Київський національний університет імені Тараса Шевченка

**Лабораторна робота №1**

**З навчального курсу «Розробка ПЗ під мобільні платформи»**

Виконав:

студент 4 курсу

факультету кібернетики

спеціальність «Комп’ютерні науки»

групи ТТП-42

Божевський Іван Петрович

**Київ 2024**

# Зміст

# Опис завдання

Метою роботи є формування базових знань та умінь розробляти для мобільного пристрою програми обчислювального характеру, використовувати програмні засоби роботи з файлами та візуалізувати дані. За бажанням студента погодженням з боку викладача, зміст роботи може бути змінений.

Загальні вимоги:

- Має бути документ (DOCX, PDF) з описом вимог до програмного продукту (ТЗ)

- Програма повинна реалізовувати запис/читання даних з файлу або локальної БД

- Програма повинна мати пункт з інформацією про розробника та його/її фото

- Повинні бути застосовані базові принципи UX-дизайну та ергономіки для мобільних пристроїв

- Частиною результату виконання Проєкту 1 є звіт – текстовий документ, студент має відправити в Classroom і який включає:

* Титульну сторінку
* Зміст
* Опис завдання
* Короткий виклад теорії,
* Опис алгоритмів, які застосовуються для обчислень
* Опис роботи з програмою та скрін-шоти
* Список використаних джерел (мінімум 2), оформлений за ДСТУ
* Код основної частини програми у додатку(ах)

# Опис структури, змісту і вимог до Лабораторної роботи 1

## **Введення даних:**

Текст і ключ вводяться користувачем через форму або завантажуються з файлу.

## **Обробка даних:**

Шифрування та дешифрування тексту:

1. Шифр Цезаря: обчислення результату з використанням зсуву, що визначається за допомогою слайдера.
2. Шифр Віженера: обчислення результату з використанням ключа, який вводить користувач.

## **Виведення даних:**

Виведення зашифрованого/дешифрованого тексту після обробки.

## **Збереження даних:**

1. Можливість збереження зашифрованого або дешифрованого тексту у файл.
2. Можливість завантаження тексту для шифрування/дешифрування з файлу.

## **Графічна візуалізація:**

Відображення зсуву алфавіту (шифр Цезаря) на канвасі у вигляді літер, розміщених у квадратних полях. Алфавіт до і після зсуву відображається одночасно для наглядного порівняння.

# Реалізація читання та запису тексту з файлу

У додатку використовується сервіс **FileService** для виконання операцій читання та запису тексту з/у файл. Цей сервіс забезпечує дві основні функції: завантаження тексту з файлу та збереження вмісту до файлу в обраній користувачем директорії.

## **Читання тексту з файлу**

Метод **LoadFromFileAsync** відповідає за завантаження текстових даних з файлу, який обирає користувач через діалог вибору файлу. Операції виконуються асинхронно для забезпечення плавного користувацького досвіду без блокування інтерфейсу.

1. **Вибір файлу:** Метод використовує вбудований клас **FilePicker** для відображення діалогу вибору файлу. Для підтримки різних платформ, таких як Android, WinUI, MacCatalyst, iOS, налаштовується підтримка файлів з розширенням .txt або MIME-типами text/plain.
2. **Читання вмісту файлу:** Після вибору файлу, відкривається потік на читання через **OpenReadAsync**, і за допомогою **StreamReader** вміст файлу зчитується до кінця. Весь текст повертається як рядок.
3. **Обробка помилок:** У разі виникнення винятків, наприклад, якщо файл не був обраний або виникли інші помилки, вони обробляються через блок **try-catch** із виведенням повідомлення про помилку в консоль.

## **Запис тексту у файл**

Метод **SaveToFileAsync** виконує збереження вмісту до файлу, дозволяючи користувачу обрати директорію для збереження через діалог вибору папки.

