ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ ІНФОРМАТИКИ

Допускається до захисту

Завідувач кафедри _____О.О. Ємець
(підпис)

«_____»_____2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА ДО БАКАЛАВРСЬКОЇ РОБОТИ

на тему

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ТЕМИ «КОДУВАННЯ ТЕКСТІВ ШИФРОМ ГОНСФЕЛЬДА»

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи Ніколаєнко Олексій Васильович
_____ «___»_____2020 р.
(підпис)
Науковий керівник професор, канд. фіз.-мат. наук Ємець Єлизавета Михайлівна
_____ «___»_____2020 р.
(підпис)

ПОЛТАВА 2020 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ 3
ВСТУП 4
1 ПОСТАНОВКА ЗАДАЧІ 6
2 ІНФОРМАЦІЙНИЙ ОГЛЯД 7
2.1 Огляд робіт, де розглянуто аналогічне до теми роботи завдання, їх переваги та вади 7
2.2 Основні види та призначення платформ дистанційного навчання 8
2.3 Загальні відомості про захист інформації в інформаційно-телекомунікаційних системах 19
3 ТЕОРЕТИЧНА ЧАСТИНА 23
3.1 Теоретичні відомості з теми «Кодування текстів шифром Гронсфельда» 23
3.2 Приклади застосування 25
3.3 Алгоритмізація за темою роботи 26
3.4 Розробка блок-схеми, яка підлягає реалізації 27
4 ПРАКТИЧНА ЧАСТИНА 29
4.1 Обґрунтування вибору програмних засобів 29
4.2 Опис процесу програмної реалізації 31
4.3 Опис програми 36
4.4 Необхідна користувачу програми інструкція 38
ВИСНОВКИ 42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43
ДОДАТОК А. КОД ПРОГРАМИ 45

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, символів, скорочень
Код	Набір символів (умовних позначень) для представлення інформації.
Код	Система умовних знаків (символів) для передачі, обробки та зберігання інформації (зі спілкування).
Кодування	Процес представлення інформації (повідомлення) у вигляді коду.
Шифр Гронсфельда	Поліалфавітний підстановочний шифр створений графом Гронсфельдом.
ПЗ	Програмне забезпечення

ВСТУП

Кодування буквально пронизує інформаційні технології і є центральним питанням при розв’язуванні самих різних (практично усіх задач) програмування:

- представлення даних довільної природи (наприклад, чисел, тексту, графіки) у пам’яті комп’ютера;
- захист інформації від несанкціонованого доступу;
- забезпечення перешкодозахищеності при передачі даних по каналам зв’язку;
- стиснення інформації у бази даних.

Метою роботи – розробка програмного забезпечення з теми «Кодування текстів шифром Гронсфельда».

Об’єктом розробки – процес дистанційного навчання методам кодування.

Предметом розробки – програмне забезпечення для кодування текстів шифром Гронсфельда.

Головне завдання – розробка алгоритму програмного забезпечення з теми «Кодування текстів шифром Гронсфельда» та створення програмної реалізації.

Методи розробки – методика кодування текстів шифром Гронсфельда. Серед програмного забезпечення – середовище розробки Microsoft Visual Studio, мова програмування C++ (CLR (.NET Framework)).

Новизною роботи – розробка програмного забезпечення з теми «Кодування текстів шифром Гронсфольда». Розробка даного програмного забезпечення є унікальною в своєму роді, бо не знайдено жодного схожого ПЗ з цієї теми.

Практичною цінністю роботи – створення програмного забезпечення для навчання студентів кодуванню текстів шифром Гронсфольда. Реалізований навчальний тренажер рекомендовано до використання студентами спеціальності «Комп’ютерні науки».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі програмного забезпечення. У другому розділі розглянуто навчальні тренажери, описано їх переваги та вади, а також актуальність та

призначення дистанційного навчання. У третьому розділі представлено теоретичний матеріал, описано алгоритм роботи програмного забезпечення та представлено його блок-схему. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, опис програми та інструкцію, щодо використання програмного забезпечення.

Обсяг пояснювальної записки: 57 сторінки, в тому числі основна частина – 44 сторінки, літературних джерел – 10.

1 ПОСТАНОВКА ЗАДАЧІ

Задачею роботи – розробка програмного забезпечення з теми «Кодування текстів шифром Гронсфельда».

Для виконання поставленої задачі, було висунуто основні завдання:

1. Розглянути актуальність та призначення дистанційних платформ.
2. Оглянути навчальні тренажери, виокремити переваги та вади.
3. Опрацювати навчальну літературу з теми «Кодування текстів шифром Гронсфельда».
4. Розробити алгоритм програмного засобу, який допоможе студентові закріпити знання та вміння.
5. Програмного реалізувати ПЗ з теми «Кодування текстів шифром Гронсфельда».
6. Обґрунтувати вибір програмного забезпечення.
7. Описати процес програмної реалізації.
8. Описати інструкцію, щодо використання програмного засобу.

При оформлені матеріалів роботи необхідно дотримуватися методичних рекомендацій щодо оформлення. Зміст роботи повинен відповідати завданню затвердженим керівником та завідувачем кафедри [1].

Результатами роботи мають бути опубліковані в збірнику наукових статей та матеріалах науково-практичного семінару.

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми завдання, їх переваги та вади

Під час проведення інформаційного огляду було виділено деякі програмні забезпечення:

Першим розглянуто навчальний тренажер на тему «Градієнтний метод». Під час запуску тренажера з’являється вікно із завданням в якому зазначено обчислення методом найшвидшого спуску [2,3].

Другим було розглянуто тренажер з теми «Розкриття найпростіших невизначеностей». На головній сторінці виводиться тема, автор-розробник і керівник [4,6]. Натиснувши на кнопку «Довідка» з’являється вікно з довідкою. Якщо натиснути «Типи невизначеностей» відбувається перехід до вибору завдання. Перед кожним кроком з’являється спочатку інструкція, потім умова прикладу. У кожному завданні потрібно обрати відповідь або формулу. У разі виникнення помилки з’являється відповідне повідомлення. Після проходження навчального тренажера виводиться повідомлення і відбувається перехід на головну сторінку.

Наступний навчальний тренажер був з теми «Диференціальне числення функції однієї змінної» [4] містить схожу до попереднього загальну інформацію. Також пропонує розглянути теоретичний матеріал або перейти до самого тренажеру В даному програмного застосунку студентові надається можливість запустити вже розроблений навчальний тренажер або створити власний. Якщо вибрати «Створити новий тренажер», то необхідно вказати його назву та додати запитання. Після запуску вже існуючого тренажера відображаються питання і варіанти відповідей. Спочатку перевіряться на вірність відповідь, а потім реалізується перехід до наступного питання. Після проходження також виводиться відповідне повідомлення і відбувається перехід на головну сторінку.

Останнім було розглянуто тренажер з теми «Матриці і визначники» [6], в якому відразу пропонується перейти до вибору прикладу. Якщо при виконанні

прикладу ввести значення в порожню комірку або обрати відповідну відповідь, то студента інформує чи правильно відповіли, чи ні. Також реалізовано вікно з повідомленням про помилку. Після розв’язання прикладу є можливість перейти далі або повернутися до меню.

2.2 Основні види та призначення платформ дистанційного навчання

Оскільки одним із стратегічних напрямів реформування освітньої системи України є активне використання інформаційних та комунікаційних технологій для розвитку дистанційного навчання необхідно зупинитися на дослідженні застосовування платформ дистанційного навчання, без яких організувати дистанційне навчання неможливо. Вибір платформ дистанційного навчання є дуже важливим кроком.

Платформа дистанційного навчання – це програмне забезпечення для підтримки дистанційного навчання, метою якого є створення та управління педагогічним змістом, індивідуалізоване навчання та телетьюторат. Воно включає засоби, необхідні для трьох основних користувачів – викладача, студента, адміністратора.

Тобто платформа дистанційного навчання – це центральний елемент, навколо якого збираються учасники дистанційної освіти.

У цій системі, викладач створює загальний курс навчання, використовуючи мультимедійні педагогічні ресурси, індивідуалізує його до потреб та здібностей кожного студента, та здійснює підтримку діяльності студентів.

Студент вивчає в мережі або завантажує педагогічний зміст, що йому рекомендований, організовує свою роботу, виконує вправи, він може бачити еволюцію своєї діяльності на інтерфейсі комп’ютера, виконувати завдання для самооцінки та передавати виконані завдання на перевірку викладачеві. Викладачі та студенти спілкуються індивідуально або в групі, пропонують теми для обговорення й співробітничають при вивченні або створенні загальних документів.

Адміністратор забезпечує й підтримує обслуговування системи, управляє доступами та правами викладачів і студентів, створює зв’язки із зовнішніми інформаційними системами (адміністративними документами, каталогами, педагогічними ресурсами тощо). Тобто адміністратор платформи має специфічну роль, яка відрізняється від ролі адміністратора установи.

На сьогоднішній день у світі існує значне число e-learning платформ для організації електронного навчання, які поділяються на дві великі категорії: з закритим кодом (комерційні); відкритим кодом (поширюються безкоштовно) [7,8].

Платформа для електронного навчання «Blackboard»

Світовим лідером серед розробників комерційних продуктів є американська компанія Blackboard Inc. (www.blackboard.com), яка розробила однойменну платформу для електронного навчання "Blackboard". Компанія володіє цілою лінійкою програмних продуктів, які активно використовуються по всьому світу для організації навчального процесу на всіх рівнях освіти. Особливо продукція компанії широко використовується в Північній Америці і Японії. Після придбання іншої великої компанії WebCT, що також спеціалізувалася на електронної освіті, Blackboard зміцнив свої позиції і в Європі. Недоліками цього продукту є висока вартість.

До складу системи Blackboard Learn входять:

- Blackboard Course Delivery - платформа електронного навчання, призначена для управління віртуальним навчальним середовищем і надання платформи для курсів дистанційного навчання;
- Blackboard Content Management - сховище електронних освітніх ресурсів, призначене для централізованого накопичення та структурування електронних освітніх ресурсів, а також управління доступом до них користувачів і зовнішніх додатків;
- Blackboard Community Engagement – навчальний портал, призначений для організації єдиного доступу до сервісів системи Blackboard Learn, забезпечення комунікацій і спільної роботи користувачів.

Система Blackboard забезпечує єдине інтерактивне середовище для навчання, взаємодії, обміну інформацією між студентами і викладачами та тьюторами ВНЗ. Система дозволяє управляти віртуальним навчальним середовищем, створювати електронні освітні ресурси, забезпечувати віддалений доступ до освітніх ресурсів навчального закладу, здійснювати контроль освітнього процесу, надавати платформи для курсів дистанційного навчання, накопичувати, структурувати, керувати доступом, поповнювати освітню базу, а також надавати засоби комунікації та інформування учасників.

Система Blackboard дозволяє автоматизувати наступні основні області діяльності вузу в освітньому процесі: підготовка освітніх матеріалів, дистанційне навчання, спільна науково-дослідна діяльність, облік і контроль персональних критеріїв освітнього процесу, ведення нормативно-довідкової інформації, спільна робота віддалених членів освітніх проектів.

Web-сервіси компанії Blackboard:

♦ запобігання плагіату за допомогою програми SafeAssign, що дає можливість викладачам доносити до студентів важливість академічної порядності та встановлення справжності авторства;

♦ інтеграція з платформою Facebook, що забезпечує доступ до інформації з курсу навчання, оновлень інформації, списків та оповіщень, а також можливість соціального навчання в рамках інтерфейсу Facebook.

До переваг системи можна віднести:

→ можливість роботи в єдиній системі на різних мовах;

→ можливість масштабування системи; цілодобова технічна та методична підтримка користувачів;

→ наявність гарантій якості рішень; наявність впроваджень системи в проектах з більш ніж 100 000 користувачів;

→ швидка автоматизована підготовка звітів;

→ використання єдиної централізованої бази даних;

→ інтеграція з єдиним каталогом користувачів.

