ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ

ПО ДИСЦИПЛИНЕ

«Защита информации»

Руководители \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Харченко

Москва 2023

СПИСОК ИСПОЛНИТЕЛЕЙ

Студ. группы 201-361 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е. Сильченко

**Ход работы:**

1. Для реализации генератор псевдослучайной последовательности битов на основе регистра сдвига с линейной обратной связью (РСЛОС) в конфигурации Галуа был создан класс GaloisLFSR рисунок 1.

Этот класс содержит конструктор и 2 метода.

Конструктор GaloisLFSR - принимает два массива: register, который содержит начальную последовательность битов, и taps которые указывают на те биты регистра, которые используются для вычисления обратной связи. Создает объект класса GaloisLFSR.

Метод shift - сдвигает элементы регистра на один бит вправо и вычисляет значение обратной связи. Затем вставляет вычисленное значение обратной связи в первый бит регистра.

Метод generate - генерирует последовательность псевдослучайных битов заданной длины, используя метод shift(). Возвращает массив int[], содержащий сгенерированные биты.

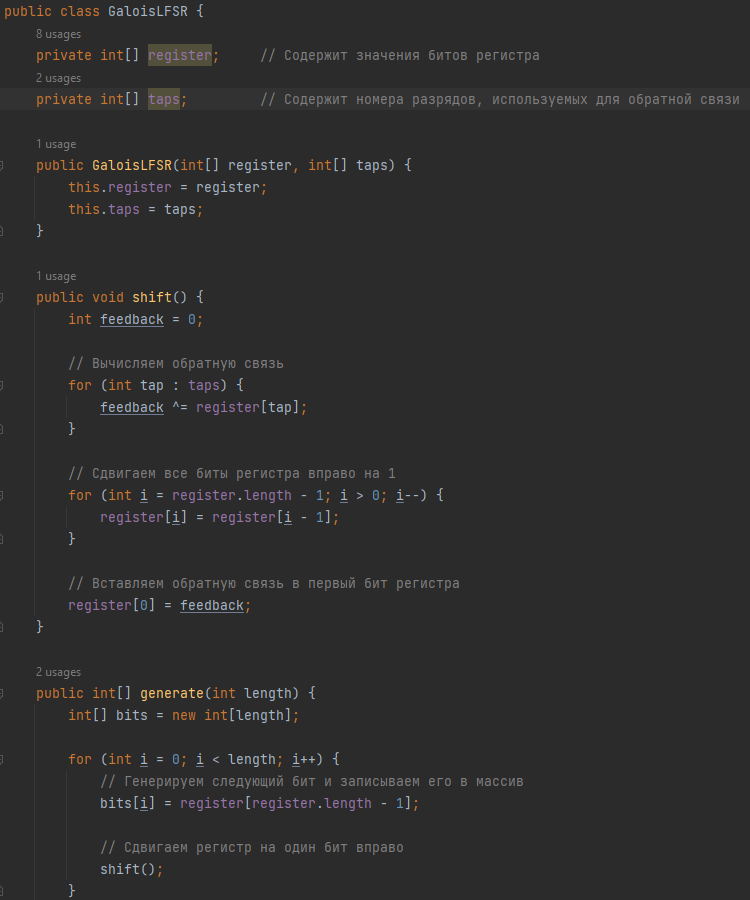


Рисунок 1 – Класс GaloisLFSR.

Далее в Main мы создаем 2 массива содержащих начальное значение регистра и образующий многочлен, создаем экземпляр класс и выводим последовательность в консоль.

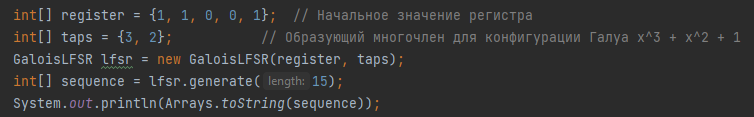


Рисунок 2 – Вывод последовательности в консоль.

