Dokumentacja Projektu PL
Jakub Strawa 300266

Środowisko testowe prostych algorytmów szyfrujących

Opis projektu:
Celem projektu jest utworzenie w Javie prostego środowiska testowego dla 4 algorytmów szyfrujących:

- Szyfr Cezara
- Szachownica Polibiusza
- Enigma
- Blowfish

Projekt wykonany jest przy użyciu biblioteki graficznej Java Swing.

Funkcjonalności:

- Możliwość zapisania do pliku output.txt podanej wiadomości razem z jej zakodowaną lub rozkodowaną wersją i sposobem kodowania.
- Możliwość wczytania wiadomości z pliku input.txt
- Skopiowanie zakodowanej/rozkodowanej wiadomości do pamięci (funkcjonalność równoważna skrótowi Ctrl + C)
- Krótki opis algorytmu w zakładce „About cipher”
- Podgląd ustawień algorytmu w przypadku Enigmy oraz szachownicy Polibiusza
- Przetłumaczenie wiadomości krok po kroku lub jednym kliknięciem

Instrukcja użytkowania:

1. Wybieramy interesujący nas algorytm
2. Wpisujemy wiadomość, którą chcemy zakodować/rozkodować, używamy tylko słów złożonych z 26 liter alfabetu łacińskiego
3. Uzupełniamy opcje algorytmu: szyfr Cezara – przesunięcie, szachownica Polibiusza – własny klucz jeśli nie chcemy używać domyślnego, Enigma – kolejność rotorów, ich początkowe ustawienia, połączenia liter (może być w postaci słowa/zdania) i deflektor, Blowfish – klucz, rodzaj danych wyjściowych
4. Wybieramy czy chcemy zakodować czy rozkodować wiadomość
5. Wybieramy czy chcemy wykonać algorytm krok po kroku
6. Naciskamy przycisk Start lub Next Step w zależności od wybranej w poprzednim punkcie opcji
7. Opcjonalnie zapisujemy wynik działania programu

Sposób wprowadzania danych:
Szyfr Cezara: słowa złożone tylko z 26 liter alfabetu łacińskiego (pozostałe znaki nie zostaną zakodowane/rozkodowane).
Szachownica Polibiusza: słowa złożone tylko z 26 liter alfabetu łacińskiego przy kodowaniu, a przy rozkodowaniu ciąg liczb odpowiadających położeniu liter w szachownicy rozdzielony spacjami, np. 12 32 41 23 44 (pozostałe znaki nie zostaną zakodowane/rozkodowane).
Enigma: słowa złożone tylko z 26 liter alfabetu łacińskiego (pozostałe znaki nie zostaną zakodowane/rozkodowane).

Blowfish: dowolne słowa, mogą zawierać znaki interpunkcyjne oraz polskie znaki (dla rozkodowania wiadomości używamy znaków otrzymanych po zakodowaniu z opcją Char), klucz złożony z dowolnych liter (nie zawiera cyfr).

Instrukcja instalacji:

1. Pobieramy aktualny branch master
2. W katalogu z pobranym branchem używamy komendy: *mvn clean install*
3. Używamy komendy: *java -jar target/cipher-ver.jar*, gdzie ver to aktualna wersja programu (aktualnie 2.0)

Opis klas:

Klasy *model zawierają informacje o algorytmach, są tworzone raz przy starcie programu. Klasy EnigmaTable i PolybiusTable tworzone są kiedy użytkownik naciśnie przycisk Show Table. Klasy About* tworzone są kiedy użytkownik naciśnie przycisk About cipher w sekcji About. Klasa Main inicjalizuje klasy z modelami algorytmów, MainView i główny kontroler, którym jest klasa MainController. Obsługuje ona wszystkie zdarzenia z MainView. Klasy Pair oraz Rotor odpowiadają za obsługę par oraz rotorów Enigmy.
Klasy Test zawierają testy wszystkich metod klas z modelami, Pair oraz Rotor.

Komentarze:

Głównym problemem programu jest sposób działania krok po kroku algorytmu Blowfish, który przetwarza jednocześnie 4 znaki, co wiąże się z jego sposobem działania. Niepotrzebne wydaje się pokazanie 18 razy przetwarzanego longa, który jest rozkodowywany dopiero po wszystkich iteracjach algorytmu.