Міністерство освіти і науки України

Київський державний коледж туризму та готельного господарства

До захисту допущений

Голова ЦК «Програмування, інформатики та комп’ютерної техніки»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(О.І. Артеменко)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

**КУРСОВИЙ ПРОЕКТ**

з дисципліни *«Інструментальні засоби візуального програмування»*

на тему: *« Розробка програми “Шифрування тексту” »*

**Пояснювальна записка**

**КП 121.04 ПЗ**

Студента 3 курсу 31-ІПЗ групи

напряму підготовки 12

спеціальності 121

Д.А.Глибовця

Керівник курсового проекту

викладач О.П. Стародуб

Оцінка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Члени комісії:** |  | *Артеменко О.І.* |
| (Підпис) | (Прізвище та ініціали) |
|  | *Стародуб О.П.* |
| (Підпис) | (Прізвище та ініціали) |
|  | *Тетерева М.Я.* |
| (Підпис) | (Прізвище та ініціали) |

Дата захисту “\_\_\_”\_\_\_\_\_\_\_\_\_\_*2020 р.*

Київ 2020

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

|  |
| --- |
| **КИЇВСЬКИЙ ДЕРЖАВНИЙ КОЛЕДЖ**  **ТУРИЗМУ ТА ГОТЕЛЬНОГО ГОСПОДАРСТВА** |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Спеціальність: | | | ***121 "Інженерія програмного забезпечення"*** | | |
| Дисципліна: | | | ***Інструментальні засоби візуального програмування*** | | |
| Курс | ***ІІІ*** | Група | ***31-ІПЗ*** | Семестр | ***6*** |

**ЗАВДАННЯ  
на курсовий проект студента**

|  |
| --- |
|  |

(прізвище, ім’я, по батькові)

|  |  |  |
| --- | --- | --- |
| 1. Тема проекту: |  | |
|  | | |
| 2. Термін захисту розробленого проекту: | | ***23 квітня 2020 р.*** |
| 3. Вимоги до проекту: |  | |
|  | | |
|  | | |
|  | | |
|  | | |
| 4. Зміст пояснювальної записки (перелік питань, які підлягають розробці) | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
| 6. Дата видачі завдання: | | ***18 лютого 2020 р.*** |

**Календарний план**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Назва етапів курсового проекту | Термін виконання | Примітки |
| 1 | Обрання та затвердження теми курсового проекту студентами | 18.02.20 |  |
| 2 | Складання плану курсового проекту | 20.02.20 |  |
| 3 | Підбір літератури по темі курсового проекту | 25.02.20 |  |
| 4 | Загальні вимоги до оформлення курсового проекту | 27.02.20 |  |
| 5 | Перевірка планів курсових проектів | 03.03.20 |  |
| 6 | Доопрацювання планів робіт згідно з зауваженнями | 05.03.20 |  |
| 7 | Порядок подання постановки власної задачі та алгоритму її розв’язання | 10.03.20 |  |
| 8 | Порядок подання програмної реалізації задачі курсового проекту | 17.03.20 |  |
| 9 | Доопрацювання програм курсового проекту згідно з зауваженнями | 19.03.20 |  |
| 10 | Перегляд змістовних частин курсового проекту | 26.03.20 |  |
| 11 | Перегляд програм з курсових проектів | 02.04.20 |  |
| 12 | Висновки і пропозиції за темами | 09.04.20 |  |
| 13 | Оформлення курсового проекту згідно вимог | 16.04.20 |  |
| 14 | Доопрацювання оформлення курсового проекту згідно із зауваженнями. | 21.04.20 |  |

|  |  |  |
| --- | --- | --- |
| Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  18 лютого 2020 р. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Прізвище, ім’я по батькові)  Стародуб Олександр Петрович  (Прізвище, ім’я по батькові) |

**ЗМІСТ**

[ВСТУП 3](#_Toc39523763)

[**1.** **ТЕОРЕТИЧНА ЧАСТИНА** 4](#_Toc39523764)

[1.1 Аналіз проблеми, методів та засобів її вирішення 4](#_Toc39523765)

[1.2 Шифр Цезаря 4](#_Toc39523766)

[1.3 . Мова для розробки програми 6](#_Toc39523767)

[1.4. Порівняння аналізу програм аналогічного функціювання 7](#_Toc39523768)

[1.5.Аналіз засобів вирішення задачі 8](#_Toc39523769)

[Висновки до першого розділу 9](#_Toc39523770)

[**2.** **Практична частина** 11](#_Toc39523771)

[2.1. Розробка загального алгоритму роботи програми 11](#_Toc39523772)

[2.2. Розробка інтерфейсної частини додатку 12](#_Toc39523773)

[2.3. Розробка програмного коду додатку 15](#_Toc39523774)

[2.4. Опис роботи з додатком 19](#_Toc39523775)

[Висновки до другого розділу 27](#_Toc39523776)

[ВИСНОВКИ 27](#_Toc39523777)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 29](#_Toc39523778)

# ВСТУП

*Актуальність теми:* на сьогоднішній день шифрування даних це одна з невід'ємних складових мережі інтернет. Захист персональних даних це те, на що витрачається значна частина бюджету при розробці будь-якого програмного додатку орієнтованого на масового користувача. Адже ніхто не хоче, щоб інформація про кредитну картку, якою ми оплачуємо послуги або продукцію в інтернет-магазинах потрапила в руки злодіїв. Саме тому зараз розробляються все складніші і складніші алгоритми шифрування даних.

*Головною метою курсового проекту:* є розробка програми для шифрування тексту.

*Задачі курсового проекту:* розробити програмний комплекс, що:

* **виконує шифрування тексту;**
* **дозволяє обрати мову шифрування та зсув за алфавітом;**
* **надає можливість запису тексту в файл;**
* **має інтуїтивно зрозумілий інтерфейс;**
* **має меню-інструкцій, щодо користування програмою;**
* **надає можливість очищення даних.**

Для реалізації поставлених задач потрібна була мова програмування, що може звертатися до пам’яті комп’ютера напряму, але при цьому мала зрозумілий синтаксис, можливість використання графічного інтерфейсу та швидкість роботи програмного додатку. Саме тому була використана мова С++ та Microsoft Visual Studio 2017.

1. **ТЕОРЕТИЧНА ЧАСТИНА**

## 1.1 Аналіз проблеми, методів та засобів її вирішення

Темою даного курсового проекту є розробка програми “Шифрування тексту”. В якості шифру було обрано шифр ROT, або по іншому Шифр Цезаря, це означає що програма має отримувати на вхід набір символів, зміщувати кожен на певну кількість літер за алфавітом, а потім зашифрований текст записувати в файл. Параметрами що задає користувач можуть бути:

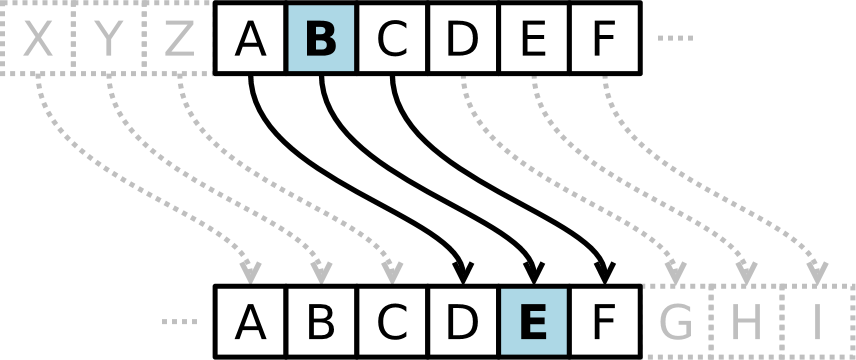
* Кількість символів, на яку відбудеться зсув;
* Мову тексту.

