Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

Высшего профессионального образования

Владимирский государственный университет

Кафедра информационных систем

информационного менеджмента

**Лабораторная работа №7**

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

Выполнил:

Ст. гр. ИСТ-114

Мочалова Алена

Проверил:

Герасимов П.К.

Владимир 2017

**1. Цель работы:**

Освоить на практике применение алгоритма шифрования с помощью аналитических преобразований.

**2. Ход работы:**

package javaapplication4;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

import java.util.Set;

import org.jblas.DoubleMatrix;

import org.jblas.Solve;

public class JavaApplication4 {

public static final String gammastring = "0123456789";

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("matrix may contain only numbers");

return false;

}

}

return true;

}

public static double[] multiplicar(double[][] A, int[] B) {

int aRows = A.length;

int aColumns = A[0].length;

int bRows = B.length;

if (aColumns != bRows) {

throw new IllegalArgumentException("A:Rows: " + aColumns + " did not match B:Columns " + bRows + ".");

}

double[] C = new double[B.length];

for (int i = 0; i < aRows; i++) { // aRow

for (int k = 0; k < aColumns; k++) { // aColumn

C[i] += A[i][k] \* B[k];

}

}

return C;

}

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

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

String alb = "abcdefghijklmnopqrstuvwxyz";

ArrayList<Character> alphabet = new ArrayList<>();

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

alphabet.add(c);

}

Scanner in = new Scanner(System.in);

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

String messagetmp = in.nextLine().toLowerCase();

char[] messagetmp2 = messagetmp.toCharArray(); //массив из которого удаляем знаки

Map<Integer, Character> mark = new LinkedHashMap<>(); // запоминаем расположение любых знаков, не попадающих в алфавит

ArrayList<Character> messages = new ArrayList<>(); //Linked - дает нам проход по массиву по порядку, не выкидывая ошибку о выходе за границы массива result

for (int i = 0; i < messagetmp2.length; i++) {

if (alphabet.contains(messagetmp2[i])) {

messages.add(messagetmp2[i]);

} else {

mark.put(i, messagetmp2[i]);

}

}

char[] message = new char[messages.size()]; //массив СООБЩЕНИЯ без знаков

for (int i = 0; i < messages.size(); i++) {

message[i] = messages.get(i);

}

System.out.println("Enter the size of matrix");

String keytmp = reader.readLine();

while (!chkey(keytmp)) {

keytmp = reader.readLine();

}

int kolvo = Integer.parseInt(keytmp);

while (kolvo > message.length) {

System.out.println("Enter smaller size PLEASE");

kolvo = in.nextInt();

}

boolean flag = false;

while (message.length % kolvo != 0) {

kolvo--;

flag = true;

}

if (flag) {

System.out.println("We are sorry. We changed your matrix size down to " + kolvo + " for good working program ...");

}

double matrix[][] = new double[kolvo][kolvo];

System.out.println("Enter elements of matrix");

for (int i = 0; i < kolvo; i++) {

for (int j = 0; j < kolvo; j++) {

keytmp = reader.readLine();

while (!chkey(keytmp)) {

keytmp = reader.readLine();

}

matrix[i][j] = Integer.parseInt(keytmp);

}

}

int alph[] = new int[message.length];

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

alph[i] = alphabet.indexOf(message[i]);

System.out.print(alph[i] + " ");

}

System.out.println();

int perem = alph.length;

int shag = 0;

List<int[]> encode = new ArrayList();

int[] messagePart = new int[0];

while (perem >= kolvo) {

messagePart = Arrays.copyOfRange(alph, shag, (shag + kolvo));

shag += kolvo;

perem -= kolvo;

double[] C = multiplicar(matrix, messagePart);

int[] intArray = new int[C.length];

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

intArray[i] = (int) Math.round(C[i]);

}

encode.add(intArray);

int k = 0;

}

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

ArrayList<String> result = new ArrayList<>();

Set<Map.Entry<Integer, Character>> set = mark.entrySet();

int decoded[] = new int[message.length];

int q = 0;

for (int[] is : encode) {

for (int i : is) {

result.add(Integer.toString(i));

decoded[q] = i;

q++;

}

}

double matrixTMP[][] = new double[matrix.length][matrix.length];

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

for (int j = 0; j < matrix.length; j++) {

matrixTMP[i][j] = matrix[i][j];

}

}

DoubleMatrix smth = new DoubleMatrix(matrixTMP);

DoubleMatrix inverseMatrix = Solve.pinv(smth);

matrixTMP = inverseMatrix.toArray2();

System.out.println("Inversed matrix");

for (int i = 0; i < matrixTMP[0].length; i++) {

System.out.println();

for (int j = 0; j < matrixTMP[0].length; j++) {

System.out.print(matrixTMP[i][j] + " ");

}

System.out.println();

}

perem = decoded.length;

shag = 0;

List<int[]> decode = new ArrayList();

messagePart = new int[0];

while (perem >= kolvo) {

messagePart = Arrays.copyOfRange(decoded, shag, (shag + kolvo));

shag += kolvo;

perem -= kolvo;

double[] C = multiplicar(matrixTMP, messagePart);

int[] intArray = new int[C.length];

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

intArray[i] = (int) Math.round(C[i]);

}

decode.add(intArray);

int k = 0;

}

for (String str : result) {

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

}

for (Map.Entry<Integer, Character> me : set) {

result.add(me.getKey(), me.getValue().toString()); //проставляем знаки в нужных местах

}

System.out.println();

System.out.println("And the codes of our words again");

ArrayList<String> decodedresult = new ArrayList<>();

for (int[] is : decode) {

for (int i : is) {

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

decodedresult.add(alphabet.get(i).toString());

}

}

System.out.println();

System.out.println("And the encoded message");

for (Map.Entry<Integer, Character> me : set) {

decodedresult.add(me.getKey(), me.getValue().toString()); //проставляем знаки в нужных местах

}

for (String str : decodedresult) {

System.out.print(str);

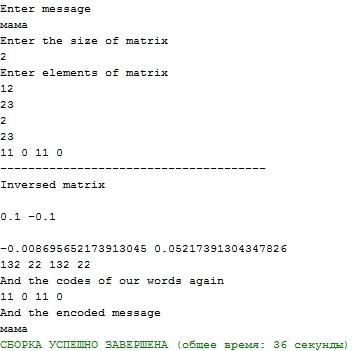
}

System.out.println();

}

}

**3. Результат работы программы**

****

**4. Вывод**

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