965	0.13278	19
33	11.85878	1	7	0.10687	0.13706	20
34	11.96565	2	6	0.843	0.03019	14
35	11.99584	1	7	0.81281	1.76823	21
36	12.80865	2	6	0.843	0.95542	15
37	13.65165	2	5	0.843	0.11242	16
38	13.76407	1	6	0.73058	1.25974	22
39	14.49465	2	5	0.843	0.52916	17
40	15.02381	1	6	0.31384	0.41126	23
41	15.33765	2	5	0.843	0.09742	18
42	15.43507	1	6	0.74558	0.42697	24
43	15.86204	1	7	0.31861	2.7999	25
44	16.18065	2	6	0.843	2.48129	19
45	17.02365	2	5	0.843	1.63829	20
46	17.86665	2	4	0.843	0.79529	21
47	18.66194	1	5	0.04771	0.66707	26
48	18.70965	2	4	0.843	0.61936	22
49	19.32901	1	5	0.22364	1.37455	27
50	19.55265	2	4	0.843	1.15091	23
51	20.39565	2	3	0.843	0.30791	24
52	20.70356	1	4	0.53509	0.31539	28
53	21.01895	1	5	0.2197	2.95818	29
54	21.23865	2	4	0.843	2.73848	25
55	22.08165	2	3	0.843	1.89548	26
56	22.92465	2	2	0.843	1.05248	27
57	23.76765	2	1	0.843	0.20948	28
58	23.97713	1	2	0.63352	0.48637	30
59	24.4635	1	3	0.14715	0.54359	31
60	24.61065	2	2	0.843	0.39644	29
61	25.00709	1	3	0.44656	0.08265	32
62	25.08974	1	4	0.36391	0.59318	33
63	25.45365	2	3	0.843	0.22927	30

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	25.68292	1	4	0.61373	0.05361	34
65	25.73653	1	5	0.56012	0.68963	35
66	26.29665	2	4	0.843	0.12951	31
67	26.42616	1	5	0.71349	0.20026	36
68	26.62642	1	6	0.51323	0.61653	37
69	27.13965	2	5	0.843	0.1033	32
70	27.24295	1	6	0.7397	0.66071	38
71	27.90366	1	7	0.07899	2.44416	39
72	27.98265	2	6	0.843	2.36517	33
73	28.82565	2	5	0.843	1.52217	34
74	29.66865	2	4	0.843	0.67917	35
75	30.34782	1	5	0.16383	0.06358	40
76	30.4114	1	6	0.10025	0.73198	41
77	30.51165	2	5	0.843	0.63173	36
78	31.14338	1	6	0.21127	2.1371	42
79	31.35465	2	5	0.843	1.92583	37
80	32.19765	2	4	0.843	1.08283	38
81	33.04065	2	3	0.843	0.23983	39
82	33.28048	1	4	0.60317	1.52383	43
83	33.88365	2	3	0.843	0.92066	40
84	34.72665	2	2	0.843	0.07766	41
85	34.80431	1	3	0.76534	0.3297	44
86	35.13401	1	4	0.43564	0.02391	45
87	35.15792	1	5	0.41173	2.27357	46
88	35.56965	2	4	0.843	1.86184	42
89	36.41265	2	3	0.843	1.01884	43
90	37.25565	2	2	0.843	0.17584	44
91	37.43149	1	3	0.66716	0.76791	47
92	38.09865	2	2	0.843	0.10075	45
93	38.1994	1	3	0.74225	1.14695	48
94	38.94165	2	2	0.843	0.4047	46
95	39.34635	1	3	0.4383	2.20468	49
96	39.78465	2	2	0.843	1.76638	47
97	40.62765	2	1	0.843	0.92338	48
98	41.47065	2	0	-1	0.08038	49
99	41.55103	1	1	0.843	1.58448	50
100	42.39403	2	0	-1	0.74148	50

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.16365	0	0	0.16365	0.843	1.00665
2	0.3273	1	0.67935	1.00665	0.843	1.84965
3	0.88161	2	0.96804	1.84965	0.843	2.69265
4	1.66234	2	1.03031	2.69265	0.843	3.53565
5	1.8899	2	1.64575	3.53565	0.843	4.37865
6	2.16631	3	2.21234	4.37865	0.843	5.22165
7	2.78153	3	2.44012	5.22165	0.843	6.06465
8	4.06099	3	2.00366	6.06465	0.843	6.90765
9	4.46095	3	2.4467	6.90765	0.843	7.75065
10	5.20999	4	2.54066	7.75065	0.843	8.59365
11	5.70346	4	2.89019	8.59365	0.843	9.43665
12	6.58404	4	2.85261	9.43665	0.843	10.27965
13	6.72446	5	3.55519	10.27965	0.843	11.12265
14	7.04306	5	4.07959	11.12265	0.843	11.96565
15	7.42333	6	4.54232	11.96565	0.843	12.80865
16	7.96186	6	4.84679	12.80865	0.843	13.65165
17	11.10073	4	2.55092	13.65165	0.843	14.49465
18	11.20597	4	3.28868	14.49465	0.843	15.33765
19	11.726	5	3.61165	15.33765	0.843	16.18065
20	11.85878	6	4.32187	16.18065	0.843	17.02365
21	11.99584	6	5.02781	17.02365	0.843	17.86665
22	13.76407	5	4.10258	17.86665	0.843	18.70965
23	15.02381	5	3.68584	18.70965	0.843	19.55265
24	15.43507	5	4.11758	19.55265	0.843	20.39565
25	15.86204	6	4.53361	20.39565	0.843	21.23865