Ліцензія надається на 12 місяців. Вартість залежить від кількості користувачів від 38 000 $.

Система дистанційного навчання «Прометей»

Система дистанційного навчання «Прометей» (розробник: ООО «Віртуальні технології в освіті», Росія), є платформою, за допомогою якої можна створити віртуальний університет та організувати дистанційне навчання з великою кількістю студентів, автоматизуючи при цьому весь навчальний процес від вступу до видачі диплома.

«Прометей» має таку модульну архітектуру: навчальний портал, реєстрація, контроль за оплатою за навчання, керування групами, календарний план, бібліотеки, тестування, спілкування.

Платформа відзначена сертифікатом Міністерства освіти Росії про відповідність вимогам до СДН. Налаштування інтерфейсу можливий п’ятьма мовами (російська, українська, казахська, англійська та іспанська). Переваги:

→ простота освоєння та експлуатація;

→ відсутність ліцензій на клієнтські місця;

→ використання методики онлайн - навчання;

→ висока продуктивність та масштабність відповідно до зростання кількості користувачів і навантаження;

→ 10 видів тестів (можливість використання графіки та мультимедіа);

→ мінімальні вимоги до сервера та клієнтських місць;

→ дозволяє об’єднати декілька систем в єдине освітнє середовище;

→ інтегрує з кадровими, бухгалтерськими, інформаційними ресурсами;

→ встановлюється протягом одного дня.

Система дистанційного навчання на основі платформи «Прометей» успішно використовують у державних та корпоративних установах, ВНЗ Росії, України та інших країнах СНД.

Крім комерційних систем організації електронного навчання з закритими кодами існують і так звані open source («відкритим вихідним кодом») рішення. Їх відмітною особливістю є те, що вихідні коди цих програм відкриті для користувачів

і допускають будь-які виправлення, модифікацію і доповнення. Згідно ліцензії, за якої поширюються на ці 32 продукти, вони абсолютно безкоштовні і такими залишаться. На сьогоднішній день існують кілька десятків платформ електронного навчання, побудованих за принципом відкритих джерел. Для дослідження були відібрані дев'ять найбільш популярних відкритих платформ і проведено зіставлення їх можливостей.

Платформа дистанційного навчання ATutor

ATutor є веб-орієнтованою системою керування навчанням (Learning Management System, LMS). Програмний продукт є простим у встановленні, налаштуванні та підтримці для системних адміністраторів; викладачі (інструктори) можуть досить легко створювати та переносити навчальні матеріали та запускати свої онлайн-курси. А оскільки система є модульна, тобто складається з окремих функціональних одиниць — модулів, то вона відкрита для модернізації і розширення функціональних можливостей. Програма розробляється та підтримується з 2001 року Ґреґом Ґеєм (Greg Gay), Джоелом Кроненберґом (Joel Kronenberg), Гайді Гейзелтон (Heidi Hazelton) із Дослідницького центру адаптивних технологій Університету Торонто (Adaptive Technology Resource Centre, University of Toronto). Система ATutor поширюється на основі GNU General Public License (GPL), 33 яка, зокрема, дозволяє вільно використовувати, змінювати та доповнювати програму.

В ATutor визначено 3 типи користувачів (студенти, інструкторивикладачі та адміністратори). Система надає різним категоріям користувачів різні можливості.

Для студентів:

1. Редагування персональної інформації (студент має можливість редагувати персональну інформацію, включаючи можливість завантаження власного фото, зміни паролю та адреси електронної пошти);

2. Перегляд існуючих курсів та запис на них (студент може переглядати список курсів, відправляти запит на отримання прав доступу до них);

3. Використання навчальних курсів (студент має можливість переглядати в повному об'ємі інформацію у навчальному курсі, на який він записаний, з

можливістю пакетного завантаження навчальних матеріалів, якщо це дозволено інструктором курсу);

4. Тестування та опитування (студенти в рамках навчального курсу можуть проходити тестування або анонімні опитування, переглядати результати тестувань);

5. Засоби спілкування (система дистанційного навчання володіє такими засобами зв'язку між учасниками навчального процесу: синхронними (чати, телеконференції, дошки (whiteboards), асинхронними (оголошення, форуми, внутрішні повідомлення, електронна пошта, блоги, вікі, коментарі в файлообміннику);

6. Групи та файлообмінник (студенти можуть завантажувати та обмінюватись файлами в рамках навчального курсу або своєї групи);

7. Пошук (ефективна система пошуку в межах навчального курсу, всіх курсів та зовнішніх джерел інформації (пошук по TILE).

Для інструкторів (викладачів).

Інструктори, окрім можливостей студентів, мають додаткові інструменти для ефективного створення навчальних курсів в системі ATutor.

Зокрема:

1. Навчальний курс (викладачі мають можливість створювати навчальні курси в межах системи, визначати права доступу до них та інші властивості);

2. Матеріал (створення навчальних матеріалів у навчальному курсі з використанням вбудованого редактора матеріалів, керування навчальними матеріалами (структура, період доступу), та перегляд статистики використання матеріалів. Можливість експорту та імпорту навчальних матеріалів у формат обміну навчальними матеріалами SCORM);

3. Файловий менеджер (завантаження на сервер необхідних навчальних матеріалів, наприклад, текстів лекцій, практичних занять, тощо у різноманітних форматах (Microsoft Word, PDF, DJVU) з наступним використанням у навчальних матеріалах. Передбачена можливість пакетного завантаження файлів);

4. Тести (широкі можливості щодо створення і керування тестами, запитаннями, організація бази даних питань курсу, попередній перегляд 35 тестів,

перегляд спроб складання тестів користувачами, можливість їх оцінювання, перегляд статистики по тестах);

5. Запис на курс, групи (керування записом на курс, перегляд записаних на курс студентів та керування їх правами у межах курсу. Можливість призначення асистентів та випускників курсу. Створення груп у межах курсу та керування ними);

6. Електронна пошта курсу (дозволяє розсилати повідомлення різним категоріям студентів: усім зареєстрованим у даному курсі, тільки привілейованим студентам, випускникам, тим, кому в запису на курс було відмовлено, або студентам окремих груп);

7. Резервна копія курсу (можливість створення резервних копій курсу, відновлення курсу з резервної копії);

8. Оголошення (дає можливість додавати, видаляти та редагувати оголошення для студентів курсу. Оголошення відображаються на домашній сторінці курсу і можуть розсилатися через RSS (якщо така функція увімкнена у властивостях курсу);

9. Опитування (за допомогою цього інструменту можна організовувати неоцінювані опитування студентів з метою з’ясування їх думки з тих чи інших питань);

10. Словник (цей пункт дозволяє вводити і редагувати словникові терміни. Терміни, які використовуються в матеріалі, легше вводити через редактор матеріалу);

11. Список літератури (цей засіб дає можливість вказувати список джерел, обов’язковість та термін ознайомлення з ними);

12. Статистика (цей інструмент показує дані про те, як користуються курсом студенти та незареєстровані користувачі).

Для адміністраторів:

1. Керування користувачами (можливість керування користувачами системи, та їх правами);

2. Керування курсами (можливість керування курсами системи, резервними копіями);

з. Керування загальними параметрами системи (можливість керування загальними параметрами системи, зокрема темами оформлення, мовою інтерфейсу тощо).

Серед ВУЗів України систему ATutor використовує Тернопільський національний технічний університет імені Івана Пулюя.

Платформа дистанційного навчання Dokeos

Dokeos – платформа побудови сайтів дистанційного навчання, заснована на гілці (fork) Claroline (версії 1.4.2.). Гілка являє собою клон вільно поширюваного програмного продукту, створений з метою змінити додаток-оригінал в тому чи іншому напрямку.

Dokeos безкоштовний і залишиться таким, оскільки ліцензія Claroline (GNU/GPL) припускає, що гілки підпадають під ту ж ліцензію. Оскільки гілка була виділена недавно, обидва додатки зараз відносно схожі один на 37 одного, хоча деякі відмінності в ергономіці, побудові інтерфейсу, функціоналі вже починають проявлятися.

Dokeos - результат роботи деяких членів первісної команди розробників Claroline, які задумали: змінити орієнтацію додатку. Справа в тому, що Claroline прекрасно адаптована для університетського середовища, що виражається в підтримці великої кількості учнів та курсів. Dokeos, більше орієнтований на професійну клієнтуру, наприклад, на персонал підприємства.

Система Dokeos має великий набір психологічних та організаційно-технічних можливостей, а саме:

- створення та підтримка онлайн-курсів; облік і контроль успішності;
- можливість постійного оновлення і доповнення змісту курсу;
- модульність – кожен модуль окремо можна редагувати, робити відкритим або прихованим, а також експортувати у вигляді SCORM;
- можливість поділу студентів на малі групи;
- використання вбудованих мультимедійних додатків для ілюстрації змісту зображеннями, анімацією, звуком і відео;

- різноманітні засоби комунікації з викладачем та іншими студентами: пошта, чат, форум, обмін файлами, відео конференції.

Також є можливість проводити онлайн анкетування студентів про ефективність навчання і отримувати дані у вигляді порівняльних таблиць.

До переваг Dokeos належать:

- Автоматизація навчання;
- оптимізація для мобільних пристроїв та планшетів; відео конференції є базовим функціоналом, що дуже важливо для ефективного дистанційного курсу або інтернет-тренінгу;
- наявність інструментарію для створення колективних проектних робіт і вікі-документів;
- можливість створювати різноспрямовані тести, а саме: тести множинного вибору з одним або декількома правильними відповідями, завдання на зіставлення або вибудовування елементів по порядку, завдання на заповнення пропусків, завдання на маркування різних областей малюнка, а також питання для вільної відповіді;
- можливість сортувати надіслані письмові роботи, обмежувати терміни виконання; відкривати або закривати студентам доступ до робіт однокурсників;
- менеджер звітів, що надає можливість отримати як глобальний звіт про успішність студентів за курсом, так і детальний звіт про успішність кожного студента, а також додаткової інформації про те, як часто і як довго студент працював з дистанційним курсом.

Система Dokeos отримала високу оцінку зарубіжних фахівців в області дистанційної освіти. Особливо наголошується функціональність і простота використання системи, сумісність з різними операційними системами, 39 ергономічність та економічність. Крім того, система постійно розвивається, додаються нові інтерактивні інструменти створення контенту та організації процесу навчання.

Платформа дистанційного навчання Moodle

Moodle (модульне об'єктно-орієнтоване динамічне навчальне середовище), яке може використовуватися як платформа для електронного, в тому числі дистанційного навчання. Moodle — це безкоштовна, відкрита (Open Source) система управління навчанням. Вона реалізує філософію «педагогіки соціального конструктивізму» та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання.

Moodle перекладена на десятки мов, в числі й на українську. Система використовується у 175 країнах світу.

Головним розробником системи є Martin Dougiamas з Австралії. Цей проект є відкритим та в ньому бере участь і велика кількість інших розробників.

В сучасному інформаційному суспільстві Moodle набуває все більшого поширення. Сьогодні система використовується не лише в закладах вищої школи, а й загально-освітніх школах, некомерційних організаціях, приватних компаніях, індивідуальними викладачами і навіть, батьками, що самостійно навчають дітей. Цьому сприяє те, що система придатна для використання не тільки в варіанті роботи в глобальних мереж, а й легко адаптується під самодостатню платформу для створення локальних однокористувацьких офлайн навчальних ресурсів, та ресурсів, здатних повноцінно функціонувати в рамках локальних мереж. Цей комплекс забезпечує розробника навчального ресурсу великою кількістю інструментів, які надають можливість співпрацювати на рівнях учень – учень, учень – викладач, викладач – учень в мережному варіанті, або учень – система керування курсом, система керування курсом – учень в оф-лайн режимі роботи. При підготовці та проведенні занять на платформі Moodle викладач може використовувати її можливості, за допомогою яких організовує вивчення матеріалу таким чином, щоб форми навчання відповідали цілям та задачам конкретних занять.

