



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

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 3

по курсу «Системы массового обслуживания»

ВАРИАНТ 44

Тема: «Многоканальные 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	Теоретическая часть	7
2.1	Многоканальные СМО ($M M n$) с бесконечной очередью	7
2.2	Описание рассматриваемых СМО	8
2.3	Средства языка программирования	8
3	Результаты расчётов	9
3.1	Система массового обслуживания ($D M 13$)	9
3.2	Система массового обслуживания ($M D 13$).	14
3.3	Система массового обслуживания ($M M 13$).	19
4	Анализ результатов	24
4.1	Система массового обслуживания ($D M 13$)	24
4.2	Система массового обслуживания ($M D 13$)	26
4.3	Система массового обслуживания ($M M 13$)	28
5	Список литературы	30
6	Приложение	31

1 Задание

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени t характеризуется числом заявок, находящихся в СМО. Для всех СМО задано количество приборов n , все приборы пронумерованы.

Событием в развитии СМО является переход из одного состояния в другое. События могут быть двух типов: 1 - появление в системе новой заявки, 2— завершение обслуживания заявки прибором (при этом данный прибор освобождается, и, если есть заявки в очереди, то первая из них поступает сразу же на обслуживание в этот прибор). Если при появлении в системе новой заявки есть свободные приборы, то она сразу же принимается на обслуживание свободным прибором с наименьшим номером, в противном случае заявка становится в очередь типа FIFO.

I. Система массового обслуживания (D|M|n).

Дано:

- время между приходом заявок ΔT_3 (заданная постоянная величина);
- параметр μ показательного распределения времени обслуживания заявки каждым 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 $t_{\text{cob}}(1) = \Delta T_3$, 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.

II. Система массового обслуживания (M|D|n).

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 .

III. Система массового обслуживания (M|M|n).

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 каждым 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 $t_3(1)$ 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 $\text{Type}(l)$;
 - состояние СМО $C(l)$ после события l - минимальное оставшееся время $t_{\text{осмин}}(l)$ обслуживания приборами заявок после события l (если после события все приборы свободны, то $t_{\text{осмин}}(l) = -1$);
 - время ожидания $t_{\text{ож}}(l)$, 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 $t_3(j)$ 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 $t_{\text{о4}}(j)$
 - 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. Составить таблицу 3 с данными о приборах:

- номер прибора k ;
- общее число заявок $N(k)$, поступивших на обслуживание в данный прибор на интервале $[0, t_{\text{коб}}(100)]$;
- общее время занятости прибора $t_{\text{зан}}(k)$ на интервале $[0, t_{\text{коб}}(100)]$.
- коэффициент простоя прибора на интервале $[0, t_{\text{коб}}(100)]$ (отношение времени простоя прибора на интервале $[0, t_{\text{коб}}(100)]$ к $t_{\text{коб}}(100)$);

5. Найти:

- число заявок $J(100)$, поступивших в СМО на интервале $[0, t_{\text{коб}}(100)]$;
- число $JF(100)$ полностью обслуженных заявок на интервале $[0, t_{\text{коб}}(100)]$;
- среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{коб}}(100)]$, 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 $[0, t_{\text{коб}}(100)]$, которое находится по формуле $\bar{t}_{\text{оч}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} t_{\text{оч}}(j)$; - среднее время пребывания заявок в СМО на интервале $[0, t_{\text{коб}}(100)]$, которое находится по формуле $\bar{t}_{\text{СМО}}(100) = \frac{1}{JF(100)} \sum_{j=1}^{JF(100)} [t_{\text{коб}}(j) - t_{\text{з}}(j)]$

Для СМО $(D|M|n)$ и $(M|D|n)$ составить таблицу относительных частот пребывания СМО в состояниях следующего вида:

i	$v_i(100)$
0	$v_0(100)$
1	$v_1(100)$
...	...

где i - состояние СМО, $v_i(100)$ - отношение числа попаданий СМО в состояние i за 100 событий к 100.

Для СМО(М|М|n) найти первые значения стационарных вероятностей состояний $(r_0, r_1, r_2, \dots, r_M)$, где $M = \max\{C(l), l = 1, \dots, 100\}$ и составить таблицу относительных частот пребывания СМО в состояниях следующего вида:

i	r_i	$v_i(100)$	$ v_i(100) - r_i $
0	r_0	$v_0(100)$	$ v_0(100) - r_0 $
1	r_1	$v_1(100)$	$ v_1(100) - r_1 $
\dots	\dots	\dots	\dots
M	r_M	$v_M(100)$	$ v_M(100) - r_M $
	$\sum_{i=0}^M r_i$	$\sum_{i=0}^M v_i(100)$	$\max \{ v_i(100) - r_i \}$

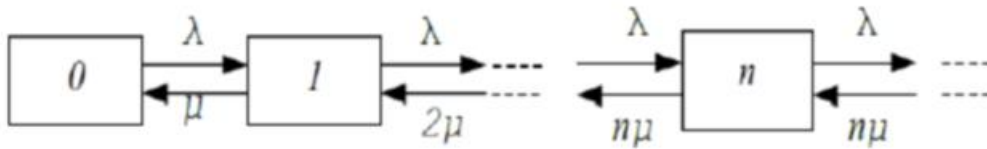
Вывод результатов проводить с округлением до 0,00001.

2 Теоретическая часть

2.1 Многоканальные СМО ($M|M|n$) с бесконечной очередью

Имеется n -канальная СМО с неограниченной очередью. Поток заявок, поступающих в СМО, имеет интенсивность λ , а поток обслуживания – интенсивность μ . Система может находиться в одном из состояний $S_0, S_1, S_2, \dots, S_k, \dots, S_n, \dots$, нумеруемых по числу заявок, находящихся в СМО: S_0 – в системе нет заявок (все каналы свободны); S_1 – занят один канал, остальные свободны; S_2 – заняты два канала, остальные свободны; ... S_k – занято k каналов, остальные свободны; ... S_n – заняты все n каналов (очереди нет); S_{n+1} – заняты все n каналов, в очереди одна заявка; ... S_{n+l} – заняты все n каналов, l заявок стоит в очереди, и т.д.

Граф состояний системы



Формулы для многоканальной СМО ($M|M|n$):

- Стационарные вероятности состояний:

$$r_0 = \left\{ \sum_{k=0}^{n-1} \frac{\rho^k}{k!} + \frac{\rho^n}{n!} \cdot \frac{1}{1-\nu} \right\}^{-1}$$

$$r_k = \frac{\rho^k}{k!} r_0, \quad 1 \leq k \leq n$$

$$r_{n+l} = \nu^l \cdot r_n = \nu^l \cdot \frac{\rho^n}{n!} \cdot r_0, \quad l \geq 1$$

где $\rho = \frac{\lambda}{\mu}$, $\nu = \frac{\lambda}{n\mu} = \frac{\rho}{n}$

- Среднее число занятых приборов:

$$\bar{k} = \rho = \frac{\lambda}{\mu}$$

- Средняя длина очереди:

$$\bar{r} = \frac{\nu r_n}{(1-\nu)^2}$$

- Среднее время пребывания заявок в очереди:

$$\bar{t}_{\text{оч}} = \frac{r_n}{n\mu(1-\nu)^2} = \frac{n\mu r_n}{(n\mu - \lambda)^2}$$

- Среднее время пребывания заявок в СМО:

$$\bar{t}_{\text{смо}} = \bar{t}_{\text{оч}} + \frac{1}{\mu}$$

2.2 Описание рассматриваемых СМО

В рассматриваемых системах массового обслуживания (СМО) состояние в любой момент времени t характеризуется числом заявок, находящихся в СМО. Для всех СМО количество приборов n , все, приборы пронумерованы.