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
26	18.66194	4	2.57671	21.23865	0.843	22.08165
27	19.32901	4	2.75264	22.08165	0.843	22.92465
28	20.70356	3	2.22109	22.92465	0.843	23.76765
29	21.01895	4	2.7487	23.76765	0.843	24.61065
30	23.97713	1	0.63352	24.61065	0.843	25.45365
31	24.4635	2	0.99015	25.45365	0.843	26.29665
32	25.00709	2	1.28956	26.29665	0.843	27.13965
33	25.08974	3	2.04991	27.13965	0.843	27.98265
34	25.68292	3	2.29973	27.98265	0.843	28.82565
35	25.73653	4	3.08912	28.82565	0.843	29.66865
36	26.42616	4	3.24249	29.66865	0.843	30.51165
37	26.62642	5	3.88523	30.51165	0.843	31.35465
38	27.24295	5	4.1117	31.35465	0.843	32.19765
39	27.90366	6	4.29399	32.19765	0.843	33.04065
40	30.34782	4	2.69283	33.04065	0.843	33.88365
41	30.4114	5	3.47225	33.88365	0.843	34.72665
42	31.14338	5	3.58327	34.72665	0.843	35.56965
43	33.28048	3	2.28917	35.56965	0.843	36.41265
44	34.80431	2	1.60834	36.41265	0.843	37.25565
45	35.13401	3	2.12164	37.25565	0.843	38.09865
46	35.15792	4	2.94073	38.09865	0.843	38.94165
47	37.43149	2	1.51016	38.94165	0.843	39.78465
48	38.1994	2	1.58525	39.78465	0.843	40.62765
49	39.34635	2	1.2813	40.62765	0.843	41.47065
50	41.55103	0	0	41.55103	0.843	42.39403

3.3 Система массового обслуживания (M|M|1).

Вариант №44.

Начальные данные: $\lambda = 1.082$, $\mu = 1.208$

Таблица 1.

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
1	0.65859	1	1	0.69189	0.65859	1
2	1.31718	1	2	0.0333	2.14112	2
3	1.35048	2	1	1.76156	2.10782	1
4	3.11204	2	0	-1	0.34626	2
5	3.4583	1	1	0.33462	0.68029	3
6	3.79292	2	0	-1	0.34567	3
7	4.13859	1	1	0.78897	0.5203	4
8	4.65889	1	2	0.26867	2.29354	5
9	4.92756	2	1	0.93393	2.02487	4
10	5.86149	2	0	-1	1.09094	5
11	6.95243	1	1	0.96465	0.12138	6
12	7.07381	1	2	0.84327	0.30723	7
13	7.38104	1	3	0.53604	0.30279	8
14	7.68383	1	4	0.23325	0.6688	9
15	7.91708	2	3	0.91425	0.43555	6
16	8.35263	1	4	0.4787	1.29588	10
17	8.83133	2	3	0.20427	0.81718	7
18	9.0356	2	2	2.9019	0.61291	8
19	9.64851	1	3	2.28899	0.35465	11
20	10.00316	1	4	1.93434	1.70988	12
21	11.71304	1	5	0.22446	0.13139	13
22	11.84443	1	6	0.09307	1.40246	14
23	11.9375	2	5	0.69376	1.30939	9
24	12.63126	2	4	0.25705	0.61563	10
25	12.88831	2	3	0.01585	0.35858	11
26	12.90416	2	2	0.63892	0.34273	12

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	13.24689	1	3	0.29619	0.37281	15
28	13.54308	2	2	0.55623	0.07662	13
29	13.6197	1	3	0.47961	1.05288	16
30	14.09931	2	2	0.55037	0.57327	14
31	14.64968	2	1	0.27693	0.0229	15
32	14.67258	1	2	0.25403	1.31734	17
33	14.92661	2	1	2.77348	1.06331	16
34	15.98992	1	2	1.71017	5.03388	18
35	17.70009	2	1	0.40542	3.32371	17
36	18.10551	2	0	-1	2.91829	18
37	21.0238	1	1	0.4029	0.16884	19
38	21.19264	1	2	0.23406	2.02867	20
39	21.4267	2	1	0.28406	1.79461	19
40	21.71076	2	0	-1	1.51055	20
41	23.22131	1	1	1.43306	0.81644	21
42	24.03775	1	2	0.61662	0.89051	22
43	24.65437	2	1	0.05495	0.27389	21
44	24.70932	2	0	-1	0.21894	22
45	24.92826	1	1	0.0314	0.74841	23
46	24.95966	2	0	-1	0.71701	23
47	25.67667	1	1	0.61946	1.50406	24
48	26.29613	2	0	-1	0.8846	24
49	27.18073	1	1	1.97615	1.12545	25
50	28.30618	1	2	0.8507	1.01063	26
51	29.15688	2	1	0.25638	0.15993	25
52	29.31681	1	2	0.09645	0.89319	27
53	29.41326	2	1	0.46058	0.79674	26
54	29.87384	2	0	-1	0.33616	27
55	30.21	1	1	0.0257	0.83078	28
56	30.2357	2	0	-1	0.80508	28
57	31.04078	1	1	0.1212	0.24413	29
58	31.16198	2	0	-1	0.12293	29
59	31.28491	1	1	0.22176	2.35765	30
60	31.50667	2	0	-1	2.13589	30
61	33.64256	1	1	0.21112	1.98122	31
62	33.85368	2	0	-1	1.7701	31
63	35.62378	1	1	0.59222	0.10422	32

