­­МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**ЗВІТ**

про виконання лабораторної роботи № 1  
з дисципліни

***«*АЛГОРИТМИ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНИХ МЕРЕЖАХ*»***

Виконав:  
студент  
групи ПМ-33  
Павло Гутковський  
Прийняв:   
Ліснічук А. Є.

Львів 2023

­­ **Лабораторна робота №1**

***Тема:***

Шифр зсуву. Процедури шифрування та дешифрування

***Мета:***

Отримати навики практичної реалізації процедур шифрування та дешифрування шифром зсуву (n = 25).

***Постановка задачі:***

1. програмно реалізувати шифр зсуву для української абетки (процедуру шифрування та дешифрування);

2. написати текст про вулицю, на якій живете (довжиною понад 100 символів – відкритий текст; розділові знаки та пробіли не використовувати) і записати його у файл;

3. прочитати відкритий текст з файлу і зашифрувати його шифром зсуву; як ключ використовувати дату свого народження (наприклад, якщо студент народився 17 лютого, то використовує ключ n = 17); результат записати у файл;

4. дешифрувати цю криптограму з метою отримання відкритого тексту;

5. повторити попередні пункти, але уже з англійською абеткою, доповненою розділовими знаками;

6. підготувати опис/звіт про виконану роботу.

***Короткі теоретичні відомості:***

Шифр зсуву відноситься до класу шифрів простої заміни. При шифруванні кожен символ вхідного тексту замінюємо на інший символ тієї ж абетки, тільки зміщений вправо на n позицій. Цей шифр використовували ще в давньоримській імперії. Зокрема, Юлій Цезар використовував зміщення на три позиції (n = 3):

тобто, латинській абетці a b c d e f g h i …. v w x y z

ставилася у відповідність зміщена абетка d e f g h i j k l …. y z a b c

причому останні літери абетки заміщали відповідно першими літерами.

Наприклад, у результаті шифрування слова ‘highway’ отримуємо криптограму ‘kljkzdb’. При такому шифруванні величина зсуву n є ключем. Число можливих ключів визначається числом літер використовуваної абетки.

При дешифруванні кожен символ шифрованого тексту (криптограми) замінюємо на інший символ тієї ж абетки, тільки зміщений у протилежному напрямку (вліво) на n позицій.

***Програмна реалізація:***

*import* java.io.\*;  
*import* java.util.InputMismatchException;  
*import* java.util.NoSuchElementException;  
*import* java.util.Scanner;  
  
*public class* Main {  
  
 *public static* String readFromFile(String filePath) {  
 StringBuilder content = *new* StringBuilder();  
  
 *try* {  
 *var* FA = *new* FileReader(filePath);  
 Scanner in = *new* Scanner(FA);  
 content.append(in.next());  
 in.close();  
 } *catch* (FileNotFoundException ex) {  
 System.out.println("Файл не створено!");  
 *return* "";  
 } *catch* (InputMismatchException ex){  
 System.out.println("Перевірте запис в файлі!");  
 *return* "";  
 } *catch* (NoSuchElementException ex) {  
 System.out.println("У файлі не достатньо елементів!");  
 *return* "";  
 }  
 *return* content.toString();  
 }  
  
 *public static void* writeToFile(String filePath, String str) {  
 *try* (PrintWriter out = *new* PrintWriter(filePath)) {  
 out.println(str);  
 } *catch* (IOException ex) {  
 System.out.println(ex.getMessage());  
 }  
 }  
  
 *public static void* main(String[] args){  
 *var* N = 25;  
 *var* ukrainianStr = "ЛьвівєдинанаУкраїнімістоуякомузбереглисяархітектурніспорудичасівРенесансуНайбільшяскравимиприкладамицьогостилюслужатьцеркваУспінняікаплицяТрьохСвятихтакожкількаіншихбудинків";  
 *var* englishStr = "Lviv is the historical capital of Galicia and Western Ukraine. It is a big cultural, political and religious centre of Ukraine. Lviv was founded in the century by Prince Danylo Romanovych";  
 writeToFile("encrypt.txt", ShiftCipher.encrypt(englishStr, N));  
 *var* decipherStr = ShiftCipher.decipher(readFromFile("encrypt.txt"), N);  
 System.out.println(decipherStr);  
 writeToFile("decipher.txt", decipherStr);  
 }  
}

