Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 1

По дисциплине СМЗКС

Тема: «Избыточное кодирование данных в информационных системах. Код Хемминга»

Выполнил:

Студент 4 курса

Группы ИИ-16

Журавлёв В.А.

Проверил:

Хацкевич М.В.

Брест, 2022

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

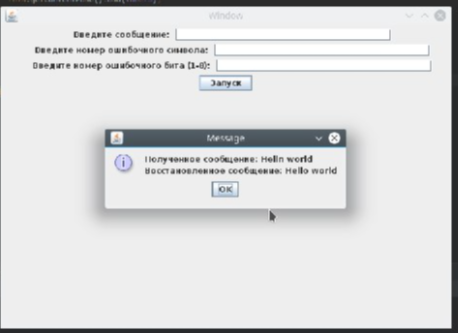
**Задание:**

Разработать приложение для кодирования/декодирования двоичной информации кодом Хемминга с минимальным кодовым расстоянием 3.

**Код программы:**

|  |
| --- |
| import java.util.Arrays; |
|  | import java.util.Scanner; |
|  |  |
|  | public class Hamming3 { |
|  |  |
|  | static int[][] matrix = { |
|  | {0, 0, 0, 1, 1, 1, 1}, |
|  | {0, 1, 1, 0, 0, 1, 1}, |
|  | {1, 0, 1, 0, 1, 0, 1} |
|  | }; |
|  |  |
|  | static int[] errorIndexMatcher = new int[]{2,4,5,6}; |
|  |  |
|  | public static void main(String[] args) { |
|  |  |
|  | Scanner scanner = new Scanner(System.in); |
|  | System.out.print("Введите текст для отправки: "); |
|  | String text = scanner.nextLine(); |
|  | System.out.print("Введите номер символа с ошибкой: "); |
|  | int errorWordNumber = (scanner.nextInt() - 1) \* 2; |
|  | System.out.print("Введите номер бита с ошибкой (1-8): "); |
|  | int errorBitNumber = scanner.nextInt() - 1; |
|  | if(errorBitNumber >= 4){ |
|  | errorWordNumber++; |
|  | errorBitNumber%=4; |
|  | } |
|  | int[][] textIntBytes = parseText(text); |
|  |  |
|  | start(textIntBytes, errorWordNumber, errorBitNumber); |
|  | } |
|  |  |
|  | public static int[][] parseText(String text) { |
|  | int[][] result = new int[text.length() \* 2][4]; |
|  | for (int i = 0; i < text.length(); i++) { |
|  | char c = text.charAt(i); |
|  | String binaryChar = Integer.toBinaryString(c); |
|  | while (binaryChar.length() < 8) { |
|  | binaryChar = "0".concat(binaryChar); |
|  | } |
|  | String left = binaryChar.substring(0, 4); |
|  | String right = binaryChar.substring(4); |
|  | for (int j = 0; j < 4; j++) { |
|  | result[i \* 2][j] = Integer.parseInt(String.valueOf(left.charAt(j))); |
|  | result[i \* 2 + 1][j] = Integer.parseInt(String.valueOf(right.charAt(j))); |
|  | } |
|  | } |
|  | return result; |
|  | } |
|  |  |
|  | public static String[] start(int[][] words, int errorWordNumber,int errorBitNumber) { |
|  | int[] resultArray = new int[words.length \* 4]; |
|  | int[] textWithErrorArray = new int[words.length \* 4]; |
|  | int resultIndex = 0; |
|  |  |
|  | // int errorWordNumber = (int) (Math.random() \* words.length); |
|  | // int errorBitNumber = (int) (Math.random() \* 4); |
|  | System.out.println(String.format("Ошибка возникла в группе #%s - %s, в бите №%s", errorWordNumber, Arrays.toString(words[errorWordNumber]), errorBitNumber + 1)); |
|  | System.out.println("------------------------------------------"); |
|  | for (int i = 0; i < words.length; i++) { |
|  | int[] word = words[i]; |
|  | int[] codeWord = new int[7]; |
