



**WYŻSZA SZKOŁA  
INFORMATYKI i ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## KOLEGIUM INFORMATYKI STOSOWANEJ

**Kierunek: INFORMATYKA**

**Specjalność: Inżynieria Danych**

Dominik Pyrzak  
Nr albumu studenta w69838

### ***Aplikacja do zarządzania hasłami***

Prowadzący: mgr inż. Ewa Zesławska

Rzeszów 2025



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>6</b>
1.1	Założenia projektu . . . . .	6
1.2	Cele projektu . . . . .	6
<b>2</b>	<b>Opis struktury projektu</b>	<b>7</b>
2.1	Diagram klas . . . . .	7
2.2	Minimalne wymagania . . . . .	7
2.3	Format plikowy . . . . .	8
<b>3</b>	<b>Plan projektu i repozytorium</b>	<b>9</b>
3.1	Repozytorium . . . . .	9
3.2	Diagram Gantta . . . . .	9
<b>4</b>	<b>Prezentacja warstwy użytkowej</b>	<b>10</b>
4.1	Dodaj nowe hasło . . . . .	10
4.2	Wyświetl wszystkie hasła . . . . .	11
4.3	Edytuj hasło . . . . .	12
4.4	Usuń hasło . . . . .	12
4.5	Zakończ . . . . .	12
4.6	Logi w pliku historia.log . . . . .	13
4.7	Plik passwordy.txt z zaszyfrowanymi hasłami . . . . .	13
<b>5</b>	<b>Podsumowanie</b>	<b>14</b>
<b>6</b>	<b>Bibliografia</b>	<b>15</b>

# Spis treści

# Rozdział 1

## Wstęp

Niniejszy rozdział opisuje założenia i cele aplikacji konsolowej w C#, służącej do zarządzania hasłami i zabezpieczania ich poprzez szyfrowanie AES. Aplikacja realizuje operacje CRUD na pliku tekstowym, co eliminuje konieczność instalowania bazy danych.

### 1.1 Założenia projektu

- Operacje CRUD: dodawanie, wyświetlanie, edycja i usuwanie wpisów,
- Szyfrowanie AES: każdy zapis hasła jest zabezpieczony kluczem 256-bitowym,
- Logowanie zdarzeń: klasa Logger rejestruje wykonane akcje w pliku logów,
- Format plikowy: passwordy.txt przechowuje dane w postaci zaszyfrowanej.

### 1.2 Cele projektu

- Praktyczne pokazanie programowania obiektowego w C# (dziedziczenie, klasa abstrakcyjna),
- Zapewnienie poufności hasel poprzez szyfrowanie,
- Udowodnienie prostego wdrożenia menedżera hasel w trybie konsolowym.

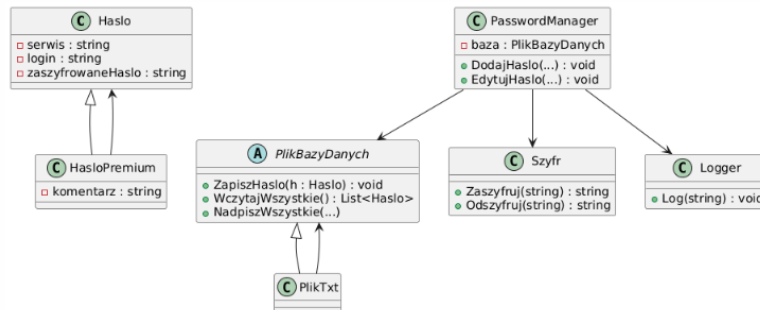
# Rozdział 2

## Opis struktury projektu

W tym rozdziale przedstawiono główne elementy architektury aplikacji, w tym diagram klas i informacje o formacie danych.

### 2.1 Diagram klas

Rysunek 2.1 prezentuje główne klasy projektu.



Rysunek 2.1: Diagram klas aplikacji

Haslo to klasa bazowa przechowująca serwis, login i zaszyfrowane hasło. HasloPremium dziedziczy po Haslo, rozszerzając je np. o pole komentarz. PlikBazyDanych jest klasą abstrakcyjną, a PlikTxt konkretną implementacją obsługi pliku. Szyfr statycznie realizuje szyfrowanie AES, Logger zapisuje logi, a PasswordManager zarządza operacjami CRUD.

### 2.2 Minimalne wymagania

- System z obsługą .NET 8.0 (Windows, Linux, macOS),

- Procesor 1 GHz lub szybszy, co najmniej 1 GB RAM,
- Około 100 MB na pliki projektu i SDK,
- Dodatkowe biblioteki: System.Security.Cryptography.

## 2.3 Format plikowy

Dane są zapisywane w pliku passwordy.txt, każdy wiersz w postaci:

`Serwis|Login|ZaszyfrowaneHaslo`

Logi zapisywane są w pliku historia.log, zawierając informacje o dacie i rodzaju operacji.



# Rozdział 3

## Plan projektu i repozytorium

### 3.1 Repozytorium

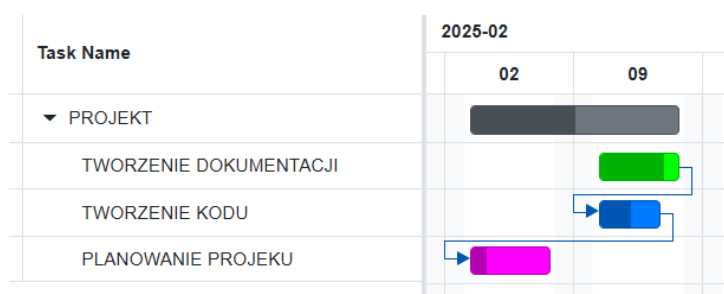
Kod źródłowy projektu jest przechowywany w systemie kontroli wersji **Git**. Aktualna wersja znajduje się w repozytorium:

<https://github.com/ProgramowaniePyrzakDominikw69838/ProjektZarządzanieHasel>

W konsoli pojawi się menu główne, z którego można wybrać między innymi dodanie, wyświetlanie, edycję i usuwanie wpisu.

