**Форма № Н-6.01**

ЗАТВЕРДЖЕНО Наказ Міністерства освіти і науки, молоді та спорту України від 29 березня 2012 року № 384

**Черкаський національний університет імені Богдана Хмельницького**

(повне найменування вищого навчального закладу)

**Кафедра програмного забезпечення автоматизованих систем**

(повна назва кафедри)

**КУРСОВА РОБОТА**

***з дисципліни “Об‘єктно-орієнтоване програмування” НА ТЕМУ «*Навчальний додаток «Шифр Гронсвельда»*»***

|  |  |
| --- | --- |
|  | Студента (ки) 2 курсу, групи КС-181 . напряму підготовки «Програмна інженерія» спеціальності«Інженерія програмного забеспечення»  Лесечка Дениса .  (прізвище та ініціали)  Керівник Супруненко Оксана Олександрівна .  кандидат технічних наук, доцент .  (посада, вчене звання, науковій ступінь, прізвище та ініціали)  Національна шкала:  Кількість балів: Оцінка: ECTS |
| Члени комісії | **(підпис) (прізвище та ініціали)**    **(підпис) (прізвище та ініціали)**    **(підпис) (прізвище та ініціали)** |

м. Черкаси – 2020 рік

Зміст

[Вступ 3](#_Toc41516629)

[Розділ 1. Огляд алгоритмів що використовується для створення програми 4](#_Toc41516630)

[1.1 Реалізація програмного продукту через picturebox 4](#_Toc41516631)

[Висновок до першого розділу 5](#_Toc41516632)

[Розділ 2. Алгоритми для реалізації задачі 6](#_Toc41516633)

[2.1. Блок схема алгоритму 6](#_Toc41516634)

[2.2 Структурна схема програмного продукту 6](#_Toc41516635)

[2.3 Алгоритм шифрування 7](#_Toc41516636)

[Висновок до другого розділу 9](#_Toc41516637)

[Розділ 3. Реалізація та створення програмного продукту 10](#_Toc41516638)

[3.1 Реалізація ієрархії класів 10](#_Toc41516639)

[3.2 Реалізація принципів ООП 11](#_Toc41516640)

[3.3 Тестування програми 11](#_Toc41516641)

[Висновки 14](#_Toc41516642)

[Список використаних джерел 15](#_Toc41516643)

[ДОДАТОК Б. Блок схема алгоритму шифрування 16](#_Toc41516644)

[ДОДАТОК В Блок сеха алгоритму підстановки ключа 18](#_Toc41516645)

[ДОДАТОК Г Блок схема алгоритму 19](#_Toc41516646)

# Вступ

На сьогоднішній день Шифр Гронсвельда не актуальний. Тому що зараз існують більш продвинуті види шифрування.

Навчальний додаток «Шифр Гронсвельда» - це програма яка навчає користувача використовувати шифр Гронсвельда. Також для користувача схематично відображається принцип, за яким проводиться шифрування та дешифрування.

# Розділ 1. Огляд алгоритмів що використовується для створення програми

Для простоти відображення процесу шифрування будемо використовувати Windows Form. Будемо використовувати picturebox для схематичного відображення шифрування. Через textbox будемо вводити текст і ключ для шифрування.

# Реалізація програмного продукту через picturebox

Реалізація через picturebox досить добре підходить. Ідея полягає в тому що створимо масив з алфавітом і проведемо його нумерацію. І виведемо його на головний екран щоб користувач розумів як проходить шифрування і дешифрування. Під текстом що вів користувач буде відображатися ключ згідно з яким буде проводитися шифрування і дешифрування.

# Висновок до першого розділу

У першому розділі розглянули алгоритм реалізації навчального додатку «Шифр Гронсвельда». В якості головного інструменту будемо використовувати Windows Form та мову програмування C#.

# Розділ 2. Алгоритми для реалізації задачі

Для створення навчального додатку «Шифр Гронсвельда» потрібно виконати такі дії:

Створити масив Українського алфавіту;

Відображення алфавіту та порядкового номеру букв;

Відображення тексту що вів користувач і ключа під ним;

Відображення вказівки як проходить шифрування і дешифрування;

Перевірка чи правильні дані надійшли від користувача;

Після натиску кнопки «Зашифрувати» запуск алгоритму шифрування.