Результатом роботи програми є файл з зашифрованим текстом.

Так, програмне забезпечення, яке планується розробити, буде мати наступний формат. Буде створена форма на якій можна буде вказати файл в якому знаходиться оригінальний текст, файл в який потрібно записати текст, зсув тексту за алфавітом та мову тексту.

1.2 Шифр Цезаря

**Шифр Цезаря або шифр зсуву** — симетричний моноалфавітний алгоритм шифрування, в якому кожна буква відкритого тексту заміняється на ту, що віддалена від неї в алфавіті на сталу кількість позицій. Римський імператор Юлій Цезар використовував для приватного листування шифр зсуву з ключем 3 — замість літери A підставляв D, замість B — E і так далі.



Як і будь-який моноалфавітний шифр, Цезарів вразливий до частотного криптоаналізу. Навіть легше зламати його простим перебором, бо кількість можливих ключів зовсім невелика — для українського алфавіту 32 варіанти. Метод Цезаря не надає майже ніякого криптографічного захисту, але він ліг в основу дещо складніших алгоритмів, наприклад шифру Віженера. Варіант шифру зсуву ROT13 використовується в англомовному сегменті інтернету для приховування спойлерів, розгадок головоломок тощо.

**Принцип дії**

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

Якщо зіставити кожному символу алфавіту його порядковий номер (нумеруючи з 0), то шифрування і дешифрування можна виразити формулами:

**y=(x+k) mod n**

**x=(y-k) mod n**

де

**x** — порядковий номер символу відкритого тексту,

**y** — порядковий номер символу шифрованого тексту,

**n** — потужність алфавіту,

**k** — ключ.

**Приклад**

Припустимо, що, використовуючи шифр Цезаря, з ключем, який дорівнює 3, необхідно зашифрувати словосполучення «ШИФР ЦЕЗАРЯ».

Для цього зрушимо алфавіт так, щоб він починався з четвертої букви (Г). Отже, беручи вихідний алфавіт

АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ,

і зміщуючи всі літери вліво на 3, отримуємо відповідність:  
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  
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 А Б В,  
де Г=А, Д=В, Е=Г, і т. д.

Використовуючи цю схему, відкритий текст «ШИФР ЦЕЗАРЯ» перетворюється на «ЮЙЧУ ЩЗЇГУВ». Для того, щоб одержувач повідомлення міг відновити вихідний текст, необхідно повідомити йому, що ключ — 3.

## . Мова для розробки програми

Розглянемо більш детально переваги використання *С++* для створення програми.

C++ - мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в *AT&T Bell Laboratories* (Мюррей-Хілл, Нью-Джерсі) 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на *C++* у 1983 р. Базується на мові *С*. Вперше описана стандартом *ISO/IEC* *14882:1998*, найбільш актуальним же є стандарт *ISO/IEC* *14882:2014*.

Мова *С++* багато в чому є надмножиною *С*. Нові можливості *С++* включають оголошення у вигляді виразів, перетворення типів у вигляді функцій, оператори *new* і *delete*, тип *bool*, посилання, розширене поняття константності та змінності, функції, що підставляються, аргументи за умовчанням, перевизначення, простори імен, класи (включаючи і всі пов'язані з класами можливості, такі як успадкування, функції-члени (методи), віртуальні функції, абстрактні класи і конструктори), перевизначення операторів, шаблони, оператор *::*, обробку винятків, динамічну ідентифікацію і багато що інше. *С++* є також мовою строгого типування і накладає більше вимагань щодо дотримання типів, порівняно з *С*.

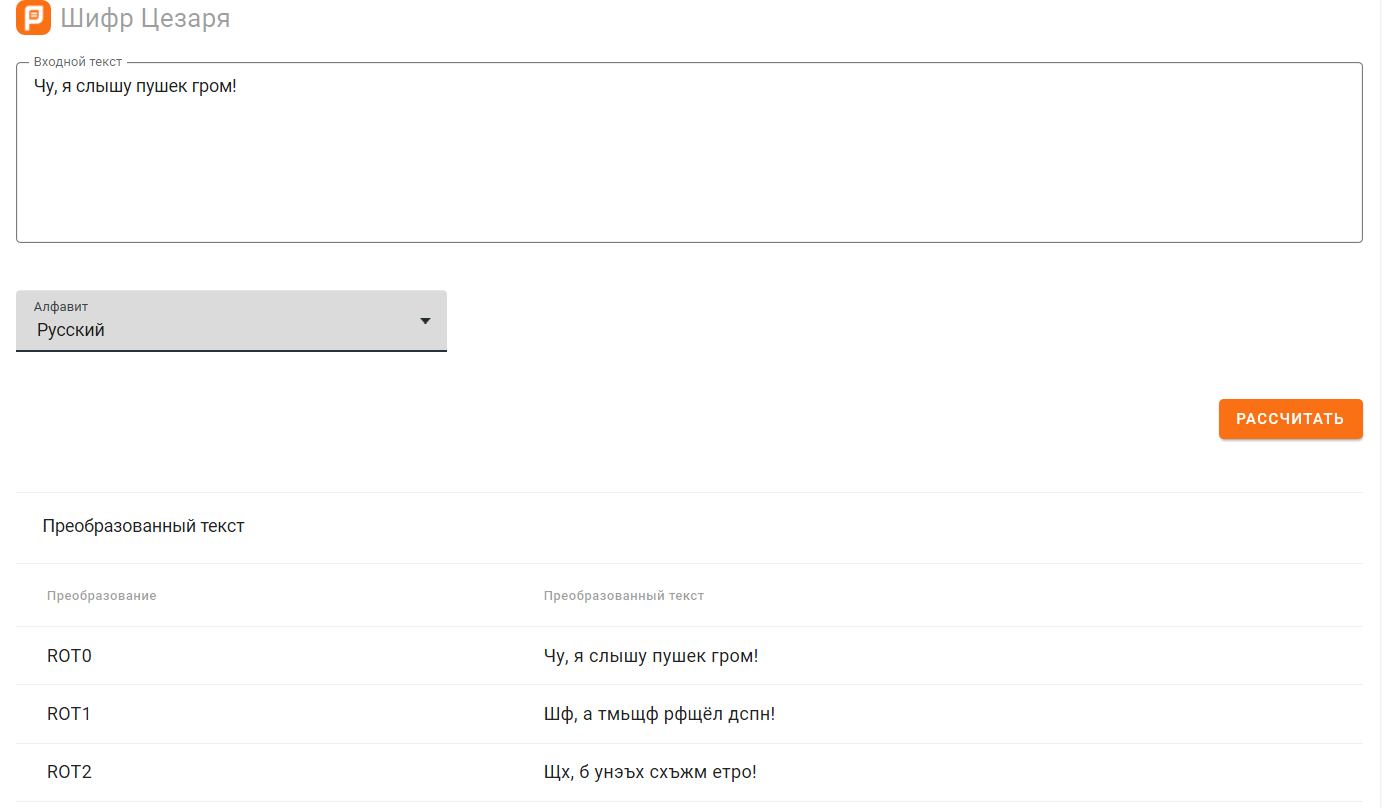
Більшість розробників мають бажання використовувати сучасні мови програмування, які є достатньо потужними та гнучкими, однією з них є *С++*, що забезпечує доступ до всіх функціональних можливостей системи, дозволяє взаємодіяти з існуючими програмами і легко працювати з виникаючими *Web*-стандартами.

