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

Дисциплина: Моделирование

Отчёт по лабораторной работе №4
Вариант 18

Студент: Шибанова Д.А.

Группа: ИУ7-72Б

Преподаватель: Рудаков И.В.

Москва, 2019

Содержание

Условие лабораторной работы

Смоделировать систему, состоящую из генератора, очереди и обслуживающего аппарата (ОА). Генерация заявок происходит по равномерному закону распределения. Для ОА используется закон распределения Гаусса. Определить оптимальный объём памяти, при котором не будет потери заявок. С заданной вероятностью заявка может вернуться в очередь после её обслуживания. Исследовать систему Δt-методом и событийным методом.

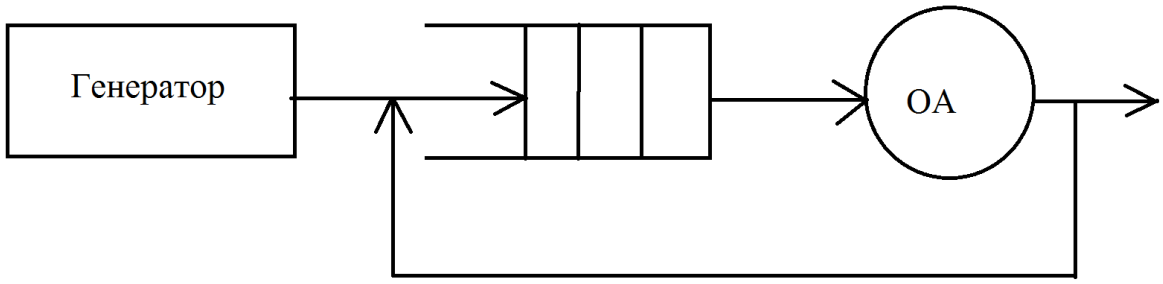

Рис. 1. Схема работы СМО.

Теоретическая часть

1. Распределения

1.1. Равномерное распределение

Случайная величина имеет равномерное распределение на отрезке $[a;b]$, где $a, b \in \mathbb{R}$, если её плотность распределения $f_X(x)$ имеет следующий вид:

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & x \in [a;b] \\ 0, & x \notin [a;b] \end{cases}. \tag{1}$$

Интегрируя функцию плотности распределения, можно получить соответствующую её функцию распределения:

$$F_X(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & x < b \\ 1, & x \geq b \end{cases}. \tag{2}$$

1.2. Распределение Гаусса

Случайная величина имеет распределение Гаусса (нормальное распределение), если её функция плотности распределения $f_X(x)$ имеет следующий вид:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{3}$$

Функция распределения для данного закона:

$$F_X(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}}\, dt. \tag{4}$$

Функцию распределения также можно представить в следующем виде:

$$F_X(x) = \frac{1}{2}[1 + erf(\frac{x-\mu}{\sqrt{2\sigma^2}})] \tag{5}$$

2. Протяжка модельного времени.

Основная функция протягивания модельного времени состоит в реализации алгоритма функционирования. Имитация взаимодействия отдельных устройств системы происходит с помощью управляющей программы.

Управляющая программа реализуется в основном по двум принципам:

1) принцип Δt;

2) событийный принцип.

Также возможно применять комбинирующий метод, сочетающий в себе два указанных принципа.

2.1. Метод Δt.

Принцип Δt заключается в последовательном анализе состояний всех блоков в момент $t + \Delta t$ по заданному состоянию блоков в момент t. При этом новое состояние блоков определяется в соответствии с их алгоритмическим описанием с учётом действующих случайных факторов, задаваемых распределениями вероятности. В результате такого анализа принимается решение о том, какие общесистемные события должны имитироваться программной моделью на данный момент времени.

Достоинством данного метода является равномерность протягивания модельного времени.

Основной недостаток этого принципа заключается в значительных затратах машинного времени на реализацию моделирования системы. При недостаточно малом Δt появляется опасность пропуска отдельных событий в системе, что исключает возможность получения адекватных результатов при моделировании.

2.2. Событийный метод.

Характерным свойством систем обработки информации является то, что состояние отдельных устройств изменяется в дискретные моменты времени, совпадающие с моментами времени поступления сообщений в систему, временем поступления окончания задачи, времени поступления аварийных сигналов и т.д. Поэтому моделирование и продвижение времени можно проводить с использованием событийного принципа. При его использовании состояние всех блоков системы анализируется лишь в момент появления или наступления какого-либо события. Момент поступления следующего события определяется минимальным значением из списка будущих событий, представляющего собой совокупность моментов ближайшего изменения состояния каждого из блоков системы.