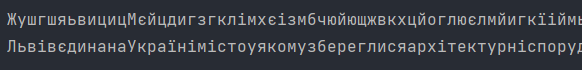
*import* java.util.stream.IntStream;  
  
*public class* ShiftCipher {  
 *private static final char*[][] UKRAINIAN\_ALPHABET = {  
 {'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Є', 'Ж', 'З', 'И', 'І', 'Ї', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ю', 'Я'},  
 {'а', 'б', 'в', 'г', 'д', 'е', 'є', 'ж', 'з', 'и', 'і', 'ї', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ь', 'ю', 'я'}  
 };  
 *private static final char*[][] ENGLISH\_ALPHABET = {  
 {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'},  
 {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'},  
 {' ', ',', '.', ';', ':', '!', '?', '(', ')', '[', ']', '\'', '\"', '-'}  
 };  
  
 *public static* String encrypt(String stringToEncrypt, *int* bias) {  
 *return* moveTo(stringToEncrypt, bias);  
 }  
  
 *public static* String decipher(String stringToEncrypt, *int* bias) {  
 *return* moveTo(stringToEncrypt, -bias);  
 }  
  
 *public static* String moveTo(String stringToEncrypt, *int* bias) {  
 *var* encryptString = *new* StringBuilder();  
 *var* isUkraineAlphabet = Character.UnicodeBlock.of(stringToEncrypt.charAt(0)) == Character.UnicodeBlock.CYRILLIC;  
  
 *for* (*int* i = 0; i < stringToEncrypt.length(); i++)  
 encryptString.append(encryptChar(stringToEncrypt.charAt(i), bias, isUkraineAlphabet));  
  
 *return* encryptString.toString();  
 }  
  
 *private static char* encryptChar(*char* charToEncrypt, *int* bias, *boolean* isUkraineAlphabet) {  
 *var* alphabet = isUkraineAlphabet ? UKRAINIAN\_ALPHABET : ENGLISH\_ALPHABET;  
 *var* alph = !Character.isLetter(charToEncrypt) ? alphabet[2] : Character.isUpperCase(charToEncrypt) ? alphabet[0] : alphabet [1];  
 *var* position = IntStream.range(0, alph.length)  
 .filter(i -> alph[i] == charToEncrypt)  
 .findFirst()  
 .orElse(-1);  
 *var* delta = ((bias > 0 && position + Math.abs(bias) < alph.length) || (bias < 0 && position >= Math.abs(bias)))  
 ? position + bias  
 : position - (bias / Math.abs(bias)) \* alph.length + bias;  
  
 *return* alph[delta];  
 }  
}

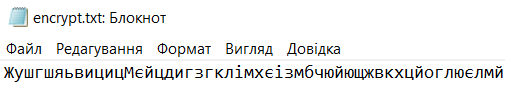
***Опис реалізації:***

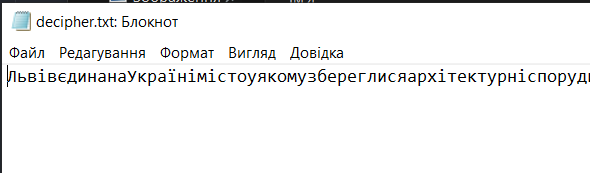
Створюмо клас ShiftCipher який місить методи encrypt та decrypt для шифрування та дешифрування. Допоміжні поля з алфавітами мов для шифрування та метод encryptChar що шифрує певний символ.

***Результати виконання:***

Для українського алфавіту







Для англійського алфавіту