Основні характеристики системи, які дозволили їй стати визнаним лідером серед програмного забезпечення цього типу:

- розширена функціональність (викладення матеріалів, перевірка знань, аналіз активності студентів, простота оновлення контенту;

- можливість створення копій, висока стійкість);
- низька вартість впровадження - сама система безкоштовна, відсутні обмеження за кількістю ліцензій на слухачів (студентів) та підтримуваних курсів. Витрати на впровадження системи, розробку курсів і супровід - мінімальні, вони не потребують спеціальних технічних знань (адмініструвати систему здатний користувач з поглибленими знаннями в області мережних технологій, а при створенні курсу визначальний характер мають тільки знання в тій області, по якій створюється курс, з технічних знань для автора достатньо мати навики впевненого користувача комп'ютера);
- наявність вбудованих засобів розробки та редагування навчального контента, інтеграції різноманітних освітніх матеріалів різного призначення та підтримка міжнародного стандарту SCORM - основи обміну електронними курсами, що забезпечує перенесення ресурсів в інші системи (з інших систем); модульність – наявність в навчальних курсах набору блоків матеріалу, які можуть бути використані в інших курсах;
- зручність та простота використання - інтуїтивно зрозумілий інтерфейс та технологія навчання (можливість легко знайти меню допомоги, простота переходу від одного розділу до іншого, можливість підказок інструктора, тощо;
- наявність вебсайту moodle.org, який виступає в ролі централізованого джерела інформації, дискусій та співпраці серед користувачів Moodle - системних адміністраторів, викладачів, дослідників, проектувальників і, звісно, розробників. Подібно Moodle, сайт постійно розвивається, щоб забезпечувати потреби суспільства.

2.3. Загальні відомості про захист інформації в інформаційно-телекомунікаційних системах

Стрімкий розвиток засобів обчислювальної техніки і відкритих мереж передачі даних зумовило їх широке поширення в повсякденному житті і підприємницької діяльності [9].

Однак останні досягнення людської думки в області комп'ютерних технологій пов'язані з появою не тільки персональних комп'ютерів, мереж передачі даних і електронних грошей, а й таких понять, як хакер, інформаційна зброя, комп'ютерні віруси та ін.

На практиці загрози ІТС можуть бути реалізовані безпосереднім впливом на інформацію, що представляє інтерес для кінцевих користувачів подібних систем, так і на інформаційні ресурси та телекомунікаційні служби, які забезпечуються в рамках цієї ІТС. Наприклад, існує поширений вид атаки через Internet - шторм помилкових запитів на TCP (Transmission Control Protocol - протокол управління передачею) - з'єднання, що приводить до того, що система тимчасово припиняє обслуговування віддалених користувачів.

Під інформаційною безпекою будемо розуміти стан захищеності оброблюваних, збережених і переданих в ІТС даних від незаконного ознайомлення, перетворення і знищення (як крайній випадок модифікації), а також стан захищеності інформаційних ресурсів від впливів, спрямованих на порушення їх працездатності.

Основними завданнями захисту інформації, яка призначена для користувача, є забезпечення [9]:

- конфіденційності інформації;
- цілісності інформації;
- достовірності інформації;
- оперативності доступу до інформації;
- юридичної значимості інформації, представленої у вигляді електронного документа;
- невідстежуваності дій клієнта.

Конфіденційність інформації - це її властивість бути доступною тільки обмеженому колу користувачів ІТС, в якій циркулює ця інформація.

Під цілісністю інформації розуміють властивість її або програмного забезпечення зберігати свою структуру і / або вміст в процесі передачі і / або зберігання. розглядаючи питання передачі інформації у вигляді повідомлення через

мережу, можна прийти до висновку, що кожне повідомлення за своїм смисловим змісту утворює певний клас. Іншими словами, сенс кінцевого повідомлення залишиться таким же, як і початкового, навіть якщо форма подання інформації в електронному вигляді суттєво зміниться. Таким чином, кожне повідомлення на будь-якій мові матиме свій клас еквівалентності, і для даного випадку властивість збереження цілісності можна сформулювати наступним чином: передане повідомлення М вважається зберігає цілісність, якщо отримане в результаті передачі повідомлення М належить класу еквівалентності повідомлення М.

Достовірність інформації - це властивість, яке виражається в суворій приналежності об'єкту, який є її джерелом, або тому об'єкту, від якого ця інформація прийнята.

Оперативність - це здатність інформації або деякого інформаційного ресурсу бути доступним для кінцевого користувача відповідно до його тимчасовими потребами.

Юридична значимість означає, що документ має юридичну силу. З цією метою суб'єкти, які потребують підтвердження юридичної значимості переданого повідомлення, домовляються про повсюдне прийняття деяких атрибутів інформації, що виражають її здатність бути юридично значимої. Дана властивість інформації особливо актуально в системах електронних платежів, де здійснюється операція з переказу грошових коштів. Виходячи зі сказаного, можна сформулювати деякі вимоги до атрибутів інформації, що виражає її властивість бути юридично значимої. інформацію необхідно сформувати таким чином, щоб з формальної точки зору було визначено ясно, що тільки відправник, якому належить даний платіжний документ, міг його створити.

Невідстежуваність - це здатність здійснювати деякі дії в ІТС непомітно для інших об'єктів. Актуальність даної вимоги стала очевидною завдяки появі таких понять, як електронні гроші і Internet-banking. Так, для авторизації доступу до електронної платіжної системи користувач повинен надати деякі відомості, однозначно його ідентифікують. У міру стрімкого розвитку даних систем може з'явитися реальна небезпека, що, наприклад, всі платіжні операції будуть

контролюватися, тим самим виникнуть умови для тотального стеження за користувачами ІТС.

Існує кілька шляхів вирішення проблеми невідстежуваності:

- заборона за допомогою законодавчих актів всякого тотального стеження за користувачами ІТС;

- застосування криптографічних методів для підтримки невідстежуваності.

Як вже говорилося, інформаційна безпека може розглядатися не тільки по відношенню до деяких конфіденційних відомостей, а й по відношенню до здатності ІТС виконувати задані функції.

Основні завдання, які вирішуються в рамках інформаційної безпеки по відношенню до працездатності ІТС, повинні забезпечувати захист від:

- порушення функціонування телекомунікаційної системи, що виражається у впливі на інформаційні канали, канали сигналізації, управління і віддаленого завантаження баз даних комутаційного обладнання, системне і прикладне програмне забезпечення;

- несанкціонованого доступу до інформаційних ресурсів і від спроб використання ресурсів мережі, що призводять до витоку даних, порушення цілісності мережі та інформації, зміни функціонування підсистеми розподілу інформації, доступності баз даних;

- руйнування вбудованих і зовнішніх засобів захисту;

- неправомірних дій користувачів і обслуговуючого персоналу мережі.

Пріоритети серед перерахованих завдань інформаційної безпеки визначаються індивідуально для кожної конкретної ІТС та залежать від вимог, що пред'являються безпосередньо до інформаційних систем.

Слід врахувати, що з точки зору державних структур заходи щодо захисту в першу чергу покликані забезпечити конфіденційність, цілісність і доступність інформації. Зрозуміло, що для режимних державних організацій на першому місці завжди стоїть конфіденційність відомостей, а цілісність розуміється виключно як їх незмінність. Комерційним структурам, ймовірно, найважливіше цілісність і доступність даних та послуг з їх обробці. У порівнянні з державними, комерційні

організації більш відкриті і динамічні, тому ймовірні загрози для них відрізняються не тільки кількістю, але і якістю.

Для вирішення завдання інформаційної безпеки в ІТС необхідно:

- захистити інформацію при її зберіганні, обробці та передачі по мережі;

- підтвердити справжність об'єктів даних і користувачів (Аутентифікація сторін, що встановлюють зв'язок);

- виявлення і попередження порушення цілісності об'єктів даних;

- захистити конфіденційну інформацію від витоку і від впроваджених електронних пристроїв знімання інформації;

- захистити програмні продукти від впровадження програмних закладок і вірусів;

- захистити від несанкціонованого доступу до інформаційних ресурсів і технічних засобів мережі, в тому числі і до засобів управління, щоб запобігти зниження рівня захищеності інформації і самої мережі в цілому;

- організувати потрібні заходи, спрямовані на забезпечення схоронності конфіденційних даних;

- захистити технічні пристрої і приміщення.

Конкретна реалізація загальних принципів забезпечення інформаційної безпеки може виражатися в організаційних або технічних заходів захисту інформації.

Слід зазначити, що обсяг заходів щодо захисту оброблюваних і переданих даних залежить, перш за все, від величини можливої шкоди. Ця величина може виражатися у прямій (Наприклад, витрати на покупку нового програмного забезпечення в разі порушення його цілісності) або в опосередкованій (Наприклад, витрати від простою інформаційної системи банку) формі. Правда, в деяких ситуаціях розрахувати величину збитку важко (наприклад, випадку витоку державної таємниці).

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Теоретичні відомості з теми «Кодування текстів шифром Гронсфельда»

Шифри складної заміни

Шифри складної заміни називають багатоалфавітними, так як для шифрування кожного символу вихідного повідомлення застосовують свій шифр простої заміни. Багатоалфавітна підстановка послідовно і циклічно змінює використовувані алфавіти.

При r-алфавітній підстановці символ x_0 вихідного повідомлення замінюється символом y_0 з алфавіту B_0, символ x_1 - символом y_1 з алфавіту B_1, і так далі, символ x_{r-1} замінюється символом y_{r-1} з алфавіту B_{r-1}, символ x_r замінюється символом y_r знову з алфавіту B_0, і т.д.

Загальна схема багатоалфавітної підстановки для випадку r = 4 показана в таблиці 3.1.

Таблиця 3.1 Загальна схема багатоалфавітної підстановки

Вхідний символ:	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
Алфавіт підстановки:	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0	B_1

Схема r-алфавітної підстановки для випадку r = 4.

Ефект використання багатоалфавітної підстановки полягає в тому, що забезпечується маскування природної статистики вихідного мови, так як конкретний символ з вихідного алфавіту А може бути перетворений в кілька різних символів шифрувальних алфавітів B_j. Ступінь забезпечується захисту теоретично пропорційна довжині періоду г в послідовності використовуваних алфавітів B_j.

Багатоалфавітні шифри заміни запропонував і ввів в практику криптографії Леон Батист Альберті, який також був відомим архітектором і теоретиком мистецтва. Його книга "Трактат про шифр", написана в 1566 р, представляла собою

перший в Європі наукова праця по криптології. Крім шифру багатоалфавітної заміни, Альберті також детально описав пристрої з обертових коліс для його реалізації. Криптологи усього світу шанують Л. Альберті основоположником криптології.

Шифр Гронсфельда

Цей шифр складної заміни, званий шифром Гронсфельда, є модифікацією шифру Цезаря числовим ключем. Для цього під буквами вихідного повідомлення записують цифри числового ключа. Якщо ключ коротше повідомлення, то його запис циклічно повторюють. Шифртекст отримують приблизно, як в шифрі Цезаря, але відраховують за алфавітом на три букви (як це робиться в шифрі Цезаря), а вибирають ту букву, яка зміщена за алфавітом на відповідну цифру ключа. Наприклад, застосовуючи в якості ключа групу з чотирьох початкових цифр числа e (підстави натуральних логарифмів), а саме 2718, отримуємо для вихідного повідомлення ВОСТОЧНЫЙ ЭКСПРЕСС наступний шифртекст (таблиця 3.2):

Таблиця 3.2 Шифрування Гронсфельдом

Повідомлення	В	О	С	Т	О	Ч	Н	Ы	Й		Э	К	С	П	Р	Е	С	С
Ключ	2	7	1	8	2	7	1	8	2		7	1	8	2	7	1	8	2
Шифртекст	Д	Х	Т	Ь	Р	Ю	О	Г	Л		Д	Л	Щ	С	Ч	Ж	Щ	У

Щоб зашифрувати першу букву повідомлення В, використовуючи першу цифру ключа 2, потрібно відрахувати другу по порядку букву від В в алфавіті (таблиця 3.3) виходить перша буква шифртекста Д.