# 2.1. Блок схема алгоритму

Блок-схема навчального додатку «Шифр Гронсвельда» наведена у додатку.

# 2.2 Структурна схема програмного продукту

Завданням курсової роботи є написання навчального додатку «Шифр Гронсвельда». Основною вимогою до цієї програми є начити користувача використовувати шифр Гронсфельда. Можливість ввести текст і ключ. В лівій частині форми розміщений picturebox, в якому зображено масив афлавіту, процес шифрування і дешифрування. В правій частині зверху поля для введеня тексту і ключа для шифруваня, а в нижче поля для дешифрування.

# 2.3 Алгоритм шифрування

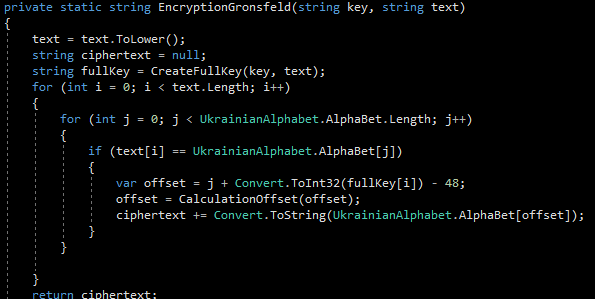
Після натискання на кнопку «Зашифрувати» програма запускає алгоритм шифрування (рис 2.1) 

Рисунок 2.1– алгоритм шифрування

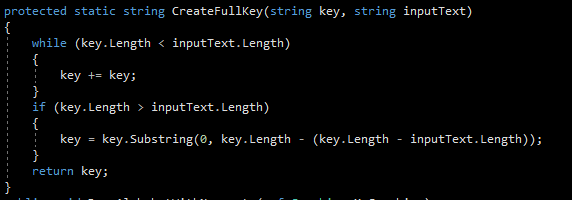
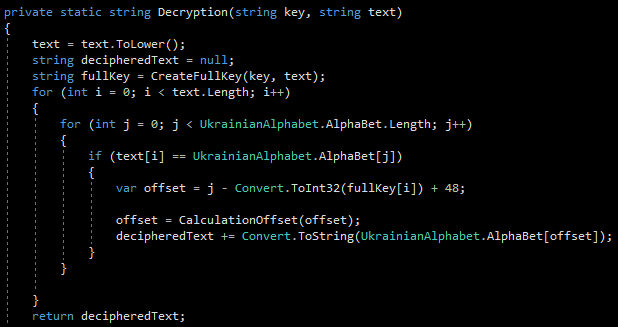
В тілі алгоритму шифрування запускається метод метод підназвою CreateFullKey(рис. 2.2) в якому дублюэться ключ доки не закінчаться символи тексту який ввели для шифрування.

Рисунок 2.2 – метод підстановки ключа

Після натискання на кнопку «Дешифрувати» програма запускає алгоритм дешифрування (рис 2.3)

Рисунок 2.3 – алгоритм дешифрування

# Висновок до другого розділу

В другому розділу була визначена структурана схема програми, побудована блок-схема основного алгоритмну так його сутність. Сутність алгоритму показана в скріншотах методів.

# Розділ 3. Реалізація та створення програмного продукту

Під час реалізації створюється всі основні фрагменти, кнопки керування та навігації в них.

# 3.1 Реалізація ієрархії класів

Під час реалізації програми були створені такі класи (рис.3.1)

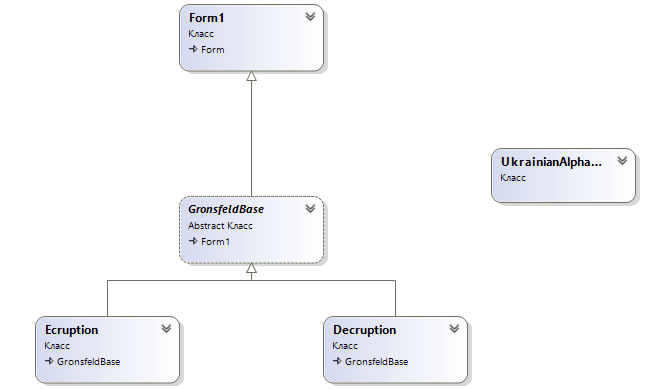
* GronsfeldBase абстрактний клас, який э батьком для двох класів і реалізує для них 3 метода СreateFullKey, DrawAlphabetWithNumerate, CalculationOffset.
* Ecruption клас, який успадковується від класу GronsfeldBase та виконує метод шифрування
* Decruption клас, який успадковується від класу GronsfeldBase та виконує метод дешифрування
* UkrainianAlphabet клас в якому створений char массив в якому записані українські символи які потрібні для роботи програми.