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	35.728	1	2	0.488	1.47177	33
65	36.216	2	1	0.02615	0.98377	32
66	36.24215	2	0	-1	0.95762	33
67	37.19977	1	1	1.82342	0.73038	34
68	37.93015	1	2	1.09304	0.68146	35
69	38.61161	1	3	0.41158	0.37251	36
70	38.98412	1	4	0.03907	2.50859	37
71	39.02319	2	3	0.23706	2.46952	34
72	39.26025	2	2	2.00156	2.23246	35
73	41.26181	2	1	0.38027	0.2309	36
74	41.49271	1	2	0.14937	6.75028	38
75	41.64208	2	1	0.94258	6.60091	37
76	42.58466	2	0	-1	5.65833	38
77	48.24299	1	1	0.956	1.7982	39
78	49.19899	2	0	-1	0.8422	39
79	50.04119	1	1	1.9032	0.41304	40
80	50.45423	1	2	1.49016	1.25836	41
81	51.71259	1	3	0.2318	2.09852	42
82	51.94439	2	2	0.41681	1.86672	40
83	52.3612	2	1	2.37913	1.44991	41
84	53.81111	1	2	0.92922	1.35941	43
85	54.74033	2	1	0.05944	0.43019	42
86	54.79977	2	0	-1	0.37075	43
87	55.17052	1	1	0.12424	1.60022	44
88	55.29476	2	0	-1	1.47598	44
89	56.77074	1	1	0.79898	2.24121	45
90	57.56972	2	0	-1	1.44223	45
91	59.01195	1	1	1.83935	1.30065	46
92	60.3126	1	2	0.5387	0.12998	47
93	60.44258	1	3	0.40872	0.07967	48
94	60.52225	1	4	0.32905	0.51903	49
95	60.8513	2	3	0.05348	0.18998	46
96	60.90478	2	2	0.84324	0.1365	47
97	61.04128	1	3	0.70674	0.49472	50
98	61.536	1	4	0.21202	0.03457	51
99	61.57057	1	5	0.17745	0.19607	52
100	61.74802	2	4	0.03396	0.01862	48

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.65859	0	0	0.65859	0.69189	1.35048
2	1.31718	1	0.0333	1.35048	1.76156	3.11204
3	3.4583	0	0	3.4583	0.33462	3.79292
4	4.13859	0	0	4.13859	0.78897	4.92756
5	4.65889	1	0.26867	4.92756	0.93393	5.86149
6	6.95243	0	0	6.95243	0.96465	7.91708
7	7.07381	1	0.84327	7.91708	0.91425	8.83133
8	7.38104	2	1.45029	8.83133	0.20427	9.0356
9	7.68383	3	1.35177	9.0356	2.9019	11.9375
10	8.35263	3	3.58487	11.9375	0.69376	12.63126
11	9.64851	2	2.98275	12.63126	0.25705	12.88831
12	10.00316	3	2.88515	12.88831	0.01585	12.90416
13	11.71304	4	1.19112	12.90416	0.63892	13.54308
14	11.84443	5	1.69865	13.54308	0.55623	14.09931
15	13.24689	2	0.85242	14.09931	0.55037	14.64968
16	13.6197	2	1.02998	14.64968	0.27693	14.92661
17	14.67258	1	0.25403	14.92661	2.77348	17.70009
18	15.98992	1	1.71017	17.70009	0.40542	18.10551
19	21.0238	0	0	21.0238	0.4029	21.4267
20	21.19264	1	0.23406	21.4267	0.28406	21.71076
21	23.22131	0	0	23.22131	1.43306	24.65437
22	24.03775	1	0.61662	24.65437	0.05495	24.70932
23	24.92826	0	0	24.92826	0.0314	24.95966
24	25.67667	0	0	25.67667	0.61946	26.29613
25	27.18073	0	0	27.18073	1.97615	29.15688
26	28.30618	1	0.8507	29.15688	0.25638	29.41326

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
27	29.31681	1	0.09645	29.41326	0.46058	29.87384
28	30.21	0	0	30.21	0.0257	30.2357
29	31.04078	0	0	31.04078	0.1212	31.16198
30	31.28491	0	0	31.28491	0.22176	31.50667
31	33.64256	0	0	33.64256	0.21112	33.85368
32	35.62378	0	0	35.62378	0.59222	36.216
33	35.728	1	0.488	36.216	0.02615	36.24215
34	37.19977	0	0	37.19977	1.82342	39.02319
35	37.93015	1	1.09304	39.02319	0.23706	39.26025
36	38.61161	2	0.64864	39.26025	2.00156	41.26181
37	38.98412	3	2.27769	41.26181	0.38027	41.64208
38	41.49271	1	0.14937	41.64208	0.94258	42.58466
39	48.24299	0	0	48.24299	0.956	49.19899
40	50.04119	0	0	50.04119	1.9032	51.94439
41	50.45423	1	1.49016	51.94439	0.41681	52.3612
42	51.71259	2	0.64861	52.3612	2.37913	54.74033
43	53.81111	1	0.92922	54.74033	0.05944	54.79977
44	55.17052	0	0	55.17052	0.12424	55.29476
45	56.77074	0	0	56.77074	0.79898	57.56972
46	59.01195	0	0	59.01195	1.83935	60.8513
47	60.3126	1	0.5387	60.8513	0.05348	60.90478
48	60.44258	2	0.4622	60.90478	0.84324	61.74802
49	60.52225	3	1.22577	61.74802	0.03396	61.78198
50	61.04128	2	-1	-1	-1	-1
51	61.536	3	-1	-1	-1	-1
52	61.57057	4	-1	-1	-1	-1