Событием в развитии СМО является переход из одного состояния в другое. 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|13)

Вариант №44.

Начальные данные: $\Delta T_3 = 0.21$, $\mu = 0.405$

Таблица 1.

l	$t_{\text{собр}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
1	0.21	1	1	1.3588	0.21	1
2	0.42	1	2	1.1488	0.21	2
3	0.63	1	3	0.53371	0.21	3
4	0.84	1	4	0.32371	0.21	4
5	1.05	1	5	0.11371	0.21	5
6	1.16371	2	4	0.40509	0.09629	3
7	1.26	1	5	0.3088	0.21	6
8	1.47	1	6	0.0988	0.21	7
9	1.5688	2	5	1.16726	0.1112	1
10	1.68	1	6	1.05606	0.21	8
11	1.89	1	7	0.84606	0.21	9
12	2.1	1	8	0.63606	0.21	10
13	2.31	1	9	0.42606	0.21	11
14	2.52	1	10	0.21606	0.21	12
15	2.73	1	11	0.00606	0.21	13
16	2.73606	2	10	0.44868	0.20394	5
17	2.94	1	11	0.24474	0.21	14
18	3.15	1	12	0.03474	0.21	15
19	3.18474	2	11	0.21337	0.17526	10
20	3.36	1	12	0.03811	0.21	16
21	3.39811	2	11	0.2319	0.17189	7
22	3.57	1	12	0.06001	0.21	17
23	3.63001	2	11	0.26114	0.14999	16
24	3.78	1	12	0.11115	0.21	18
25	3.89115	2	11	0.00999	0.09885	11
26	3.90114	2	10	0.26278	0.08886	17

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	3.99	1	11	0.17392	0.21	19
28	4.16392	2	10	0.63865	0.03608	9
29	4.2	1	11	0.60257	0.21	20
30	4.41	1	12	0.39257	0.21	21
31	4.62	1	13	0.18257	0.21	22
32	4.80257	2	12	0.31628	0.02743	4
33	4.83	1	13	0.28885	0.21	23
34	5.04	1	14	0.07885	0.21	24
35	5.11885	2	13	0.31407	0.13115	24
36	5.25	1	14	0.18292	0.21	25
37	5.43292	2	13	0.01045	0.02708	25
38	5.44337	2	12	0.18784	0.01663	6
39	5.46	1	13	0.17121	0.21	26
40	5.63121	2	12	0.1167	0.03879	12
41	5.67	1	13	0.07791	0.21	27
42	5.74791	2	12	0.1642	0.13209	8
43	5.88	1	13	0.03211	0.21	28
44	5.91211	2	12	0.10495	0.17789	19
45	6.01706	2	11	0.16687	0.07294	26
46	6.09	1	12	0.09393	0.21	29
47	6.18393	2	11	0.03608	0.11607	13
48	6.22001	2	10	0.03545	0.07999	18
49	6.25546	2	9	0.1377	0.04454	21
50	6.3	1	10	0.09316	0.21	30
51	6.39316	2	9	0.19637	0.11684	14
52	6.51	1	10	0.07953	0.21	31
53	6.58953	2	9	0.21615	0.13047	22
54	6.72	1	10	0.08568	0.21	32
55	6.80568	2	9	0.43452	0.12432	25
56	6.93	1	10	0.3102	0.21	33
57	7.14	1	11	0.1002	0.21	34
58	7.2402	2	10	0.65502	0.1098	29
59	7.35	1	11	0.37055	0.21	35
60	7.56	1	12	0.16055	0.21	36
61	7.72055	2	11	0.17467	0.04945	35
62	7.77	1	12	0.12522	0.21	37
63	7.89522	2	11	0.00519	0.08478	31

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	7.90041	2	10	0.11507	0.07959	28
65	7.98	1	11	0.03548	0.21	38
66	8.01548	2	10	0.69457	0.17452	15
67	8.19	1	11	0.17365	0.21	39
68	8.36365	2	10	0.3464	0.03635	39
69	8.4	1	11	0.31005	0.21	40
70	8.61	1	12	0.10005	0.21	41
71	8.71005	2	11	0.41308	0.10995	34
72	8.82	1	12	0.30313	0.21	42
73	9.03	1	13	0.09313	0.21	43
74	9.12313	2	12	0.19089	0.11687	30
75	9.24	1	13	0.07402	0.21	44
76	9.31402	2	12	0.08487	0.13598	42
77	9.39889	2	11	0.35095	0.05111	33
78	9.45	1	12	0.29984	0.21	45
79	9.66	1	13	0.08984	0.21	46
80	9.74984	2	12	0.27168	0.12016	23
81	9.87	1	13	0.15152	0.21	47
82	10.02152	2	12	0.49221	0.05848	32
83	10.08	1	13	0.43373	0.21	48
84	10.29	1	14	0.22373	0.21	49
85	10.5	1	15	0.01373	0.21	50
86	10.51373	2	14	0.38257	0.19627	49
87	10.71	1	15	0.1863	0.21	51
88	10.8963	2	14	0.06279	0.0237	50
89	10.92	1	15	0.03909	0.21	52
90	10.95909	2	14	0.11686	0.17091	51
91	11.07595	2	13	0.09517	0.05405	52
92	11.13	1	14	0.04112	0.21	53
93	11.17112	2	13	0.07064	0.16888	53
94	11.24176	2	12	0.05753	0.09824	24
95	11.29929	2	11	0.4973	0.04071	52
96	11.34	1	12	0.45659	0.21	54
97	11.55	1	13	0.24659	0.21	55
98	11.76	1	14	0.03659	0.21	56
99	11.79659	2	13	0.1144	0.17341	56
100	11.91099	2	12	0.65534	0.05901	38

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.21	0	0	0.21	1.3588	1.5688
2	0.42	0	0	0.42	5.01292	5.43292
3	0.63	0	0	0.63	0.53371	1.16371
4	0.84	0	0	0.84	3.96257	4.80257
5	1.05	0	0	1.05	1.68606	2.73606
6	1.26	0	0	1.26	4.18337	5.44337
7	1.47	0	0	1.47	1.92811	3.39811
8	1.68	0	0	1.68	4.06791	5.74791
9	1.89	0	0	1.89	2.27392	4.16392
10	2.1	0	0	2.1	1.08474	3.18474
11	2.31	0	0	2.31	1.58115	3.89115
12	2.52	0	0	2.52	3.11121	5.63121
13	2.73	0	0	2.73	3.45393	6.18393
14	2.94	0	0	2.94	3.45316	6.39316
15	3.15	0	0	3.15	4.86548	8.01548
16	3.36	0	0	3.36	0.27001	3.63001
17	3.57	0	0	3.57	0.33114	3.90114
18	3.78	0	0	3.78	2.44001	6.22001
19	3.99	0	0	3.99	1.92211	5.91211
20	4.2	0	0	4.2	0.91885	5.11885
21	4.41	0	0	4.41	1.84546	6.25546
22	4.62	0	0	4.62	1.96953	6.58953
23	4.83	0	0	4.83	4.91984	9.74984
24	5.04	1	0.07885	5.11885	6.12291	11.24176
25	5.25	1	0.18292	5.43292	1.37276	6.80568
26	5.46	0	0	5.46	0.55706	6.01706