Любое событие в этом подходе можно описать с использованием пяти осей:

1) первая ось – момент появления события от источника информации;

2) вторая ось – момент освобождения обслуживающего аппарата (ОА);

3) третья ось – моменты сбора статистики;

4) четвёртая ось – время окончания моделирования;

5) пятая ось – текущее время.

С помощью этих осей задаются интервалы обслуживания сообщений и соответствующие моменты. На основе этих данных формируется список будущих событий (СБС).

В общем виде метод можно описать следующим образом:

1) для блоков, порождающих события (активных блоков), заводят свой элемент в одномерном массиве (в СБС);

2) в СБС заносят время ближайшего события от любого активного блока;

3) активизируется программный имитатор источника событий и вырабатывает псевдослучайную величину, которая определяет момент появления первого сообщения от источника сообщений и которую помещают в СБС;

4) активизируется программный имитатор для выработки выличины для ОА, которую тоже необходимо занести в СБС;

5) в момент первого сбора статистики определяется стандартный шаг сбора статистики, который заносится в СБС;

6) в СБС заносится момент окончания моделирования;

7) выполняется протяжка времени:
   а) в СБС определяется минимальное числовое значение и его номер;
   б) реализуется событие, порождаемое блоком с соответствующим номером;
   в) реализуется событие, связанное с появлением нового сообщения;
   г) сообщение записывается в память и вырабатывается момент появления следующего события, это время записывается в ячейку СБС вместо предыдущего момента;
   д) новый поиск с новыми сборами статистики, пока минимальное время не станет равно времени момента окончания моделирования.

При этом задача блока сбора статистики заключается в накоплении численных значений, которые необходимы для вычисления заданных параметров моделируемой системы. Как правило, при моделировании систем массового обслуживания (СМО) к таким значениям относят:

1) среднее время ожидания в очереди – для каждого сообщения время ожидания в очереди представляется как разность между моментом, когда сообщение было выбрано на обработку обслуживающим аппаратом, и моментом, когда оно пришло в систему от источника информации;

2) среднее значение длины очереди;

3) коэффициент загрузки ОА – отношение времени работы ОА к общему времени моделирования;

4) вероятность потери сообщений.

Примеры работы

Описание программы

Для работы программы был сформирован пользовательский интерфейс с использованием программы *QtDesigner*. Созданная форма была сконвертирована из формата *.ui* в формат *.py* с использованием *pyuic*5 через консоль и команду *pyuic5window.ui – omainwindow.py* для дальнейшего её использования в программе.

Лабораторная работа включает в себя 3 файла на языке программирования *Python*:

1) *mainwindow.py* – файл описания формы интерфейса;

2) *Queue.py* – файл описания генерации;

3) *main.py* – основной файл решения, к которому подключаются другие для использования их содержимого.

mainwindow.py – генерируемый автоматически файл. Любое изменение формы требует нового запуска конвертации для получения данного файла.

Queue.py – задаёт класс работы с очередью.

main.py описывает класс работы с интерфейсом (получение параметров и наступление события нажатия кнопки) и запускает программу на выполнение. Содержит также принципы протяжки модельного времени.

Запуск программы

На рисунке 2 показано состояние программы при исходном запуске. Для параметров в форме интерфейса предложены параметры по умолчанию, но непосредственный расчёт происходит только в результате нажатия пользователем на кнопку "Посчитать до этого момента результирующие поля остаются пустыми.

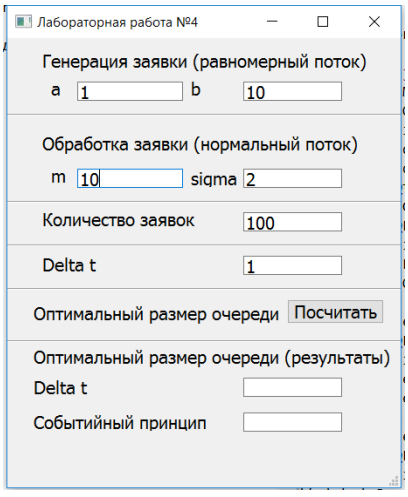

Рис. 2. Исходный вид.