4 Анализ результатов

4.1 Система массового обслуживания (D|M|1)

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$: 56;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$: 44;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$: 3.98;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 2.09552;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 3.00615;
- Коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$: 0.15958;

Таблица 3.

i	$\nu(i)$
0	0.18
1	0.26
2	0.11
3	0.05
4	0.04
5	0.04
6	0.02
7	0.05
8	0.06
9	0.02
10	0.05
11	0.08
12	0.04

4.2 Система массового обслуживания (M|D|1)

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$: 50;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$: 50;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$: 4.12;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 2.66487;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 3.50787;
- Коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$: 0.0019;

Таблица 3.

0	0.02
1	0.04
2	0.11
3	0.18
4	0.21
5	0.22
6	0.16
7	0.06

4.3 Система массового обслуживания (M|M|1)

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$: 52;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$: 48;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$: 1.7;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 0.60178;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 1.43874;
- Коэффициент простоя прибора на интервале $[0, t_{\text{cob}}(100)]$: 0.38786;

Таблица 3.

0	0.19
1	0.34
2	0.22
3	0.13
4	0.08
5	0.03
6	0.01

5 Список литературы

1. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. – М.: ЛКИ, 2021 – 400 с.
2. Кирпичников А.П. Методы прикладной теории массового обслуживания.– М.: URSS, 2018 – 224 с.
3. Ивченко Г.И., Каштанов В.А., Коваленко И.Н. Теория массового обслуживания. – М.: URSS, 2012 – 304 с.
4. Смирнов С.Н. Введение в прикладную теорию массового обслуживания. – М.: Гелиос АРВ, 2016 – 176 с.
5. Лобузов А.А., Гумляева С.Д., Норин Н.В. Задачи по теории случайных процессов. – М.: МИРЭА, 1993 – 68 с.
6. Алпатов Ю. Н. Моделирование процессов и систем управления. – СПб: Лань, 2021 - 140 с.
7. Самусевич Г. А. Моделирование процессов функционирования СМО. – М.: Издательство Юрайт, 2021 — 117 с.

6 Приложение

Листинг 1: main.py

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import math
4 import plotly.graph_objects as go
5 import copy
6 import xlsxwriter
7 import pandas as pd
8 from texttable import Texttable
9 from tabulate import tabulate
10 import latextable
11
12 def bmatrix(b):
13     ##b is ndarray of numpy
14     """Returns a LaTeX bmatrix
15
16     :a: numpy array
17     :returns: LaTeX bmatrix as a string
18     """
19     a = np.asarray(b)
20     if len(a.shape) > 2:
21         raise ValueError('bmatrix can at most display two dimensions')
22     temp_string = np.array2string(a, formatter={'float_kind': lambda x: "
        {:.5f}"}.format(x)})
23
24     lines = temp_string.replace('[', '').replace(']', '').splitlines()
25     rv = [r'$\begin{pmatrix}$']
26     rv += [' ' + ' & '.join(l.split()) + r'\'\' ' for l in lines]
27     rv += [r'$\end{pmatrix}$']
28     return '\n'.join(rv)
29
30 def savetable(array, numberOfTable):
31     table = Texttable()
32
33     table.set_cols_align(["c"] * len(array[0]))
34     table.set_cols_dtype(['t'] * len(array[0]))
35     table.set_deco(Texttable.HEADER | Texttable.VLINES | Texttable.HLINES)
36     table.add_rows(array)
37

```

```

38     path = "C:/Users/Danila/Documents/Study/7 semestor/Queuing systems/2-
        nd lab/Report/table_" + str(numberOfTable) + ".tex"
39     my_file = open(path, 'w+')
40     my_file.write(latextable.draw_latex(table))
41     my_file.close()
42
43 def gen_wait_time(lambda_m):
44     return round(np.random.exponential(1/lambda_m), 5)
45
46 def gen_serv_time(myu):
47     return round(np.random.exponential(1/myu), 5)
48
49 class SMO:
50     def __init__(self, m_flag, m_delt_T=0, m_delt_proc=0, m_lambda=0, m_myu=0):
51         self.active_app=1 #номер заявки обрабатываемой СМО(следующей на покидание
            СМО)
52         self.event_counter=1 #счётчик событий
53         self.flag=m_flag #Флаг - вид СМО: 1(D,M,1); 2(M,D,1); 3(M,M,1)
54         self.SMO_table=[] #Таблица данных СМО
55         self.queue=[] #очередь заявок по номерам в СМО
56         self.SMO_counter_app=1 #количество заявок в СМО
57         self.m_Application=[] #список заявок
58         self.SMO_condition=[0,1]
59         self.idle_time=0 #время простоя
60         self.SMO_counter_avr=self.SMO_counter_app #параметр для нахождения
            среднего числа заявок в СМО
61         if self.flag==1:
62             if m_delt_T==0 or m_myu==0:
63                 print("incorrect parameters entered")
64                 return
65             self.delt_T=m_delt_T #постоянное время ожидания заявки
66             self.myu_m=m_myu #параметр для генерации времени обслуживания заявки по
                закону показательного распределения
67             self.time_event_now=m_delt_T #время текущего события
68             self.wait_app_time=m_delt_T #оставшееся время ожидания заявки
69             self.remaining_time=round(gen_serv_time(self.myu_m), 5) #
                оставшееся время обслуживания(в момент инициализации равно времени
                обслуживания)
70             self.m_Application.append(Appliccation(round(self.
                time_event_now, 5), 0, round(self.time_event_now, 5), round(self.
                remaining_time, 5)))
71             self.SMO_table.append([self.event_counter, round(self.
                time_event_now, 5), 1, self.SMO_counter_app, round(self.
                remaining_time, 5), round(self.wait_app_time, 5), self.
                active_app]) #номер события, время события, тип события, кол-во заявок

```

*в СМО, оставшееся время обработки, оставшееся время ожидания заявки,
номер заявки виновной в событии*