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
27	5.67	0	0	5.67	7.73433	13.40433
28	5.88	0	0	5.88	2.02041	7.90041
29	6.09	0	0	6.09	1.1502	7.2402
30	6.3	0	0	6.3	2.82313	9.12313
31	6.51	0	0	6.51	1.38522	7.89522
32	6.72	0	0	6.72	3.30152	10.02152
33	6.93	0	0	6.93	2.46889	9.39889
34	7.14	0	0	7.14	1.57005	8.71005
35	7.35	0	0	7.35	0.37055	7.72055
36	7.56	0	0	7.56	7.31119	14.87119
37	7.77	0	0	7.77	10.62705	18.39705
38	7.98	0	0	7.98	3.93099	11.91099
39	8.19	0	0	8.19	0.17365	8.36365
40	8.4	0	0	8.4	4.92875	13.32875
41	8.61	0	0	8.61	2.2863	10.8963
42	8.82	0	0	8.82	0.49402	9.31402
43	9.03	0	0	9.03	3.53633	12.56633
44	9.24	0	0	9.24	4.94253	14.18253
45	9.45	0	0	9.45	5.26098	14.71098
46	9.66	0	0	9.66	0.85373	10.51373
47	9.87	0	0	9.87	1.92659	11.79659
48	10.08	0	0	10.08	0.99595	11.07595
49	10.29	1	0.22373	10.51373	0.44536	10.95909
50	10.5	2	0.3963	10.8963	0.27482	11.17112
51	10.71	2	0.24909	10.95909	4.95224	15.91133
52	10.92	2	0.15595	11.07595	0.22334	11.29929
53	11.13	1	0.04112	11.17112	1.77613	12.94725
54	11.34	0	0	11.34	2.14014	13.48014
55	11.55	0	0	11.55	3.65135	15.20135
56	11.76	1	0.03659	11.79659	1.93727	13.73386

3.2 Система массового обслуживания (M|D|13).

Вариант №44.

Начальные данные: $\lambda = 5.203$, $T_{об} = 2.468$

Таблица 1.

l	$t_{cob}(l)$	$Type(l)$	$C(l)$	$t_{ост}$	$t_{ожз}$	$j(l)$
1	0.60867	1	1	2.468	0.02144	1
2	0.63011	1	2	2.44656	0.18013	2
3	0.81024	1	3	2.26643	0.2594	3
4	1.06964	1	4	2.00703	0.11954	4
5	1.18918	1	5	1.88749	0.00763	5
6	1.19681	1	6	1.87986	0.04539	6
7	1.2422	1	7	1.83447	0.29624	7
8	1.53844	1	8	1.53823	0.10011	8
9	1.63855	1	9	1.43812	0.6757	9
10	2.31425	1	10	0.76242	0.01047	10
11	2.32472	1	11	0.75195	0.10622	11
12	2.43094	1	12	0.64573	0.01339	12
13	2.44433	1	13	0.63234	0.2067	13
14	2.65103	1	14	0.42564	0.07466	14
15	2.72569	1	15	0.35098	0.21568	15
16	2.94137	1	16	0.1353	0.13537	16
17	3.07667	2	15	0.02144	7e-05	14
18	3.07674	1	16	0.02137	0.22669	17
19	3.09811	2	15	0.18013	0.20532	15
20	3.27824	2	14	0.2594	0.02519	16
21	3.30343	1	15	0.23421	0.09662	18
22	3.40005	1	16	0.13759	0.11254	19
23	3.51259	1	17	0.02505	0.06808	20
24	3.53764	2	16	0.11954	0.04303	17
25	3.58067	1	17	0.07651	0.42216	21
26	3.65718	2	16	0.00763	0.34565	18

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	3.66481	2	15	0.04539	0.33802	19
28	3.7102	2	14	0.29624	0.29263	20
29	4.00283	1	15	0.00361	0.02469	22
30	4.00644	2	14	0.10011	0.02108	21
31	4.02752	1	15	0.07903	0.02497	23
32	4.05249	1	16	0.05406	0.16037	24
33	4.10655	2	15	0.6757	0.10631	22
34	4.21286	1	16	0.56939	0.07614	25
35	4.289	1	17	0.49325	0.03384	26
36	4.32284	1	18	0.45941	0.325	27
37	4.64784	1	19	0.13441	0.05581	28
38	4.70365	1	20	0.0786	0.25199	29
39	4.78225	2	19	0.01047	0.17339	23
40	4.79272	2	18	0.10622	0.16292	24
41	4.89894	2	17	0.01339	0.0567	25
42	4.91233	2	16	0.63234	0.04331	26
43	4.95564	1	17	0.58903	0.00202	30
44	4.95766	1	18	0.58701	0.05029	31
45	5.00795	1	19	0.53672	0.2709	32
46	5.27885	1	20	0.26582	0.15504	33
47	5.43389	1	21	0.11078	0.28455	34
48	5.54467	2	20	0.02144	0.17377	27
49	5.56611	2	19	0.18013	0.15233	28
50	5.71844	1	20	0.0278	0.2863	35
51	5.74624	2	19	0.2594	0.2585	29
52	6.00474	1	20	0.0009	0.02752	36
53	6.00564	2	19	0.11954	0.02662	30
54	6.03226	1	20	0.09292	0.13054	37
55	6.12518	2	19	0.00763	0.03762	31
56	6.13281	2	18	0.04539	0.02999	32
57	6.1628	1	19	0.0154	0.09608	38
58	6.1782	2	18	0.29624	0.08068	33
59	6.25888	1	19	0.21556	0.74768	39
60	6.47444	2	18	0.10011	0.53212	34
61	6.57455	2	17	0.6757	0.43201	35
62	7.00656	1	18	0.24369	0.15114	40
63	7.1577	1	19	0.09255	0.27069	41

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	7.25025	2	18	0.01047	0.17814	36
65	7.26072	2	17	0.10622	0.16767	37
66	7.36694	2	16	0.01339	0.06145	38
67	7.38033	2	15	0.63234	0.04806	39
68	7.42839	1	16	0.58428	0.00643	42
69	7.43482	1	17	0.57785	0.56254	43
70	7.99736	1	18	0.01531	0.37372	44
71	8.01267	2	17	0.02144	0.35841	40
72	8.03411	2	16	0.18013	0.33697	41
73	8.21424	2	15	0.2594	0.15684	42
74	8.37108	1	16	0.10256	0.0262	45
75	8.39728	1	17	0.07636	0.13279	46
76	8.47364	2	16	0.11954	0.05643	43
77	8.53007	1	17	0.06311	0.10905	47
78	8.59318	2	16	0.00763	0.04594	44
79	8.60081	2	15	0.04539	0.03831	45
80	8.63912	1	16	0.00708	0.01157	48
81	8.6462	2	15	0.29624	0.00449	46
82	8.65069	1	16	0.29175	0.04083	49
83	8.69152	1	17	0.25092	0.03206	50
84	8.72358	1	18	0.21886	0.14384	51
85	8.86742	1	19	0.07502	0.12482	52
86	8.94244	2	18	0.10011	0.0498	47
87	8.99224	1	19	0.05031	0.01365	53
88	9.00589	1	20	0.03666	0.39237	54
89	9.04255	2	19	0.6757	0.35571	48
90	9.39826	1	20	0.31999	0.3792	55
91	9.71825	2	19	0.01047	0.05921	49
92	9.72872	2	18	0.10622	0.04874	50
93	9.77746	1	19	0.05748	0.28079	56
94	9.83494	2	18	0.01339	0.22331	51
95	9.84833	2	17	0.63234	0.20992	52
96	10.05825	1	18	0.42242	0.02274	57
97	10.08099	1	19	0.39968	0.1033	58
98	10.18429	1	20	0.29638	0.08896	59
99	10.27325	1	21	0.20742	0.05194	60
100	10.32519	1	22	0.15548	0.02161	61