1. **Вибір папки:** Метод використовує **FolderPicker** для вибору папки, в яку буде збережено файл. Користувач обирає папку за допомогою стандартного діалогу системи.
2. **Запис даних:** Після вибору папки, створюється файл з ім'ям, вказаним користувачем. За допомогою класу **StreamWriter**, вміст (текст) записується у файл асинхронно.
3. **Обробка помилок:** Як і у випадку з читанням, винятки обробляються за допомогою блоку **try-catch**. Якщо виникає помилка, користувач отримує відповідне повідомлення через **DisplayAlert**.

Завдяки цьому сервісу, користувач може зручно завантажувати текстові файли для подальшого шифрування/дешифрування, а також зберігати результат до файлу.



# Теоретичні відомості про шифр Цезаря та шифр Віженера

## Шифр Цезаря

**Шифр Цезаря** – це один із найпростіших і найстаріших методів симетричного шифрування. Його назва походить від імені римського імператора Гая Юлія Цезаря, який використовував цей метод для захисту своїх військових повідомлень. Шифр полягає в циклічному зміщенні кожної літери алфавіту на певну кількість позицій.

## Основні принципи роботи:

1. **Алфавіт**: Для шифру Цезаря використовується фіксований алфавіт (наприклад, український або англійський). Всі літери цього алфавіту розглядаються як послідовність символів.
2. **Ключ**: Ключом шифру є ціле число, яке визначає кількість позицій, на яку кожна літера тексту зміщується в алфавіті.
3. **Шифрування**: Кожна літера тексту замінюється на ту, що розташована через вказану кількість позицій далі в алфавіті. Якщо зміщення виходить за межі алфавіту, воно повторюється циклічно з початку.
4. **Дешифрування**: Для розшифрування повідомлення виконується зворотнє зміщення на ту ж кількість позицій.

## Формула шифрування:

де:

* C – зашифрований символ,
* P – індекс символу відкритого тексту,
* K – ключ (зсув),
* N – кількість символів в алфавіті.

## Формула дешифрування:

Де P – розшифрований символ.

## Приклад:

Припустимо, використовується український алфавіт і ключ K=3. Для слова «СЛОВО» після шифрування отримаємо «УНСҐУ».

## Переваги шифру Цезаря:

* Простота реалізації.
* Швидкість шифрування та дешифрування.

## Недоліки:

* Низька стійкість до атак, оскільки ключ легко підібрати методом перебору (всього N можливих варіантів).
* Вразливий до частотного аналізу, оскільки кожна літера шифрується однаково.

## Шифр Віженера

**Шифр Віженера** – це більш складний метод симетричного шифрування, який розв'язує деякі недоліки шифру Цезаря. Вперше запропонований французьким криптографом Блезом де Віженером у XVI столітті, цей шифр використовує кілька різних зсувів для шифрування кожної літери тексту, завдяки чому стає значно стійкішим до атак.

## Основні принципи роботи:

1. **Алфавіт**: Як і в шифрі Цезаря, використовується фіксований алфавіт.
2. **Ключ**: Ключ у шифрі Віженера складається з декількох літер. Цей ключ повторюється стільки разів, скільки потрібно, щоб відповідати довжині тексту.
3. **Шифрування**: Кожна літера тексту зміщується на кількість позицій, яка визначається відповідною літерою ключа. Кожна літера ключа може задавати свій окремий зсув.
4. **Дешифрування**: Для розшифрування використовується той самий ключ, де кожна літера тексту зміщується у зворотний бік.

## Формула шифрування:

де:

Ci – зашифрований символ,

Pi – індекс символу відкритого тексту,

Kj – індекс символу ключа (обирається циклічно),

N – кількість символів в алфавіті.

## Формула дешифрування:

## Переваги шифру Віженера:

* Вищий рівень криптостійкості у порівнянні з шифром Цезаря.
* Менш вразливий до частотного аналізу, оскільки одна і та ж літера може шифруватися по-різному залежно від позиції в тексті.