Рисунок 3.1. Діаграма класів

# 3.2 Реалізація принципів ООП

Наслідування – це механізм утворення нових [класів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) на основі використання вже існуючих. При цьому властивості та функціональність батьківського класу переходять до класу нащадка (дочірнього). В програмі два класа такі як Ecruption, Decruption, які наслідують батьківський класс GronsfeldBase (рис. 3.2)

Рисунок 3.2 Наслідування класу GronsfeldBase

Інкапсуляцію використовуємо для зручного використання алфавіту без змоги внести зміни так як вони не потрібні (рис.3.3)

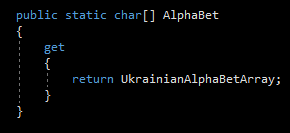


Рисунок 3.3 Приклад ікапсуляції в класі GronsfeldBase

Ecruption

GronsfeldBase

Decruption

В роботі використовуємо поліморфізм підтипів – це коли під одним ім’ям може використовуватися декілька типів від одного базаового. В цій програмі він зустрічається в (додаток А).

# 3.3 Тестування програми

Поділимо тестування на етапи:

–Шифруваня

–Дешифрування

Для шифрування потрібно ввести текст в поле “Введіть текст який потрібно зашифрувати (тільки українською мовою)” і ключ в поле “Введіть цифровий ключ” натиснути кнопку «Зашифрувати» (рис.3.4-3.5)

