



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе 4 по курсу "Моделирование"

Студент Тэмүүжин Янжинлхам

Группа ИУ7И-776 (ИУ7-73Б)

Преподаватель Рудаков И. В.

Москва — 2022 г.

Задание

Промоделировать систему, состоящую из генератора, памяти и обслуживающего аппарата. Генератор подает сообщения, распределенные по равномерному закону, они приходят в память и выбираются на обработку по закону Пуассона. Количество заявок задано. Предусмотреть случай, когда обработанная заявка возвращается обратно в очередь. Определить оптимальную длину очереди, при которой не будет потерянных сообщений. Реализовать двумя способами: используя пошаговый и событийный подходы

Результаты работы программы $N = 10000$

$$a = 0$$

$$b = 8$$

$$\Delta t = 0.01$$

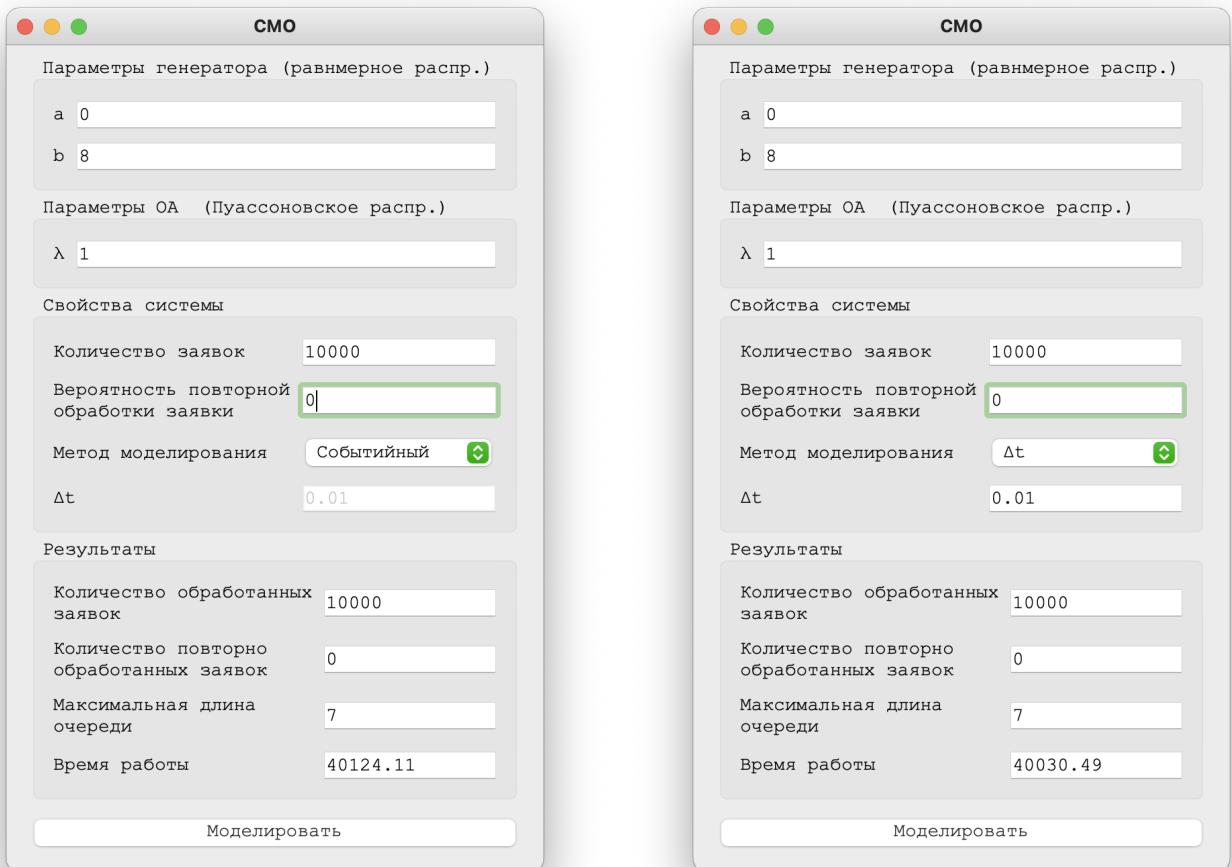


Рис. 1: Пример работы программы при $p = 0$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

a 0
b 8

Параметры ОА (Пуассоновское распр.)

λ 1

Свойства системы

Количество заявок 10000
Вероятность повторной обработки заявки 0.25
Метод моделирования Событийный
Δt 0.01

Результаты

Количество обработанных заявок 10000
Количество повторно обработанных заявок 2450
Максимальная длина очереди 7
Время работы 30163.82

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0
b 8

Параметры ОА (Пуассоновское распр.)