Ще у 1990-х роках *С++* стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор.

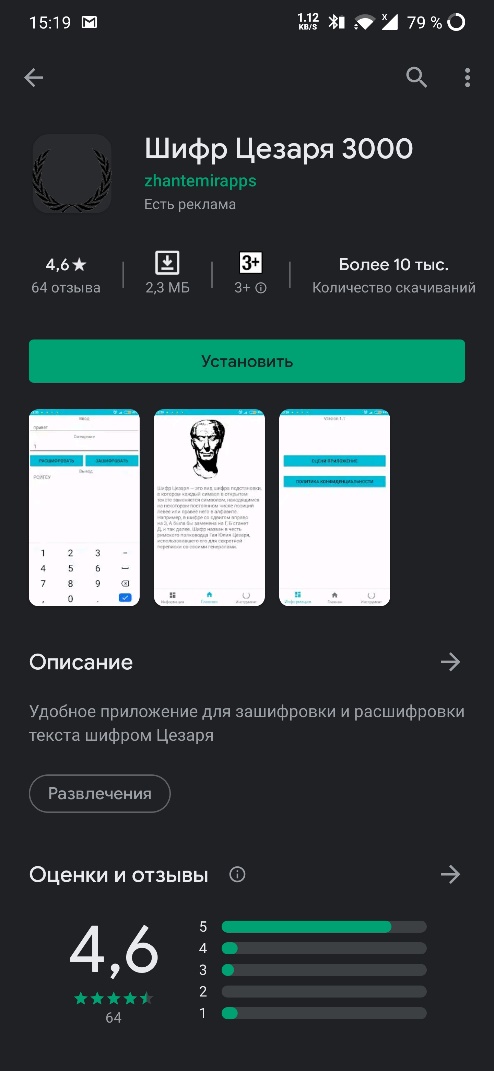
## 1.4. Порівняння аналізу програм аналогічного функціювання

До програм, що виконують шифрування на методом ROT відносяться:

1. Онлайн калькулятори в мережі Інтернет, що дозволяють отримати текст зі зміщенням на будь яку кількість символів. Приклад такої програми зображено на Р ис. 1.1*.* Недоліком таких програм є вимога безпосереднього доступу до мережі і неможливість збереження отриманих результатів у файл;
2. Мобільні шифратори, як наприклад Шифр Цезаря 3000 Рис 1.2.



*Рис. 1.1. Онлайн калькулятор для шифру Цезаря*



*Рис. 1.2. Мобільний додаток для шифру цезаря*

## 1.5.Аналіз засобів вирішення задачі

Проектована програма повинна шифрувати текст в залежності від вказаного зсуву та мови, записувати текст в окремий файл.

Отже, в наведеному списку знаходяться основні компоненти, які будуть додані в програму. Дані вікна будуть мати все оформлення для шифрування даних, таке як:

* Компонент для вибору файлу з оригінальним текстом;
* Компонент для вибору файлу, в який буде записано зашированний текст;
* Компонент для вибору мови тексту, який буде зашифровано;
* Поле в якому вказано зсув за алфавітом;
* Головне меню програми.

Кожен з даних компонентів буде мати свою функціональність, тож можна проаналізувати кожний компонент та розписати за що він відповідатиме.

Вибір файлу з оригінальним текстом:

* + Для вибору файлу, в який користувач попередньо записує текст, який потрібно зашифрувати;

Вибір файлу, в який буде записано зашифрований файл:

* + Для вибору файлу, в який буде записно зашифрований файл;

Вибір мови тексту:

* + Для забезпечення правильного шифрування даних;

Поле в якому вказано зсув:

* + У цьому полі буде вказано на скільки знаків потрібно змістити положення кожної літери, відносно номера в алфавіті;

Головне меню програми:

* + Можливість очищення форми;
  + Довідка.

Отже, визначено основні методи та події, які буде потрібно сформувати та розробити у проектованому проекті для вирішення поставлених задач.

Безліч проблемних питань вирішиться за умови вибору для створення проекту мови С++, зокрема, інтегрованого середовища розробки Microsoft Visual Studio.

## Висновки до першого розділу

При виконанні даного розділу курсового проекту було проаналізовано поставлене завдання. На основі первинного аналізу визначено подальшу розробку програми.

Виконано порівняльний аналіз програм аналогічного призначення, визначено основні виконувані та найбільш необхідні функції.

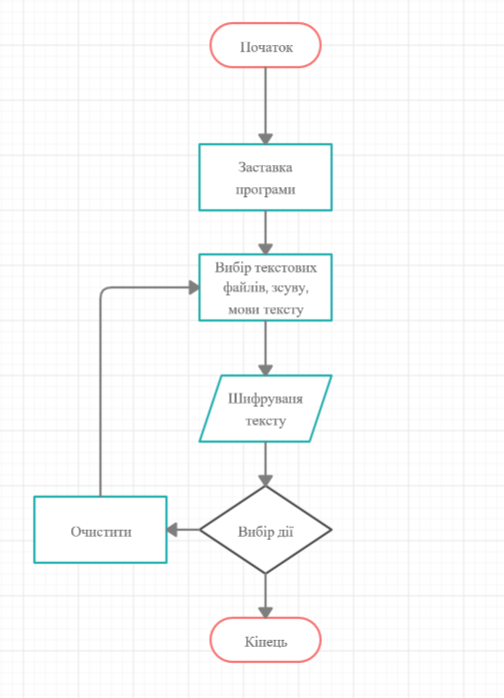
Визначено схему створюваного програмного забезпечення − організацію основних компонентів програми та основні події, які будуть прив’язані до основних компонентів.

Результатом проведеного аналізу є розробка технічного завдання – перелік інтерфейсних особливостей програми, її основного та додаткового функціоналів, що підлягають розробці.

1. **Практична частина**

2.1. Розробка загального алгоритму роботи програми

На даному етапі курсового проекту чітко сформовано задачі, які повинно виконувати розроблене програмне забезпечення. Розроблено методи вирішення поставлених задач та вигляд інтерфейсної частини. Також, отримано уявлення про алгоритм роботи програми. Реалізована схема загального алгоритму роботи програми, що зображена на *Рис. 2.1.*



*Рис. 2.1. Схема загального алгоритму роботи програми*

Опис загального алгоритму роботи програми наведено нижче.

Початок програми включає в себе безпосередньо запуск додатку. При запуску програми з’являється заставка, після неї з’являється форма, де користувач зможе обрати дію та розміри матриць. Після вибору потрібних налаштувань користувач може приступити до роботи. Спочатку потрібно вказати шлях до файлу з текстом та до файлу в який буде записано зашифрований текст.

