Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Защита информации и надёжность информационных систем»

Отчёт по лабораторной работе №3

ИЗБЫТОЧНОЕ КОДИРОВАНИЕ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ. КОД ХЕММИНГА

Выполнил: Лешук Д. И.

ФИТ 3 курс 7 группа

Преподаватель: Николайчук А.Н

Минск 2024

**Цель:** приобретение практических навыков кодирования/ декодирования двоичных данных при использовании кода Хемминга.

**Практическое задание:**

1. На основе информационного сообщения, представленного символами русского/английского алфавитов, служебными символами и цифрами, содержащегося в некотором текстовом файле сформировать информационное сообщение в двоичном виде; длина сообщения в бинарном виде должна быть не менее 16 символов. Для выполнения этого задания будем использовать коды ASCII символов алфавита.

2. Для полученного информационного слова построить проверочную матрицу Хемминга (значение минимального кодового расстояния принято равным 3).

3. Используя построенную матрицу, вычислить избыточные символы (слово Xr).

4. Принять исходное слово со следующим числом ошибок: 0, 1, 2. Позиция ошибки определяется (генерируется) случайным образом.

5. Для полученного слова Yn = Yk, Yr, используя уже известную проверочную матрицу Хемминга, вновь вычислить избыточные символы (обозначим их Yr’), используя выражение (4.6).

6. Вычислить и проанализировать синдром. В случае, если анализ синдрома показал, что информационное сообщение было передано с ошибкой (или 2 ошибками), сгенерировать унарный вектор ошибки Еn = е1, е2, …, еn и исправить одиночную ошибку, проанализировать ситуацию при возникновении ошибки в 2 битах.

**Выполнение работы:**

Определим наше входное слово – пусть оно будет преобразованием произвольных двух символов в ASCII, например, строка «h6» будет иметь вид: «0110100000110110». Далее определим функцию, которая будет строить матрицу Хемминга – рисунок 1. В начале мы вычисляем все необходимые нам величины, такие как число строк (r), количество столбцов матрицы – k + r, где k – длина нашего информационного слова (она равна 16). После заполняем матрицу значениями 0,1.

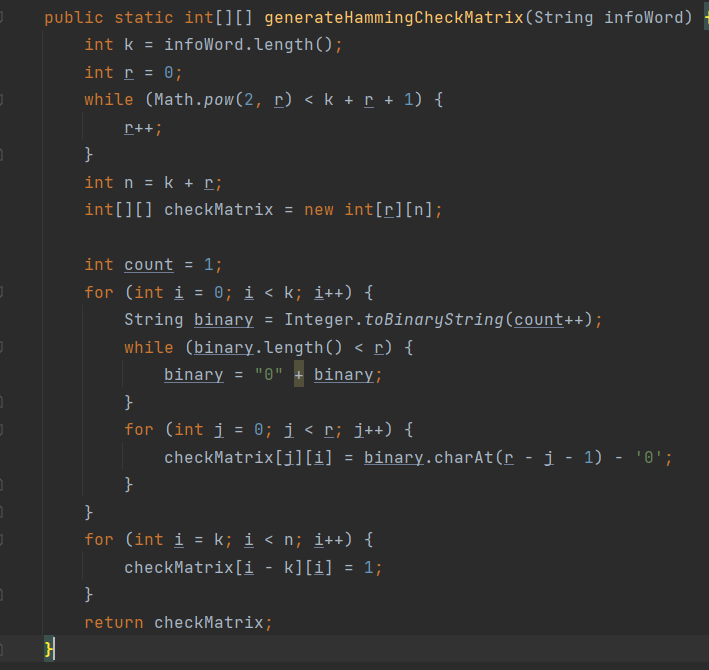


Рисунок 1 – Код функции для создания матрицы Хемминга

В итоге у нас получилась матрица с характеристиками (21,16) – рисунок 2.

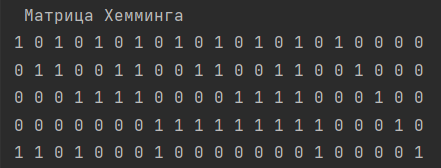


Рисунок 2 – Матрица Хемминга

Далее определяем функцию для вычисления избыточных битов, должна получиться последовательность из 5 символов – рисунок 3. Мы выполняем умножение нашего исходного слова на матрицу (операция XOR).

