**Міністерство освіти України**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет інформатики та обчислювальної техніки**

**ЗВІТ**

до лабораторної роботи № 1

з дисципліни “ООП”

**“Основні концепції об’єктно-орієнтованого програмування”**

Виконав:

Студент групи ІП-71

Козирєв Олександр Анатолійович

Київ 2018

ЗМІСТ

Вступ………………………………………………………………………………3

1. Ціль лабораторної роботи………………………………………….………….4
2. Постановка задачі……………………………………………………………...5
3. Теоретичні положення…………………………………………………………6

3.1 Шифр Цезаря……………………………………………………………….6

3.2 Шифр ROT1………........…………………………………………………...6

3.3 Шифр Атбаш….……………………………………………………………6

1. UML-діаграма класів…………………………………………………………..8
2. Вихідний код програми………………………………………………………..9
3. Приклади роботи програми…………………………………………………..11

Висновки………………………………………………………………………...12

ВСТУП

**Об'єктно-орієнтоване програмування** (ООП) — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Це одна з моделей програмування.

Кожен об’єкт в цій моделі є незалежним, і він здатний отримувати, обробляти дані та відправляти ці дані іншим об’єктам. В ООП використано моделі успадкування, модульності, поліморфізму та інкапсуляції.

Основним поняттям ООП є об'єкт. Об'єкт можна визначити як певну сукупність даних (характеристик об'єкта) та методів роботи з ними. Для класифікації об'єктів у ООП використовують класи. Клас служить зразком для створення об'єкту, тобто об'єкт є нічим іншим, ніж копією класу.

Кожен об'єкт має процедури і функції (те, що він уміє виконувати), які служать для роботи з даними об'єкта. Ці процедури і функції називаються методами.

Існування ООП можливе завдяки трьом основним парадигмам, на яких базується саме ООП:

* Інкапсуляція
* Успадкування
* Поліморфізм

1. ЦІЛЬ ЛАБОРАТОРНОЇ РОБОТИ

**Ціль роботи** – вивчити основні концепції об’єктно-орієнтованого програмування. Вивчити особливості використання класів і об’єктів, а також особливості використання конструкторів.

1. ПОСТАНОВКА ЗАДАЧІ

Створити клас “Криптографія” з закритим елементом *t* - який являє собою деякий текст. Передбачити у класі:

* конструктор ініціалізації для вводу тексту з клавіатури (задати деякий текст за замовчуванням).

Клас повинен містити методи для шифрування і розшифрування тексту за трьома алгоритмами і більше. Зімітувати атаку крипто аналітика (є зашифрований текст, робиться припущення щодо методу, яким він зашифрований, підбирається ключ і розшифровується текст). Порівняти час, витрачений на підбір ключів різних методів.

Визначити деструктор класу.

1. ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Шифрування - спосіб приховування, що застосовується для зберігання інформації в надійних джерелах або передачі її по незахищеним каналам зв’язку.

Для виконання даної лабораторної роботи були використані 3 найпростіші алгоритми шифрування текстового повідомлення: шифр Цезаря, шифр ROT1 та шифр Атбаш.

* 1. Шифр Цезаря

**Шифр Цезаря** — симетричний алгоритм шифрування підстановками. Використовувався римським імператором Юлієм Цезарем для приватного листування.

Принцип дії полягає в тому, щоб циклічно зсунути алфавіт, а ключ — це кількість літер, на які робиться зсув.

* 1. Шифр ROT1

**Шифр ROT1** — простий буквений підстановочний шифр, який заміняє літеру 1 буквою в абетці після неї. ROT-1 є різновидом шифру Цезаря, розробленого в Стародавньому Римі.

* 1. Шифр Атбаш

**Атба́ш** — простий шифр підстановки для івриту. Даним алгоритмом зашифровано частину біблійних текстів.

Правило шифрування полягає у заміні i-тої літери абетки літерою з номером n − i + 1, де n — кількість літер в алфавіті. Таким чином, перша буква алфавіту замінюється останньою, друга — передостанньою і так далі. Приклад для латинського алфавіту виглядає так:

**Вхідний текст:** abcdefghijklmnopqrstuvwxyz

**Зашифрований текст:** ZYXWVUTSRQPONMLKJIHGFEDCBA

Застосування алгоритму до українського алфавіту:

**Вхідний текст:** а б в г ґ 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

**Зашифрований текст** я ю ь щ ш ч ц х ф у т с р п о н м л к й ї і и з ж є е д ґ г в б а

Походження слова «Атбаш» пояснюється принципом заміни літер. Слово אתבש складено з літер «Алеф», «Тав», «Бет» та «Шин», тобто першої та останньої, другої та передостанньої літер давньосемітського алфавіту.