## Недоліки:

* Уразливий до більш складних атак, таких як атака Касіскі або атака за повторенням ключа.
* Довжина ключа має велике значення для надійності шифру.

# Опис реалізації шифрування у додатку

EncryptionService виконує основну функцію шифрування та дешифрування тексту за допомогою двох методів: **Шифр Цезаря** і **Шифр Віженера**. Цей сервіс реалізує вимогу шифрування і дешифрування введеного користувачем тексту.

## Шифр Цезаря

1. **CaesarEncrypt(text, shift)**: Шифрує введений текст, зміщуючи кожну букву на вказану кількість позицій у алфавіті. Шифр враховує специфіку українського алфавіту та циклічне зміщення букв.
2. **CaesarDecrypt(cipherText, shift)**: Виконує зворотнє зміщення для дешифрування тексту.

Алгоритм використовує метод **ShiftText**, який:

* Обробляє кожен символ тексту і зміщує його на вказану кількість позицій у алфавіті.
* Визначає, чи є символ літерою українського алфавіту, і проводить відповідне зміщення.

## Шифр Віженера

1. **VigenereEncrypt(text, key)**: Шифрує текст за допомогою ключа, де кожна літера ключа визначає зміщення для відповідної літери тексту.
2. **VigenereDecrypt(cipherText, key)**: Дешифрує текст, зворотно зміщуючи кожну літеру на відповідну кількість позицій, вказану ключем.

Алгоритм працює з використанням українського алфавіту, де кожен символ зміщується залежно від значення ключа. Сервіс також перевіряє наявність букви в алфавіті і пропускає символи, які не є літерами.

## Додаткові методи

* **GetTransformedCaesarAlphabet(shift)**: Повертає трансформований алфавіт після застосування шифру Цезаря зі зміщенням.
* **GetTransformedVignereAlphabet(key)**: Повертає трансформований алфавіт після застосування шифру Віженера з вказаним ключем.

Цей сервіс забезпечує реалізацію вимоги до шифрування і дешифрування тексту, а також надає можливості для візуалізації алфавітів до і після шифрування.



# Код сторінок та їх ViewModel

Кожна сторінка має свої відповідні ViewModel, що відповідають за бізнес-логіку та обробку взаємодії з користувачем.

**Сторінки**:

* **CaesarPage**: Сторінка для роботи з шифром Цезаря. Включає такі елементи як поле для введення тексту, слайдер для вибору зсуву, кнопки для шифрування та дешифрування, кнопки для завантаження та збереження файлів, а також візуалізація шифрованого алфавіту через SkiaSharp.
* **VigenerePage**: Сторінка для роботи з шифром Віженера. Має поле для введення тексту, поле для введення ключа, кнопки для шифрування та дешифрування, завантаження та збереження файлів, а також візуалізацію через SkiaSharp.

**ViewModel**:

* Кожна сторінка має свою ViewModel (CaesarPageVM та VigenerePageVM), яка реалізує логіку шифрування та дешифрування, роботу з файлами, а також ініціалізацію даних на сторінці. За допомогою команд RelayCommand забезпечується обробка натискань кнопок.
* У ViewModel інтегрована підтримка графіки, де за допомогою класів CaesarCipherDrawable та VignereCipherDrawable візуалізується шифрований алфавіт.

ViewModel працюють з сервісами EncryptionService для виконання шифрування та FileService для обробки файлів.