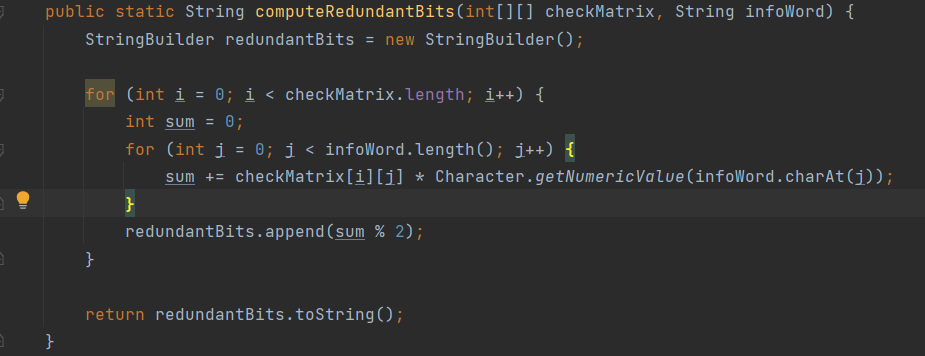


Рисунок 3 – Функция вычисления избыточных битов

Для нашего слова без ошибок избыточные биты получились равными «01001». В итоге закодированное слово равно: «011010000011011001001». Следующим шагом внедрим функцию, которая будет инвертировать нужное нам количество бит в исходном слове на случайных позициях – рисунок 4.

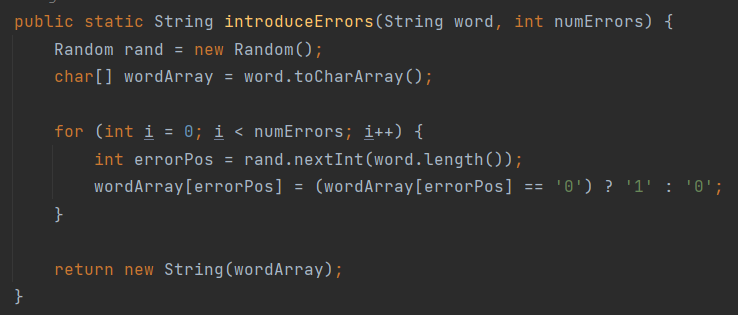


Рисунок 4 – Функция генерации ошибок в исходном сообщении

В первый раз генерируем 1 ошибку, получаем сообщение «0110101000110110». Избыточные биты для этого сообщения равны «10100». В случае с двумя ошибками – «0110101010110110», «01101». Также определим функцию для вычисления вектора ошибок и их исправления – рисунок 5.

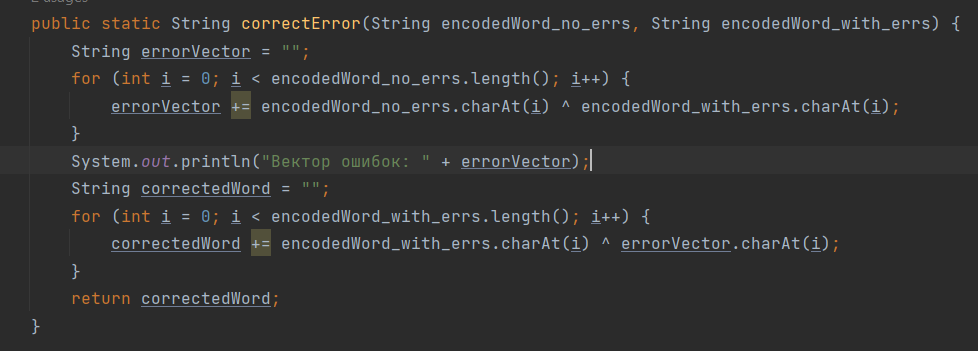


Рисунок 5 – Функция исправления ошибок

В итоге получаем такой результат – рисунок 6.

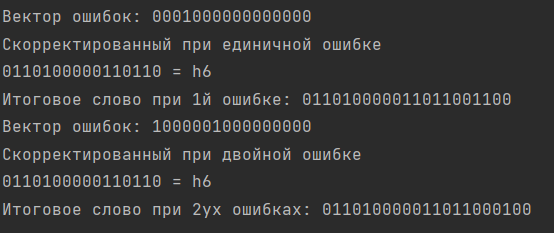


Рисунок 6 – Результат исправления

Мы успешно исправили исходное сообщение, несмотря на то, что значение избыточных битов получилось разным, мы их не учитываем в преобразовании исправленного бинарного сообщения обратно в символьное.

**Вывод:** В результате лабораторной работы были приобретены теоретические знания о помехоустойчивом кодировании с использованием избыточной информации, приобретены практические навыки разработки функций для взаимодействия с двоичными данными в информационных системах с использованием кода Хемминга – линейного блочного кода.