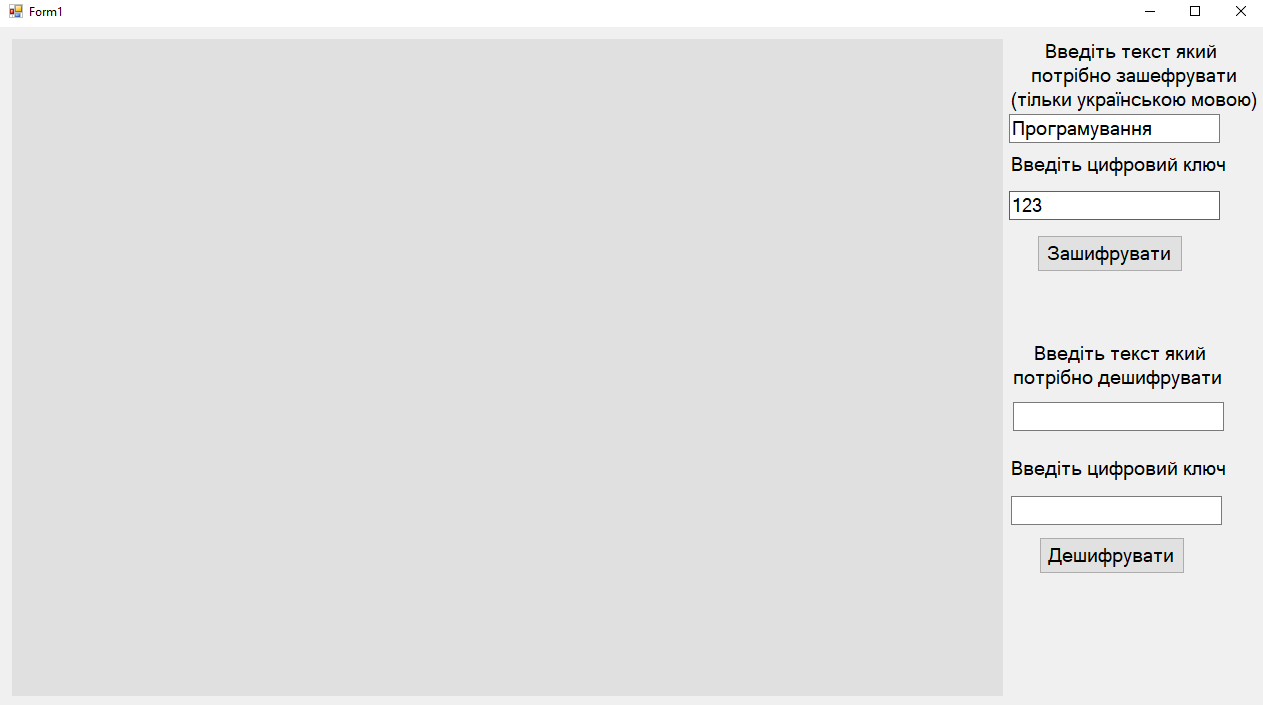


Рисунок 3.4 – Запуск програми і введення тексту

Рисунок 3.5 - Приклад шифрування

Для дешифрування потрібно вести зашифрований текст в поле “Введіть текст який потрібно дешифрувати” і ключ в поле “Введіть цифровий ключ” натиснути кнопку дешифрувати.(рис. 3.6)

Рисунок 3.6. Приклад дешифрування

# Висновки

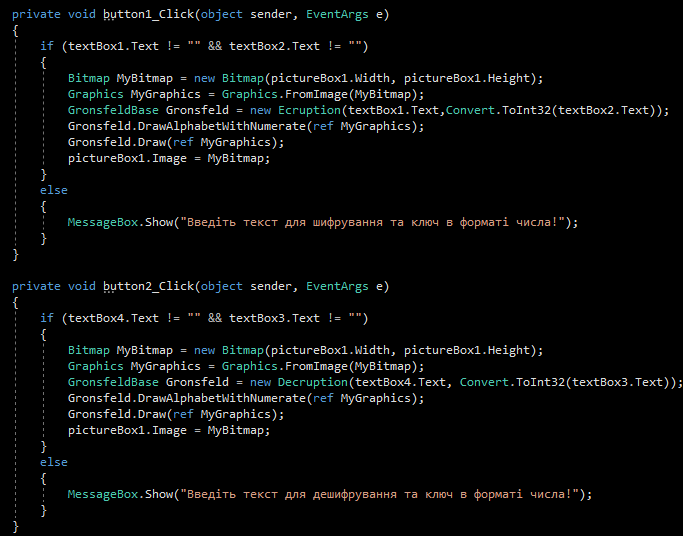
При написанні курсової роботи були закріплені навички програмування на мові C# та використання Windows Form.

Під час виконання курсової роботи був створений зручний інтерфейс в якому з легкістю зрозуміє, що де вводити і як працює шифрування.

При подальшому розвитку програми можна замінити UI на інший, додати схожі методи шифрування, зробити інтерфейс ще краще.

# Список використаних джерел

1. Успадкування <https://uk.wikipedia.org/wiki/Успадкування_(програмування)> (Перевірено: 27.05.2020)
2. Поліморфізм <https://uk.wikipedia.org/wiki/Поліморфізм_(програмування)> (Перевірено: 27.05.2020)
3. Інкапсуляція <https://uk.wikipedia.org/wiki/Інкапсуляція_(програмування)> (Перевірено: 27.05.2020)
4. Windows Forms <https://docs.microsoft.com/ru-ru/visualstudio/designers/windows-forms-designer-overview?view=vs-2019> (Перевірено: 27.05.2020)

ДОДАТОК А. Реалізація Поліморфізму підтипів

# ДОДАТОК Б. Блок схеми алгоритму шифрування

Початок

key, text

fullKey = СreateFullKey(key, text);

i=0; i< text.Leght; i++

j:(0,Ukrainian.Lenght)

text[i] == UkrainianAlphabet

offset = CalculationOffset(offset)

ciphertext += Convert.ToString(UkrainianAlphabet.AlphaBet[offset])

offset = j + Convert.ToInt32

return ciphertext

Кінець

Початок

key.Lenght<inputText.Lenght

key+=key

key = key.Substring(0,key.Length-(key.Lenght-inputText.Lenght))

return key

key.Lenght>inputText.Lenght

Кінець

# 

Початок

iputOffset>=UkrainianAlphabet.Alphabet.Lenght

iputOffset<0

iputOffset<0

iputOffset>=UkrainianAlphabet.Alphabet.Lenght

iputOffset = iputOffset - UkrainianAlphabet.Alphabet.Lenght

iputOffset = iputOffset + UkrainianAlphabet.Alphabet.Lenght

return ipuOffset

Кінець