1. Результат был представлен в виде точечной диаграммы, где по горизонтали отложены порядковые номера генерируемых битов, а по вертикали – их значения.

Для отображения диаграммы был создан класс LFSRDiagram, который наследуется от класса Panel и предназначен для отображения диаграммы.

Этот класс содержит два метода:

- Конструктор LFSRDiagram - принимает массив sequence, который содержит последовательность битов регистра. Создает объект класса LFSRDiagram.

- Метод paint - отображает диаграмму на панели. Принимает объект Graphics g, который используется для рисования, определяет размеры и ширину полосы на основе размера панели и длины последовательности. Затем отрисовывает оси координат и точки, соответствующие элементам регистра. Радиус окружности каждой точки равен половине ширины полосы, а координата Y вычисляется на основе значения бита.

Класс представлен на рисунке 3.

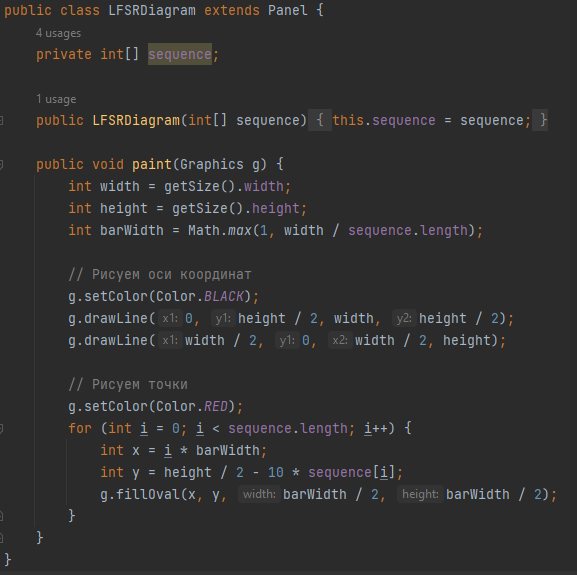


Рисунок 3 - класс LFSRDiagram.

В методе Main создается объект LFSRDiagram с использованием последовательности sequence. Далее создается объект Frame, который представляет окно для отображения диаграммы. В этом объекте добавляется обработчик событий для закрытия окна, который завершает выполнение программы при закрытии окна. Затем объект диаграммы добавляется в объект frame, размер окна устанавливается на 500 x 300, и окно становится видимым с помощью метода setVisible.

Представлен на рисунке 4.

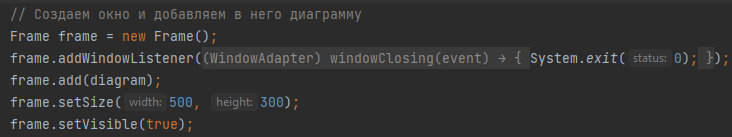


Рисунок 4 – Создание диаграммы.

1. С помощью критерия x^2 оценивается качество генерируемой последовательности рисунок 4.

Переменная numBins, которая хранит количество групп, на которые разбивается последовательность для проверки соответствия распределению равномерному. В данном случае у нас есть только 2 возможных значения (0 и 1), поэтому numBins установлено на 2.

С помощью элементов массива sequence вычисляются частоты появления символов и сохраняются в массив observedFrequencies. Затем вычисляется ожидаемая частота символов и для каждой группы, проводится x^2 тест для проверки соответствия полученных частот равномерному распределению. Результат теста сохраняется в переменную chiSquared.

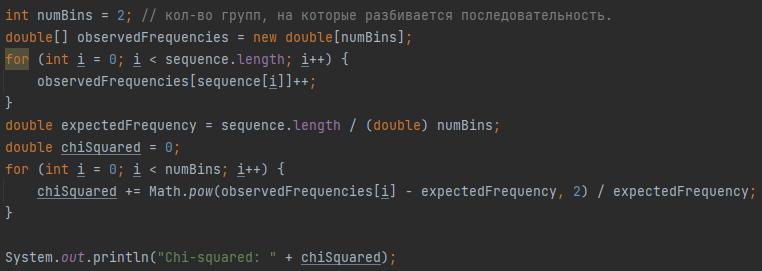


Рисунок 4 – Проверка x^2.

