**Console Card Manager**

Aplikacja służy do przechowywania danych kart debetowych/kredytowych takich jak:  
- numer karty  
- data ważności  
- imię i nazwisko posiadacza karty

Dane kart są przechowywane w pliku wykorzystującym format JavaScript Object Notation (json) oraz są zaszyfrowane przy pomocy algorytmu AES z wykorzystaniem 128 bitowego klucza. Obsługa pliku w powyższym formacie oraz szyfrowanie zostało zrealizowane przy pomocy gotowych darmowych bibliotek: JSON for Modern C++ (<https://github.com/nlohmann/json>) oraz Plusaes (<https://github.com/kkAyataka/plusaes>).

Aplikacja umożliwia wykonanie poniższych operacji:  
- wyświetlenie zapisanych kart  
- dodanie nowej karty  
- usunięcie wybranej karty  
- usunięcie przeterminowanych kart  
- usunięcie wszystkich kart

W momencie uruchomienia programu następuje odczyt danych zapisanych w pliku, ich deszyfracja i zapisanie do pamięci operacyjnej w postaci listy dwukierunkowej obiektów klasy “Card”. Jest to jedyna sytuacja, gdy wykorzystujemy odczyt pliku. Wszystkie operacje modyfikujące powyższą listę powodują nadpisanie pliku jej obecną zawartością.

Klasy:

**Card** – klasa mająca na celu przechowywać dane o karcie kredytowej/debetowej. Nie posiada dodatkowej logiki, poza zestawem getterów do bezpiecznego odczytu zapisanych danych. Klasa jako identyfikator obiektu (id) używa 4 ostatnie cyfry numeru karty.

**CardManager** - klasa implementująca główną funkcjonalność programu. Przechowuje dwukierunkową listę obiektów typu “Card”.

Do komunikacji z użytkownikiem używa wskaźnika do klasy czysto wirtualnej CardManagerUserInterface, co docelowo ma ułatwić ewentualną zmianę wykorzystywanego interfejsu.

Modyfikacje listy kart zapisuje wykorzystując wskaźnik do obiektu typu FileHandler.

Poza implementacją głównych funkcjonalności programu zawiera również zestaw metod służących do weryfikacji danych karty. Implementuje alogrytm Luhna do sprawdzenia numeru karty kredytowej.

**CardManagerUserInterface –** klasa czysto wirtualna definiująca podstawowe metody wymagane do obsługi programu.

**CardManagerUICommandLine –** klasa implementująca **CardManagerUserInterface** przy wykorzystaniu linii komend jako metody komunikacji z użytkownikiem.

**FileHandler –** klasa umożliwiająca zapis i odczyt pliku z danymi o kartach bazująca na bibliotece JSON for Modern C++. Przy każdej operacji odczytu/zapisu dane są deszyfrowane/szyfrowane do czego jest wykorzystywany wskaźnik na obiekt typu EncryptionHandler.

**EncryptionHandler** – klasa służąca do szyfrowania/deszyfrowania danych wykorzystująca bibliotekę Plusaes.

Przykłady use case:

Start programu:  
Tworzenie obiektów->odczyt danych z plików->zapis danych do pamięci operacyjnej || brak pliku/danych - brak akcji

Wyświetlanie danych:  
Odczyt danych z pamięci operacyjnej->Wyświetlenie przy pomocy User Interface

Dodanie nowej karty:  
Procedura dla każdej danej w CardManager:  
- pobierz informację przy pomocy user interface  
- zweryfikuj informację przy pomocy zaimplementowanej logiki  
Następnie tworzony jest obiekt typu Card i dodawany do listy. Zmodyfikowana lista jest zapisywana do pliku.

Operacje usuwania wybranej karty  
- pobierz informację o cardId do usunięcia (przy wykorzystaniu user interface)  
- znajdź kartę z cardId  
- (powodzenie) usuń kartę z listy i zapisz obraz zmodyfikowanej listy do pliku   
- (niepowodzenie) wyświetl komunikat

Mateusz Pazdro