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.60867	0	0	0.60867	2.468	3.07667
2	0.63011	0	0	0.63011	2.468	3.09811
3	0.81024	0	0	0.81024	2.468	3.27824
4	1.06964	0	0	1.06964	2.468	3.53764
5	1.18918	0	0	1.18918	2.468	3.65718
6	1.19681	0	0	1.19681	2.468	3.66481
7	1.2422	0	0	1.2422	2.468	3.7102
8	1.53844	0	0	1.53844	2.468	4.00644
9	1.63855	0	0	1.63855	2.468	4.10655
10	2.31425	0	0	2.31425	2.468	4.78225
11	2.32472	0	0	2.32472	2.468	4.79272
12	2.43094	0	0	2.43094	2.468	4.89894
13	2.44433	0	0	2.44433	2.468	4.91233
14	2.65103	1	0.42564	3.07667	2.468	5.54467
15	2.72569	2	0.37242	3.09811	2.468	5.56611
16	2.94137	3	0.33687	3.27824	2.468	5.74624
17	3.07674	3	0.4609	3.53764	2.468	6.00564
18	3.30343	2	0.35375	3.65718	2.468	6.12518
19	3.40005	3	0.26476	3.66481	2.468	6.13281
20	3.51259	4	0.19761	3.7102	2.468	6.1782
21	3.58067	4	0.42577	4.00644	2.468	6.47444
22	4.00283	2	0.10372	4.10655	2.468	6.57455
23	4.02752	2	0.75473	4.78225	2.468	7.25025
24	4.05249	3	0.74023	4.79272	2.468	7.26072
25	4.21286	3	0.68608	4.89894	2.468	7.36694
26	4.289	4	0.62333	4.91233	2.468	7.38033

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
27	4.32284	5	1.22183	5.54467	2.468	8.01267
28	4.64784	6	0.91827	5.56611	2.468	8.03411
29	4.70365	7	1.04259	5.74624	2.468	8.21424
30	4.95564	4	1.05	6.00564	2.468	8.47364
31	4.95766	5	1.16752	6.12518	2.468	8.59318
32	5.00795	6	1.12486	6.13281	2.468	8.60081
33	5.27885	7	0.89935	6.1782	2.468	8.6462
34	5.43389	8	1.04055	6.47444	2.468	8.94244
35	5.71844	7	0.85611	6.57455	2.468	9.04255
36	6.00474	7	1.24551	7.25025	2.468	9.71825
37	6.03226	7	1.22846	7.26072	2.468	9.72872
38	6.1628	6	1.20414	7.36694	2.468	9.83494
39	6.25888	6	1.12145	7.38033	2.468	9.84833
40	7.00656	5	1.00611	8.01267	2.468	10.48067
41	7.1577	6	0.87641	8.03411	2.468	10.50211
42	7.42839	3	0.78585	8.21424	2.468	10.68224
43	7.43482	4	1.03882	8.47364	2.468	10.94164
44	7.99736	5	0.59582	8.59318	2.468	11.06118
45	8.37108	3	0.22973	8.60081	2.468	11.06881
46	8.39728	4	0.24892	8.6462	2.468	11.1142
47	8.53007	4	0.41237	8.94244	2.468	11.41044
48	8.63912	3	0.40343	9.04255	2.468	11.51055
49	8.65069	3	1.06756	9.71825	2.468	12.18625
50	8.69152	4	1.0372	9.72872	2.468	12.19672
51	8.72358	5	1.11136	9.83494	2.468	12.30294
52	8.86742	6	0.98091	9.84833	2.468	12.31633
53	8.99224	6	-1	-1	-1	-1
54	9.00589	7	-1	-1	-1	-1
55	9.39826	7	-1	-1	-1	-1
56	9.77746	6	-1	-1	-1	-1
57	10.05825	5	-1	-1	-1	-1
58	10.08099	6	-1	-1	-1	-1
59	10.18429	7	-1	-1	-1	-1
60	10.27325	8	-1	-1	-1	-1
61	10.32519	9	-1	-1	-1	-1

3.3 Система массового обслуживания (М|М|13).

Вариант №44.

Начальные данные: $\lambda = 5.203$, $\mu = 0.405$

Таблица 1.

l	$t_{\text{собр}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
1	0.10921	1	1	3.47964	0.09382	1
2	0.20303	1	2	0.12719	0.46524	2
3	0.33022	2	1	3.25863	0.33805	2
4	0.66827	1	2	0.08427	0.08142	3
5	0.74969	1	3	0.00285	0.42644	4
6	0.75254	2	2	2.83631	0.42359	3
7	1.17613	1	3	2.41272	0.22314	5
8	1.39927	1	4	2.18958	0.1912	6
9	1.59047	1	5	1.99838	0.12387	7
10	1.71434	1	6	0.44497	0.03114	8
11	1.74548	1	7	0.41383	0.73516	9
12	2.15931	2	6	1.24722	0.32133	8
13	2.48064	1	7	0.92589	0.31035	10
14	2.79099	1	8	0.27828	0.26775	11
15	3.05874	1	9	0.01053	0.1312	12
16	3.06927	2	8	0.33726	0.12067	11
17	3.18994	1	9	0.21659	0.24611	13
18	3.40653	2	8	0.18232	0.02952	9
19	3.43605	1	9	0.1528	0.04397	14
20	3.48002	1	10	0.10883	0.2667	15
21	3.58885	2	9	0.22529	0.15787	1
22	3.74672	1	10	0.06742	0.62103	16
23	3.81414	2	9	0.01677	0.55361	14
24	3.83091	2	8	0.15046	0.53684	15
25	3.98137	2	7	0.25742	0.38638	16
26	4.23879	2	6	0.00713	0.12896	4