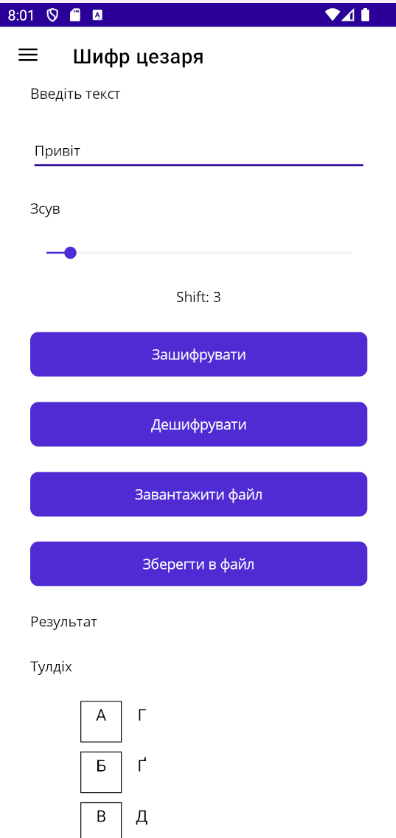


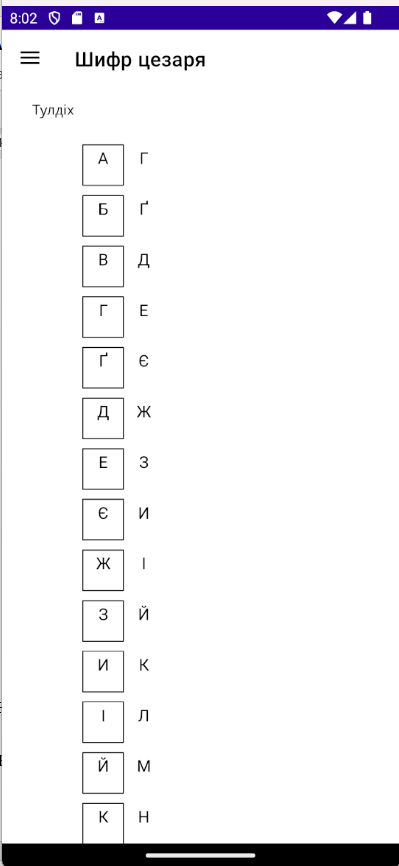
# Опис роботи програми

Ця програма дозволяє шифрувати та дешифрувати текст за допомогою двох класичних методів шифрування — **Шифр Цезаря** та **Шифр Віженера**. Крім того, програма надає можливість зберігати зашифрований текст у файл і завантажувати текст з файлу для подальшої обробки. Візуалізація результатів шифрування відображається на графічному полотні для кращого розуміння змін у шифрованому тексті.

## Основні сторінки програми

1. **Шифр Цезаря (Caesar Cipher Page)**:
   * Поле для введення тексту: Введіть текст, який необхідно зашифрувати або дешифрувати.
   * Слайдер для вибору зсуву: Виберіть величину зсуву для шифру Цезаря (від 0 до 33 для українського алфавіту).
   * Кнопки:
     + **Зашифрувати**: Натисніть для шифрування введеного тексту.
     + **Дешифрувати**: Натисніть для дешифрування зашифрованого тексту.
     + **Завантажити файл**: Завантажує текст з файлу для обробки.
     + **Зберегти в файл**: Зберігає результат шифрування або дешифрування у файл.
   * Відображення результату: Після натискання на кнопки шифрування/дешифрування результат буде відображено на сторінці.
   * **Візуалізація шифру**: У графічному полі відображається алфавіт з початковими літерами та їх зашифрованими еквівалентами.





1. **Шифр Віженера (Vigenere Cipher Page)**:
   * Поле для введення тексту: Введіть текст для шифрування або дешифрування.
   * Поле для введення ключа: Введіть ключ, який буде використано для шифру Віженера.
   * Кнопки:
     + **Зашифрувати**: Натисніть для шифрування тексту за допомогою ключа.
     + **Дешифрувати**: Натисніть для дешифрування зашифрованого тексту.
     + **Завантажити файл**: Завантажує текст з файлу для шифрування/дешифрування.
     + **Зберегти в файл**: Зберігає результат у файл.
   * **Візуалізація шифру Віженера**: Графічне полотно відображає зашифровані літери відповідно до ключа.