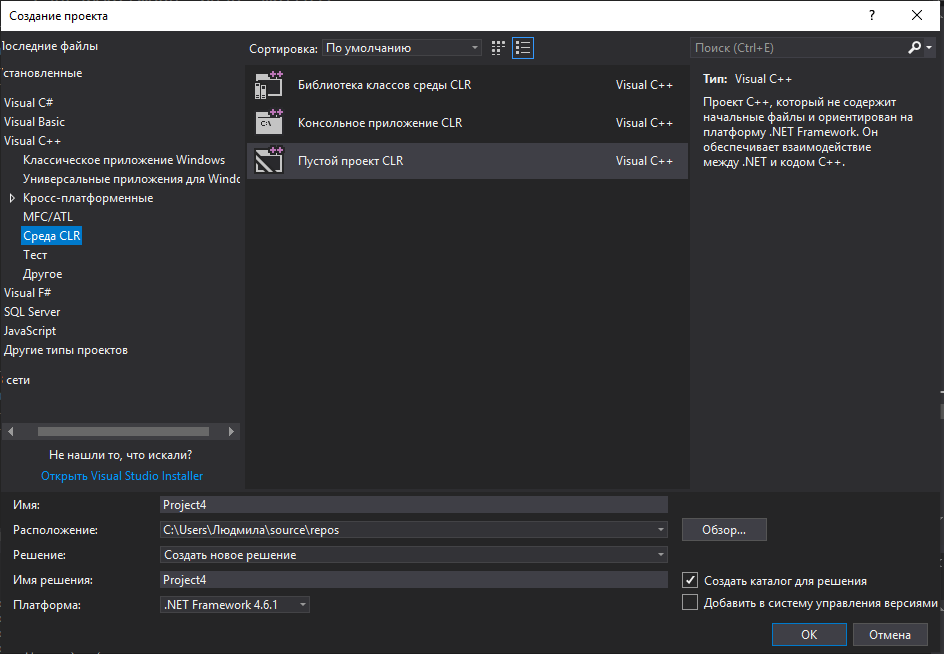
Після того як було обрано файли користувач повинен обрати мову оригінального тексту. На вибір надається англійська, російська та українська мови. Потім користувачу стане доступна можливість ввести зсув. Зсув також обмежений кількістю літер в алфавіті вказаної мови. Якщо буде вказано число, більше за кількість літер, воно буде зменшене до максимально можливого для коректної роботи програми.

Якщо всі елементи форми заповнені вірно при натисканні на кнопку «Зашифрувати» текст, що знаходиться в першому файлі буде зашифровано згідно вказаного зсуву та записано в другий файл. При цьому все, що знаходилось в другому файлі до запису буде втрачено. Якщо користувач захоче повністю очистити форму він може викликати контекстне меню натисканням правої клавіши миші або відповідною кнопкою в головному меню форми. Для виходу з форми користувачу потрібно натиснути відповідну клавішу на формі.

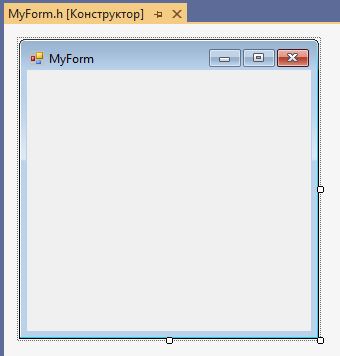
2.2. Розробка інтерфейсної частини додатку

На даному етапі розробки курсового проекту відбувається перехід до безпосередньої реалізації спроектованого програмного продукту. На даний момент вирішені всі організаційні та технічні питання з рахунку розроблюваного програмного забезпечення. Чітко сформовано задачі, які повинно виконувати програмне забезпечення, вирішені питання по їх реалізації. Так, притримуючись технічного завдання та вище оговорених положень можна переходити до розробки інтерфейсу додатку.

Для початку, створено новий проект в *MS Visual Studio*, в якому буде проводитися розробка нового програмного забезпечення. Етап створення нового проекту наведено на *Рис*. 2.2.

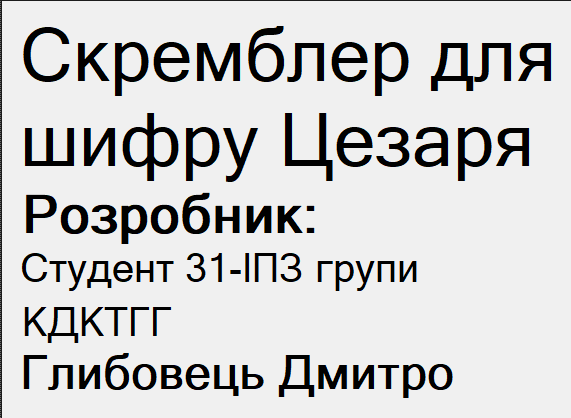


*Рис. 2.2. Створення нового проекту MS Visual Studio*

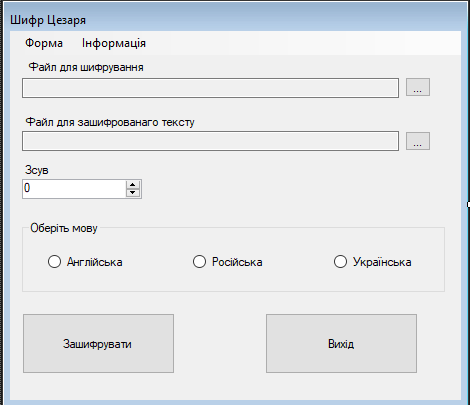
В результаті створення нового проекту майстром з’явиться пуста форма, з якої і починається розробка потрібного програмного забезпечення. Початковий вигляд проекту наведено на *Рис*. *2.3.*

*Рис. 2.3. Пустий проект*

Можна переходити до розробки програмного забезпечення, а саме, створення інтерфейсу додатку. Для цього потрібно створити нову форму, на яких розмістити елементи інтерфейсу. Як вже зазначалося, розробка інтерфейсу починається з аналізу розробленого технічного завдання та оговореної функціональності додатку.



*Рис. 2.4. Заставка програмного проекту*



*Рис. 2.5. Вікно шифратора*

2.3. Розробка програмного коду додатку

На даному етапі виконання курсового проекту створено інтерфейсну частину додатку і лишилося розробити програмний код розроблюваного програмного забезпечення.

Лістинг програми буде наведено в додатку. Код алгоритму шифрування:

void CaesarOut(int k, string r, string w)

{

ifstream input(r, std::ifstream::in | std::ifstream::binary);

ofstream output(w, std::ifstream::out | std::ifstream::binary);

char buff;

int iter;

char RU[] =

{ 'А','Б','В','Г','Д','Е','Ё','Ж','З','И',

'Й','К','Л','М','Н','О','П','Р','С',

'Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ',

'Ы','Ь','Э','Ю','Я' };

char ru[] =

{ 'а','б','в','г','д','е','ё','ж','з','и',

'й','к','л','м','н','о','п','р','с',

'т','у','ф','x','ц','ч','ш','щ','ъ',

'ы','ь','э','ю','я' };

char UA[] =

{ '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' };

char ua[]

{ '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' };

char ENG[] =

{ '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' };

char eng[] =

{ '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' };

while (input.eof() == false) {

buff = input.get();

if (buff == ' ')

output << ' ';

if (buff == '\n')

output << '\n';

if (radioButton1->Checked == true)

{

for (int i = 0; i < 26; i++)

{

if (buff == ENG[i])

{

if (i + k >= 26)

{

buff = ENG[i - (26 - k)];

output << buff;

break;

}

else

{

buff = ENG[i + k];

output << buff;

break;

}

}

if (buff == eng[i])