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
27	4.24592	2	5	1.46336	0.12183	13
28	4.36775	1	6	1.34153	0.38188	17
29	4.74963	1	7	0.95965	0.31913	18
30	5.06876	1	8	0.64052	0.86924	19
31	5.70928	2	7	0.02193	0.22872	10
32	5.73121	2	6	0.33457	0.20679	5
33	5.938	1	7	0.12778	0.16732	20
34	6.06578	2	6	0.04058	0.03954	19
35	6.10532	1	7	0.00104	0.07643	21
36	6.10636	2	6	0.03027	0.07539	17
37	6.13663	2	5	0.02897	0.04512	21
38	6.1656	2	4	0.79497	0.01615	12
39	6.18175	1	5	0.77882	1.34427	22
40	6.96057	2	4	0.86475	0.56545	7
41	7.52602	1	5	0.2993	0.43274	23
42	7.82532	2	4	0.32628	0.13344	22
43	7.95876	1	5	0.19284	0.08882	24
44	8.04758	1	6	0.00918	0.28646	25
45	8.05676	2	5	0.09484	0.27728	25
46	8.1516	2	4	0.25154	0.18244	20
47	8.33404	1	5	0.0691	0.21657	26
48	8.40314	2	4	0.07251	0.14747	23
49	8.47565	2	3	0.37037	0.07496	18
50	8.55061	1	4	0.29541	0.01914	27
51	8.56975	1	5	0.27627	0.94658	28
52	8.84602	2	4	2.30717	0.67031	6
53	9.51633	1	5	0.17891	0.25245	29
54	9.69524	2	4	1.45795	0.07354	29
55	9.76878	1	5	0.11612	0.73163	30
56	9.8849	2	4	1.26829	0.61551	30
57	10.50041	1	5	0.65278	0.66067	31
58	11.15319	2	4	0.49629	0.00789	26
59	11.16108	1	5	0.4884	0.35734	32
60	11.51842	1	6	0.13106	0.12318	33
61	11.6416	1	7	0.00788	0.28366	34
62	11.64948	2	6	0.44551	0.27578	27
63	11.92526	1	7	0.16973	0.6736	35

l	$t_{\text{cob}}(l)$	$Type(l)$	$C(l)$	$t_{\text{ост}}$	$t_{\text{ожз}}$	$j(l)$
64	12.09499	2	6	0.31844	0.50387	24
65	12.41343	2	5	0.05076	0.18543	28
66	12.46419	2	4	0.18817	0.13467	33
67	12.59886	1	5	0.0535	0.12303	36
68	12.65236	2	4	1.00887	0.06953	35
69	12.72189	1	5	0.5636	0.14859	37
70	12.87048	1	6	0.41501	0.04847	38
71	12.91895	1	7	0.36654	0.21102	39
72	13.12997	1	8	0.15552	0.1257	40
73	13.25567	1	9	0.02982	0.30784	41
74	13.28549	2	8	0.37574	0.27802	37
75	13.56351	1	9	0.09772	0.18484	42
76	13.66123	2	8	0.48301	0.08712	36
77	13.74835	1	9	0.39589	0.13775	43
78	13.8861	1	10	0.25814	0.0249	44
79	13.911	1	11	0.23324	0.08658	45
80	13.99758	1	12	0.14666	0.06317	46
81	14.06075	1	13	0.08349	0.29313	47
82	14.14424	2	12	0.27715	0.20964	31
83	14.35388	1	13	0.06751	0.2348	48
84	14.42139	2	12	0.56281	0.16729	40
85	14.58868	1	13	0.39552	0.03018	49
86	14.61886	1	14	0.36534	0.18688	50
87	14.80574	1	15	0.17846	0.27634	51
88	14.9842	2	14	0.04966	0.09788	50
89	15.03386	2	13	0.07285	0.04822	51
90	15.08208	1	14	0.02463	0.0419	52
91	15.10671	2	13	0.33697	0.01727	52
92	15.12398	1	14	0.3197	0.16052	53
93	15.2845	1	15	0.15918	0.04128	54
94	15.32578	1	16	0.1179	0.24907	55
95	15.44368	2	15	0.15504	0.13117	53
96	15.57485	1	16	0.02387	0.05788	56
97	15.59872	2	15	0.00779	0.03401	54
98	15.60651	2	14	0.08235	0.02622	55
99	15.63273	1	15	0.05613	0.23305	57
100	15.68886	2	14	0.44867	0.17692	56

Таблица 2.

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
1	0.10921	0	0	0.10921	3.47964	3.58885
2	0.20303	0	0	0.20303	0.12719	0.33022
3	0.66827	0	0	0.66827	0.08427	0.75254
4	0.74969	0	0	0.74969	3.4891	4.23879
5	1.17613	0	0	1.17613	4.55508	5.73121
6	1.39927	0	0	1.39927	7.44675	8.84602
7	1.59047	0	0	1.59047	5.3701	6.96057
8	1.71434	0	0	1.71434	0.44497	2.15931
9	1.74548	0	0	1.74548	1.66105	3.40653
10	2.48064	0	0	2.48064	3.22864	5.70928
11	2.79099	0	0	2.79099	0.27828	3.06927
12	3.05874	0	0	3.05874	3.10686	6.1656
13	3.18994	0	0	3.18994	1.05598	4.24592
14	3.43605	0	0	3.43605	0.37809	3.81414
15	3.48002	0	0	3.48002	0.35089	3.83091
16	3.74672	0	0	3.74672	0.23465	3.98137
17	4.36775	0	0	4.36775	1.73861	6.10636
18	4.74963	0	0	4.74963	3.72602	8.47565
19	5.06876	0	0	5.06876	0.99702	6.06578
20	5.938	0	0	5.938	2.2136	8.1516
21	6.10532	0	0	6.10532	0.03131	6.13663
22	6.18175	0	0	6.18175	1.64357	7.82532
23	7.52602	0	0	7.52602	0.87712	8.40314
24	7.95876	0	0	7.95876	4.13623	12.09499
25	8.04758	0	0	8.04758	0.00918	8.05676
26	8.33404	0	0	8.33404	2.81915	11.15319

j	$t_3(j)$	$q(j)$	$t_{оч}(j)$	$t_{ноб}$	$t_{обсл}$	$t_{коб}$
27	8.55061	0	0	8.55061	3.09887	11.64948
28	8.56975	0	0	8.56975	3.84368	12.41343
29	9.51633	0	0	9.51633	0.17891	9.69524
30	9.76878	0	0	9.76878	0.11612	9.8849
31	10.50041	0	0	10.50041	3.64383	14.14424
32	11.16108	0	0	11.16108	8.29455	19.45563
33	11.51842	0	0	11.51842	0.94577	12.46419
34	11.6416	0	0	11.6416	7.65509	19.29669
35	11.92526	0	0	11.92526	0.7271	12.65236
36	12.59886	0	0	12.59886	1.06237	13.66123
37	12.72189	0	0	12.72189	0.5636	13.28549
38	12.87048	0	0	12.87048	5.45739	18.32787
39	12.91895	0	0	12.91895	2.52473	15.44368
40	13.12997	0	0	13.12997	1.29142	14.42139
41	13.25567	0	0	13.25567	1.72853	14.9842
42	13.56351	0	0	13.56351	2.03521	15.59872
43	13.74835	0	0	13.74835	1.28551	15.03386
44	13.8861	0	0	13.8861	2.25143	16.13753
45	13.911	0	0	13.911	3.46575	17.37675
46	13.99758	0	0	13.99758	1.10913	15.10671
47	14.06075	0	0	14.06075	8.96309	23.02384
48	14.35388	0	0	14.35388	1.33498	15.68886
49	14.58868	0	0	14.58868	8.36216	22.95084
50	14.61886	1	0.36534	14.9842	1.76737	16.75157
51	14.80574	2	0.22812	15.03386	7.94222	22.97608
52	15.08208	1	0.02463	15.10671	3.80194	18.90865
53	15.12398	1	0.3197	15.44368	2.06161	17.50529
54	15.2845	2	0.31422	15.59872	0.00779	15.60651
55	15.32578	3	0.28073	15.60651	0.60758	16.21409
56	15.57485	3	0.11401	15.68886	3.08857	18.77743
57	15.63273	2	-1	-1	-1	-1

