



**AKADEMIA GÓRNICZO-HUTNICZA**

Dokumentacja do projektu

# **Biblioteka szyfrująca i deszyfrująca dane**

z przedmiotu

## **Języki Programowania Bbięktowego**

Elektronika i Telekomunikacja

*Bartłomiej Żurad*

Piątek 9:45

prowadzący: mgr inż. Jakub Zimnol

14.01.2025

# 1. Ogólny opis projektu

Stworzona biblioteka służy do pobierania danych z trzech różnych źródeł: konsoli, pliku oraz gniazda sieciowego. Następnie dane te są szyfrowane lub deszyfrowane w zależności od potrzeby użytkownika. Po wykonaniu wybranej operacji dane są przekazywane na wyjście, tak jak w przypadku wejścia, do wyboru są trzy możliwości: wyświetlenie danych w konsoli, zapisanie w pliku lub wysłanie do odpowiedniego gniazda.

Szyfrowanie danych odbywa się przy pomocy zaimplementowanych algorytmów:

- xor – proste szyfrowanie symetryczne (ten sam klucz odpowiada za szyfrowanie i deszyfrowanie)
- caesar – proste szyfrowanie przesunięciem znaków o określoną liczbę pozycji

użytkownik ma możliwość wyboru typu szyfrowania oraz określenia jego podstawowych parametrów takich jak: klucz szyfrowania/deszyfrowania lub liczbę przesunięć.

## 2. Struktura projektu

Projekt zawiera dwa katalogi: *inc* oraz *src*, w których odpowiednio są umieszczone pliki nagłówkowe i pliki źródłowe.

Pliki nagłówkowe:

- *inputHandler.hpp*
- *encryptClass.hpp*
- *decryptClass.hpp*
- *outputHandler.hpp*

Pliki źródłowe:

- *inputHandler.cpp*
- *encryptClass.cpp*
- *decryptClass.cpp*
- *outputHandler.cpp*
- *main.cpp*

W pliku *main.cpp* znajduje się główna logika programu wraz z interakcją z użytkownikiem. W katalogu głównym projektu znajduje się także *Makefile* odpowiadający za kompilację programu.

CELOWO POZOSTAWIONO PUSTE

### 3. Uruchomienie projektu

Aby skorzystać z funkcjonalności projektu należy go na początku skompilować, w tym celu należy użyć komendy: *make* w konsoli, znajdując się w głównym katalogu projektu. Plik *Makefile* jest tak skonfigurowany, że automatycznie dokona kompilacji, stworzy plik *main*, który odpowiada za uruchomienie programu oraz stworzy dodatkowy katalog *obj*.

W celu uruchomienia programu należy wpisać w konsolę *./main*. Dalsze działanie programu odbywa się dzięki interakcji programu z użytkownikiem, które zostanie zaprezentowane w zrzutach ekranu.

```
choose your data source:
1. console
2. file
3. socket
source: |
```

*screenshot 1 - wybór wejścia danych*

W tym momencie użytkownik wybiera źródło danych. W przypadku podania niewłaściwego nr wejścia pokazuje się komunikat „invalid source...” oraz program kończy pracę.

```
source: 5
invalid source...
```

*screenshot 2 - zły nr źródła*

W przypadku dobrze wybranego źródła dane zostaną pobrane z pliku, gniazda lub użytkownik zostanie poproszony o wpisanie danych do konsoli. Następnie dane wyświetlą się w konsoli w celu sprawdzenia przez użytkownika czy są one właściwe.

```
choose your data source:
1. console
2. file
3. socket
source: 1
enter your data: hello
loaded data: hello
```

*screenshot 3 - wejście danych (konsola)*

Następnie użytkownik ma możliwość wyboru operacji: szyfrowanie, deszyfrowanie. Po tym wyborze, następuje kolejny tym razem należy wybrać algorytm szyfrowania/deszyfrowania oraz jego parametry.

```
choose operation:
1. encryption
2. decryption
operation: 1
choose algorithm (xor/caesar): caesar
enter shift: 3
```

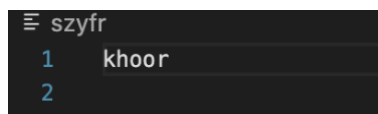
*screenshot 4 - wybranie operacji oraz parametrów algorytmu*

Na koniec użytkownik wybiera wyjście danych, w przypadku pliku lub gniazda odpowiednia należy podać ścieżkę pliku i nr gniazda sieciowego.

```
choose data output:
1. console
2. file
3. socket
output: 2
file path: szyfr
data saved to file...
```

*screenshot 5 - wybór wyjścia i zapisanie danych*

Wynik w tym przypadku zostanie zapisany w katalogu głównym projektu i wygląda następująco



```
≡ szyfr
1   khoor
2
```

*screenshot 6 - wynik szyfrowania*

Interakcja z programem jest przejrzysta i nie powinna stanowić problemu podczas użytkowania, przedstawione działanie jest tylko jednym, wybranym przykładem.

Wybór źródła danych, formy zapisu (wysłania), algorytmu (jednego z dwóch) i jego parametrów zależy od użytkownika.