

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

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:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$$

где B_i - двоичная цифра результирующего кода, полученная на i -м такте работы скремблера, A_i - двоичная цифра исходного кода, поступающая на i -м такте на вход скремблера, B_{i-3} и B_{i-5} - 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.

1	Максимальное число нулей: 4			4
2	Исходная	Скремблер	Дескремблер	
3	1	1	1	
4	1	1	1	
5	1	1	1	
6	1	0	1	
7	0	1	0	
8	0	0	0	
9	0	1	0	
10	0	0	0	
11	0	0	0	
12	0	0	0	
13	0	0	0	
14	0	1	0	
15	0	0	0	
16	0	0	0	
17	0	1	0	
18	0	0	0	
19	0	1	0	
20	0	1	0	
21	0	0	0	
22	0	0	0	
23	0	1	0	
24	0	1	0	
25	0	1	0	
26	0	1	0	
27	0	1	0	
28	0	0	0	
29	0	0	0	
30	0	0	0	
31	0	1	0	
32	0	1	0	
33	0	0	0	
34	0	1	0	
35	0	1	0	
36	0	1	0	
37	0	0	0	
38	0	1	0	
39	0	0	0	
40	0	1	0	
41	0	0	0	
42	0	0	0	
43	0	0	0	
44	0	0	0	
45	0	1	0	
46	0	0	0	
47	1	1	1	
48	1	0	1	
49	1	1	1	
50	1	1	1	
51				

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[inputArrayStr.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


```

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