```

72     return
73 elif self.flag==2:
74     if m_delt_proc==0 or m_lambda==0:
75         print("incorrect parameters entered")
76         return
77     self.delt_proc=m_delt_proc #постоянное время обслуживания заявки
78     self.lambda_m=m_lambda #параметр для генерации времени ожидания заявки
79     по закону показательного распределения
80     self.wait_app_time=round(gen_wait_time(self.lambda_m),5) #
81     оставшееся время ожидания заявки(в момент инициализации равно времени
82     ожидания)
83     self.time_event_now=self.wait_app_time #время текущего события
84     self.remaining_time=m_delt_proc #оставшееся время обслуживания
85     self.m_Application.append(Appliccation(round(self.
86         time_event_now,5),0,round(self.time_event_now,5),round(self.
87         remaining_time,5)))
88     self.SMO_table.append([self.event_counter,round(self.
89         time_event_now,5),1,self.SMO_counter_app,round(self.
90         remaining_time,5),round(self.wait_app_time,5),self.
91         active_app]) #номер события, время события, тип события, кол-во заявок
92     в СМО, оставшееся время обработки, оставшееся время ожидания заявки,
93     номер заявки виновной в событии
94     return
95 elif self.flag==3:
96     if m_lambda==0 or m_myu==0:
97         print("incorrect parameters entered")
98         return
99     self.lambda_m=m_lambda #параметр для генерации времени ожидания заявки
100    по закону показательного распределения
101    self.myu_m=m_myu #параметр для генерации времени обслуживания заявки по
102    закону показательного распределения
103    self.wait_app_time=round(gen_wait_time(self.lambda_m),5) #
104    оставшееся время ожидания заявки(в момент инициализации равно времени
105    ожидания)
106    self.remaining_time=round(gen_serv_time(self.myu_m),5) #
107    оставшееся время обслуживания(в момент инициализации равно времени
108    обслуживания)
109    self.time_event_now=self.wait_app_time #время текущего события
110    self.m_Application.append(Appliccation(round(self.
111        time_event_now,5),0,round(self.time_event_now,5),round(self.
112        remaining_time,5)))
113    self.SMO_table.append([self.event_counter,round(self.
114        time_event_now,5),1,self.SMO_counter_app,round(self.
115        remaining_time,5),round(self.wait_app_time,5),self.
116        active_app]) #номер события, время события, тип события, кол-во заявок
117    в СМО, оставшееся время обработки, оставшееся время ожидания заявки,

```

```

96         номер заявки виновной в событии
97         return
98     else:
99         print("m_flag – incorrect parameters entered")
100         return
101
102 def gen_event(self):
103     if self.flag==1:
104         if self.remaining_time>self.wait_app_time: #заявка придёт раньше,
105             чем предыдущая закончит обрабатываться
106             self.time_event_now+=self.wait_app_time
107             self.event_counter+=1
108             self.remaining_time-=self.wait_app_time
109             self.wait_app_time=self.delt_T
110             self.SMO_counter_app+=1
111             if (self.SMO_counter_app+1)>len(self.SMO_condition):
112                 self.SMO_condition.append(1)
113             else:
114                 self.SMO_condition[self.SMO_counter_app]+=1
115                 self.queue.append(len(self.m_Application)+1)
116                 self.SMO_counter_avr+=self.SMO_counter_app #параметр для
117                     нахождения среднего числа заявок в СМО
118                 self.m_Application.append(Appliccation(round(self.
119                     time_event_now,5),(len(self.queue)+1),-1,-1))
120                 self.SMO_table.append([self.event_counter,round(self.
121                     time_event_now,5),1,self.SMO_counter_app,round(self.
122                     remaining_time,5),round(self.wait_app_time,5),len(self.
123                     m_Application)]) #номер события, время события, тип события,
124                     кол-во заявок в СМО, оставшееся время обработки, оставшееся время
125                     ожидания заявки, номер заявки виновной в событии
126
127     else:
128         #СМО закончила обрабатывать заявку и либо берёт из очереди, либо стоит и
129         ждёт
130         if len(self.queue)>0:
131             self.event_counter+=1
132             self.time_event_now+=self.remaining_time
133             self.wait_app_time-=self.remaining_time
134             self.SMO_counter_app-=1
135             self.SMO_condition[self.SMO_counter_app]+=1
136             helper=(self.queue).pop(0)
137             self.SMO_counter_avr+=self.SMO_counter_app #параметр для
138                 нахождения среднего числа заявок в СМО
139             self.remaining_time=round(gen_serv_time(self.myu_m),5)
140                 #оставшееся время обслуживания(в момент начала обслуживания
141                 равно времени обслуживания)

```