4 Анализ результатов

4.1 Система массового обслуживания (D|M|13)

Таблица 3.

k	$N(k)$	$t_{\text{зан}}$	$k_{\text{прост}}$
1	4	11.37811	0.04474
2	4	11.31556	0.04999
3	6	10.93588	0.08187
4	4	10.9234	0.08291
5	5	10.20908	0.14289
6	5	9.67548	0.18768
7	4	9.88667	0.16995
8	7	8.67656	0.27155
9	7	7.74167	0.35004
10	2	9.3522	0.21483
11	2	7.80492	0.34473
12	4	8.16647	0.31438
13	2	4.85052	0.59277

- 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 $[0, t_{\text{cob}}(100)]$: 7.14;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 0.03101;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 2.11271;

Таблица 4.

i	$\nu(i)$
0	0.0
1	0.01
2	0.01
3	0.01
4	0.02
5	0.03
6	0.02
7	0.01
8	0.01
9	0.05
10	0.13
11	0.2
12	0.23
13	0.16
14	0.08
15	0.03

4.2 Система массового обслуживания (M|D|13)

Таблица 3.

k	$N(k)$	$t_{\text{зан}}$	$k_{\text{прост}}$
1	4	9.71652	0.05895
2	4	9.69508	0.06103
3	4	9.51495	0.07847
4	4	9.25555	0.1036
5	4	9.13601	0.11517
6	4	9.12838	0.11591
7	4	9.08299	0.12031
8	4	8.78675	0.149
9	4	8.68664	0.15869
10	4	8.01094	0.22414
11	4	8.00047	0.22515
12	4	7.89425	0.23544
13	4	7.88086	0.23673

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$: 61;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$: 39;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$: 16.04;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 0.52977;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 2.9774;

Таблица 4.

i	$\nu(i)$
0	0.0
1	0.01
2	0.01
3	0.01
4	0.01
5	0.01
6	0.01
7	0.01
8	0.01
9	0.01
10	0.01
11	0.01
12	0.01
13	0.01
14	0.04
15	0.12
16	0.16
17	0.13
18	0.14
19	0.16
20	0.09
21	0.02
22	0.01

4.3 Система массового обслуживания (M|M|13)

Таблица 3.

k	$N(k)$	$t_{\text{зан}}$	$k_{\text{прост}}$
1	8	14.23558	0.09263
2	6	14.32707	0.0868
3	8	13.73004	0.12485
4	6	12.72059	0.1892
5	4	12.90928	0.17717
6	7	7.42978	0.52643
7	4	7.08342	0.54851
8	4	3.72586	0.76252
9	3	5.54005	0.64688
10	2	2.15365	0.86273
11	1	1.77786	0.88668
12	2	1.69128	0.8922
13	1	1.62811	0.89623

- Число заявок, поступивших в СМО на интервале $[0, t_{\text{cob}}(100)]$: 57;
- Число полностью обслуженных заявок на интервале $[0, t_{\text{cob}}(100)]$: 43;
- Среднее число заявок, находившихся в СМО, на интервале $[0, t_{\text{cob}}(100)]$: 5.52;
- Среднее время пребывания заявок в очереди на интервале $[0, t_{\text{cob}}(100)]$: 0.01504;
- Среднее время пребывания заявок в СМО на интервале $[0, t_{\text{cob}}(100)]$: 1.84407;

Таблица 4.

i	r_i	$\nu_i(100)$	$ \nu_i(100) - r_i $
0	0.0	0.0	0.0
1	0.0	0.02	0.02
2	0.00002	0.03	0.02998
3	0.00009	0.03	0.02991
4	0.0003	0.13	0.1297
5	0.00078	0.16	0.15922
6	0.00168	0.12	0.11832
7	0.00308	0.1	0.09692
8	0.00494	0.08	0.07506
9	0.00706	0.08	0.07294
10	0.00906	0.03	0.02094
11	0.01059	0.01	0.00059
12	0.01133	0.03	0.01867
13	0.0112	0.05	0.0388
14	0.01107	0.06	0.04893
15	0.01094	0.05	0.03906
16	0.01081	0.02	0.00919
	0.09295	1	0.15922

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 savetable(array, numberOfTable):
13     table = Texttable()
14
15     table.set_cols_align(["c"] * len(array[0]))
16     table.set_cols_dtype(['t'] * len(array[0]))
17     table.set_deco(Texttable.HEADER | Texttable.VLINES | Texttable.HLINES)
18     table.add_rows(array)
19
20     path = "C:/Users/Danila/Documents/Study/7 semestor/Queuing systems/3-
        hd lab/Report/table_" + str(numberOfTable) + ".tex"
21     my_file = open(path, 'w+')
22     my_file.write(latextable.draw_latex(table))
23     my_file.close()
24     return
25
26 def gen_wait_time(lambda_m):
27     return round(np.random.exponential(1/lambda_m), 5)
28
29 def gen_serv_time(myu):
30     return round(np.random.exponential(1/myu), 5)
31
32 class SMO:
33     def __init__(self, m_flag, n, m_delt_T = 0, m_delt_proc = 0, m_lambda =
        0, m_myu = 0):
34         self.event_counter = 1 #счётчик событий
35         self.SMO_table = [] #Таблица данных СМО (Таблица 1)
36         self.queue = [] #очередь заявок по номерам в СМО
37         self.SMO_counter_app = 1 #количество заявок в СМО

```