|  |  |
|  | for (int j = 0; j < 4; j++) { |
|  | textWithErrorArray[i\*4 + j] = word[j]; |
|  | } |
|  |  |
|  | //(r1, r2, x, r3, x, x, x) |
|  |  |
|  | codeWord[2] = word[0]; |
|  | codeWord[4] = word[1]; |
|  | codeWord[5] = word[2]; |
|  | codeWord[6] = word[3]; |
|  |  |
|  | //r1=u3+u5+u7 |
|  | //r2=u3+u6+u7 |
|  | //r3=u5+u6+u7 |
|  | codeWord[0] = (codeWord[2] + codeWord[4] + codeWord[6]) % 2; |
|  | codeWord[1] = (codeWord[2] + codeWord[5] + codeWord[6]) % 2; |
|  | codeWord[3] = (codeWord[4] + codeWord[5] + codeWord[6]) % 2; |
|  |  |
|  | //добавляем ошибку |
|  | if (errorWordNumber == i) { |
|  | System.out.println("Отправляемое сообщение: " + Arrays.toString(codeWord)); |
|  |  |
|  | codeWord[errorIndexMatcher[errorBitNumber]] ^= 1; |
|  | textWithErrorArray[i\*4 + errorBitNumber] ^= 1; |
|  |  |
|  | System.out.println("Полученное сообщение: " + Arrays.toString(codeWord)); |
|  | } |
|  |  |
|  | int resultErrorIndex = 0; |
|  | for (int k = 0; k < matrix.length; k++) { |
|  | int result = 0; |
|  | for (int j = 0; j < codeWord.length; j++) { |
|  | result += codeWord[j] \* matrix[k][j]; |
|  | } |
|  | result %= 2; |
|  | resultErrorIndex += Math.pow(2, 2 - k) \* result; |
|  | } |
|  |  |
|  |  |
|  | if (resultErrorIndex >= 1) { |
|  | System.out.println(String.format("Обнаружена ошибка в бите №%s", (resultErrorIndex))); |
|  | codeWord[resultErrorIndex - 1] ^= 1; |
|  | } |
|  | if (i == errorWordNumber) { |
|  | System.out.println("Исходное сообщение: " + codeWord[2] + "" + codeWord[4] + "" + codeWord[5] + "" + codeWord[6]); |
|  | System.out.println("------------------------------------------"); |
|  | } |
|  | resultArray[resultIndex] = codeWord[2]; |
|  | resultArray[resultIndex + 1] = codeWord[4]; |
|  | resultArray[resultIndex + 2] = codeWord[5]; |
|  | resultArray[resultIndex + 3] = codeWord[6]; |
|  | resultIndex += 4; |
|  | } |
|  |  |
|  | System.out.print("Полученное сообщение: "); |
|  | StringBuilder message = new StringBuilder(); |
|  | for (int i = 0; i < words.length / 2; i++) { |
|  | char c = 0; |
|  | for (int j = 0; j < 8; j++) { |
|  | c += Math.pow(2, 7 - j) \* textWithErrorArray[i \* 8 + j]; |
|  | } |
|  | System.out.print(c); |
|  | message.append(c); |
|  | } |
|  |  |
|  | System.out.println(); |
|  |  |
|  | System.out.print("Восстановленное сообщение: "); |
|  | StringBuilder restoredMessage = new StringBuilder(); |
|  | for (int i = 0; i < words.length / 2; i++) { |
|  | char c = 0; |
|  | for (int j = 0; j < 8; j++) { |
|  | c += Math.pow(2, 7 - j) \* resultArray[i \* 8 + j]; |
|  | } |
|  | System.out.print(c); |
|  | restoredMessage.append(c); |
|  | } |
|  |  |
|  | return new String[]{message.toString(), restoredMessage.toString()}; |
|  |  |
|  | } |
|  | } |

**Результаты работы программы:**

****

**Вывод:** приобрел практические навыки кодирования/декодирования двоичных данных при использовании кода Хемминга.