Таблиця 3.3 Приклад шифрування

В	Г	Д
	1	2

Слід зазначити, що шифр Гронсфельда розкривається відносно легко, якщо врахувати, що в числовому ключі кожна цифра має тільки десять значень, а значить,

є лише десять варіантів прочитання кожної букви шифртекста. З іншого боку, шифр Гронсфельда допускає подальші модифікації, що поліпшують його стійкість, зокрема подвійне шифрування різними числовими ключами.

Шифр Гронсфельда є по суті окремий випадок системи шифрування Вижинера.

3.2 Приклади застосування

Приклад 1. Припустимо, ми хочемо зашифрувати слово «ТАЙНА», використовуючи ключ «103». Записуємо циклічно під словом ТАЙНЫ наш ключ, після чого зрушуємо за алфавітом кожну букву на стільки букв вперед, скільки вказано нижче, отримаємо (таблиця 3.4):

Таблиця 3.4 Розв’язання прикладу 1.

Т	А	Й	Н	А
1	0	3	1	0
У	А	М	О	А

Приклад 2. Є англійські символи + та пробіл, зашифруємо SOURCE CODE (переклад - «вихідний код»). Перші 6 символів шифрованого тексту як і раніше будуть TSWTDI, як в прикладі з слово SOURCE і алфавітом без пробілу. При цьому ми застосували один раз всі цифри ключа 1422, також довелося вдруге задіяти 1 і 4.

Далі за алгоритмом шифру Гронсфельда задіємо двійку. На черзі пробіл, він 26-ий, якщо A - символ номер 0. (26 + 2) mod 27 = 1, тобто замість пробілу ставимо B. Тепер C і друга двійка в ключі. Якщо A - номер 0, то C - номер 2. (2 + 2) mod 27 = 4, тобто це E. Далі шифруємо O, всі цифри ключа використані, знову починаємо зі старшою (найлівішій) цифри, тобто потрібен зсув на 1, замість O буде P. І так далі (таблиця 3.5).

Таблиця 3.5 Вихідні дані до прикладу 2

Повідомлення	S	O	U	R	C	E	_	C	O	D	E
Ключ	1	4	2	2	1	4	2	2	1	4	2
Шифрування	T	S	W	T	D	I	B	E	P	H	G

Приклад 3. Тепер знову англійська з пропуском, але текст: MY FAT CAT, ключ: 143 (таблиця 3.6).

Таблиця 3.6 Вихідні дані прикладу 3

Повідомлення	M	Y	_	F	A	T	_	C	A	T
Ключ	1	4	3	1	4	3	1	4	3	1
Шифрування	N	B	C	G	E	W	A	G	D	U

Даний приклад застосування шифру Гронсфельда - яскрава демонстрація того, як шифри складної заміни затирають статистику входження символів у відкритий текст. Якби у нас був шифр простої заміни, то однакові символи замінилися б однаково. Особливо негативно це позначилося б на прогалинах: дуже часто зустрічається символ шифрованого тексту - напевно «маска» пробілу. За шифру Гронсфельда в нашому прикладі прогалини замінені по-різному.

3.3 Алгоритмізація за темою роботи

Після запуску програмного забезпечення, перед користувачем відкривається вікно, на якому відображено інтерфейс програми. Інтерфейс програми містить: поле для введення повідомлення (на англійській мові), поле для введення ключа та поле з результатом шифрування або дешифрування.

Крок 1. Користувачеві необхідно ввести повідомлення (англійською мовою), яке буде в подальшому зашифроване або дешифроване.

Після того як введено повідомлення перехід до кроку 2.

Крок 2. Користувач необхідно ввести ключ для шифрування.

Крок 3. Обрати зашифрувати чи дешифрувати повідомлення.

Крок 4. Отримати результат шифрування або дешифрування.

3.4 Розробка блок-схеми, яка підлягає програмуванню

Для детальнішого огляду програмного забезпечення з теми «Кодування текстів шифром Гронсфельда», розроблено блок-схему роботи.

Робота програмного забезпечення розпочинається з введення повідомлення для шифрування або дешифрування. Далі вводиться ключ та отримується результат. Блок-схема алгоритму зображена на рисунку 3.1.

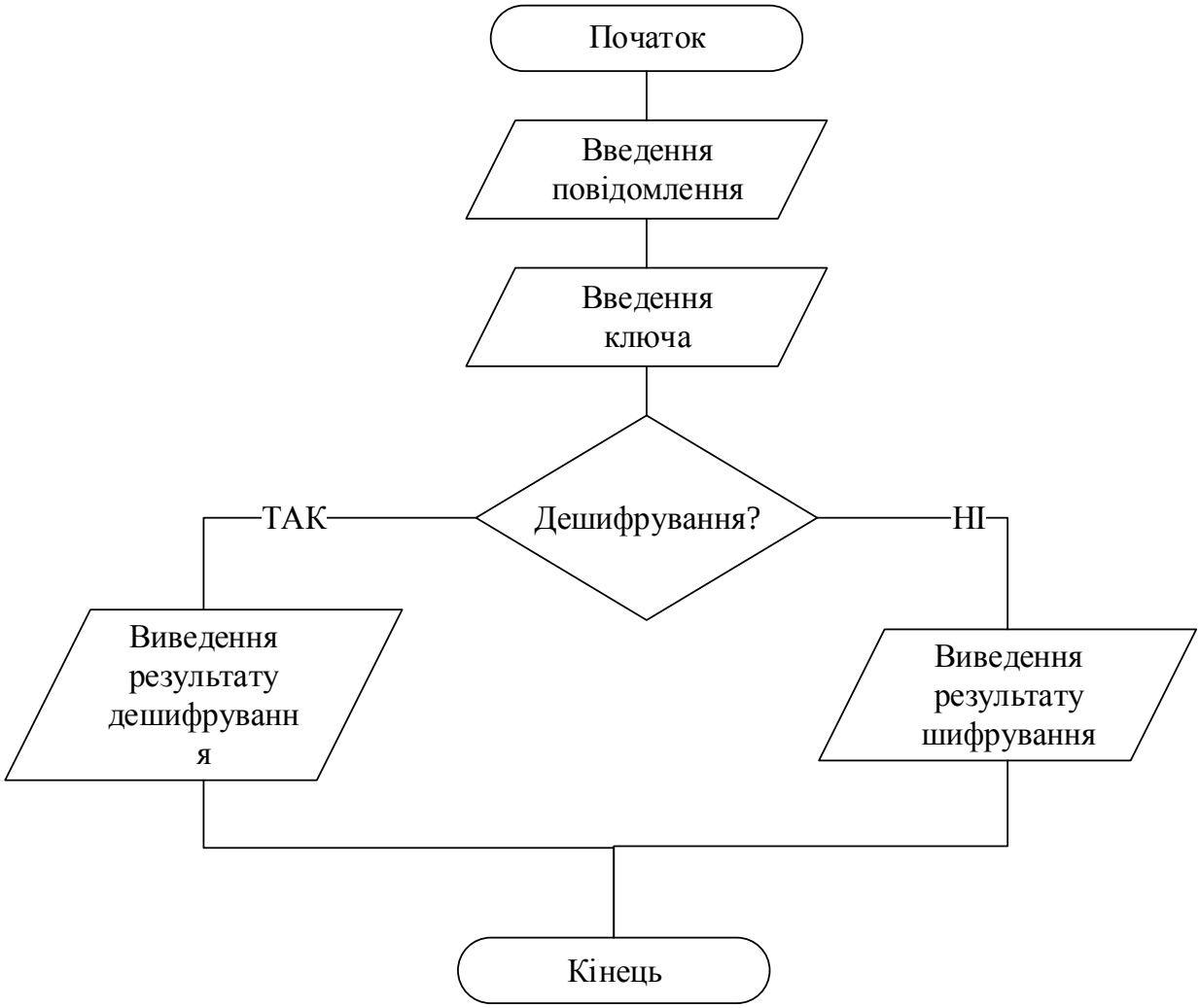

Рисунок 3.1 – Блок-схема алгоритму

4 ПРАКТИЧНА ЧАСТИНА

4.1 Обґрунтування вибору програмних засобів

.NET Framework - програмна платформа, випущена компанією Microsoft в 2002 році. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), яке підходить для різних мов програмування. Функціональні можливості CLR доступні в будь-яких мовах програмування, що використовують це середовище.

Основною ідеєю при розробці .NET Framework було забезпечення свободи розробника за рахунок надання йому можливості створювати додатки різних типів, здатні виконуватися на різних типах пристроїв і в різних середовищах. Другим принципом стала орієнтація на системи, що працюють під управлінням сімейства операційних систем Microsoft Windows.

Програма для .NET Framework, написана будь-якою мовою програмування, що підтримується, спочатку перекладається компілятором в єдиний для .NET проміжний байт-код Common Intermediate Language (CIL) (раніше називався Microsoft Intermediate Language, MSIL). У термінах .NET виходить збірка, англ. assembly. Потім код або виконується віртуальною машиною Common Language Runtime (CLR), або транслюється утилітою NGen.exe в виконуваний код для конкретного цільового процесора. Використання віртуальної машини є переважним, оскільки позбавляє розробників необхідності піклуватися про особливості апаратної частини. У разі використання віртуальної машини CLR вбудований в неї JIT-компілятор «на льоту» (just in time) перетворює проміжний байт-код в машинні коди потрібного процесора. Сучасна технологія динамічної компіляції дозволяє досягти високого рівня швидкодії. Віртуальна машина CLR також сама піклується про базову безпеку, управління пам'яттю та систему винятків, виконуючи частину роботи за розробника.

Архітектура .NET Framework описана і опублікована в специфікації Common Language Infrastructure (CLI), яка розроблена Microsoft та затверджена 39 ISO і

ECMA. У CLI описані типи даних .NET, формат метаданих про структуру програми, система виконання байт-коду і багато іншого [10].

.NET є багаторівневим, модульним і ієрархічним. Кожен рівень .NET Framework є шаром абстракції. Мови .NET - це верхній і найбільш абстрагований від інших рівень. Common Language Runtime (CLR) - це нижній рівень, найменш абстрагований і найбільш близький до вихідного середовища. Це важливо, тому що CLR тісно пов'язана з операційним середовищем для управління додатками. .NET Framework розділений на модулі, кожен з яких має свою особливу сферу відповідальності. Оскільки вищі рівні запитують служби тільки з більш низьких рівнів, .NET є ієрархічним. Загальна архітектура .NET Framework показана на рисунку 4.1.