{

if (i + k >= 26)

{

buff = eng[i - (26 - k)];

output << buff;

break;

}

else

{

buff = eng[i + k];

output << buff;

break;

}

}

}

}

if (radioButton2->Checked == true)

{

setlocale(LC\_ALL, "Russian");

for (int i = 0; i < 33; i++)

{

if (buff == RU[i])

{

if (i + k >= 33)

{

buff = RU[i - (33 - k)];

output << buff;

break;

}

else

{

buff = RU[i + k];

output << buff;

break;

}

}

if (buff == ru[i])

{

if (i + k >= 33)

{

buff = ru[i - (33 - k)];

output << buff;

break;

}

else

{

buff = ru[i + k];

output << buff;

break;

}

}

}

}

if (radioButton3->Checked == true)

{

setlocale(LC\_ALL, "Ukrainian");

for (int i = 0; i < 33; i++)

{

if (buff == UA[i])

{

if (i + k >= 33)

{

buff = UA[i - (33 - k)];

output << buff;

break;

}

else

{

buff = UA[i + k];

output << buff;

break;

}

}

if (buff == ua[i])

{

if (i + k >= 33)

{

buff = ua[i - (33 - k)];

output << buff;

break;

}

else

{

buff = ua[i + k];

output << buff;

break;

}

}

}

}

}

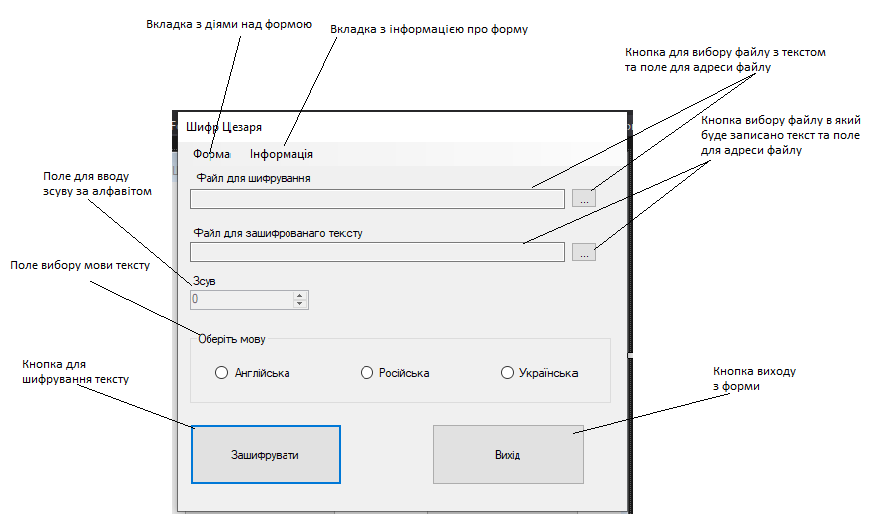
input.close();

output.close();

}

2.4. Опис роботи з додатком

Опис роботи з додатком включає в себе розробку текстової та ілюстраційної частин, які пояснюють принципи роботи з додатком та опис елементів інтерфейсу для його користувачів. При запуску додатку перш за все викликається вікно-заставка. Потім відкривається вікно шифратора в якому користувач може вказати файл з текстом для шифрування, файл в який буде записано текст, мову оригінального тексту, зсув літер за алфавітом. Після введення всіх необхідних даних зашифрований текст буде записано в відповідний файл. На наступних рисунках наведені зображення вікон та їх опис.

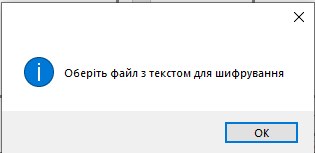
**

*Рис. 2.7. Вікно шифратора*

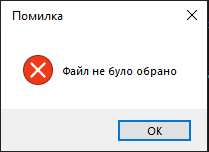
На даному етапі розробки курсового проекту маємо створене програмне забезпечення для виконання різноманітних дій над матрицями. Залишилося провести його тестування. На меті проведення тестування є використання створеного програмного продукту в реальних умовах та перевірка отриманих результатів роботи і порівняння з їх очікуваними.

Після запуску програми викликається заставка, через декілька секунд з’являється вікно програми з налаштуваннями.

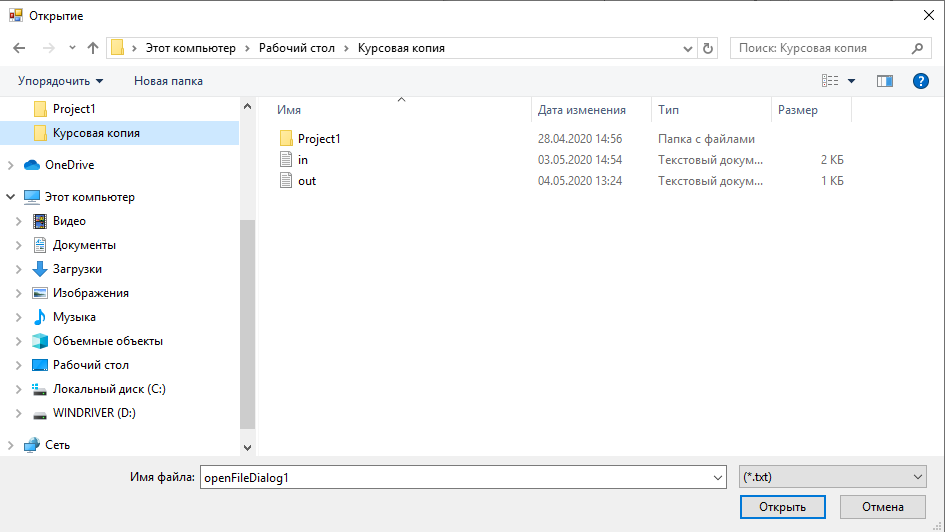
Далі обираємо файл з оригінальним текстом, при натисканні на кнопку вибору користувач отримає повідомлення про те, що саме потрібно обрати (Рис. 2.8), якщо натиснути кнопку відміна користувач отримає повідомлення про те, що файл не було обрано (Рис. 2.9). Користувач може обрати лише файли формату .txt. Всі інші файли відображуватись не будуть.



*Рис. 2.8. Повідомлення про необхідну дію*

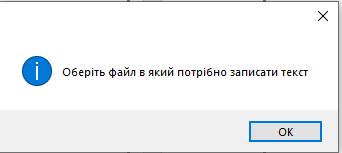


*Рис. 2.9. Повідомлення про не обраний файл*



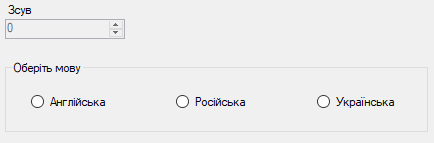
*Рис. 2.10. Меню вибору файлу з текстом*

Після вибору файлу з оригінальним текстом, користувач повинен обрати файл, в який буде записано змінений текст. При натисканні на кнопку вибору користувач отримає повідомлення про те, що саме потрібно обрати (Рис. 2.11), якщо натиснути кнопку відміна користувач отримає повідомлення про те, що файл не було обрано (Рис. 2.9). Користувач може обрати лише файли формату .txt. Всі інші файли відображуватись не будуть.