```

128         self.m_Application[helper-1].start_serv(round(self.
            time_event_now,5),round(self.remaining_time,5))
129         self.SMO_table.append([self.event_counter,round(self.
            time_event_now,5),2,self.SMO_counter_app,round(self.
            remaining_time,5),round(self.wait_app_time,5),self
            .active_app]) #номер события, время события, тип события,
кол-во заявок в СМО, оставшееся время обработки, оставшееся
время ожидания заявки, номер заявки виновной в событии
130         self.active_app=helper
131     elif self.remaining_time!= -1:
132         self.event_counter+=1
133         self.time_event_now+=self.remaining_time
134         self.wait_app_time-=self.remaining_time
135         self.SMO_counter_app-=1
136         self.SMO_condition[self.SMO_counter_app]+=1
137         self.remaining_time=-1 #оставшееся время обслуживания(в
момент начала обслуживания равно времени обслуживания)
138         self.SMO_table.append([self.event_counter,round(self.
            time_event_now,5),2,self.SMO_counter_app,round(self.
            remaining_time,5),round(self.wait_app_time,5),self
            .active_app]) #номер события, время события, тип события,
кол-во заявок в СМО, оставшееся время обработки, оставшееся
время ожидания заявки, номер заявки виновной в событии
139     else:
140         self.idle_time+=self.wait_app_time
141         self.time_event_now+=self.wait_app_time
142         self.event_counter+=1
143         self.SMO_counter_app+=1
144         self.SMO_condition[self.SMO_counter_app]+=1
145         self.SMO_counter_avr+=self.SMO_counter_app #параметр для
нахождения среднего числа заявок в СМО
146         self.remaining_time=round(gen_serv_time(self.myu_m),5)
            #оставшееся время обслуживания(в момент начала обслуживания
равно времени обслуживания)
147         self.wait_app_time=self.delt_T
148         self.m_Application.append(Appliccation(round(self.
            time_event_now,5),0,round(self.time_event_now,5),
            round(self.remaining_time,5)))
149         self.active_app=len(self.m_Application)
150         self.SMO_table.append([self.event_counter,round(self.
            time_event_now,5),1,self.SMO_counter_app,round(self.
            remaining_time,5),round(self.wait_app_time,5),self
            .active_app]) #номер события, время события, тип события,
кол-во заявок в СМО, оставшееся время обработки, оставшееся
время ожидания заявки, номер заявки виновной в событии
151
152     elif self.flag==2:

```



```

153 |         if self.remaining_time>self.wait_app_time: #заявка придёт раньше,
        чем предыдущая закончит обрабатываться
154 |             self.time_event_now+=self.wait_app_time
155 |             self.event_counter+=1
156 |             self.remaining_time-=self.wait_app_time
157 |             self.wait_app_time=round(gen_wait_time(self.lambda_m),5) #
        оставшееся время ожидания заявки(в момент инициализации равно
        времени ожидания)
158 |             self.SMO_counter_app+=1
159 |             if (self.SMO_counter_app+1)>len(self.SMO_condition):
160 |                 self.SMO_condition.append(1)
161 |             else:
162 |                 self.SMO_condition[self.SMO_counter_app]+=1
163 |             self.SMO_counter_avr+=self.SMO_counter_app #параметр для
        нахождения среднего числа заявок в СМО
164 |             self.queue.append(len(self.m_Application)+1)
165 |             self.m_Application.append(Appliccation(round(self.
        time_event_now,5),(len(self.queue)+1),-1,-1))
166 |             self.SMO_table.append([self.event_counter,round(self.
        time_event_now,5),1,self.SMO_counter_app,round(self.
        remaining_time,5),round(self.wait_app_time,5),len(self.
        m_Application)]) #номер события, время события, тип события,
        кол-во заявок в СМО, оставшееся время обработки, оставшееся время
        ожидания заявки, номер заявки виновной в событии
167 |         else:
168 |             #СМО закончила обрабатывать заявку и либо берёт из очереди, либо стоит и
        ждёт
169 |             if len(self.queue)>0:
170 |                 self.event_counter+=1
171 |                 self.time_event_now+=self.remaining_time
172 |                 self.wait_app_time-=self.remaining_time
173 |                 self.SMO_counter_app-=1
174 |                 self.SMO_condition[self.SMO_counter_app]+=1
175 |                 self.SMO_counter_avr+=self.SMO_counter_app #параметр для
        нахождения среднего числа заявок в СМО
176 |                 helper=(self.queue).pop(0)
177 |                 self.remaining_time=self.delt_proc #оставшееся время
        обслуживания(в момент начала обслуживания равно времени
        обслуживания)
178 |                 self.m_Application[helper-1].start_serv(round(self.
        time_event_now,5),round(self.remaining_time,5))
179 |                 self.SMO_table.append([self.event_counter,round(self.
        time_event_now,5),2,self.SMO_counter_app,round(self.
        remaining_time,5),round(self.wait_app_time,5),self.
        active_app]) #номер события, время события, тип события,
        кол-во заявок в СМО, оставшееся время обработки, оставшееся
        время ожидания заявки, номер заявки виновной в событии

```