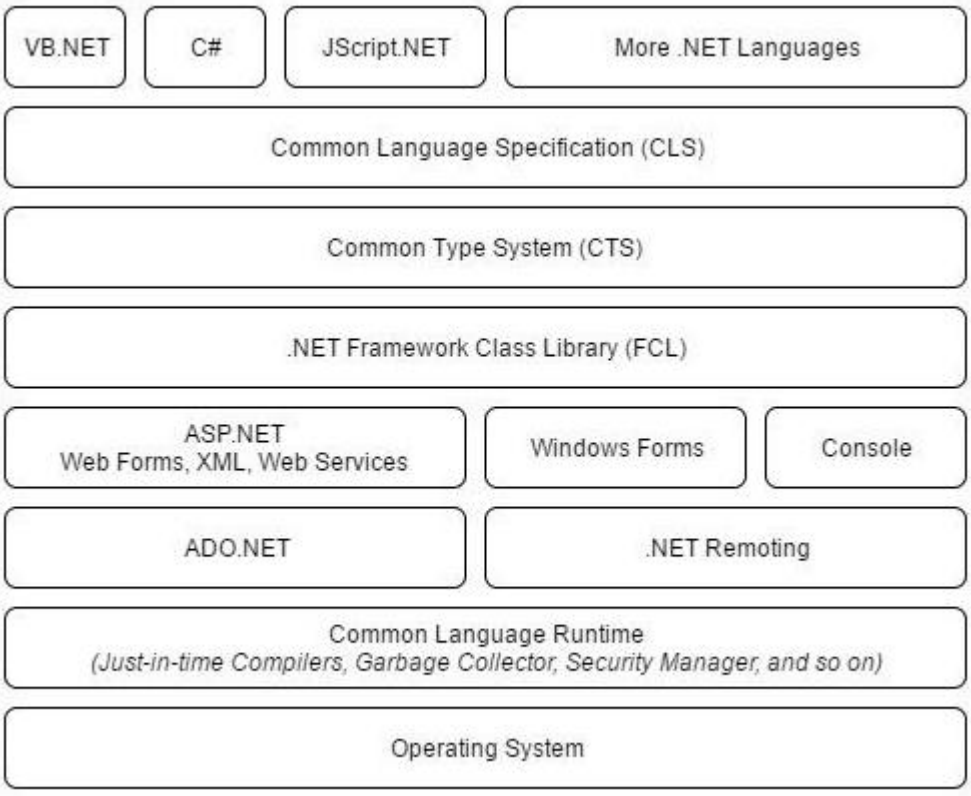

Рисунок 4.1 – Архітектора платформи .NET

Об'єктні класи .NET, доступні для всіх підтримуваних мов програмування, містяться в бібліотеці Framework Class Library (FCL). У FCL входять класи 40

Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation та інші (рис. 4.2). Ядро FCL називається Base Class Library (BCL).

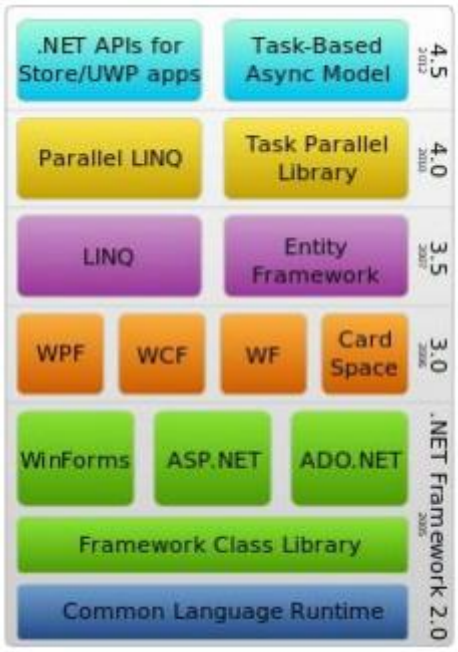

Рисунок 4.2 – Стек технологій .NET Framework [10]

4.2 Опис процесу програмної реалізації

Першим кроком програмної реалізації було розроблення графічного інтерфейсу (рисунок 4.3).

Так на форму було додано наступні елементи:

- поля з інформацією;
- поля для введення даних;
- кнопки для отримання результатів.

Другим кроком створення програмного забезпечення було підключення бібліотеки для подальшого використання всіх елементів. `#include <string>`

```
#include <iostream>
#include <string.h>
```

```
#include <cstdlib>
#include <conio.h>
#include "windows.h"
#include <string>
```

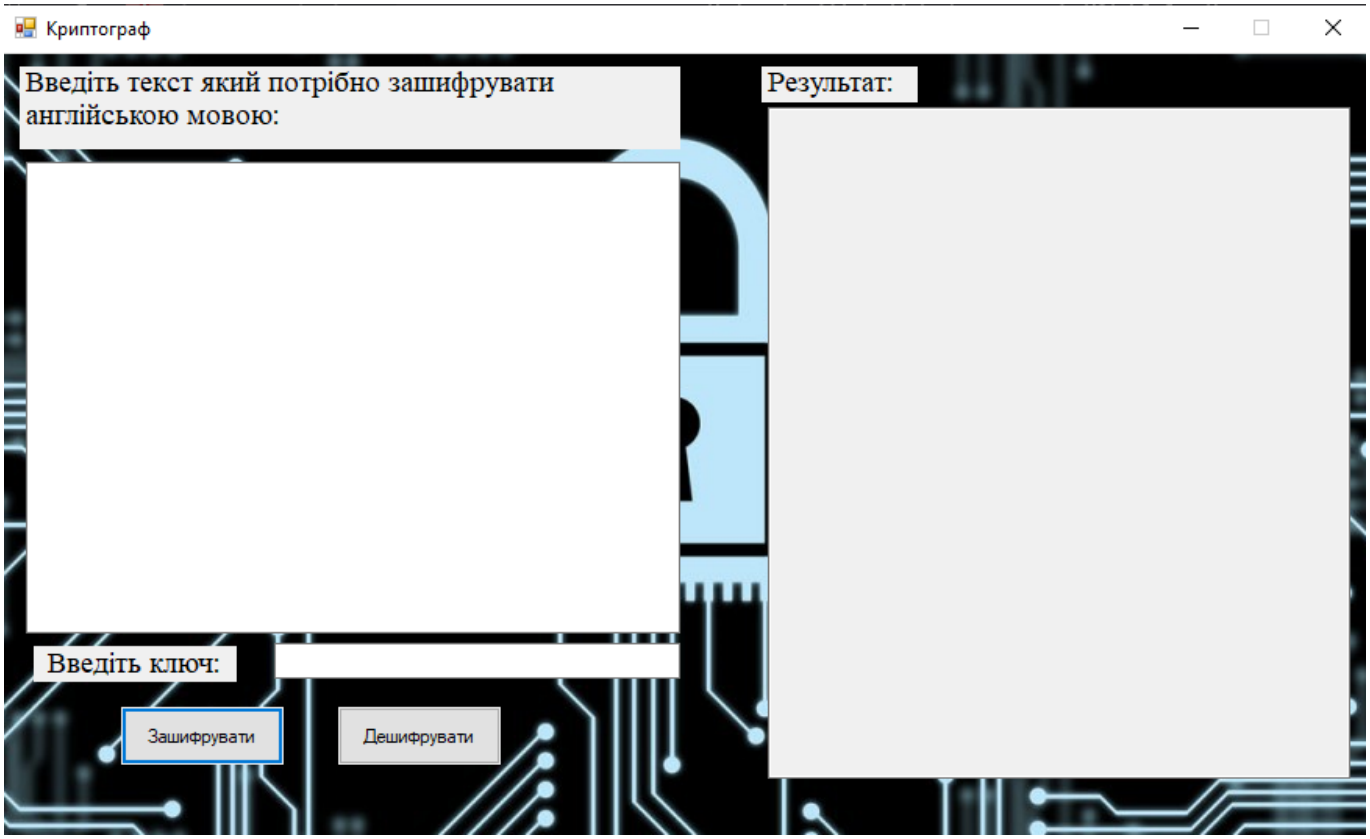

Рисунок 4.3 – Представлення графічного інтерфейсу програмної реалізації

Ініціалізація і створення форми відбувається за допомогою наступного коду.

```
void InitializeComponent(void)
        {
            System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(MyForm::typeid));
            this->button1 = (gcnew System::Windows::Forms::Button());
            this->button2 = (gcnew System::Windows::Forms::Button());
            this->textBox1 = (gcnew System::Windows::Forms::TextBox());
```

```
this->textBox2 = (gcnew System::Windows::Forms::TextBox());
this->label1 = (gcnew System::Windows::Forms::Label());
this->textBox3 = (gcnew System::Windows::Forms::TextBox());
this->label2 = (gcnew System::Windows::Forms::Label());
this->label3 = (gcnew System::Windows::Forms::Label());
this->label4 = (gcnew System::Windows::Forms::Label());
this->SuspendLayout();
//
// button1
//
this->button1->Location = System::Drawing::Point(77, 419);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(104, 37);
this->button1->TabIndex = 0;
this->button1->Text = L"Зашифрувати";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
    // textBox1
                    //
this->textBox1->AccessibleDescription = L"";
this->textBox1->AccessibleName = L"qwe";
this->textBox1->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
this->textBox1->Location = System::Drawing::Point(16, 70);
this->textBox1->MinimumSize = System::Drawing::Size(419,
302);
this->textBox1->Multiline = true;
```

```
                this->textBox1->Name = L"textBox1";
                this->textBox1->Size = System::Drawing::Size(419, 302);
                this->textBox1->TabIndex = 2;
                this->textBox1->Tag = L"";
                this->textBox1->TextChanged += gcnew
System::EventHandler(this, &MyForm::textBox1_TextChanged);
        // label1
                //
                this->label1->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 14.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(204)));
                this->label1->Location = System::Drawing::Point(12, 9);
                this->label1->Name = L"label1";
                this->label1->Size = System::Drawing::Size(423, 53);
                this->label1->TabIndex = 4;
                this->label1->Text = L"Введіть текст який потрібно
зашифрувати англійською мовою:";
                this->label1->Click += gcnew System::EventHandler(this,
&MyForm::label1_Click);
        //
                // MyForm
                //
                this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                this->ClientSize = System::Drawing::Size(876, 501);
                this->Controls->Add(this->label3);
                this->Controls->Add(this->label2);
                this->Controls->Add(this->textBox3);
```

```
this->Controls->Add(this->label1);
this->Controls->Add(this->textBox2);
this->Controls->Add(this->textBox1);
this->Controls->Add(this->button2);
this->Controls->Add(this->button1);
this->Controls->Add(this->label4);
this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
this->MaximizeBox = false;
this->Name = L"MyForm";
this->Text = L"Криптограф";
this->ResumeLayout(false);
this->PerformLayout();
}
```

Оголошення змінних:

```
System::String^ B;
System::String^ C;
System::String^ D;
B = MyForm::textBox1->Text;
C = MyForm::textBox3->Text;
int b = B->Length;
int c = C->Length;
```

Наступник кроком було розроблено перевірку повідомлення з довжиною ключа.

```
if (b >= c)
{
for (int i = 0; i < (b / c); i++)
{
D = D + C;
}
```

```
            for (int j = 0; j < (b % c); j++)
            {
                D = D + C[j];
            }
        }
        else
        {
            D = C;
        }
```

Виведення результатів шифрування або дешифрування реалізовано наступним чином.

```
for (int i = 0; i < b; i++)
            {
                Bb.push_back(char((B[i]) + (D[i] - 48)));
            }
            String^ Bbb = gcnew String(Bb.c_str());
            MyForm::textBox2->Text = System::Convert::ToString(Bbb);
```

Щоб елементи керування функціонували потрібним чином для кожної було створено відповідну подію. Подія button1_Click(System::Object^ sender, System::EventArgs^ e) спрацьовує при натисненні на першу кнопку (Зашифрувати) і виводить результат шифрування повідомлення з відповідним ключем. Подія button2_Click(System::Object^ sender, System::EventArgs^ e) спрацьовує при натисненні на другу кнопку (Дешифрувати) і виводить результат дешифрування відповідного повідомлення з ключем.

4.3 Опис програми

Продемонструємо роботу програмного забезпечення на наступному прикладі 1 з пункту 3.2. На рисунку 4.4 представлено шифрування, а на рисунку 4.5 – дешифрування.

