Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Отчет защищен

с оценкой *\_\_\_\_\_*

Преподаватель

А.Н. Вологин

«20» октября 2022

**КОДИРОВАНИЕ**

Отчет о лабораторной работе №2

по дисциплине «Компьютерные сети»

ЯГТУ 09.03.04 – 002 ЛР

Отчет выполнил

студент группы ЦПИ-21

Д.В. Аристов

«07» октября 2022

2022

Цель работы: изучить механизм работы логического кодирования на основе метода скремблирования. Рекомендации к выполнению: отчет оформлять согласно СТО, в отчете должны быть отображены формулы расчетов и результат, а также сделан вывод о проделанной работе.

Задание: а) Прогнать последовательность 1111(40 нулей)1111 через стандартный скремблер и дескремблер.

б) Придумать свой скремблер, который покажет лучшее значение, чем стандартный, с исходной последовательностью.

Стандартный скремблер:

,

где -двоичная цифра результирующего кода, полученная на *i-м* такте работы скремблера, -двоичная цифра исходного кода, поступающая на *i-м* такте на вход скремблера, и -двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера, соответственно на 3 и на 5 тактов ранее текущего такта. Члены выражения объединены знаком операции исключающего ИЛИ (сложение по модулю 2). Определить закодированную последовательность для исходной последовательности.

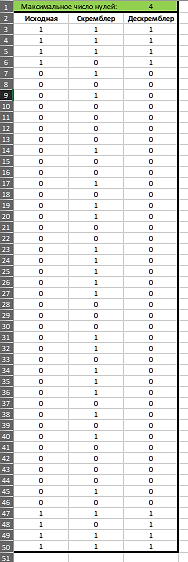


Рисунок 1 – решение в excel

*public void* scramblerDefault(String input) {

String outputDefault = "";  
 String outputDescramblerDefault = "";  
 *int* maxCountZero = 1;  
 *int* maxCountOne = 1;  
 String[] inputArrayStr = input.split("");  
 *int*[] inputArrayInt = *new int*[inputArrayStr.length];  
 *int*[] outputArrayInt = *new int*[inputArrayInt.length];  
  
 *for* (*int* i = 0; i < inputArrayStr.length; i++) {  
 inputArrayInt[i] = Integer.*parseInt*(inputArrayStr[i]);  
 }  
  
 *// СКРЕМБЛИРОВАНИЕ  
 for* (*int* i = 0; i < inputArrayInt.length; i++) {  
 *if* (i < 3) {  
 outputArrayInt[i] = inputArrayInt[i];  
 } *else if* (i >= 3 && i < 5) {  
 outputArrayInt[i] = (inputArrayInt[i] + outputArrayInt[i-3]) % 2;  
 } *else* {  
 outputArrayInt[i] = (inputArrayInt[i] + outputArrayInt[i-3] + outputArrayInt[i-5]) % 2;  
 }  
 outputDefault += outputArrayInt[i];  
 }  
  
 *// ДЕСКРЕМБЛИРОВАНИЕ  
 for* (*int* i = 0; i < outputArrayInt.length; i++) {  
 *if* (i < 3) {  
 outputDescramblerDefault += outputArrayInt[i];  
 } *else if* (i >= 3 && i < 5) {  
 outputDescramblerDefault += (outputArrayInt[i] + outputArrayInt[i-3]) % 2;  
 } *else* {  
 outputDescramblerDefault += (outputArrayInt[i] + outputArrayInt[i-3] + outputArrayInt[i-5]) % 2;  
 }  
 }  
  
 *// ПОДСЧЕТ МАКСИМАЛЬНОЙ ПОВТОРЯЮЩЕЙСЯ ПОСЛЕДОВАТЕЛЬНОСТИ  
 int* tempOne = 1, tempZero = 1;  
 *for* (*int* i = 1; i < outputArrayInt.length; i++) {  
 *if* (outputArrayInt[i] == outputArrayInt[i-1] && outputArrayInt[i] == 1) {  
 tempOne++;  
 } *else* {  
 *if* (tempOne > maxCountOne) maxCountOne = tempOne;  
 tempOne = 1;  
 }  
  
 *if* (outputArrayInt[i] == outputArrayInt[i-1] && outputArrayInt[i] == 0) {  
 tempZero++;  
 } *else* {  
 *if* (tempZero > maxCountZero) maxCountZero = tempZero;  
 tempZero = 1;  
 }  
 }  
  
 System.***out***.println("Базовый скремблер:");  
 System.***out***.println("Последовательность до скремблирования: " + input);  
 System.***out***.println("Последовательность после скремблирования: " + outputDefault);  
 System.***out***.println("Последовательность после дескремблирования: " + outputDescramblerDefault);  
 System.***out***.println("Максимальная последовательность нулей: " + maxCountZero + "; Максимальная последовательность единиц: " + maxCountOne);  
 System.***out***.println();  
}

Листинг программы, реализующий базовый скремблер

На вход функции scramblerDefault() подается кодовая последовательность в виде строки. Затем из этой строки получается массив строк, где каждый элемент — это число входной последовательности в виде строки. Затем получаем массив чисел, где каждый элемент цифра последовательности. Затем происходит процесс скремблирования и дескремблирования с учетом формулы и вывод результатов выполнения программы в консоль.