λ 1

Свойства системы

Количество заявок 10000
Вероятность повторной обработки заявки 0.25
Метод моделирования Δt
Δt 0.01

Результаты

Количество обработанных заявок 10000
Количество повторно обработанных заявок 2487
Максимальная длина очереди 15
Время работы 30073.19

Моделировать

Рис. 2: Пример работы программы при $p = 0.25$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 1

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0.5
Метод моделирования: Событийный
Δt: 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 4959
Максимальная длина очереди: 15
Время работы: 20036.11

Моделировать

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 1

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0.5
Метод моделирования: Δt
Δt: 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 4960
Максимальная длина очереди: 15
Время работы: 20204.48

Моделировать

Рис. 3: Пример работы программы при $p = 0.5$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 1

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Событийный

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7494

Максимальная длина очереди 127

Время работы 10202.66

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 1

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Δt

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7500

Максимальная длина очереди 127

Время работы 10155.41

Моделировать

Рис. 4: Пример работы программы при $p = 0.75$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	1
-----------	---

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.95
Метод моделирования	Событийный
Δt	0.01

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	9500
Максимальная длина очереди	2047
Время работы	9925.05

Моделировать

СМО

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	1
-----------	---

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.95
Метод моделирования	Δt
Δt	0.01

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	9489
Максимальная длина очереди	4095
Время работы	10128.15

Моделировать

Рис. 5: Пример работы программы при $p = 0.95$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

Параметры ОА (Пуассоновское распр.)

Свойства системы

Количество заявок	<input type="text" value="10000"/>
Вероятность повторной обработки заявки	<input type="text" value="1"/>
Метод моделирования	Событийный
Δt	<input type="text" value="0.01"/>

Результаты

Количество обработанных заявок	<input type="text" value="10000"/>
Количество повторно обработанных заявок	<input type="text" value="10000"/>
Максимальная длина очереди	<input type="text" value="4095"/>
Время работы	<input type="text" value="10065.87"/>

Моделировать

СМО

Параметры генератора (равномерное распр.)

Параметры ОА (Пуассоновское распр.)

Свойства системы

Количество заявок	<input type="text" value="10000"/>
Вероятность повторной обработки заявки	<input type="text" value="1"/>
Метод моделирования	Δt
Δt	<input type="text" value="0.01"/>

Результаты

Количество обработанных заявок	<input type="text" value="10000"/>
Количество повторно обработанных заявок	<input type="text" value="10000"/>
Максимальная длина очереди	<input type="text" value="4095"/>
Время работы	<input type="text" value="10033.78"/>

Моделировать

Рис. 6: Пример работы программы при $p = 1$, $\lambda = 1$

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 3

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0
Метод моделирования: Событийный
Δt: 0.1

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 0
Максимальная длина очереди: 15
Время работы: 40056.55

Моделировать

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 3

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0
Метод моделирования: Δt
Δt: 0.1

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 0
Максимальная длина очереди: 15
Время работы: 40343.60

Моделировать

Рис. 7: Пример работы программы при $p = 0$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	3
-----------	---

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.25
Метод моделирования	Событийный
Δt	0.1

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	2500
Максимальная длина очереди	255
Время работы	30180.30

Моделировать

СМО

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	3
-----------	---

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.25
Метод моделирования	Δt
Δt	0.1

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	2500
Максимальная длина очереди	255
Время работы	29791.80

Моделировать

Рис. 8: Пример работы программы при $p = 0.25$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 3

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0.5
Метод моделирования: Событийный
Δt: 0.1

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	5000
Максимальная длина очереди	4095
Время работы	30126.63

Моделировать

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ: 3

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0.5
Метод моделирования: Δt
Δt: 0.1

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	5000
Максимальная длина очереди	4095
Время работы	30126.63

Моделировать

Рис. 9: Пример работы программы при $p = 0.5$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Событийный

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7500

Максимальная длина очереди 8191

Время работы 29879.55

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Δt

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7486

Максимальная длина очереди 8191

Время работы 30025.90

Моделировать

Рис. 10: Пример работы программы при $p = 0.75$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.95

Метод моделирования Событийный

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 9493

Максимальная длина очереди 8191

Время работы 30077.78

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.95

Метод моделирования Δt

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 9471

Максимальная длина очереди 8191

Время работы 29740.90

Моделировать

Рис. 11: Пример работы программы при $p = 0.95$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 1

Метод моделирования Событийный

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 10000

Максимальная длина очереди 8191

Время работы 29877.57

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 3

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 1

Метод моделирования Δt

Δt 0.1

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 10000

Максимальная длина очереди 8191

Время работы 29750.00

Моделировать

Рис. 12: Пример работы программы при $p = 1$, $\lambda = 3$

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ : 10

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0
Метод моделирования: Событийный
 Δt : 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 0
Максимальная длина очереди: 16383
Время работы: 99597.94