1. UML-ДІАГРАМА КЛАСІВ

|  |
| --- |
| Crypto |
| -t : string  -alphabet : array |
| +encryptRot1( ) : string  +decryptRot1(encrMsg : string) : string  +encryptCeasar(n : number) : string  +decryptCeasar(encrMsg : string) : string  +encryptAtbash( ) : string  +decryptAtbash(encrMsg : string) : string |

1. ВИХІДНИЙ КОД ПРОГРАМИ
2. Файл crypto.js

'use strict';

const rl = require('readline-sync');

const { PerformanceObserver, performance } = require('perf\_hooks');

class Crypto {

constructor() {

let \_t = rl.question('Type your message (or press Enter to default): ') || 'lorem ipsum dolor sit lorem ipsum dolor sit lorem ipsum dolor sit lorem ipsum dolor sit lorem ipsum dolor sit lorem ipsum dolor sit';

let 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'];

this.encryptRot1 = () => {

return this.encryptCaesar(1);

};

this.decryptRot1 = (encrMsg) => {

return encrMsg.split(' ')

.map(word => word.split('')

.map(letter => alphabet.indexOf(letter) - 1 >= 0 ? alphabet[alphabet.indexOf(letter) - 1] : alphabet[alphabet.length + alphabet.indexOf(letter) -1]

).join('')

).join(' ');

};

this.encryptCaesar = (n = Math.floor(Math.random() \* 26)) => {

let text = \_t.split(' ');

let encryptedText = text

.map(word => word.split('')

.map(letter => alphabet[(alphabet.indexOf(letter) + n) % alphabet.length])

.join(''))

.join(' ');

return encryptedText;

};

this.decryptCaesar = (encrMsg) => {

let n = 0;

while (encrMsg != \_t) {

encrMsg = encrMsg.split(' ')

.map(word => word.split('')

.map(letter => alphabet.indexOf(letter) - 1 >= 0 ? alphabet[alphabet.indexOf(letter) - 1] : alphabet[alphabet.length + alphabet.indexOf(letter) -1]

).join('')

).join(' ');

++n;

}

return {encrMsg, n};

};

this.encryptAtbash = () => {

let reverseAlphabet = alphabet.join('').split('').reverse();

let text = \_t.split(' ');

let encryptedText = text

.map(word => word.split('')

.map(letter => reverseAlphabet[alphabet.indexOf(letter)])

.join(''))

.join(' ');

return encryptedText;

};

this.decryptAtbash =(encrMsg) => {

let reverseAlphabet = alphabet.join('').split('').reverse();

encrMsg = encrMsg.split(' ')

.map(word => word.split('')

.map(letter => alphabet[reverseAlphabet.indexOf(letter)])

.join(''))

.join(' ');

return encrMsg;

};

}

benchmarkFn(fnEnc, fnDec) {

let start = performance.now();

let enc = fnEnc();

for (let i = 0; i < 10000; i++) {

fnDec(enc);

}

let finish = performance.now();

console.log(`Average ${fnDec.name} speed is ${((finish - start) / 1000).toFixed(3)} ms`);

}

benchmark() {

console.log('BENCHMARKING...');

this.benchmarkFn(this.encryptRot1, this.decryptRot1);

this.benchmarkFn(this.encryptAtbash, this.decryptAtbash);

this.benchmarkFn(this.encryptCaesar, this.decryptCaesar);

console.log('BENCHMARK FINISHED');

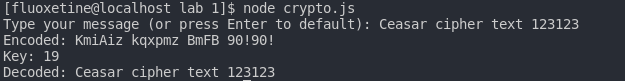
}

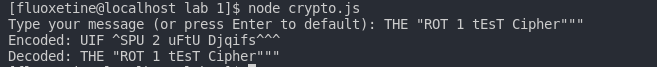
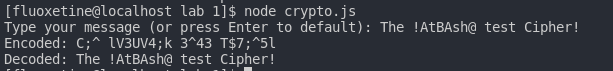
}

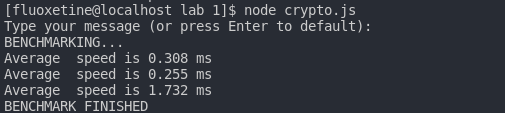
let crypto = new Crypto();

crypto.benchmark();

1. ПРИКЛАДИ РОБОТИ ПРОГРАМИ
2. Шифр Цезаря



1. Шифр ROT1
2. Шифр Атбаш
3. Benchmark



ВИСНОВКИ

У рамках даної лабораторної роботи було надано можливість ознайомитися з основними принципами побудови програми за об’єктно- орієнтованою парадигмою програмування. Для виконання цього виду роботи мною була обрана мова програмування JavaScript.

У роботі мені вдалося вдало використати інструменти ООП та виконати усі поставлені перед мною задачі.