Примеры

Программа способна как использовать корректно введённые данные, так и обрабатывать ошибочные данные. При этом обработка некорректных данных разделена на два типа подходов:

1) редактирование исходных данных при проверке;

2) вывод пользователю сообщения с предупреждением, после закрытия которого можно исправить ошибочные данные, указанные в сообщении.

На рисунке 3 показаны выведенные программой результаты для параметров по умолчанию.

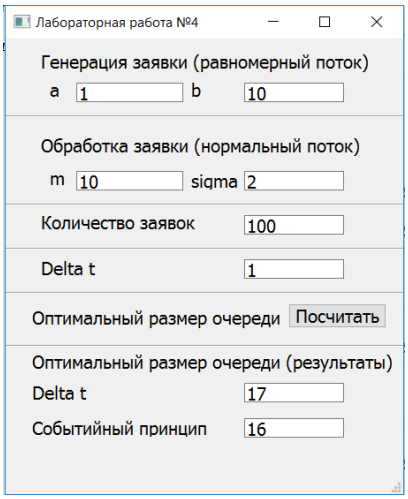

Рис. 3. Пример вывода для параметров по умолчанию.

Рисунок 4 отражает поведение программы в случае, если $a > b$. Программа обработает данную ситуацию, поменяв местами значения параметров.

Ситуация, когда $a < 0$, или $b < 0$, или $a < 0, b < 0$ также не будет выдывать ошибку: как видно из рисунка 5, программа произведёт расчёты. При этом значения будут взяты по модулю.

На рисунках 6, 7 и 8 показаны ситуации, когда параметры распределения Гаусса принимают отрицательные значения. Как и в случае с параметрами равномерного распределения, программа не выдаст ошибку и возьмёт модули введённых пользователем значений.

Программа обрабатывает отрицательное количество заявок (рисунок 9), выводя соответствующее предупреждение (10).

Обработка отрицательного значения Δt (рисунок 11) аналогична обработке ввода отрицательного числа заявок: программа выведет соответствующее сообщение (рисунок 12).

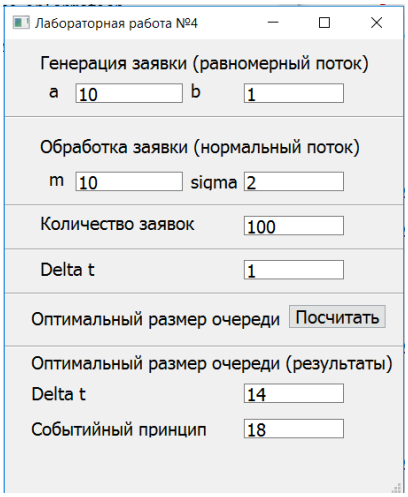

Рис. 4. Ситуация $a > b$.

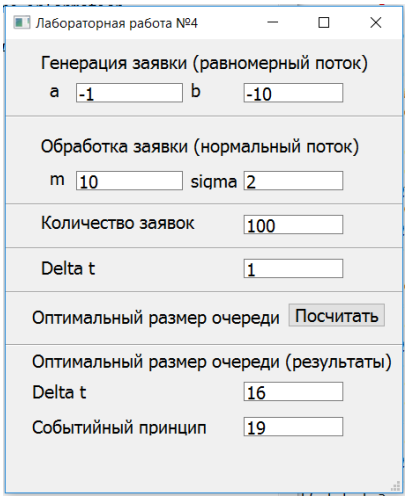

Рис. 5. Отрицательные параметры равномерного распределения.

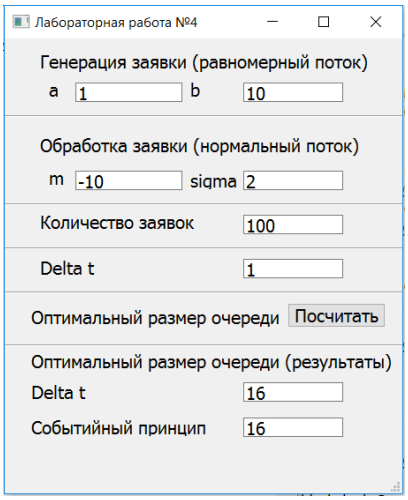

Рис. 6. Отрицательное математическое ожидание в нормальном распределении.