Моделировать

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ : 10

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 0
Метод моделирования: Δt
 Δt : 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 0
Максимальная длина очереди: 16383
Время работы: 100342.47

Моделировать

Рис. 13: Пример работы программы при $p = 0$, $\lambda = 10$

СМО

Параметры генератора (равномерное распр.)

a 0
b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000
Вероятность повторной обработки заявки 0.25
Метод моделирования Событийный
Δt 0.01

Результаты

Количество обработанных заявок 10000
Количество повторно обработанных заявок 2500
Максимальная длина очереди 32767
Время работы 100006.50

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0
b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000
Вероятность повторной обработки заявки 0.25
Метод моделирования Δt
Δt 0.01

Результаты

Количество обработанных заявок 10000
Количество повторно обработанных заявок 2426
Максимальная длина очереди 32767
Время работы 100348.47

Моделировать

Рис. 14: Пример работы программы при $p = 0.25$, $\lambda = 10$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.5

Метод моделирования Событийный

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 5000

Максимальная длина очереди 32767

Время работы 99798.20

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.5

Метод моделирования Δt

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 4999

Максимальная длина очереди 32767

Время работы 99905.44

Моделировать

Рис. 15: Пример работы программы при $p = 0.5$, $\lambda = 10$

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Событийный

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7500

Максимальная длина очереди 32767

Время работы 100621.38

Моделировать

СМО

Параметры генератора (равномерное распр.)

a 0

b 8

Параметры ОА (Пуассоновское распр.)

λ 10

Свойства системы

Количество заявок 10000

Вероятность повторной обработки заявки 0.75

Метод моделирования Δt

Δt 0.01

Результаты

Количество обработанных заявок 10000

Количество повторно обработанных заявок 7425

Максимальная длина очереди 32767

Время работы 100269.25

Моделировать

Рис. 16: Пример работы программы при $p = 0.75$, $\lambda = 10$

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	10
-----------	----

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.95
Метод моделирования	Событийный
Δt	0.01

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	9500
Максимальная длина очереди	32767
Время работы	100057.33

Параметры генератора (равномерное распр.)

a	0
b	8

Параметры ОА (Пуассоновское распр.)

λ	10
-----------	----

Свойства системы

Количество заявок	10000
Вероятность повторной обработки заявки	0.95
Метод моделирования	Δt
Δt	0.01

Результаты

Количество обработанных заявок	10000
Количество повторно обработанных заявок	9500
Максимальная длина очереди	32767
Время работы	100248.64

Рис. 17: Пример работы программы при $p = 0.95$, $\lambda = 10$

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ : 10

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 1
Метод моделирования: Событийный
 Δt : 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 10000
Максимальная длина очереди: 32767
Время работы: 100206.84

Моделировать

СМО

Параметры генератора (равномерное распр.)

a: 0
b: 8

Параметры ОА (Пуассоновское распр.)

λ : 10

Свойства системы

Количество заявок: 10000
Вероятность повторной обработки заявки: 1
Метод моделирования: Δt
 Δt : 0.01

Результаты

Количество обработанных заявок: 10000
Количество повторно обработанных заявок: 10000
Максимальная длина очереди: 32767
Время работы: 100374.87