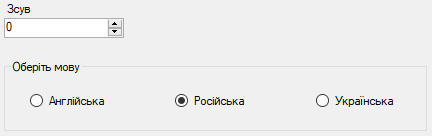


*Рис. 2.11. Повідомлення про необхідну дію*

Далі користувачу потрібно обрати мову оригінального тексту. Якщо не обрати мову, користувач не зможе вказати зсув(Рис. 2.12; 2.13).

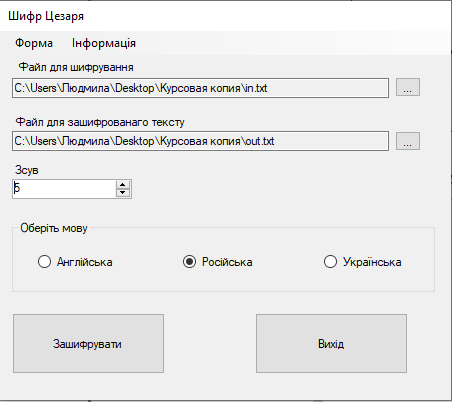


*Рис. 2.12. Поле для зсуву не активне*



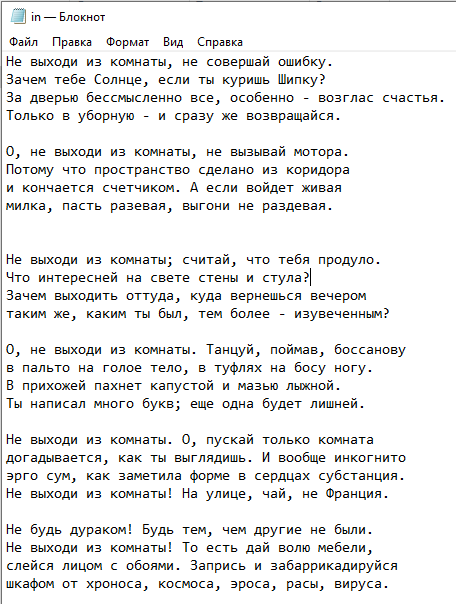
*Рис. 2.13. Поле зсуву активне тільки тоді, коли обрано мову тексту*

Після вибору мови, користувач повинен ввести зсув літер за алфавітом.

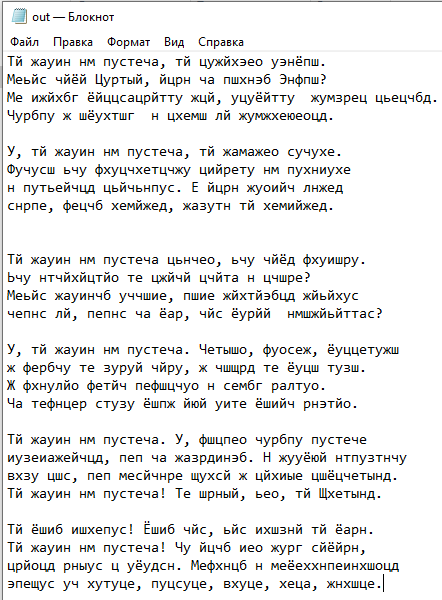


*Рис. 2.14. Повністю заповнена форма*

Далі, якщо всі дані введено вірно, відбудеться шифрування.

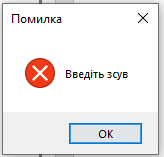


*Рис 2.15. Файл з оригінальним текстом.*

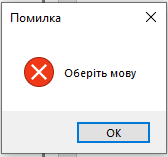


*Рис 2.16. Зашифрований файл.*

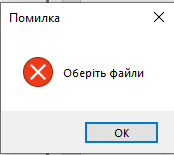
Якщо користувач не обрав якийсь елемент, але натиснув кнопку «Зашифрувати», він отримає повідомлення.



*Рис 2.17. Повідомлення про не введений зсув*



*Рис 2.18. Повідомлення про не обрану мову*



*Рис 2.19. Повідомлення про не обрані файли*

Кнопка «Вихід» закриває форму.

Висновки до другого розділу

В даному розділі курсового проекту проведено розробку загального і функціонального алгоритмів програми. Маючи повне уявлення про задачі, які повинно виконувати програмне забезпечення та методи їхнього вирішення, наступним кроком буде безпосередньо реалізація програмного продукту.

Розроблено загальний алгоритм функціонування роботи програми, в якому розбили проектований додаток на складові частини. При розробці функціонального алгоритму сформовано уявлення про методи розробки програмного коду додатку. При розробці інтерфейсної частини в Windows Forms змоделювали потрібні вікна для програмного забезпечення та наділили їх інтерфейсними одиницями. На заключному етапі розділу перейшли до розробки програмного коду спроектованого додатку.

Опис роботи з додатком включає в себе розробку текстової та ілюстраційної частин, які пояснюють принципи роботи з додатком та опис елементів інтерфейсу для його користувачів. При описі було роз’яснено призначення кожного з елементів інтерфейсу усіх наявних у проекті вікон.

Розроблену програму було протестовано. Результати тестування показали, що програма працює коректно та виконує усі функції, що планувалися.

ВИСНОВКИ

В результаті виконання курсового проекту отримано програмне забезпечення, яке по функціональності повністю відповідає поставленим на початку проектування вимогам.

Програма є простою у використанні з інтуїтивно зрозумілим україномовним інтерфейсом.

Створене програмне забезпечення шифрує текст, в залежності від введених користувачем параметрів. Також відбувається перевірка введених даних користувачем, очищення цих даних та зберігання отриманих результатів у текстовий документ. Для зрозумілості роботи з програмою викликається довідка.

Було проведено тестування створеного програмного продукту, яке показало, що спроектоване програмне забезпечення працює коректно.

Таким чином, можна зробити висновки, що створений програмний продукт повністю відповідає поставленим вимогам та є достатньо швидкодіючим, не займаючи великих ресурсів комп’ютера.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Культин Н.В. «Основи програмування в *Visual C++* 2010» СПб.:БХВ-Петербург, 2010. — 384 с.
2. Культин Н.В. «Microsoft Visual C++ в задачах і прикладах» СПб.:БХВ-Петербург, 2010. — 272 с.
3. Вільна енциклопедія «Програмування по-українськи» [Електронний ресурс] : [Веб-сайт]. – Електронні дані. - Режим доступу: http://programming.in.ua (дата звернення 22.04.2020)

4. Вільний форум «stackoverflow.com» [Електронний ресурс] : [Веб-сайт]. – Електронні дані. - Режим доступу: <https://ru.stackoverflow.com/questions/737102/Шифр-Цезаря-c> (дата звернення 04.05.2020)

# ДОДАТКИ

Додаток А. Лістинг шифратора.