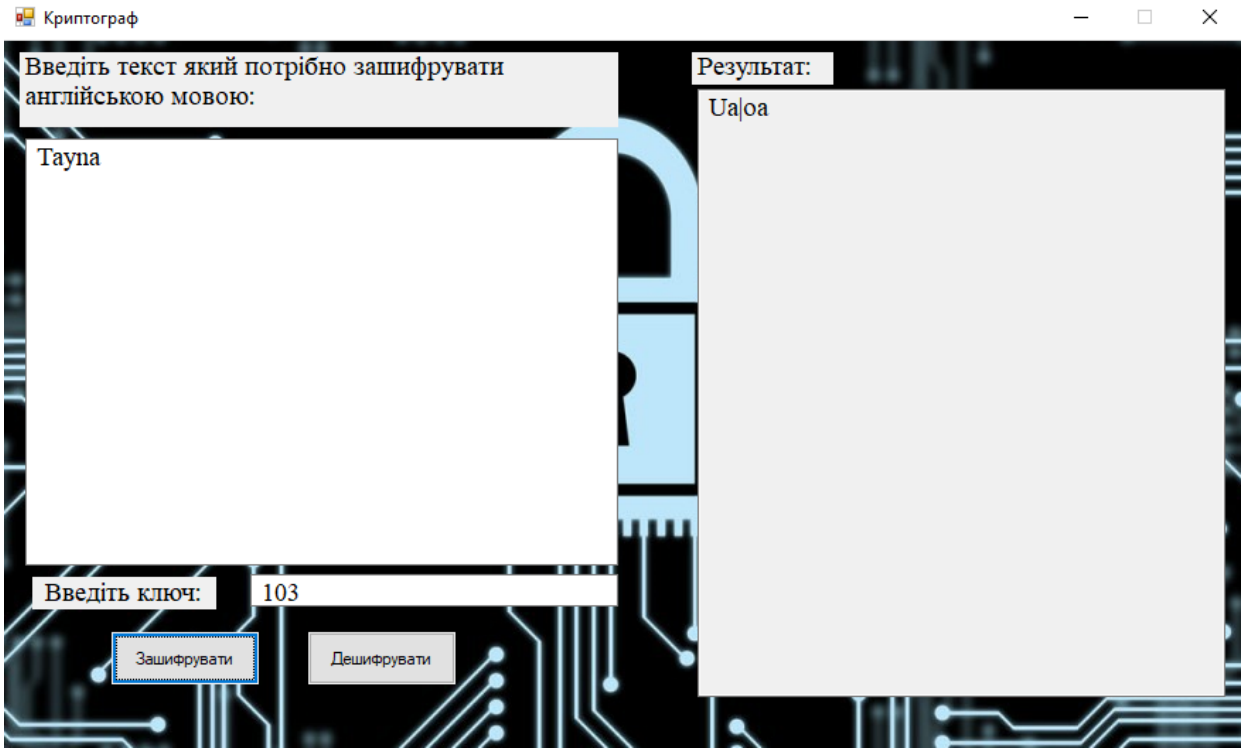

Рисунок 4.4 – Шифрування прикладу 1

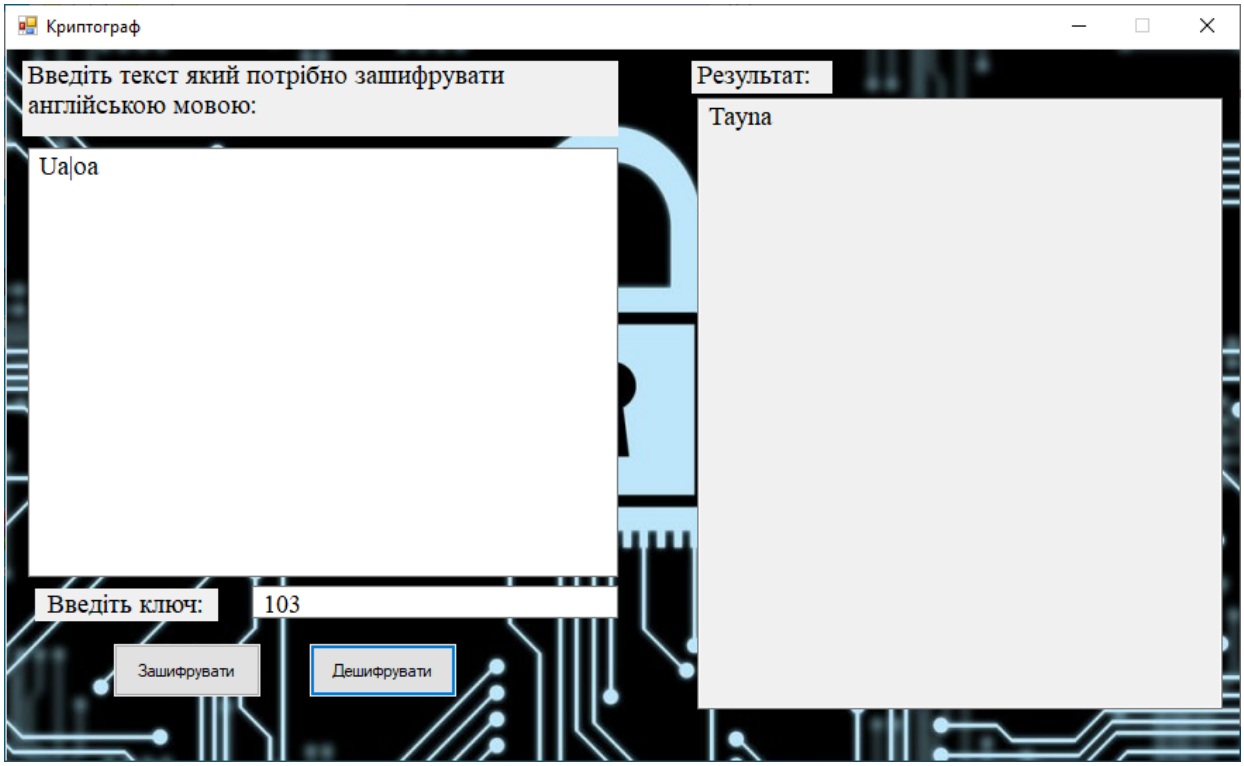

Рисунок 4.5 – Дешифрування прикладу 1

На рисунку 4.6 продемонстровано шифрування прикладу 2 з пункту 3.2. Рисунок 4.7 – містить інформацію про дешифрування прикладу 2.

На рисунку 4.8 – шифрування прикладу 3 з пункту 3.2, а на рис. 4.9 – дешифрування прикладу 3.

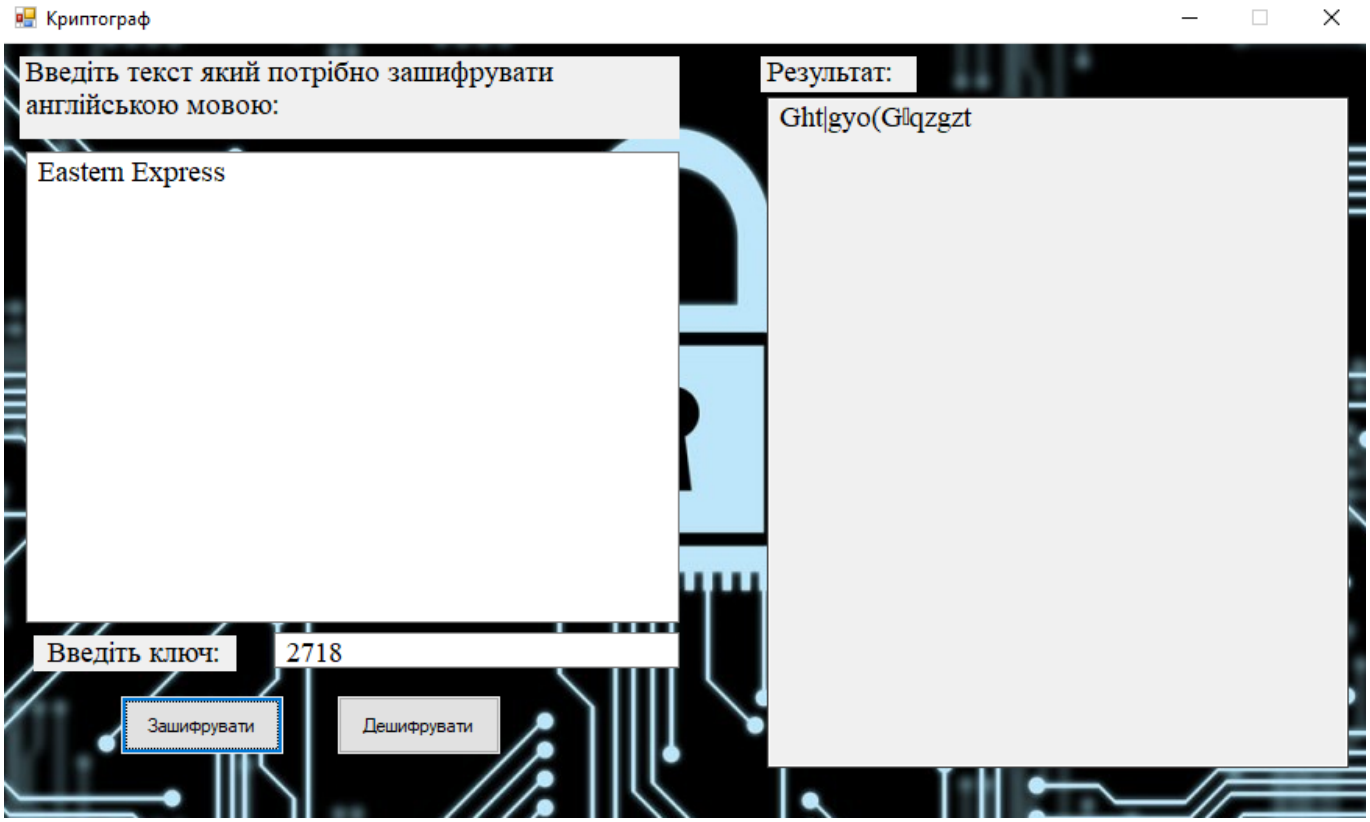

Рисунок 4.6 – Шифрування прикладу 2

4.4 Необхідна користувачу програми інструкція

Спочатку необхідно відкрити .exe файл для запуску програмного забезечення. Після того як програма була запущена, перед користувачем з’являється графічний інтерфейс (рисунок 4.3).

Для того, щоб програма видала результат, необхідно заповнити поле з повідомленням (рисунок 4.10). Також необхідно заповнити поле з ключем (рисунок 4.11).