### 3.2 Diagram Gantta

Rysunek 3.1 przedstawia uproszczony harmonogram realizacji projektu w formie diagramu Gantta. Każdy etap może być dostosowywany do bieżących potrzeb lub priorytetów.



Rysunek 3.1: Uproszczony diagram Gantta prezentujący kolejne etapy prac

## Rozdział 4

# Prezentacja warstwy użytkowej

Aplikacja uruchamia się w trybie konsolowym i wyświetla menu zawierające operacje:

1. Dodaj nowe hasło
2. Wyświetl wszystkie hasła
3. Edytuj hasło
4. Usuń hasło
5. Zakończ

### 4.1 Dodaj nowe hasło

Użytkownik wprowadza serwis, login i jawne hasło. Klasa Szyfr konwertuje hasło do Base64, a PlikTxt dopisuje wpis do pliku `passwordy.txt`. Rysunek 4.1 obrazuje konsolę podczas dodawania.

```

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór: 1
Serwis: facebook
Login: dominik
Hasło (jawne): pyrzak

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór: 2
facebook -> login: dominik, hasło: pyrzak

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór:

```

Rysunek 4.1: Zrzut ekranu: proces dodawania nowego hasła

## 4.2 Wyświetl wszystkie hasła

Opcja wyświetlania odczytuje wiersze z pliku i odszyfrowuje hasła, aby ukazać użytkownikowi w formie jawnej. Rysunek 4.2 prezentuje przykładowy wynik.

```

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór: 2
facebook -> login: dominik, hasło: pyrzak123
gmail -> login: dominik, hasło: pyrzak

```

Rysunek 4.2: Zrzut ekranu: wyświetlanie wszystkich haseł

## 4.3 Edytuj hasło

Edycja wpisu (rys. 4.3) pozwala zmienić login lub hasło w istniejącym rekordzie zidentyfikowanym po nazwie serwisu.

```
Wybór: 1
Serwis: facebook
Login: dominik
Hasło (jawne): pyrzak

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór: 4
Serwis do edycji: facebook
Obecny login: dominik
Nowy login (ENTER = bez zmian):
Obecne hasło: pyrzak
Nowe hasło (ENTER = bez zmian): pyrzak123
Zapisano zmiany.
```

Rysunek 4.3: Zrzut ekranu: edycja wpisu

## 4.4 Usuń hasło

W opcji usuwania (rys. 4.4), użytkownik podaje serwis. Program usuwa dopasowany rekord z pliku. Logger zapisuje informację o usunięciu w pliku `historia.log`.

```
5) Wyjście
Wybór: 3
Podaj serwis do usunięcia: facebook
Usunięto 1 rekord(ów).

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór:
```

Rysunek 4.4: Zrzut ekranu: usunięcie wybranego wpisu

## 4.5 Zakończ

Opcja zakończenia kończy działanie programu. Rysunek 4.5 przedstawia krótki komunikat wyświetlany użytkownikowi.

```

--- MENEDŻER HASEŁ ---
1) Dodaj hasło
2) Lista haseł
3) Usuń hasło
4) Edytuj hasło
5) Wyjście
Wybór: 5
Koniec. Papa!

```

Rysunek 4.5: Zrzut ekranu: zakończenie programu

## 4.6 Logi w pliku historia.log

Rysunek 4.6 ukazuje przykładowe logi, w których zapisano informacje o dodaniu, edycji oraz usunięciu wpisu.

```

03.03.2025 05:46:38 :: Dodano hasło -> serwis: facebook
03.03.2025 05:47:21 :: Usunięto 1 rekordów (serwis=facebook)
03.03.2025 05:48:04 :: Dodano hasło -> serwis: facebook
03.03.2025 05:48:18 :: Edytowano serwis facebook

```

Rysunek 4.6: Przykładowy widok pliku historia.log

## 4.7 Plik passwordy.txt z zaszyfrowanymi hasłami

Rysunek 4.7 przedstawia fragment pliku passwordy.txt, gdzie zamiast jawnych haseł widać jedynie ciągi Base64. Bez znajomości klucza AES nie jest możliwy odczyt faktycznych treści.

```

facebook|dominik|+PKyzwBm15w07of0JlPnCQ==
gmail|dominik|6FLkZIMPWTfoZyffQSeK+w==

```

Rysunek 4.7: Zaszyfrowane hasła w pliku passwordy.txt

## Rozdział 5

### Podsumowanie

Każda opcja menu (dodawanie, wyświetlanie, edycja, usuwanie, zakończenie) została przedstawiona wraz z przykładami zrzutów ekranu. Dodatkowo zaprezentowano zawartość pliku `historia.log` i pliku `passwordy.txt`, co pozwala zrozumieć, w jaki sposób aplikacja zarządza i zabezpiecza dane. Mechanizm szyfrowania AES zapewnia poufność haseł, a prosta struktura klas (dziedziczenie `HasloPremium`, abstrakcyjna `PlikBazyDanych`) umożliwia rozbudowę funkcjonalności bez zmian w głównej logice `PasswordManager`.

## Rozdział 6

## Bibliografia

# Bibliografia

- [1] Troelsen A., Japikse P., *Pro C# 9 with .NET 5*, Apress, 2021.