#include <string>

#include <fstream>

#include <iostream>

#include <streambuf>

#pragma once

namespace Project1 {

using namespace std;

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::OpenFileDialog^ openFile1;

protected:

protected:

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::OpenFileDialog^ openFile2;

private: System::Windows::Forms::GroupBox^ groupBox1;

private: System::Windows::Forms::RadioButton^ radioButton3;

private: System::Windows::Forms::RadioButton^ radioButton2;

private: System::Windows::Forms::RadioButton^ radioButton1;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ формаToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ очиститиToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ іноформаціяToolStripMenuItem;

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ очиститиФормуToolStripMenuItem;

private: System::Windows::Forms::NumericUpDown^ numericUpDown1;

private: System::Windows::Forms::ToolStripMenuItem^ інструкціяToolStripMenuItem;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->openFile1 = (gcnew System::Windows::Forms::OpenFileDialog());

this->button1 = (gcnew System::Windows::Forms::Button());

this->openFile2 = (gcnew System::Windows::Forms::OpenFileDialog());

this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());

this->radioButton3 = (gcnew System::Windows::Forms::RadioButton());

this->radioButton2 = (gcnew System::Windows::Forms::RadioButton());

this->radioButton1 = (gcnew System::Windows::Forms::RadioButton());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->button4 = (gcnew System::Windows::Forms::Button());

this->label3 = (gcnew System::Windows::Forms::Label());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->формаToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->очиститиToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->іноформаціяToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->інструкціяToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this->components));

this->очиститиФормуToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->numericUpDown1 = (gcnew System::Windows::Forms::NumericUpDown());

this->groupBox1->SuspendLayout();

this->menuStrip1->SuspendLayout();

this->contextMenuStrip1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDown1))->BeginInit();

this->SuspendLayout();

//

// openFile1

//

this->openFile1->FileName = L"openFileDialog1";

this->openFile1->Filter = L"(\*.txt)|\*.txt";

//

// button1

//

this->button1->Location = System::Drawing::Point(12, 282);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(153, 61);

this->button1->TabIndex = 0;

this->button1->Text = L"Зашифрувати";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// openFile2

//

this->openFile2->FileName = L"openFileDialog1";

this->openFile2->Filter = L"(\*.txt)|\*.txt";

//

// groupBox1

//

this->groupBox1->Controls->Add(this->radioButton3);

this->groupBox1->Controls->Add(this->radioButton2);

this->groupBox1->Controls->Add(this->radioButton1);

this->groupBox1->Location = System::Drawing::Point(12, 190);

this->groupBox1->Name = L"groupBox1";

this->groupBox1->Size = System::Drawing::Size(423, 72);

this->groupBox1->TabIndex = 1;

this->groupBox1->TabStop = false;

this->groupBox1->Text = L"Оберіть мову";

//

// radioButton3

//

this->radioButton3->AutoSize = true;

this->radioButton3->Location = System::Drawing::Point(312, 32);

this->radioButton3->Name = L"radioButton3";

this->radioButton3->Size = System::Drawing::Size(84, 17);

this->radioButton3->TabIndex = 2;

this->radioButton3->TabStop = true;

this->radioButton3->Text = L"Українська";

this->radioButton3->UseVisualStyleBackColor = true;

this->radioButton3->CheckedChanged += gcnew System::EventHandler(this, &MyForm::radioButton3\_CheckedChanged);

//

// radioButton2

//

this->radioButton2->AutoSize = true;

this->radioButton2->Location = System::Drawing::Point(171, 32);

this->radioButton2->Name = L"radioButton2";

this->radioButton2->Size = System::Drawing::Size(76, 17);

this->radioButton2->TabIndex = 1;

this->radioButton2->TabStop = true;

this->radioButton2->Text = L"Російська";

this->radioButton2->UseVisualStyleBackColor = true;

this->radioButton2->CheckedChanged += gcnew System::EventHandler(this, &MyForm::radioButton2\_CheckedChanged);

//

// radioButton1

//

this->radioButton1->AutoSize = true;

this->radioButton1->Location = System::Drawing::Point(26, 32);

this->radioButton1->Name = L"radioButton1";

this->radioButton1->Size = System::Drawing::Size(81, 17);

this->radioButton1->TabIndex = 0;

this->radioButton1->TabStop = true;

this->radioButton1->Text = L"Англійська";

this->radioButton1->UseVisualStyleBackColor = true;

this->radioButton1->CheckedChanged += gcnew System::EventHandler(this, &MyForm::radioButton1\_CheckedChanged);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(12, 47);

this->textBox1->Name = L"textBox1";

this->textBox1->ReadOnly = true;

this->textBox1->Size = System::Drawing::Size(377, 20);

this->textBox1->TabIndex = 2;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(12, 100);

this->textBox2->Name = L"textBox2";

this->textBox2->ReadOnly = true;

this->textBox2->Size = System::Drawing::Size(377, 20);

this->textBox2->TabIndex = 3;

//

// button2

//

this->button2->Location = System::Drawing::Point(395, 46);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(26, 20);

this->button2->TabIndex = 4;

this->button2->Text = L"...";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(395, 100);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(26, 20);

this->button3->TabIndex = 5;

this->button3->Text = L"...";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(15, 29);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(123, 13);

this->label1->TabIndex = 6;

this->label1->Text = L"Файл для шифрування";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 84);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(177, 13);

this->label2->TabIndex = 7;

this->label2->Text = L"Файл для зашифрованаго тексту";

//

// button4

//

this->button4->Location = System::Drawing::Point(255, 282);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(153, 61);

this->button4->TabIndex = 9;

this->button4->Text = L"Вихід";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4\_Click);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(12, 132);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(31, 13);

this->label3->TabIndex = 11;

this->label3->Text = L"Зсув";

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(2) {

this->формаToolStripMenuItem,

this->іноформаціяToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(451, 24);

this->menuStrip1->TabIndex = 12;

this->menuStrip1->Text = L"menuStrip1";

//

// формаToolStripMenuItem

//

this->формаToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(1) { this->очиститиToolStripMenuItem });

this->формаToolStripMenuItem->Name = L"формаToolStripMenuItem";

this->формаToolStripMenuItem->Size = System::Drawing::Size(57, 20);

this->формаToolStripMenuItem->Text = L"Форма";

//

// очиститиToolStripMenuItem

//

this->очиститиToolStripMenuItem->Name = L"очиститиToolStripMenuItem";

this->очиститиToolStripMenuItem->Size = System::Drawing::Size(168, 22);

this->очиститиToolStripMenuItem->Text = L"Очистити форму";

this->очиститиToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::очиститиToolStripMenuItem\_Click);

//

// іноформаціяToolStripMenuItem

//

this->іноформаціяToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(1) { this->інструкціяToolStripMenuItem });

this->іноформаціяToolStripMenuItem->Name = L"іноформаціяToolStripMenuItem";

this->іноформаціяToolStripMenuItem->Size = System::Drawing::Size(83, 20);

this->іноформаціяToolStripMenuItem->Text = L"Інформація";

//

// інструкціяToolStripMenuItem

//

this->інструкціяToolStripMenuItem->Name = L"інструкціяToolStripMenuItem";

this->інструкціяToolStripMenuItem->Size = System::Drawing::Size(130, 22);

this->інструкціяToolStripMenuItem->Text = L"Інструкція";

this->інструкціяToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::інструкціяToolStripMenuItem\_Click);

//

// contextMenuStrip1

//

this->contextMenuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(1) { this->очиститиФормуToolStripMenuItem });

this->contextMenuStrip1->Name = L"contextMenuStrip1";

this->contextMenuStrip1->Size = System::Drawing::Size(169, 26);

//

// очиститиФормуToolStripMenuItem

//

this->очиститиФормуToolStripMenuItem->Name = L"очиститиФормуToolStripMenuItem";

this->очиститиФормуToolStripMenuItem->Size = System::Drawing::Size(168, 22);

this->очиститиФормуToolStripMenuItem->Text = L"Очистити форму";

this->очиститиФормуToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::очиститиФормуToolStripMenuItem\_Click);