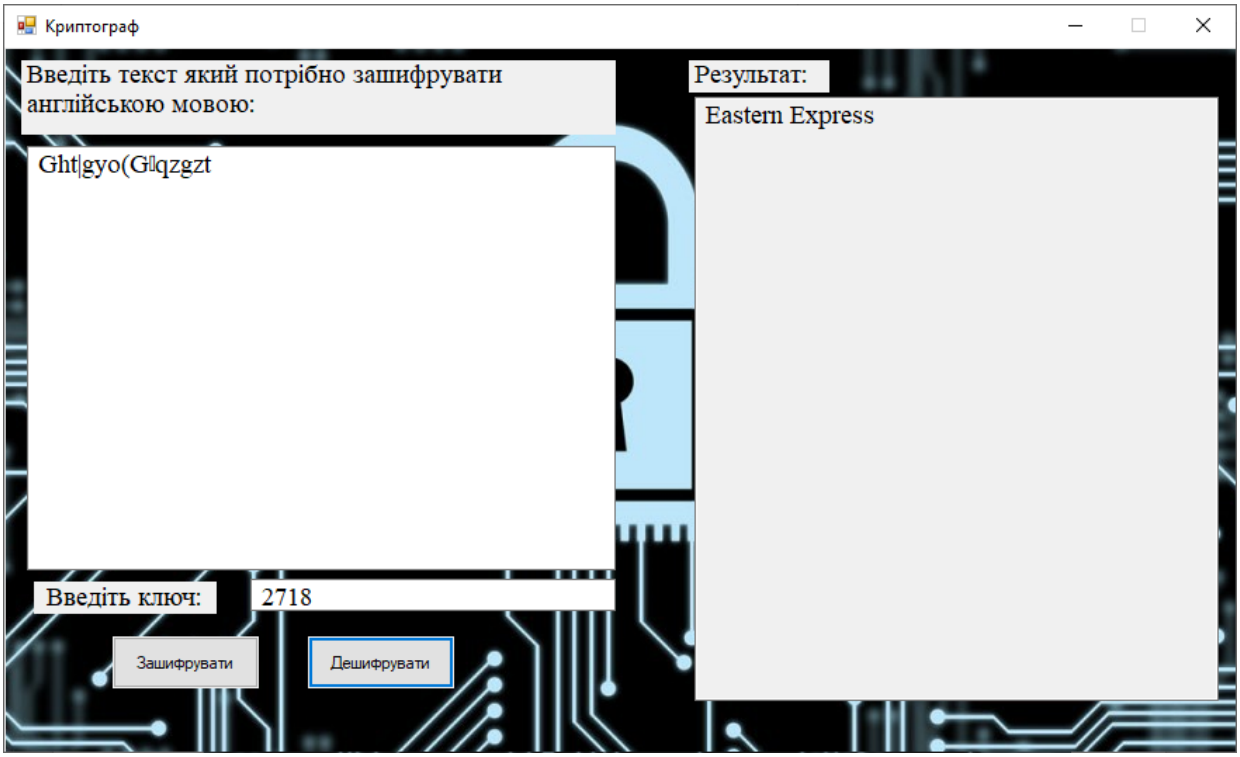

Рисунок 4.7 – Дешифрування прикладу 2

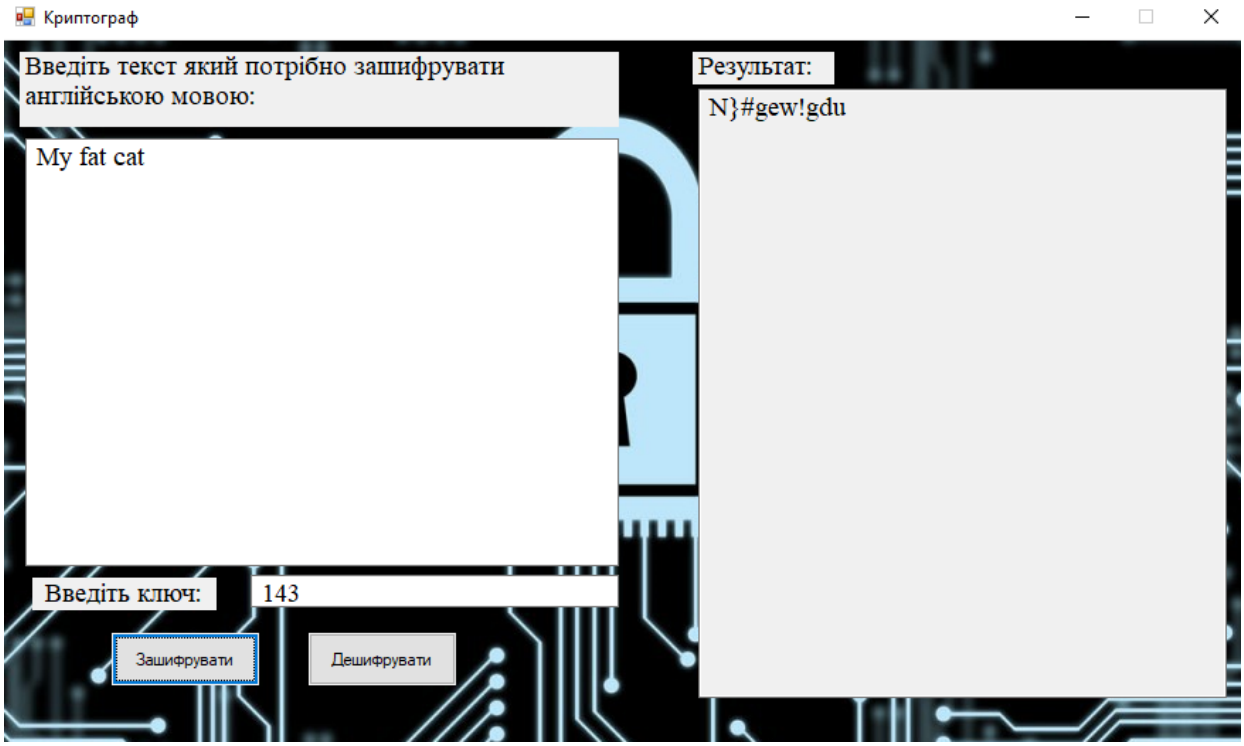

Рисунок 4.8 – Шифрування прикладу 3

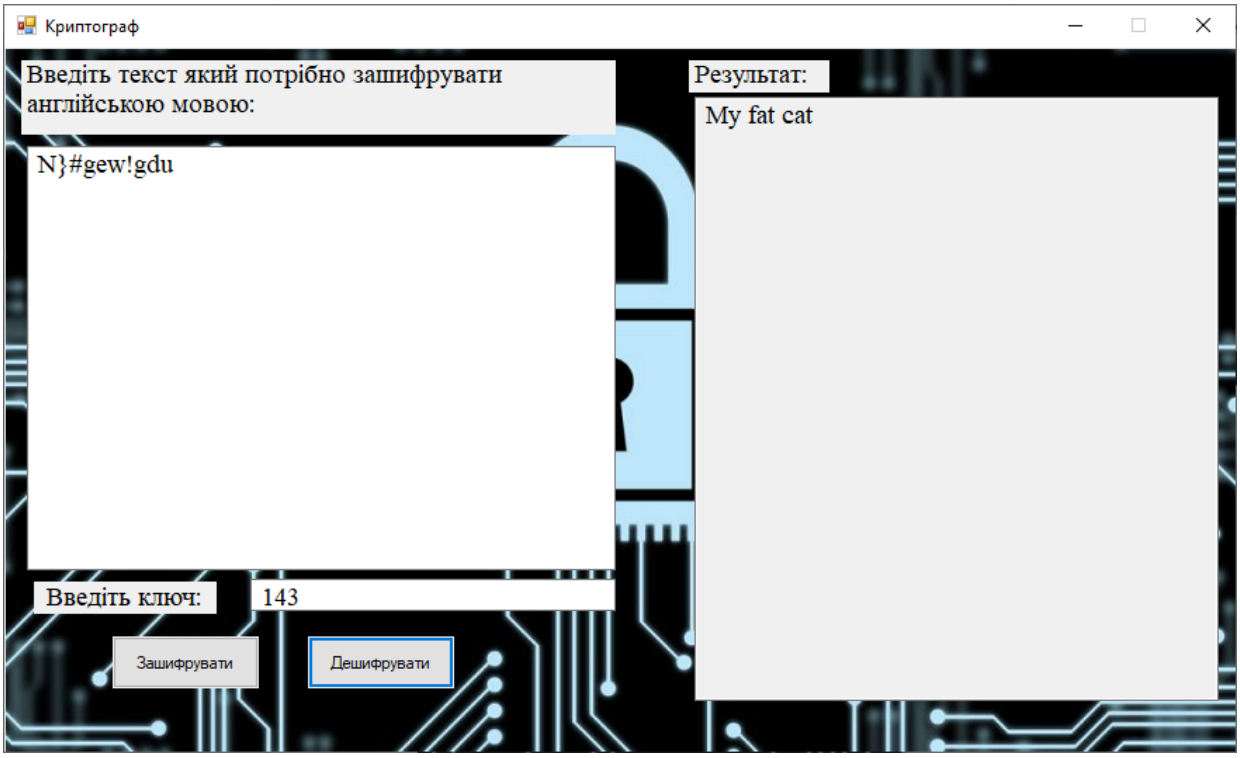

Рисунок 4.9 – Дешифрування прикладу 3

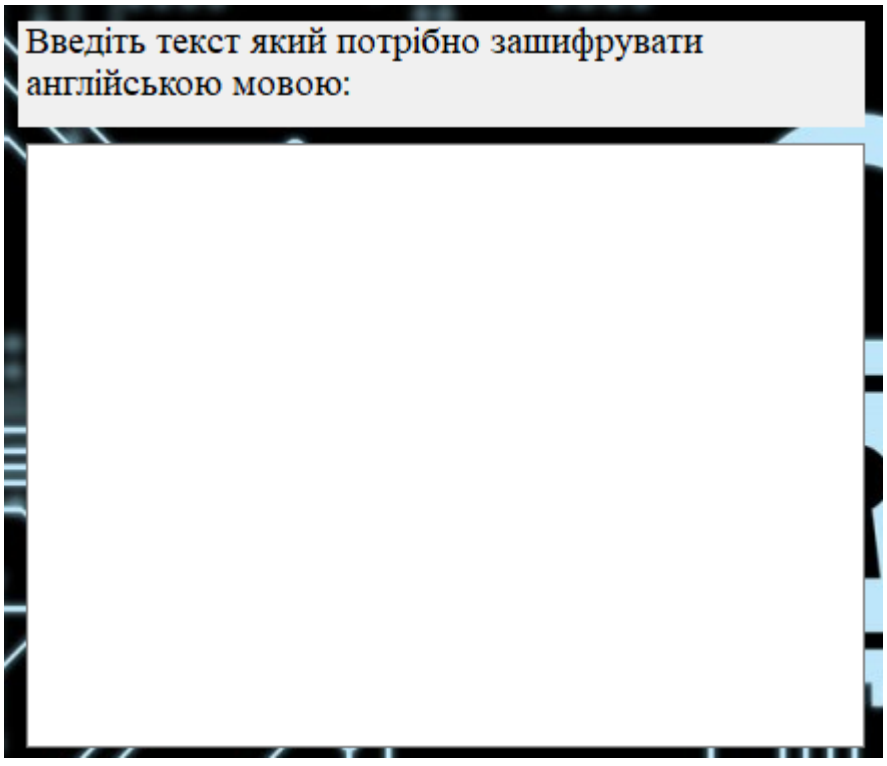

Рисунок 4.10 – Поле для ведення повідомлення

Рисунок 4.11 – Поле для введення ключа

Після того як дані введені необхідно обрати зашифрувати їх чи дешифрувати (рисунок 4.12).

Рисунок 4.12 – Кнопки для вибору

Після вибору, отримуємо результат шифрування або дешифрування. Дані результати виводяться в поле, яке представлене на рисунку 4.13.

Рисунок 4.14 – Поле з виводом результатів

ВИСНОВКИ

В результаті роботи виконані поставлені задачі:

- Розглянуто актуальність та призначення дистанційних платформ.
- Оглянуто навчальні тренажери, виокремлено переваги та вади.
- Опрацьовано навчальну літературу з теми «Кодування текстів шифром Гронсфельда».
- Розроблено алгоритм програмного засобу.
- Програмного реалізований програмне забезпечення з теми «Кодування текстів шифром Гронсфельда».
- Обґрунтовано вибір програмного забезпечення.
- Описано процес програмної реалізації.
- Описано інструкцію, щодо використання програмного засобу

До розглянутих навчальних тренажерів відносяться тренажери з дисциплін «Методи оптимізації та дослідження операцій», «Математичний аналіз» та «Алгебра і геометрія»:

- ♦ навчальний тренажер з теми «Градієнтний метод»;
- ♦ навчальний тренажер з теми «Розкриття найпростіших невизначеностей»;
- ♦ навчальний тренажер з теми «Диференційне числення функції з однієї змінної»;
- ♦ навчальний тренажер з теми «Матриця і визначники».