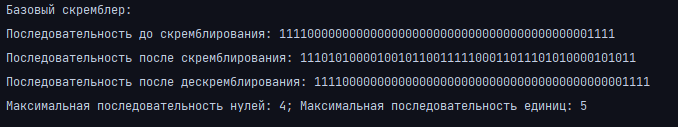


Рисунок 2 – результат выполнения программы базового скремблера

*б) Собственный скремблер:*

*Дескремблер:*

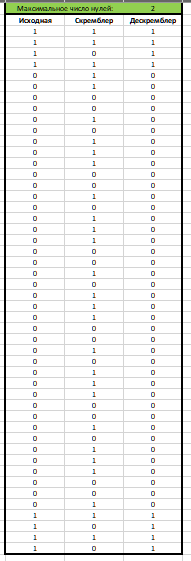


Рисунок 3 – решение в excel

*public void* scramblerMy(String input) {  
 String outputMy = "";  
 String outputDescramblerMy = "";  
 *int* maxCountZero = 1;  
 *int* maxCountOne = 1;  
 String[] inputArrayStr = input.split("");  
 *int*[] inputArrayInt = *new int*[inputArrayStr.length];  
 *int*[] outputArrayInt = *new int*[inputArrayInt.length];  
  
 *for* (*int* i = 0; i < inputArrayStr.length; i++) {  
 inputArrayInt[i] = Integer.*parseInt*(inputArrayStr[i]);  
 }  
  
 *// СКРЕМБЛИРОВАНИЕ  
 for* (*int* i = 0; i < inputArrayInt.length; i++) {  
 *if* (i < 2) {  
 outputArrayInt[i] = inputArrayInt[i];  
 } *else if* (i == 2) {  
 outputArrayInt[i] = (inputArrayInt[i] + outputArrayInt[i-2]) % 2;  
 } *else* {  
 outputArrayInt[i] = (inputArrayInt[i] + outputArrayInt[i-2] + outputArrayInt[i-3]) % 2;  
 }  
 outputMy += outputArrayInt[i];  
 }  
  
 *// ДЕСКРЕМБЛИРОВАНИЕ  
 for* (*int* i = 0; i < outputArrayInt.length; i++) {  
 *if* (i < 2) {  
 outputDescramblerMy += outputArrayInt[i];  
 } *else if* (i == 2) {  
 outputDescramblerMy += (outputArrayInt[i] + outputArrayInt[i-2]) % 2;  
 } *else* {  
 outputDescramblerMy += (outputArrayInt[i] + outputArrayInt[i-2] + outputArrayInt[i-3]) % 2;  
 }  
 }  
  
 *// ПОДСЧЕТ МАКСИМАЛЬНОЙ ПОВТОРЯЮЩЕЙСЯ ПОСЛЕДОВАТЕЛЬНОСТИ  
 int* tempOne = 1, tempZero = 1;  
 *for* (*int* i = 1; i < outputArrayInt.length; i++) {  
 *if* (outputArrayInt[i] == outputArrayInt[i-1] && outputArrayInt[i] == 1) {  
 tempOne++;  
 } *else* {  
 *if* (tempOne > maxCountOne) maxCountOne = tempOne;  
 tempOne = 1;  
 }  
  
 *if* (outputArrayInt[i] == outputArrayInt[i-1] && outputArrayInt[i] == 0) {  
 tempZero++;  
 } *else* {  
 *if* (tempZero > maxCountZero) maxCountZero = tempZero;  
 tempZero = 1;  
 }  
 }  
  
 System.***out***.println("Собственный скремблер:");  
 System.***out***.println("Последовательность до скремблирования: " + input);  
 System.***out***.println("Последовательность после скремблирования: " + outputMy);  
 System.***out***.println("Последовательность после дескремблирования: " + outputDescramblerMy);  
 System.***out***.println("Максимальная последовательность нулей: " + maxCountZero + "; Максимальная последовательность единиц: " + maxCountOne);  
 System.***out***.println();  
}

Листинг программы, реализующий собственный скремблер

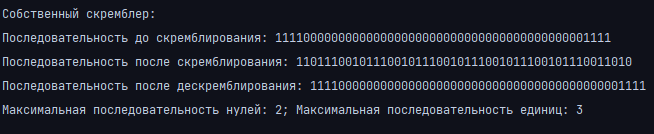


Рисунок 4 – результат выполнения программы базового скремблера

Вывод: в течение данной лабораторной работы были изучены механизмы логического кодирования на основе метода скремблирования. Кроме того, был получен навык самостоятельного составления скремблера и дескремблера.