```

180         self.active_app=helper
181     elif self.remaining_time!= -1:
182         self.event_counter+=1
183         self.time_event_now+=self.remaining_time
184         self.wait_app_time-=self.remaining_time
185         self.SMO_counter_app-=1
186         self.SMO_condition[self.SMO_counter_app]+=1
187         self.remaining_time=-1 #оставшееся время обслуживания(в
                                момент начала обслуживания равно времени обслуживания)
188         self.SMO_table.append([self.event_counter,round(self.
                                time_event_now,5),2,self.SMO_counter_app,round(self
                                .remaining_time,5),round(self.wait_app_time,5),self
                                .active_app]) #номер события, время события, тип события,
                                кол-во заявок в СМО, оставшееся время обработки, оставшееся
                                время ожидания заявки, номер заявки виновной в событии
189     else:
190         self.idle_time+=self.wait_app_time
191         self.time_event_now+=self.wait_app_time
192         self.event_counter+=1
193         self.SMO_counter_app+=1
194         self.SMO_condition[self.SMO_counter_app]+=1
195         self.SMO_counter_avr+=self.SMO_counter_app #параметр для
                                нахождения среднего числа заявок в СМО
196         self.remaining_time=self.delt_proc #оставшееся время
                                обслуживания(в момент начала обслуживания равно времени
                                обслуживания)
197         self.wait_app_time=round(gen_wait_time(self.lambda_m
                                ,5)
198         self.m_Application.append(Appliccation(round(self.
                                time_event_now,5),0,round(self.time_event_now,5),
                                round(self.remaining_time,5)))
199         self.active_app=len(self.m_Application)
200         self.SMO_table.append([self.event_counter,round(self.
                                time_event_now,5),1,self.SMO_counter_app,round(self
                                .remaining_time,5),round(self.wait_app_time,5),self
                                .active_app]) #номер события, время события, тип события,
                                кол-во заявок в СМО, оставшееся время обработки, оставшееся
                                время ожидания заявки, номер заявки виновной в событии
201
202
203     else:
204         if self.remaining_time>self.wait_app_time:#заявка придёт раньше,
                                чем предыдущая закончит обрабатываться
205             self.time_event_now+=self.wait_app_time
206             self.event_counter+=1
207             self.remaining_time-=self.wait_app_time

```

```

208         self.wait_app_time=round(gen_wait_time(self.lambda_m),5) #
           оставшееся время ожидания заявки(в момент инициализации равно
           времени ожидания)
209         self.SMO_counter_app+=1
210         if (self.SMO_counter_app+1)>len(self.SMO_condition):
211             self.SMO_condition.append(1)
212         else:
213             self.SMO_condition[self.SMO_counter_app]+=1
214         self.SMO_counter_avr+=self.SMO_counter_app #параметр для
           нахождения среднего числа заявок в СМО
215         self.queue.append(len(self.m_Application)+1)
216         self.m_Application.append(Appliccation(round(self.
           time_event_now,5),(len(self.queue)+1),-1,-1))
217         self.SMO_table.append([self.event_counter,round(self.
           time_event_now,5),1,self.SMO_counter_app,round(self.
           remaining_time,5),round(self.wait_app_time,5),len(self.
           m_Application)]) #номер события, время события, тип события,
           кол-во заявок в СМО, оставшееся время обработки, оставшееся время
           ожидания заявки, номер заявки виновной в событии

218     else:
219         #СМО закончила обрабатывать заявку и либо берёт из очереди, либо стоит и
           ждёт
220         if len(self.queue)>0:
221             self.event_counter+=1
222             self.time_event_now+=self.remaining_time
223             self.wait_app_time-=self.remaining_time
224             self.SMO_counter_app-=1
225             self.SMO_condition[self.SMO_counter_app]+=1
226             self.SMO_counter_avr+=self.SMO_counter_app #параметр для
           нахождения среднего числа заявок в СМО
227             helper=(self.queue).pop(0)
228             self.remaining_time=round(gen_serv_time(self.myu_m),5)
           #оставшееся время обслуживания(в момент начала обслуживания
           равно времени обслуживания)
229             self.m_Application[helper-1].start_serv(round(self.
           time_event_now,5),round(self.remaining_time,5))
230             self.SMO_table.append([self.event_counter,round(self.
           time_event_now,5),2,self.SMO_counter_app,round(self.
           remaining_time,5),round(self.wait_app_time,5),self.
           active_app]) #номер события, время события, тип события,
           кол-во заявок в СМО, оставшееся время обработки, оставшееся
           время ожидания заявки, номер заявки виновной в событии
231             self.active_app=helper
232         elif self.remaining_time!=-1:
233             self.event_counter+=1
234             self.time_event_now+=self.remaining_time
235             self.wait_app_time-=self.remaining_time

```

