

Analiza projektowa projektu Symulator Enigmy

Marcin Maślach

11 czerwca 2018

1. Spis klas:

- Enigma
- Alphabet
- Rotor
- Plugboard
- Reflector
- User
- Settings

2. Tekstowy opis analizy.

Projekt został stworzony na wzór oryginalnej Enigmy używanej podczas wojny. Zawiera on klasy o nazwach poszczególnych komponentów oraz klasy pomocnicze. Po włączeniu, zostajemy zapytani o klucz i tekst, które muszą składać się ze znaków występujących w alfabecie podstawowym. Klucz dodatkowo musi zawierać trzy elementy, czyli liczbę odpowiadającą liczbie rotorów używanych w enigmie. Następnie pytani jesteśmy o ustawienia komponentów maszyny. Mamy do wyboru wpisanie własnych kombinacji lub ustawienia podstawowe, zapisane w kodzie źródłowym. Po uzupełnieniu całości maszyna koduje naszą wiadomość. Przy ponownym użyciu tych samych ustawień i wpisaniu zakodowanej wiadomości, otrzymamy odkodowany tekst.

Alphabet

Jest to klasa pozwalająca utworzyć odpowiednią listę znaków wchodzących w skład występujących w maszynie alfabetów. Za pomocą funkcji *check_len_alphabet* sprawdzamy czy alfabet ma odpowiednią długość. Dzięki *repeat_alphabet* na naszej liście nie ma powtórzeń. Używając tych funkcji możemy sprawdzić, czy wpisane przez użytkownika ustawienia są dopuszczalne. Funkcja *take_index* pobiera index podanego znaku. *take_sign* zwraca znak stojący na wpisanym miejscu listy. *len_alphabet* zwraca długość alfabetu, a *__str__* zwraca listę znaków.

Plugboard

Jest to klasa pozwalająca nam zamienić znaki parami. Funkcja *plugboard_encode* pobiera

znak z listy składającej się z wyrazów i zamienia go na znak który jest z nim połączony. Zwraca listę indexów zakodowanych znaków, pobranych z alfabetu podstawowego. *plug-board_decode* pobiera index alfabetu podstawowego, a następnie zmienia go na znak z nim połączony. Zwraca listę liter.

Reflector

Dzięki temu elementowi przechodzimy z *encode* do *decode*. Funkcja *reflect* pobiera index znaku z alfabetu podstawowego ostatniego rotora, a następnie zwraca index tego znaku z alfabetu podstawowego.

Rotor

Jest to główny element kodujący enigmy. Funkcja *rotor_encode* pobiera index znaku do zakodowania, znajduje znak o tym indexie w swoim alfabecie charakterystycznym, czyta jaki to znak i oddaje jego index z alfabetu podstawowego. Funkcja *rotor_decode* działa w podobny sposób, jednak kolejność alfabetów charakterystycznego i podstawowego jest zamieniona względem poprzedniej funkcji.

Settings

Za pomocą tej klasy ustawiamy wszystkie elementy kodujące maszyny. *use_key* tworzy alfabet podstawowy dla każdego z rotorów. Zaczyna się on od znaku z kodu który został wpisany przez użytkownika. *set_characteristic* ustawia alfabet charakterystyczny każdego, zgodnie z wybranym przez użytkownika kluczem. *change_key* zmienia klucz gdy występują w nim powtórzenia. Zapobiega to nie zmienionym znakom, oraz utrudnia rozszyfrowanie wiadomości. *rotate_key* obraca wartość pierwszej litery klucza o jeden po wpisaniu każdego kolejnego znaku. *check_key* sprawdza zgodność wpisanego klucza.

User

Klasa ta ma atrybut jedynej, nie pasującej do klasy settings, zmiennej, a mianowicie wpisywaną wiadomość. *original_text* sprawdza czy każdy znak wiadomości jest możliwy do zakodowania przez enigmę. *len_text* zwraca długość wiadomości. Funkcja ta jest używana do sprawdzenia, czy przy kodowaniu nie wystąpił błąd.

Enigma

Klasa ta zbiera wszystkie informacje potrzebne do szyfrowania. *set_machine* ustawia ostatecznie komponenty kodujące i zmienia te ustawienia przy każdym znaku wiadomości. *encode* i *decode* działają podobnie, z tą różnicą, że pierwsza z nich pobiera znak wpisany przez użytkownika i działa od Łącznicy do Reflektora, a *decode* odwraca tą drogę zwracając ostatecznie zakodowaną literę.

Zidentyfikowane wzorce projektowe:

Szablon - szkieletem algorytmu jest klasa Enigma. Została zapisana tu niezmienna część algorytmu.

Adapter - dzięki klasie Settings tworzą się połączenia między rotorami gdzie alfabety podstawowe i charakterystyczne ustawiają się w odpowiedniej kolejności uwzględniając aktualną sytuację wewnątrz enigmy.