//

// numericUpDown1

//

this->numericUpDown1->Enabled = false;

this->numericUpDown1->Location = System::Drawing::Point(12, 148);

this->numericUpDown1->Name = L"numericUpDown1";

this->numericUpDown1->Size = System::Drawing::Size(120, 20);

this->numericUpDown1->TabIndex = 13;

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(451, 369);

this->ContextMenuStrip = this->contextMenuStrip1;

this->ControlBox = false;

this->Controls->Add(this->numericUpDown1);

this->Controls->Add(this->label3);

this->Controls->Add(this->button4);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->button3);

this->Controls->Add(this->button2);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->groupBox1);

this->Controls->Add(this->button1);

this->Controls->Add(this->menuStrip1);

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedToolWindow;

this->MainMenuStrip = this->menuStrip1;

this->Name = L"MyForm";

this->Text = L"Шифр Цезаря";

this->groupBox1->ResumeLayout(false);

this->groupBox1->PerformLayout();

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->contextMenuStrip1->ResumeLayout(false);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDown1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

string SystemToStr(String ^s)

{

using namespace Runtime::InteropServices;

const char\* ptr = (const char\*)(Marshal::StringToHGlobalAnsi(s)).ToPointer();

return string(ptr);

}

void CaesarOut(int k, string r, string w)

{

ifstream input(r, std::ifstream::in | std::ifstream::binary);

ofstream output(w, std::ifstream::out | std::ifstream::binary);

char buff;

int iter;

char RU[] =

{ 'А','Б','В','Г','Д','Е','Ё','Ж','З','И',

'Й','К','Л','М','Н','О','П','Р','С',

'Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ',

'Ы','Ь','Э','Ю','Я' };

char ru[] =

{ 'а','б','в','г','д','е','ё','ж','з','и',

'й','к','л','м','н','о','п','р','с',

'т','у','ф','x','ц','ч','ш','щ','ъ',

'ы','ь','э','ю','я' };

char UA[] =

{ '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' };

char ua[]

{ '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' };

char ENG[] =

{ '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' };

char eng[] =

{ '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' };

while (input.eof() == false) {

buff = input.get();

if (buff == ' ')

output << ' ';

if (buff == '\n')

output << '\n';

if (buff == '?')

output << '?';

if (buff == '.')

output << '.';

if (buff == '!')

output << '!';

if (buff == ',')

output << ',';

if (radioButton1->Checked == true)

{

for (int i = 0; i < 26; i++)

{

if (buff == ENG[i])

{

if (i + k >= 26)

{

buff = ENG[i - (26 - k)];

output << buff;

break;

}

else

{

buff = ENG[i + k];

output << buff;

break;

}

}

if (buff == eng[i])

{

if (i + k >= 26)

{

buff = eng[i - (26 - k)];

output << buff;

break;

}

else

{

buff = eng[i + k];

output << buff;

break;

}

}

}

}

if (radioButton2->Checked == true)

{

setlocale(LC\_ALL, "Russian");

for (int i = 0; i < 33; i++)

{

if (buff == RU[i])

{

if (i + k >= 33)

{

buff = RU[i - (33 - k)];

output << buff;

break;

}

else

{

buff = RU[i + k];

output << buff;

break;

}

}

if (buff == ru[i])

{

if (i + k >= 33)

{

buff = ru[i - (33 - k)];

output << buff;

break;

}

else

{

buff = ru[i + k];

output << buff;

break;

}

}

}

}

if (radioButton3->Checked == true)

{

setlocale(LC\_ALL, "Ukrainian");

for (int i = 0; i < 33; i++)

{

if (buff == UA[i])

{

if (i + k >= 33)

{

buff = UA[i - (33 - k)];

output << buff;

break;

}

else

{

buff = UA[i + k];

output << buff;

break;

}

}

if (buff == ua[i])

{

if (i + k >= 33)

{

buff = ua[i - (33 - k)];

output << buff;

break;

}

else

{

buff = ua[i + k];

output << buff;

break;

}

}

}

}

}

input.close();

output.close();

}

#pragma endregion

public: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

int move;

string read, write;

read = SystemToStr(openFile1->FileName);

write = SystemToStr(openFile2->FileName);

if (numericUpDown1->Text == "0")

{

MessageBox::Show("Введіть зсув", "Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else

{

move = Convert::ToInt64(numericUpDown1->Text);

}

if (radioButton1->Checked == false && radioButton2->Checked == false && radioButton3->Checked == false)

{

MessageBox::Show("Оберіть мову", "Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

if (textBox1->Text == "" || textBox2->Text == "")

{

MessageBox::Show("Оберіть файли", "Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

CaesarOut(move, read, write);

}

public: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

{

MessageBox::Show("Оберіть файл з текстом для шифрування", "", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

openFile1->ShowDialog();

if (openFile1->FileName == "openFileDialog1")

{

MessageBox::Show("Файл не було обрано", " Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

openFile1->ShowDialog();

}

else

{

textBox1->Text = openFile1->FileName;

}

}

public: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

{

MessageBox::Show("Оберіть файл в який потрібно записати текст", "", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

openFile2->ShowDialog();

if (openFile2->FileName == "openFileDialog1")

{

MessageBox::Show("Файл не було обрано", "Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);

openFile2->ShowDialog();

}

else

{

textBox2->Text = openFile2->FileName;

}

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e)

{

Application::Exit();

}

private: System::Void розробникиToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void очиститиФормуToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

textBox1->Text = "";

textBox2->Text = "";

numericUpDown1->Text = "0";

radioButton1->Checked = false;

radioButton2->Checked = false;

radioButton3->Checked = false;

numericUpDown1->Enabled = false;

}

private: System::Void очиститиToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

textBox1->Text = "";

textBox2->Text = "";

numericUpDown1->Text = "0";

radioButton1->Checked = false;

radioButton2->Checked = false;

radioButton3->Checked = false;

numericUpDown1->Enabled = false;

}

private: System::Void radioButton1\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)

{

numericUpDown1->Text = "0";

if (radioButton1->Checked == true)

{

numericUpDown1->Maximum = 25;

this->numericUpDown1->Enabled = true;

}

else

{

if (radioButton2->Checked == true || radioButton2->Checked == true)

{

numericUpDown1->Maximum = 32;

this->numericUpDown1->Enabled = true;

}

}

}

private: System::Void radioButton2\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)

{

numericUpDown1->Text = "0";

if (radioButton1->Checked == true)

{

numericUpDown1->Maximum = 25;

this->numericUpDown1->Enabled = true;

}

else

{

if (radioButton2->Checked == true)

{

numericUpDown1->Maximum = 32;

this->numericUpDown1->Enabled = true;

}

else

{

if (radioButton3->Checked == true)

{

numericUpDown1->Maximum = 32;

this->numericUpDown1->Enabled = true;

}

}

}

}

private: System::Void radioButton3\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)

{

numericUpDown1->Text = "0";

if (radioButton1->Checked == true)

{

numericUpDown1->Maximum = 25;

this->numericUpDown1->Enabled = true;

}

else

{

if (radioButton2->Checked == true)

{

numericUpDown1->Maximum = 32;

this->numericUpDown1->Enabled = true;

}

else

{

if (radioButton3->Checked == true)

{

numericUpDown1->Maximum = 32;

this->numericUpDown1->Enabled = true;

}

}

}

};

private: System::Void інструкціяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

Help::ShowHelp(this,"Help/NewProject.chm");

}

};

}