```

236         self.SMO_counter_app-=1
237         self.SMO_condition[self.SMO_counter_app]+=1
238         self.remaining_time-=1 #оставшееся время обслуживания(в
момент начала обслуживания равно времени обслуживания)
239         self.SMO_table.append([self.event_counter,round(self.
            time_event_now,5),2,self.SMO_counter_app,round(self
            .remaining_time,5),round(self.wait_app_time,5),self
            .active_app]) #номер события, время события, тип события,
кол-во заявок в СМО, оставшееся время обработки, оставшееся
время ожидания заявки, номер заявки виновной в событии
240     else:
241         self.idle_time+=self.wait_app_time
242         self.time_event_now+=self.wait_app_time
243         self.event_counter+=1
244         self.SMO_counter_app+=1
245         self.SMO_condition[self.SMO_counter_app]+=1
246         self.SMO_counter_avr+=self.SMO_counter_app #параметр для
нахождения среднего числа заявок в СМО
247         self.remaining_time=round(gen_serv_time(self.myu_m),5)
#оставшееся время обслуживания(в момент начала обслуживания
равно времени обслуживания)
248         self.wait_app_time=round(gen_wait_time(self.lambda_m)
            ,5)
249         self.m_Application.append(Appliccation(round(self.
            time_event_now,5),0,round(self.time_event_now,5),
            round(self.remaining_time,5)))
250         self.active_app=len(self.m_Application)
251         self.SMO_table.append([self.event_counter,round(self.
            time_event_now,5),1,self.SMO_counter_app,round(self
            .remaining_time,5),round(self.wait_app_time,5),self
            .active_app]) #номер события, время события, тип события,
кол-во заявок в СМО, оставшееся время обработки, оставшееся
время ожидания заявки, номер заявки виновной в событии
252
253
254     return 1
255
256 def save_data(self):
257     Table_2=[]
258     time_que_avr=0
259     time_SMO_avr=0
260     Table_3=[]
261     j=0
262     while 1:
263         if (j<len(self.m_Application) and self.m_Application[j].
            time_end_serv!=-1):

```

```

264         time_SMO_avr+=round((self.m_Application[j].time_end_serv-
265                               self.m_Application[j].time_coming),5)
266         j+=1
267     else:
268         break
269
270     for i in range(len(self.SMO_condition)):
271         Table_3.append([i,round(self.SMO_condition[i]/self.
272                                event_counter,5)])
273
274     for i in range(len(self.m_Application)):
275         time_que_avr+=round(self.m_Application[i].time_in_queue,5)
276         Table_2.append([i+1,round(self.m_Application[i].time_coming,5)
277                           ,self.m_Application[i].number_in_queue,round(self.
278                               m_Application[i].time_in_queue,5), round(self.m_Application
279                               [i].time_start_serv,5),round(self.m_Application[i].
280                               time_serv,5), round(self.m_Application[i].time_end_serv,5)
281                               ])
282
283     print("СМО имеет вид ")
284     if self.flag==1:
285         savetable(self.SMO_table,1_1)
286         savetable(Table_2,1_2)
287         savetable(Table_3,1_3)
288         print("(D|M|1)")
289     elif self.flag==2:
290         savetable(self.SMO_table,2_1)
291         savetable(Table_2,2_2)
292         savetable(Table_3,2_3)
293         print("(M|D|1)")
294     else:
295         savetable(self.SMO_table,3_1)
296         savetable(Table_2,3_2)
297         savetable(Table_3,3_3)
298         print("(M|M|1)")
299
300     if self.SMO_table[self.event_counter-1][4]!=-1:
301         j-=1
302     if self.SMO_table[self.event_counter-1][2]!=2:
303         time_SMO_avr-=round((self.m_Application[j].time_end_serv-self.
304                               m_Application[j].time_coming),5)
305
306     print("\число заявок :")
307     print(len(self.m_Application))

```

```

299     print("\числоп полностьюобслуженныхзаявок :")
300     print(j)
301     print("\среднееп числозаявоквсистеме :")
302     print(round(self.SMO_counter_avr/self.event_counter,5))
303     print("\среднееп времяпребываниязаявоквочереди :")
304     print(round(time_que_avr/j,5))
305     print("\среднееп времяпребываниязаявоквСМОнаинтервале :")
306     print(round(time_SMO_avr/j,5))
307     print("\коэффициентп простояприбора :")
308     print(round(self.idle_time/self.time_event_now,5))
309
310
311
312 class Application:
313     def __init__(self, m_time_coming, m_number_in_queue, m_time_start_serv,
314                 m_time_serv):
315         self.time_coming=m_time_coming
316         self.number_in_queue=m_number_in_queue
317         if m_number_in_queue==0:
318             self.time_in_queue=0
319         else:
320             self.time_in_queue=-1
321         self.time_start_serv=m_time_start_serv
322         self.time_serv=m_time_serv
323         if m_time_start_serv==-1:
324             self.time_end_serv=-1
325         else:
326             self.time_end_serv=m_time_start_serv+m_time_serv
327     def start_serv(self, m_time_start_serv, m_time_serv):
328         self.time_in_queue=m_time_start_serv-self.time_coming
329         self.time_start_serv=m_time_start_serv
330         self.time_serv=m_time_serv
331         self.time_end_serv=m_time_start_serv+m_time_serv
332
333 def SMO_start(number_of_SMO, number_of_events, m_delt_T=0, m_delt_proc=0,
334             m_lambda=0, m_myu=0):
335     my_SMO=SMO(number_of_SMO, m_delt_T, m_delt_proc, m_lambda, m_myu)
336     while my_SMO.event_counter<number_of_events:
337         my_SMO.gen_event()
338     my_SMO.save_data()
339     return 1
340
341 delt_T=0.829

```

```
340 delt_process=0.843
341 lambda_m=1.082
342 myu=1.208
343
344 SMO_start(1,100,delt_T,delt_process,lambda_m,myu)
345
346 SMO_start(2,100,delt_T,delt_process,lambda_m,myu)
347
348 SMO_start(3,100,delt_T,delt_process,lambda_m,myu)
```