



**AKADEMIA GÓRNICZO-HUTNICZA**

Dokumentacja do projektu

**Baza danych do zarządzania długami i  
użytkownikami**

z przedmiotu

**Języki Programowania Obiektowego**

Elektronika i Telekomunikacja 3 rok

*Jakub Woźniak*

Piątek 15:00

prowadzący: Jakub Zimnol

Kraków 06.01.2025r

## 1. Opis projektu

Realizowany przeze mnie projekt to aplikacja reprezentująca bazę danych długów wśród użytkowników. Jej podstawową funkcjonalnością jest zarządzanie nowymi/istniejącymi klientami oraz zarządzanie przypisanymi do nich długami. Całość została napisana w języku C++ i składa się z 3 plików (z rozszerzeniem hpp) zawierających definicję oraz implementację klas oraz 1 pliku testowego main.cpp. Aplikację tą można wykorzystać w codziennym życiu w grupie znajomych, przyjaciół czy rodziny.

## 2. Opis stworzonych klas

### - Debt

Klasa Debt reprezentuje dług z różnymi atrybutami, takimi jak identyfikator, opis, kwota, odbiorca oraz status płatności. Jest ona częścią przestrzeni nazw „jw”. W jej skład wchodzi konstruktor, gettery oraz publiczne metody pozwalające na zmianę danych w format tekstowy oraz powrotne przekształcenie ich na obiekt klasy.

### - User

Klasa User reprezentuje użytkownika z różnymi atrybutami, takimi jak identyfikator, nazwa użytkownika oraz tajny klucz (hasło). Jest ona również częścią przestrzeni nazw „jw”. W jej skład wchodzi konstruktor, gettery oraz publiczne metody do przekształcania danych jak w klasie Debt. Różnicą natomiast jest prywatna metoda realizująca podstawowe szyfrowanie oraz odczytywanie haseł użytkowników korzystając z kodowania XOR.

### - Database

Klasa Database jest klasą nadrzędną do klas User oraz Debt, ponieważ jest odpowiedzialna za zarządzanie nimi jak w bazie danych. Posiada prywatne pola typu map, które magazynują w sobie obiekty Debt oraz User oraz prywatną metodę username\_exist do wyszukiwania nazwy szukanego użytkownika. Wśród publicznych metod, znajdują się takie funkcje jak dodawanie użytkowników, wyświetlanie długów, oznaczanie długów jako zapłacone oraz przede wszystkim odczyt oraz zapis do plików tekstowych. Klasa ta również znajduje się w przestrzeni nazw „jw”.

## 3. Uruchomienie aplikacji

Aplikację uruchamia się poprzez skompilowanie pliku testowego main.cpp oraz włączenie powstałego pliku wykonywalnego. Program ten skanuje czy istnieją pliki bazy danych w formacie txt, jeśli nie, użytkownik musi się zarejestrować, aby przejść do logowania oraz dalszych funkcji. Przykładowe działanie można zobaczyć na zdjęciach poniżej:

```
Files not found. Starting with empty database
2. Register
3. Exit

Choose an option:
2
-----
Username: Jakub
Password: test123
User registered with ID: 1

1. Login
2. Register
3. Exit

Choose an option:
```

```
Choose an option:
1
-----
Here are all the users:
ID: 1, Name: Jakub
ID: 2, Name: Jakub2
-----

Which user ID: 2
Description: Test aplikacji
Amount: 89.92
Debt added successfully

1. Add debt
2. Display my debts
3. Mark debt as paid
4. Logout
```

```
Username: Jakub2
Password: test123

1. Add debt
2. Display my debts
3. Mark debt as paid
4. Logout

Choose an option:
2
-----
-ID: 1, Title: Test aplikacji, Amount: 89.92 zł, Receiver: Jakub, Paid: No

1. Add debt
2. Display my debts
3. Mark debt as paid
4. Logout
```