Моделировать

Рис. 18: Пример работы программы при $p = 1$, $\lambda = 10$

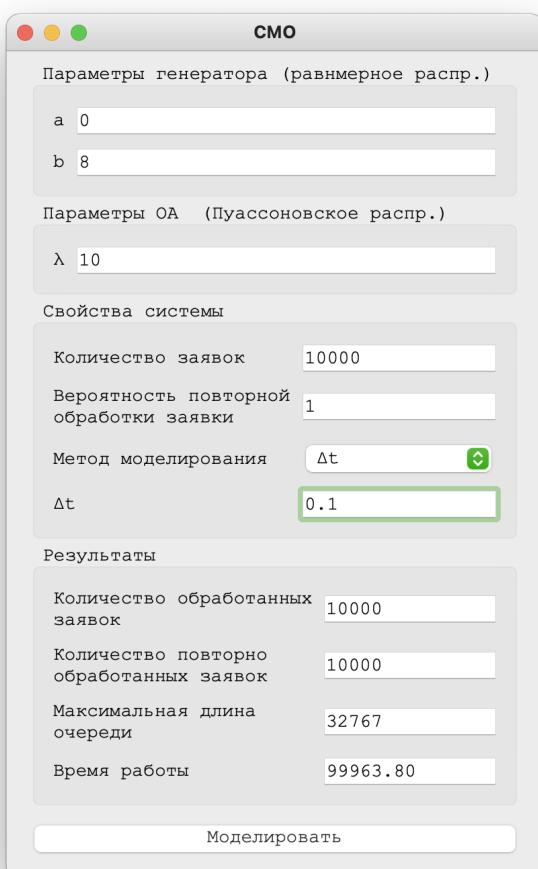


Рис. 19: Пример работы программы при $p = 1$, $\lambda = 10$, $\Delta t = 0.1$

1 Код программы

Листинг 1.1: modeller.py

```
1 import numpy.random as nr
2 from scipy.stats import poisson
3 import sys
4
5 PRB = 1000
6
7 class UniformGenerator:
8     def __init__(self, a, b):
9         if not 0 <= a <= b:
10             raise ValueError('0 <= a <= b')
11         self._a = a
12         self._b = b
13
14     def next(self):
15         return nr.uniform(self._a, self._b)
16
17
18 class PoissonDistribution:
19     def __init__(self, lmbda):
20         self._lmbda = lmbda
21
22     def next(self):
23         return poisson.rvs(self._lmbda, size=1)[0]
24
25
26 class RequestGenerator:
27     def __init__(self, generator):
28         self._generator = generator
29         self._receivers = set()
30
31     def next_time_period(self):
32         return self._generator.next()
33
34     def emit_request(self):
35         for receiver in self._receivers:
36             receiver.receive_request()
37
38     def add_receiver(self, receiver):
39         self._receivers.add(receiver)
40
41     def remove_receiver(self, receiver):
42         try:
```

```

43         self._receivers.remove(receiver)
44     except KeyError:
45         print("Error removing receiver")
46         sys.exit(0)
47     # pass
48
49
50 class RequestProcessor(RequestGenerator):
51     def __init__(self, generator, reenter_probability=0):
52         super().__init__(generator)
53         self._generator = generator
54         self._current_queue_size = 0
55         self._max_queue_size = 0
56         self._processed_requests = 0
57         self._reenter_probability = reenter_probability
58         self._reentered_requests = 0
59
60     @property
61     def processed_requests(self):
62         return self._processed_requests
63
64     @property
65     def max_queue_size(self):
66         return self._max_queue_size
67
68     @property
69     def current_queue_size(self):
70         return self._current_queue_size
71
72     @property
73     def reentered_requests(self):
74         return self._reentered_requests
75
76     def process(self):
77         if self._current_queue_size > 0:
78             self._processed_requests += 1
79             self._current_queue_size -= 1
80             self.emit_request()
81             if nr.random_sample() < self._reenter_probability and
82                 self._reentered_requests < self._reenter_probability * PRB:
83                 self._reentered_requests += 1
84                 self.receive_request()
85
86     def receive_request(self):
87         self._current_queue_size += 1
88         if self._current_queue_size > self._max_queue_size:
89             self._max_queue_size += self._current_queue_size

```

```

90
91     def next_time_period(self):
92         return self._generator.next()
93
94
95 class Modeller:
96     def __init__(self, uniform_a, uniform_b, expo_lambda, reenter_prop):
97         self._generator = RequestGenerator(UniformGenerator(uniform_a, uniform_b))
98
99         self._processor = RequestProcessor(PoissonDistribution(expo_lambda), reenter_prop)
100        self._generator.add_receiver(self._processor)
101
102    def event_based_modelling(self, request_count):
103        generator = self._generator
104        processor = self._processor
105
106        gen_period = generator.next_time_period()
107        proc_period = gen_period + processor.next_time_period()
108        while processor.processed_requests < request_count:
109            if gen_period <= proc_period:
110                generator.emit_request()
111                gen_period += generator.next_time_period()
112            if gen_period >= proc_period:
113                processor.process()
114                if processor.current_queue_size > 0:
115                    proc_period += processor.next_time_period()
116                else:
117                    proc_period = gen_period + processor.next_time_period()
118
119        return (processor.processed_requests, processor.reentered_requests,
120               processor.max_queue_size, proc_period)
121
122    def time_based_modelling(self, request_count, dt):
123        generator = self._generator
124        processor = self._processor
125
126        gen_period = generator.next_time_period()
127        proc_period = gen_period + processor.next_time_period()
128        current_time = 0
129        while processor.processed_requests < request_count:
130            if gen_period <= current_time:
131                generator.emit_request()
132                gen_period += generator.next_time_period()
133            if current_time >= proc_period:
134                processor.process()
135                if processor.current_queue_size > 0:
136                    proc_period += processor.next_time_period()
137                else:

```

```
138         proc_period = gen_period + processor.next_time_period()
139         current_time += dt
140
141     return (processor.processed_requests, processor.reentered_requests,
142            processor.max_queue_size, current_time)
```