**Постановка задачи.**

Разработать программное средство, представленное по правилам хорошего стиля программирования. Сделать вывод о проделанной работе.

Сгенерировать последовательность 100 случайных чисел  с экспоненциальным (показательным) законом распределения с параметром  .

Сформировать новую последовательность, состоящую из элементов:  .

Для полученной последовательности вычислить среднее значение, дисперсию и вывести последовательность на печать в виде гистограммы, разделив диапазон на 10 интервалов.

***Справочный материал для выполнения работы:***

Математическое ожидание: ;

Дисперсия: ;

Равномерный закон распределения:  , где  - случайное число диапазона ;

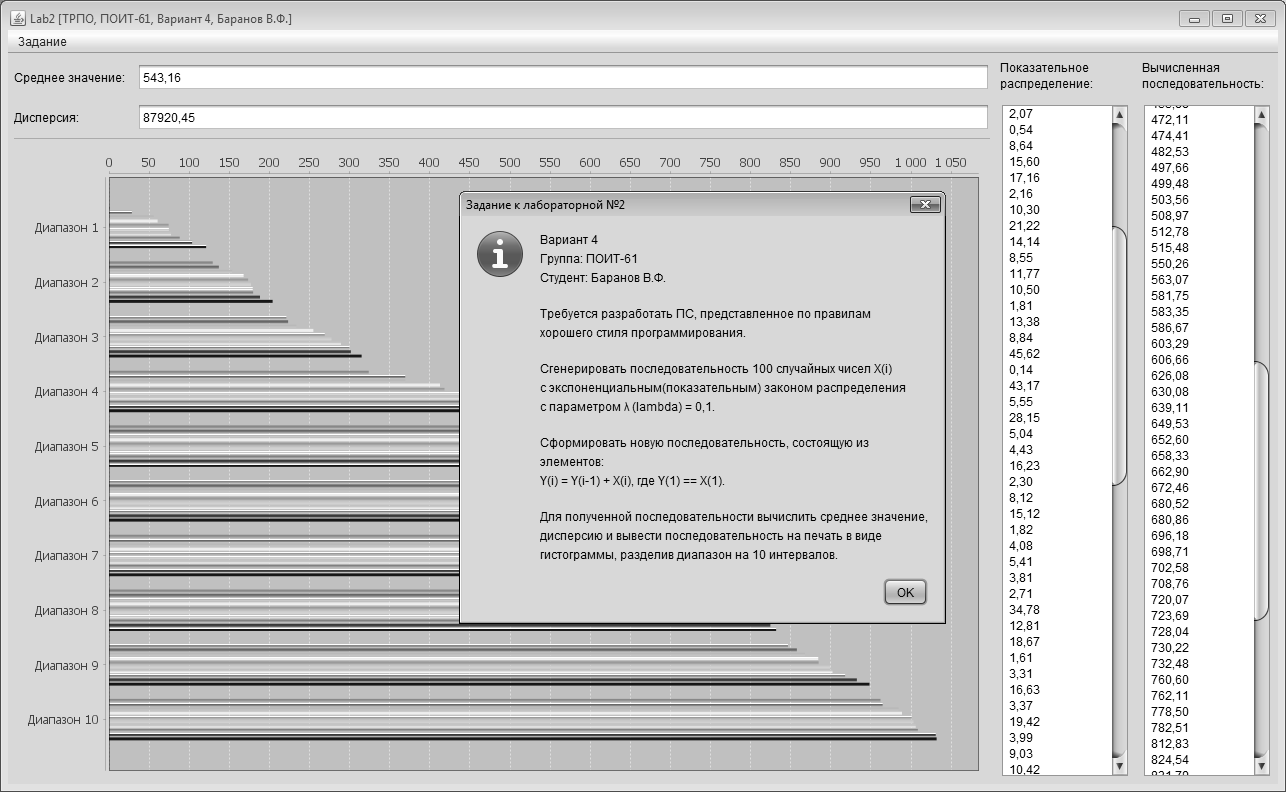
Нормальный закон распределения: ;

Экспоненциальный закон распределения: .