```

38 self.m_Application = [] #список заявок (Таблица 1)
39 self.SMO_condition = [0,1]
40 self.SMO_counter_avr = self.SMO_counter_app #параметр для нахождения
    среднего числа заявок в СМО
41 self.quantity_Unit = n
42 self.flag = m_flag
43 self.units = []
44 self.remaining_time_unit = [] #массив времени обработки заявки
45 for i in range(n):
46     self.remaining_time_unit.append(-1)
47     self.units.append(Unit(m_flag, m_delt_proc, m_myu))
48 self.remaining_time = self.units[0].start_work(0) #оставшееся время
    обслуживания(в момент инициализации равно времени обслуживания)
49 self.remaining_time_unit[0] = self.remaining_time
50 self.remaining_index = 0
51 if self.flag == 1:
52     if m_delt_T == 0 or m_myu == 0:
53         print("incorrect parameters entered")
54         return
55     self.delt_T = m_delt_T #постоянное время ожидания заявки
56     self.time_event_now = m_delt_T #время текущего события
57     self.wait_app_time = m_delt_T #оставшееся время ожидания заявки
58     self.m_Application.append(Appliccation(round(self.
        time_event_now, 5), 0, round(self.time_event_now, 5), round(self.
        remaining_time, 5), self.remaining_index))
59     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), 1]) #номер
    события, время события, тип события, кол-во заявок в СМО, оставшееся время
    обработки, оставшееся время ожидания заявки, номер заявки виновной в
    событии
60     return
61 elif self.flag == 2:
62     if m_delt_proc == 0 or m_lambda == 0:
63         print("incorrect parameters entered")
64         return
65     self.lambda_m = m_lambda #параметр для генерации времени ожидания
    заявки по закону показательного распределения
66     self.wait_app_time = gen_wait_time(self.lambda_m) #оставшееся
    время ожидания заявки(в момент инициализации равно времени ожидания)
67     self.time_event_now = gen_wait_time(self.lambda_m) #время
    текущего события
68     self.m_Application.append(Appliccation(round(self.
        time_event_now, 5), 0, round(self.time_event_now, 5), round(self.
        remaining_time, 5), self.remaining_index))
69     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), 1]) #номер
        события, время события, тип события, кол-во заявок в СМО, оставшееся время
        обработки, оставшееся время ожидания заявки, номер заявки виновной в
        событии
70     return
71 elif self.flag == 3:
72     if m_lambda == 0 or m_myu == 0:
73         print("incorrect parameters entered")
74         return
75     self.lambda_m = m_lambda #параметр для генерации времени ожидания
        заявки по закону показательного распределения
76     self.wait_app_time = gen_wait_time(self.lambda_m) #оставшееся
        время ожидания заявки(в момент инициализации равно времени ожидания)
77     self.time_event_now = gen_wait_time(self.lambda_m) #время
        текущего события
78     self.m_Application.append(Application(round(self.
        time_event_now, 5), 0, round(self.time_event_now, 5), round(self.
        remaining_time, 5), self.remaining_index))
79     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), 1]) #номер
        события, время события, тип события, кол-во заявок в СМО, оставшееся время
        обработки, оставшееся время ожидания заявки, номер заявки виновной в
        событии
80     return
81 else:
82     print("m_flag – incorrect parameters entered")
83     return
84
85 def min_rem_time(self):
86     min_v = max(self.remaining_time_unit)
87     index = -1
88     for i in range(self.quantity_Unit):
89         if min_v >= self.remaining_time_unit[i] and self.
            remaining_time_unit[i] >= 0 :
90             min_v = self.remaining_time_unit[i]
91             index = i
92     if min_v < 0:
93         return -1, -1
94     return min_v, index
95
96 def need_q(self):
97     check_free = min(self.remaining_time_unit)
98     if check_free > 0: #количество заявок в СМО больше чем количество приборов
99         self.queue.append(len(self.m_Application))
100         return 0, -1
101     else:

```

```

102         index = next(x[0] for x in enumerate(self.remaining_time_unit)
103                     if x[1] < 0)
104         self.remaining_time_unit[index] = self.units[index].start_work
105         (len(self.m_Application))
106         return 1, index
107
108     def gen_event(self):
109         if self.remaining_time > self.wait_app_time: #заявка придёт раньше, чем
110             предыдущая закончит обрабатываться
111             self.time_event_now += self.wait_app_time
112             self.event_counter += 1
113             for i in range(self.quantity_Unit):
114                 self.remaining_time_unit[i] -= self.wait_app_time
115             if self.flag == 1:
116                 self.wait_app_time = self.delt_T
117             else:
118                 self.wait_app_time = gen_wait_time(self.lambda_m)
119             self.SMO_counter_app += 1
120             flag_free, index_free = self.need_q()
121             if (self.SMO_counter_app+1) > len(self.SMO_condition):
122                 self.SMO_condition.append(1)
123             else:
124                 self.SMO_condition[self.SMO_counter_app] += 1
125             self.SMO_counter_avr += self.SMO_counter_app #параметр для
126                 нахождения среднего числа заявок в СМО
127             if (flag_free == 0):
128                 self.m_Application.append(Appliccation(round(self.
129                     time_event_now, 5), len(self.queue), -1, -1))
130             else:
131                 self.m_Application.append(Appliccation(round(self.
132                     time_event_now, 5), 0, round(self.time_event_now, 5), round(
133                     self.remaining_time_unit[index_free], 5), index_free))
134             self.remaining_time, self.remaining_index = self.min_rem_time
135             ()
136             self.SMO_table.append([self.event_counter, round(self.
137                 time_event_now, 5), 1, self.SMO_counter_app, round(self.
138                 remaining_time, 5), round(self.wait_app_time, 5), len(self.
139                 m_Application)]) #номер события, время события, тип события, кол-во
140                 заявок в СМО, оставшееся время обработки, оставшееся время ожидания
141                 заявки, номер заявки виновной в событии
142
143         else:
144             #СМО закончила обрабатывать заявку и либо берёт из очереди, либо стоит и ждёт
145             if len(self.queue) > 0: #тот прибор, который освободился берёт из
146                 очереди виновница заявка ушедшая

```

```

133         self.event_counter += 1
134         self.time_event_now += self.remaining_time
135         self.wait_app_time -= self.remaining_time
136         for i in range (self.quantity_Unit):
137             self.remaining_time_unit[i] -= self.remaining_time
138         self.SMO_counter_app -= 1
139         self.SMO_condition[self.SMO_counter_app] += 1
140         helper = (self.queue).pop(0)
141         self.remaining_time_unit[self.remaining_index] = self.
            units[self.remaining_index].start_work(helper)
142         self.SMO_counter_avr += self.SMO_counter_app #параметр для
            нахождения среднего числа заявок в СМО
143         self.m_Application[helper].start_serv(round(self.
            time_event_now,5),round(self.remaining_time_unit[self.
            remaining_index],5),self.remaining_index)
144         self.remaining_time, self.remaining_index = self.
            min_rem_time()
145         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),helper
            +1]) #номер события, время события, тип события, кол-во заявок в
            СМО, оставшееся время обработки, оставшееся время ожидания заявки,
            номер заявки виновной в событии
146     elif self.remaining_time != -1:
147         self.event_counter += 1
148         self.time_event_now += self.remaining_time
149         self.wait_app_time -= self.remaining_time
150         for i in range (self.quantity_Unit):
151             self.remaining_time_unit[i] -= self.remaining_time
152         self.remaining_time_unit[self.remaining_index]=-1
153         self.SMO_counter_app -= 1
154         active_app = self.units[self.remaining_index].last_app()+1
155         self.SMO_condition[self.SMO_counter_app] += 1
156         self.remaining_time, self.remaining_index = self.
            min_rem_time()
157         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),
            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
158     else:
159         self.time_event_now += self.wait_app_time
160         self.event_counter += 1
161         self.SMO_counter_app += 1

```

```

162         self.SMO_condition[self.SMO_counter_app] += 1
163         self.SMO_counter_avr += self.SMO_counter_app #параметр для
164         нахождения среднего числа заявок в СМО
165         active_app = len(self.m_Application)+1
166         self.remaining_time = self.units[0].start_work(active_app)
167         #оставшееся время обслуживания(в момент инициализации равно
168         времени обслуживания)
169         self.remaining_time_unit[0] = self.remaining_time
170         self.remaining_index = 0
171         if self.flag == 1:
172             self.wait_app_time = self.delt_T
173         else:
174             self.wait_app_time = gen_wait_time(self.lambda_m)
175         self.m_Application.append(Appliccation(round(self.
176             time_event_now,5),0,round(self.time_event_now,5),round(
177             self.remaining_time,5), self.remaining_index))
178         self.SMO_table.append([self.event_counter,round(self.
179             time_event_now,5),1,self.SMO_counter_app,round(self.
180             remaining_time,5),round(self.wait_app_time,5),
181             active_app]) #номер события, время события, тип события, кол-во
182             заявок в СМО, оставшееся время обработки, оставшееся время ожидания
183             заявки, номер заявки виновной в событи
184
185     return 1
186
187 def save_data(self):
188     Table_2 = []
189     time_que_avr = 0
190     time_SMO_avr = 0
191     Table_3 = []
192     Table_4 = []
193     j = 0
194     App_stop_in_work = {}
195
196     for i in range(len(self.m_Application)):
197         if self.m_Application[i].time_end_serv <= round(self.
198             time_event_now,5) and self.m_Application[i].time_end_serv
199             != -1:
200             time_SMO_avr += round((self.m_Application[i].time_end_serv
201                 -self.m_Application[i].time_coming),5)
202             j += 1
203         elif self.m_Application[i].time_end_serv != -1:
204             App_stop_in_work[self.m_Application[i].number_unit] = self
205                 .m_Application[i].time_end_serv-self.time_event_now #
206                 для учёта непроработавшего времени приборов
207
208     #print(j)

```

