#### Министерство образования и науки Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

информационного менеджмента

**Лабораторная работа №6**

**по дисциплине «Администрирование и безопасность информационных систем»**

Выполнил:

Ст. гр. ПРИ-114

Мочалова А.М.

Проверил:

Герасимов П.К.

Владимир 2017

**1. Цель работы:**

Освоить на практике применение алгоритма шифрования с помощью гаммирования

**2. Ход работы:**

package javaapplication3;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.UnsupportedEncodingException;

import java.math.BigInteger;

import java.util.Arrays;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

public class JavaApplication3 {

public static final String gammastring = "0123456789";

public static final String alb = "abcdefghijklmnopqrstuvwxyz";

public static void main(String[] args) throws UnsupportedEncodingException, IOException {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

String[][] alphabet = new String[2][alb.length()];

int count = 0;

for (char c : alb.toCharArray()) {

alphabet[0][count] = Character.toString(c);

byte b[] = Character.toString(c).getBytes("UTF-8");

BigInteger smth = new BigInteger(b);

alphabet[1][count] = smth.toString(2);

count++;

}

Scanner in = new Scanner(System.in);

System.out.println("Enter message");

String messagetmp = reader.readLine();

while(!(chstr(messagetmp))){

System.out.println("Enter string may contain only lowercase letters");

messagetmp = reader.readLine();

} String[][] message = new String[2][messagetmp.toCharArray().length] ;

int count3 = 0;

for (char c : messagetmp.toCharArray()) {

message[0][count3] = String.valueOf(c);

count3++;

}

String[][] gamma = new String[2][messagetmp.length()];

System.out.println("Enter gamma size");

int gammaSize = in.nextInt();

if (gammaSize > alb.length()) {

gammaSize = alb.length();

System.out.println("Decreased gamma size to " + alb.length());

}

count = 0;

System.out.println("Enter gamma numbers");

String keytmp;

while (gammaSize != 0) {

try {

keytmp = reader.readLine();

while(!chkey(keytmp)){

keytmp = reader.readLine();

}

int gammaElem = Integer.parseInt(keytmp);

gamma[0][count] = (String.valueOf(gammaElem));

byte b[] = String.valueOf(gammaElem).getBytes("UTF-8");

BigInteger smth = new BigInteger(b);

gamma[1][count] = "0" + smth.toString(2);

count++;

gammaSize--;

} catch (IOException ex) { Logger.getLogger(JavaApplication3.class.getName()).log(Level.SEVERE, null, ex);

}

}

int count2 = 0;

for (int i = count; i < (messagetmp.length()); i++) {

gamma[0][i] = (String.valueOf(gamma[0][count2]));

byte b[] = String.valueOf(gamma[0][count2]).getBytes("UTF-8");

BigInteger smth = new BigInteger(b);

gamma[1][i] = "0" + smth.toString(2);

count2++;

}

System.out.println("--------------------------------------");

for (String[] is : gamma) {

for (String i : is) {

System.out.print(i + " ");

}

System.out.println();

}

System.out.println("--------------------------------------");

for (String[] is : alphabet) {

for (String i : is) {

System.out.print(i + " ");

}

System.out.println();

}

for (int i = 0; i < messagetmp.length(); i++) {

if (Arrays.asList(alphabet[0]).contains(message[0][i])) {

message[1][i] = Integer.toBinaryString(Integer.parseInt(alphabet[1][Arrays.asList(alphabet[0]).indexOf(message[0][i])], 2)+Integer.parseInt(gamma[1][i], 2));

}

}

System.out.println("--------------------------------------");

for (String[] is : message) {

for (String i : is) {

System.out.print(i + " ");

}

System.out.println();

}

}

public static boolean chkey(String keytmp) {

for (int l = 0; l < keytmp.length(); l++) {

if (!(gammastring.contains(String.valueOf(keytmp.charAt(l))))) {

System.out.println("gamma string may contain only numbers");

return false;

}

}

return true;

}

public static boolean chstr(String keytmp) {

for (int l = 0; l < keytmp.length(); l++) {

if (!(alb.contains(String.valueOf(keytmp.charAt(l))))) {

return false;

}

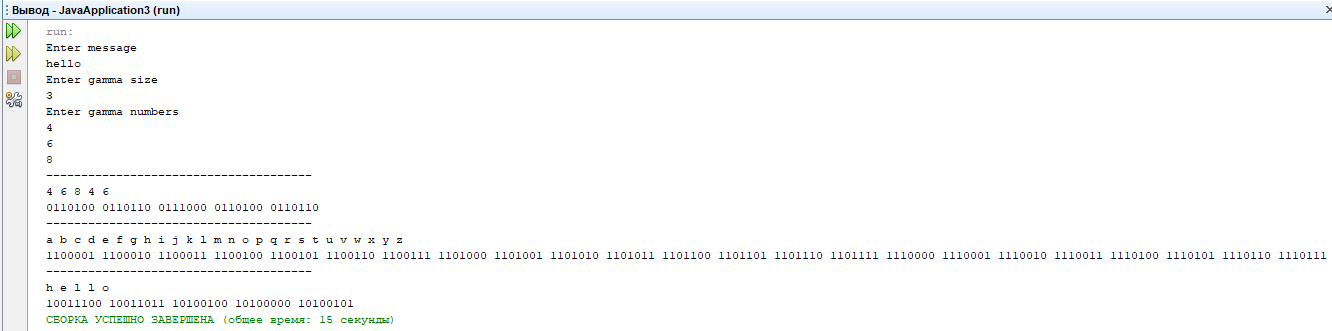
}

return true;

}

}

**3. Результат работы программы**



**4. Вывод**

В ходе выполнения работы были получены навыки написания программ по шифрованию методом гаммирования.