**Ход работы**

***Описание приложения.***

Для решения поставленной задачи, было разработано desktop-приложение на платформе Java. Для создания графического интерфейса пользователя (GUI) использовался функционал библиотеки Swing и возможности GUI-редактора среды разработки NetBeans. Ввиду отсутствия в стандартной библиотеке Swing компонентов для отображения гистограмм, для этой цели использовался опенсурсный фреймворк JFreeChart.



Приложение состоит из 5 классов и 1 интерфейс: Runner.java, Constants.java, IView.java, MainWindow.java, SequenceHandler.java, LogicController.java.

* Интерфейс IView.java содержит контракт методов отображения результатов для поставленной задачи. В случае, если при поддержке приложения потребуется реализовать view-слой с помощью консоли или веб-интерфейса, потребуется лишь добавить реализацию этого интерфейса в приложение, без значительных изменений существующего кода.
* Класс MainWindow.java является наследником класса JFrame.java и главным окном приложения (Рисунок 1). В нём реализован интерфейс IView.java для обработки действий пользователя и отображение результатов работы приложения.
* SequenceHandler.java предоставляет методы вычисления последовательностей, согласно заданию.
* LogicController.java является контроллером приложения - его методы вызываются из view-слоя. В свою очередь, он вызывает требуемые методы обработки данных из класса SequenceHandler.java и методы интерфейса IView для отображения результатов.
* Из класса Runner.java осуществляет запуск приложения, создание и инициализация классов главного окна приложения, контроллера и helper'а для вычислений.
* В классе Constants.java собраны константы и строковые литералы приложения.

Для запуска приложения требуется установленная виртуальная машина Java Runtime Environment (JRE) версии 8 или выше. Она распространяется свободно и для большинства платформ может быть загружена с сайта Oracle. Файлы фреймворка JFreeChart распространяются в каталоге приложения.

***Использованные критерии хорошего стиля программирования.***

При разработке программного средства использовался документ Code Conventions for the Java Programming Language, расположенный на сайте Oracle. В нём описаны основные соглашения по программированию на языке Java, которым следуют разработчики Sun и рекомендуют к использованию всем Java-разработчикам. Они покрывают правила именования файлов, классов, атрибутов и методов, форматирование и стиль написания кода, а также правила комментирования и документирования кода и основные практики программирования, представленные соответствующими примерами.

В связи с большим объёмом, ниже будут перечислены лишь основные пункты соглашения, использованные при разработке:

Написание названий классов приложения осуществляется в CamelCase нотации с первой буквой первого слова в верхнем регистре; констант и строковых литералов в UpperCase с нижним подчёркиванием в качестве разделителя; названий методов, полей и переменных в CamelCase с первым словом, начинающимся в нижнем регистре.

Для комментирования классов, полей, методов и параметров методов применяется функционал генератора документации javaDoc.

Названия классов, полей, методов, констант и переменных обладают смысловой нагрузкой и позволяют понять своё назначение из названия.

В исходном коде приложения использовано минимальное количество неименованных констант, во избежание анти-паттерна проектирования "magic number".

Также, за исключением сгенерированной редактором IDE части класса MainWindow, используется порядок размещения методов и полей класса в зависимости от области видимости: сначала большая видимость, затем меньшая.

Большинство строк кода приложения, за исключением сгенерированной редактором IDE части класса MainWindow, написаны с учётом рекомендации на длину строк в 80 символов.

**Вывод**

Разработанное программное средство полностью выполняет поставленную задачу. При разработке использовалось соглашение по программированию на языке Java, принятое разработчиками языка и которое рекомендуют к использованию всем Java-разработчикам. Они покрывают правила именования файлов, классов, атрибутов и методов, форматирование и стиль написания кода, а также правила комментирования и документирования кода и основные практики программирования. Всё перечисленное в достаточно полном объёме выполняет задачу написания приложения в хорошем стиле.