В алгоритмі описано по-кроково всі дії при кодуванні, що допоможе студентові закодувати або дешифрувати необхідний текст шифром Гронсфельда.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки», спеціальності 122 «Комп'ютерні науки та інформаційні технології» / О. О. Ємець // – Полтава, Кафедра ММСІ ПУЕТ, 2017 – 71 с.

2. Дистанційний курс «Математичний аналіз частина 2 (2018)» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПІЛКИ «Полтавський університет економіки і торгівлі»». – Режим доступу: http://www2.el.puet.edu.ua/st/course/view.php?id=1053

3. Потерайло О. О. Алгоритмізація тренажеру з теми «Градієнтний метод» дистанційного курсу «Методи оптимізації та дослідження операцій» / О. О. Потерайло // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10-12 березня 2016 р.). – Полтава: ПУЕТ, 2016. Режим доступу: http://dspace.puet.edu.ua/handle/123456789/2948

4. Дистанційний курс «Математичний аналіз частина 1 (2018)» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПІЛКИ «Полтавський університет економіки і торгівлі»». – Режим доступу: http://www2.el.puet.edu.ua/st/course/view.php?id=282

5. Макаренко Я.М. Тренажер з теми "Розкриття найпростіших невизначеностей" та розробка його програмного забезпечення з дистанційного навчального курсу "Математичний аналіз" / Я.М. Макаренко // Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр-.наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 204-206. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/2837

6. Дистанційний курс «Алгебра і геометрія» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПІЛКИ «Полтавський університет економіки і торгівлі»».

7. Положення про дистанційне навчання (Затверджено Наказом Міністерства освіти і науки України від 25.04.2013 № 466).

8. С.О. Сисоєва, К.П. Осадча. Системи дистанційного навчання: порівняльний аналіз навчальних можливостей. - [Електронний ресурс]. – 2011. – Режим доступу: http://www.academia.edu/931578.

9. Математические и компьютерные основы криптологии : учебное пособие / .С. Харин, В.И. Берник, .В. Матвеев, С.В. Агиевич. – Минск: Новое издание, 2003. – 382 с.: ил.

10. .NET Framework [Електронний ресурс] – Режим доступу: 88 https://ru.wikipedia.org/wiki/.NET_Framework#.D0.90.D1.80.D1.85.D0.B8.D1.82.D0.B5.D0.BA.D1.82.D1.83.D1.80.D0.B0_.NET. Дата доступу: 19.05.17

ДОДАТОК А. КОД ПРОГРАМИ

```
#include <string>
#include <iostream>
#include <string.h>
#include <cstdlib>
#include <conio.h>
#include "windows.h"
#include <string>


using namespace std;

namespace Project6 {

    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::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label4;
protected:

private:
```

```
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        System::ComponentModel::Container^ components;

#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)
        {

    System::ComponentModel::ComponentResourceManager^
resources                    =                    (gcnew
System::ComponentModel::ComponentResourceManager(MyForm::typ
eid));
            this->button1             =             (gcnew
System::Windows::Forms::Button());
            this->button2             =             (gcnew
System::Windows::Forms::Button());
            this->textBox1             =             (gcnew
System::Windows::Forms::TextBox());
            this->textBox2             =             (gcnew
System::Windows::Forms::TextBox());
            this->label1             =             (gcnew
System::Windows::Forms::Label());
```

```
            this->textBox3 = (gcnew
System::Windows::Forms::TextBox());
            this->label2 = (gcnew
System::Windows::Forms::Label());
            this->label3 = (gcnew
System::Windows::Forms::Label());
            this->label4 = (gcnew
System::Windows::Forms::Label());
            this->SuspendLayout();
            //
            // button1
            //
            this->button1->Location =
System::Drawing::Point(77, 419);
            this->button1->Name = L"button1";
            this->button1->Size = System::Drawing::Size(104,
37);
            this->button1->TabIndex = 0;
            this->button1->Text = L"Зашифрувати";
            this->button1->UseVisualStyleBackColor = true;
            this->button1->Click += gcnew
System::EventHandler(this, &MyForm::button1_Click);
            //
            // button2
            //
            this->button2->Location =
System::Drawing::Point(216, 419);
            this->button2->Name = L"button2";
            this->button2->Size = System::Drawing::Size(104,
37);
```

```
            this->button2->TabIndex = 1;
            this->button2->Text = L"Дешифрувати";
            this->button2->UseVisualStyleBackColor = true;
            this->button2->Click          +=          gcnew
System::EventHandler(this, &MyForm::button2_Click);
            //
            // textBox1
            //
            this->textBox1->AccessibleDescription = L"";
            this->textBox1->AccessibleName = L"qwe";
            this->textBox1->Font          =          (gcnew
System::Drawing::Font(L"Times    New    Roman",    14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->textBox1->Location                      =
System::Drawing::Point(16, 70);
            this->textBox1->MinimumSize                   =
System::Drawing::Size(419, 302);
            this->textBox1->Multiline = true;
            this->textBox1->Name = L"textBox1";
            this->textBox1->Size                          =
System::Drawing::Size(419, 302);
            this->textBox1->TabIndex = 2;
            this->textBox1->Tag = L"";
            this->textBox1->TextChanged      +=      gcnew
System::EventHandler(this, &MyForm::textBox1_TextChanged);
            //
            // textBox2
            //
```

```
            this->textBox2->Font          =          (gcnew
System::Drawing::Font(L"Times    New    Roman",    14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->textBox2->Location                        =
System::Drawing::Point(491, 35);
            this->textBox2->Multiline = true;
            this->textBox2->Name = L"textBox2";
            this->textBox2->ReadOnly = true;
            this->textBox2->Size                            =
System::Drawing::Size(373, 430);
            this->textBox2->TabIndex = 3;
            this->textBox2->TextChanged      +=      gcnew
System::EventHandler(this, &MyForm::textBox2_TextChanged);
            //
            // label1
            //
            this->label1->Font          =          (gcnew
System::Drawing::Font(L"Times    New    Roman",    14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->label1->Location                          =
System::Drawing::Point(12, 9);
            this->label1->Name = L"label1";
            this->label1->Size  =  System::Drawing::Size(423,
53);
            this->label1->TabIndex = 4;
```

```
            this->label1->Text = L"Введіть текст який
потрібно зашифрувати англійською мовою:";
            this->label1->Click += gcnew
System::EventHandler(this, &MyForm::label1_Click);
            //
            // textBox3
            //
            this->textBox3->Font = (gcnew
System::Drawing::Font(L"Times New Roman", 14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->textBox3->Location =
System::Drawing::Point(175, 378);
            this->textBox3->Multiline = true;
            this->textBox3->Name = L"textBox3";
            this->textBox3->Size =
System::Drawing::Size(260, 23);
            this->textBox3->TabIndex = 5;
            //
            // label2
            //
            this->label2->Font = (gcnew
System::Drawing::Font(L"Times New Roman", 14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->label2->Location =
System::Drawing::Point(21, 380);
            this->label2->Name = L"label2";
```

```
            this->label2->Size = System::Drawing::Size(130,
23);
            this->label2->TabIndex = 6;
            this->label2->Text = L"Введіть ключ:";
            this->label2->TextAlign                          =
System::Drawing::ContentAlignment::MiddleCenter;
            this->label2->Click          +=          gcnew
System::EventHandler(this, &MyForm::label2_Click);
            //
            // label3
            //
            this->label3->Font           =           (gcnew
System::Drawing::Font(L"Times    New    Roman",    14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                static_cast<System::Byte>(204)));
            this->label3->Location                           =
System::Drawing::Point(487, 9);
            this->label3->Name = L"label3";
            this->label3->Size = System::Drawing::Size(100,
23);
            this->label3->TabIndex = 7;
            this->label3->Text = L"Результат:";
            this->label3->Click          +=          gcnew
System::EventHandler(this, &MyForm::label3_Click);
            //
            // label4
            //
            this->label4->BackColor                          =
System::Drawing::Color::DodgerBlue;
```

```
            this->label4->Image                                 =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"label4.Image")));
            this->label4->Location                              =
System::Drawing::Point(1, 1);
            this->label4->Name = L"label4";
            this->label4->Size  =  System::Drawing::Size(878,
504);
            this->label4->TabIndex = 8;
            //
            // MyForm
            //
            this->AutoScaleDimensions                           =
System::Drawing::SizeF(6, 13);
            this->AutoScaleMode                                 =
System::Windows::Forms::AutoScaleMode::Font;
            this->ClientSize   =   System::Drawing::Size(876,
501);
            this->Controls->Add(this->label3);
            this->Controls->Add(this->label2);
            this->Controls->Add(this->textBox3);
            this->Controls->Add(this->label1);
            this->Controls->Add(this->textBox2);
            this->Controls->Add(this->textBox1);
            this->Controls->Add(this->button2);
            this->Controls->Add(this->button1);
            this->Controls->Add(this->label4);
            this->FormBorderStyle                               =
System::Windows::Forms::FormBorderStyle::FixedSingle;
            this->MaximizeBox = false;
```

```
            this->Name = L"MyForm";
            this->Text = L"Криптограф";
            this->ResumeLayout(false);
            this->PerformLayout();

        }
#pragma endregion


    private:                                    System::Void
textBox2_TextChanged(System::Object^                    sender,
System::EventArgs^ e) {
    }
    private:    System::Void    button1_Click(System::Object^
sender, System::EventArgs^ e) {
        System::String^ B;
        System::String^ C;
        System::String^ D;
        B = MyForm::textBox1->Text;
        C = MyForm::textBox3->Text;
        int b = B->Length;
        int c = C->Length;
        //Первое условие. Если длина вводимого слова болше,
либо равна длине ключа
        if (b >= c)
        {
            for (int i = 0; i < (b / c); i++)
            {
                D = D + C;
            }
```

```
            for (int j = 0; j < (b % c); j++)
            {
                D = D + C[j];
            }
        }
        //Если длинна ключа больше
        else
        {
            D = C;
        }

        string Bb;

        //Цикл шифрования
        for (int i = 0; i < b; i++)
        {
            Bb.push_back(char((B[i]) + (D[i] - 48)));



        }

        String^ Bbb = gcnew String(Bb.c_str());
        MyForm::textBox2->Text                                    =
System::Convert::ToString(Bbb);

    }
    private:    System::Void    label1_Click(System::Object^
sender, System::EventArgs^ e) {
    }
```

```
    private:    System::Void    label2_Click(System::Object^
sender, System::EventArgs^ e) {
    }

    private:                                    System::Void
textBox1_TextChanged(System::Object^                    sender,
System::EventArgs^ e) {

    }
private:  System::Void  label3_Click(System::Object^  sender,
System::EventArgs^ e) {
}
private:  System::Void  button2_Click(System::Object^  sender,
System::EventArgs^ e) {
    System::String^ B;
    System::String^ C;
    System::String^ D;
    B = MyForm::textBox1->Text;
    C = MyForm::textBox3->Text;
    int b = B->Length;
    int c = C->Length;
    //Первое условие. Если длина вводимого слова болше, либо
равна длине ключа
    if (b >= c)
    {
        for (int i = 0; i < (b / c); i++)
        {
            D = D + C;
        }
        for (int j = 0; j < (b % c); j++)
```

```
        {
            D = D + C[j];
        }
    }
    //Если длинна ключа больше
    else
    {
        D = C;
    }

    string Bb;

    //Цикл дешифрования
    for (int i = 0; i < b; i++)
    {
        Bb.push_back(char((B[i]) - (D[i] - 48)));


    }

    String^ Bbb = gcnew String(Bb.c_str());
    MyForm::textBox2->Text = System::Convert::ToString(Bbb);

}
};

}
```