```

192
193
194     for i in range(self.quantity_Unit):
195         if i in App_stop_in_work.keys():
196             Table_3.append([i+1,self.units[i].app_counter,round(self.
                units[i].work_time-App_stop_in_work[i],5),round((self.
                time_event_now-self.units[i].work_time+App_stop_in_work
                [i])/self.time_event_now,5)])
197         else:
198             Table_3.append([i+1, self.units[i].app_counter, round(self
                .units[i].work_time,5), round((self.time_event_now-self
                .units[i].work_time)/self.time_event_now, 5)])
199
200     if self.flag != 3:
201         for i in range(len(self.SMO_condition)):
202             Table_4.append([i,round(self.SMO_condition[i]/self.
                event_counter,5)])
203     else:
204         nyu=self.lambda_m/(self.quantity_Unit*self.units[0].myu)
205         p=self.lambda_m/self.units[0].myu
206         r_0 = (p**self.quantity_Unit)/(np.math.factorial(self.
                quantity_Unit)*(1-nyu))
207         for i in range(self.quantity_Unit):
208             r_0 += (p**i)/np.math.factorial(i)
209         r_0 = 1/r_0
210         r_k=[r_0]
211         for i in range(1, len(self.SMO_condition)):
212             if self.quantity_Unit >= i:
213                 r_k.append((p**i)*r_0/np.math.factorial(i))
214             else:
215                 r_k.append((nyu**(i-self.quantity_Unit))*r_k[self.
                quantity_Unit])
216         for i in range(len(self.SMO_condition)):
217             Table_4.append([i, round(r_k[i],5), round(self.
                SMO_condition[i]/self.event_counter,5), round(np.fabs((
                self.SMO_condition[i]/self.event_counter)-r_k[i]),5)])
218
219
220
221     for i in range(len(self.m_Application)):
222         time_que_avr += round(self.m_Application[i].time_in_queue,5)
223         Table_2.append([i+1,round(self.m_Application[i].time_coming,5)
                ,self.m_Application[i].number_in_queue,round(self.

```

```
        m_Application[i].time_in_queue,5), round(self.m_Application  
        [i].time_start_serv,5),round(self.m_Application[i].  
        time_serv,5), round(self.m_Application[i].time_end_serv,5)  
    ])
```

```
print("СМО имеет вид ")
```

```
if self.flag == 1:
```

```
    savetable(self.SMO_table,1_1)
```

```
    savetable(Table_2,1_2)
```

```
    savetable(Table_3,1_3)
```

```
    savetable(Table_4,1_4)
```

```
    print("(D|M|1)")
```

```
elif self.flag == 2:
```

```
    savetable(self.SMO_table,2_1)
```

```
    savetable(Table_2,2_2)
```

```
    savetable(Table_3,2_3)
```

```
    savetable(Table_4,2_4)
```

```
    print("(M|D|1)")
```

```
else:
```

```
    savetable(self.SMO_table,3_1)
```

```
    savetable(Table_2,3_2)
```

```
    savetable(Table_3,3_3)
```

```
    savetable(Table_4,3_4)
```

```
    print("(M|M|1)")
```

```
print("\число заявок :")
```

```
print(len(self.m_Application))
```

```
print("\число полностью обслуженных заявок :")
```

```
print(j)
```

```
print("\среднее число заявок в системе :")
```

```
print(round(self.SMO_counter_avr/self.event_counter,5))
```

```
print("\среднее время пребывания заявок в очереди :")
```

```
print(round(time_que_avr/j,5))
```

```
print("\среднее время пребывания заявок в СМО на интервале :")
```

```
print(round(time_SMO_avr/j,5))
```

```
class Application:
```

```
    def __init__(self, m_time_coming, m_number_in_queue, m_time_start_serv,  
        m_time_serv, m_number_unit=-1):
```

```
        self.time_coming = m_time_coming
```

```
        self.number_in_queue = m_number_in_queue
```

```

262     self.number_unit = m_number_unit
263     if m_number_in_queue == 0:
264         self.time_in_queue = 0
265     else:
266         self.time_in_queue = -1
267     self.time_start_serv = m_time_start_serv
268     self.time_serv = m_time_serv
269     if m_time_start_serv == -1:
270         self.time_end_serv = -1
271     else:
272         self.time_end_serv = m_time_start_serv+m_time_serv
273 def start_serv(self, m_time_start_serv, m_time_serv, m_number_unit):
274     self.time_in_queue = m_time_start_serv-self.time_coming
275     self.time_start_serv = m_time_start_serv
276     self.time_serv = m_time_serv
277     self.number_unit = m_number_unit
278     self.time_end_serv = m_time_start_serv+m_time_serv
279
280 class Unit:
281     def __init__(self, m_flag, m_delt_proc = 0, m_myu = 0):
282         self.numbers_app = []
283         self.work_time = 0
284         self.app_counter = 0 #счётчик заявок поступивших в прибор
285         self.flag = m_flag #Флаг - вид СМО: 1(D,M,1); 2(M,D,1); 3(M,M,1)
286         if m_flag == 2:
287             self.delt_proc = m_delt_proc
288         else:
289             self.myu = m_myu
290     def start_work(self, number_app):
291         self.numbers_app.append(number_app)
292         self.app_counter += 1
293         if self.flag == 2:
294             self.work_time += self.delt_proc
295             return(self.delt_proc)
296         else:
297             work_time = gen_serv_time(self.myu)
298             self.work_time += work_time
299             return(work_time)
300
301     def last_app(self):
302         return self.numbers_app[len(self.numbers_app)-1]
303
304

```

```
305 def SMO_start(number_of_SMO, number_of_events, quantity_units, m_delt_T=0,
    m_delt_proc=0, m_lambda=0, m_myu=0):
306     my_SMO=SMO(number_of_SMO, quantity_units, m_delt_T, m_delt_proc,
        m_lambda, m_myu)
307     while my_SMO.event_counter<number_of_events:
308         my_SMO.gen_event()
309     my_SMO.save_data()
310     return 1
311
312 def main():
313     delt_T=0.21
314     delt_process=2.468
315     lambda_m=1.082
316     myu=0.405
317     n=13
318     SMO_start(1, 100, n, delt_T, delt_process, lambda_m, myu)
319     SMO_start(2, 100, n, delt_T, delt_process, lambda_m, myu)
320     SMO_start(3, 100, n, delt_T, delt_process, lambda_m, myu)
321     return 1
322
323 main()
```