1. Зашифровываем изображение с использованием гаммирования с помощью Линейного Регистра Сдвига (LFSR) рисунок 5.

Загружаем изображение, далее определяем количество блоков в изображении не учитывая заголовок.

Затем для каждого блока получаем следующий 8-битный блок данных из изображения и применяем гаммирование с использованием Линейного Регистра Сдвига (LFSR).

Для гаммирования создается временный массив keyStream, который содержит биты, сгенерированные Линейным Регистром Сдвига (LFSR), затем применяется операция XOR для обеспечения шифрования данных для каждого блока.

В конце зашифрованные данные сохраняются в новый файл с именем "tux\_encrypted.bmp".

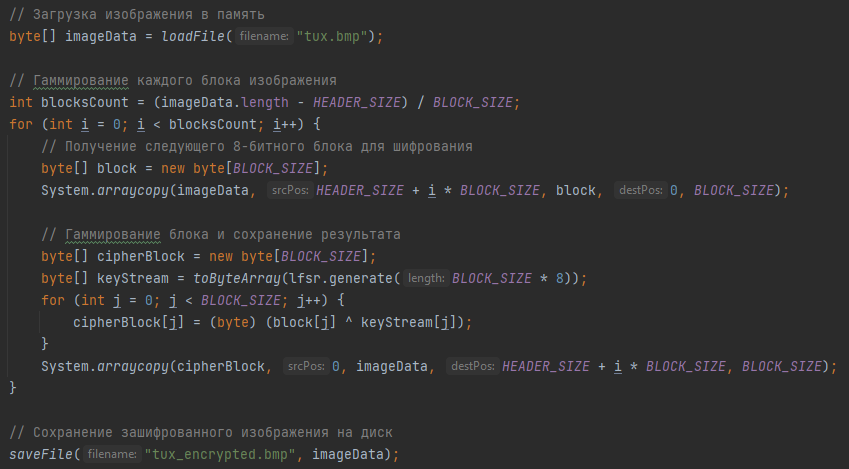


Рисунок 5 – Шифрование изображения.

Итог:

Сгенерированная последовательность битов, точечная диаграмма и оценка качества с помощью критерия x^2 представлено на рисунке 6.

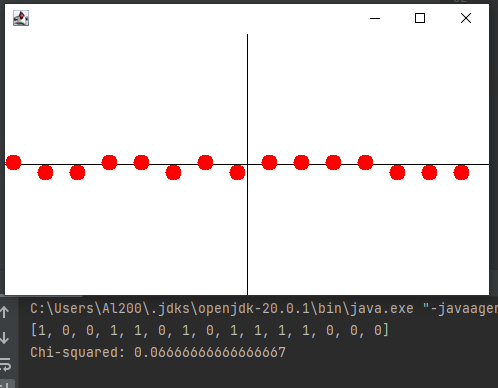


Рисунок 6 – Сгенерированная последовательность точечная диаграмма и оценка качества с помощью критерия x^2.

Зашифрованное изображение представлено на рисунке 7.

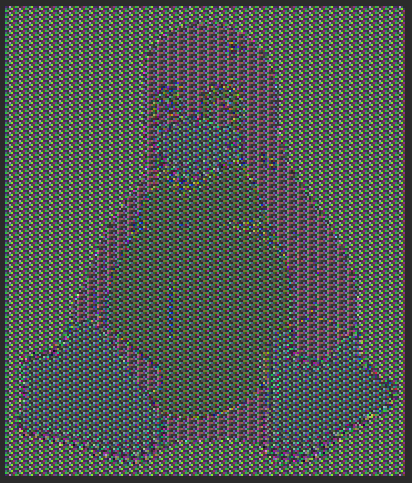


Рисунок 7 – Зашифрованное изображение.