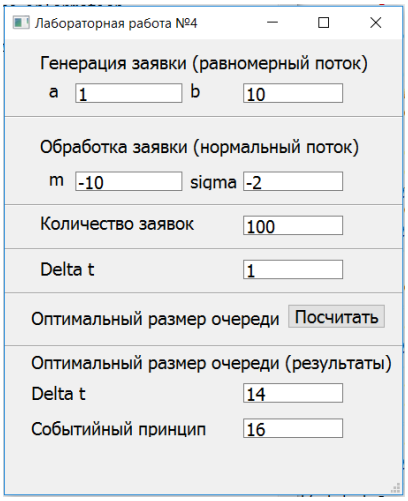

Рис. 7. Отрицательные параметры μ и σ в распредлении Гаусса.

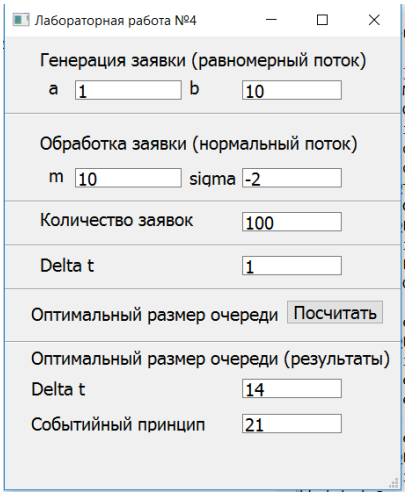

Рис. 8. Отрицательный параметр σ.

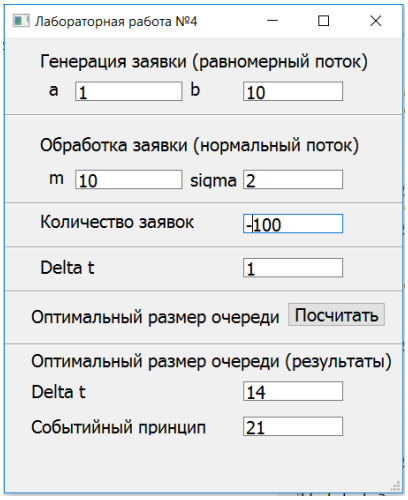

Рис. 9. Отрицательное количество заявок.

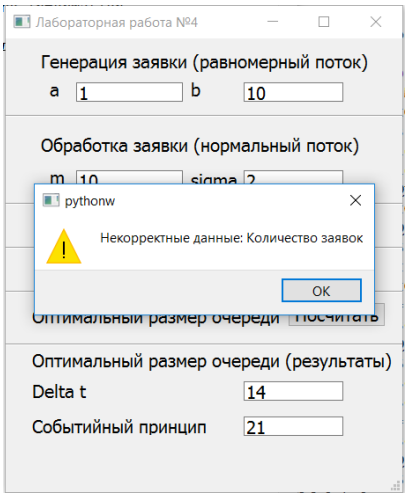

Рис. 10. Сообщение о некорректном количестве заявок.

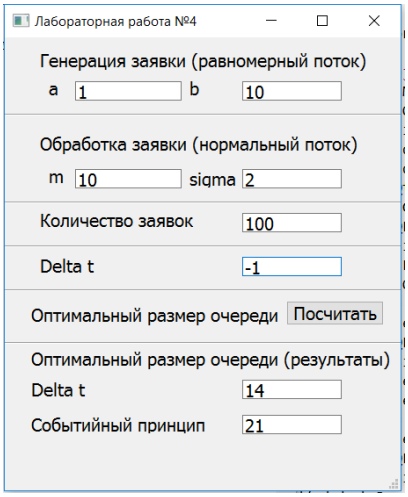

Рис. 11. Отрицательное значение Δt.

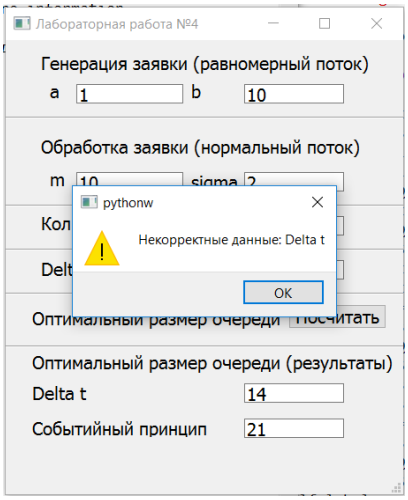

Рис. 12. Сообщение о некорректном значении Δt.