Specyfikacja Programu:

„Szyfrowanie RSA”

na projekt z przedmiotu PJP II.

Autor: **Karol Łuczyński**

1. Wprowadzenie.

- **1.1. Cel.**
- **1.2. Zakres.**
- **1.3. Definicje, akronimy i skróty.**
- **1.4. Referencje, odsyłacze do innych dokumentów.**
- **1.5. Krótki przegląd.**

2. Ogólny opis.

- **2.1. Walory użytkowe i przydatność projektowanego systemu.**
- **2.2. Ogólne możliwości projektowanego systemu.**
- **2.3. Ogólne ograniczenia.**
- **2.4. Charakterystyka użytkowników.**
- **2.5. Środowisko operacyjne.**
- **2.6. Założenia i zależności.**

3. Specyficzne wymagania.

- **3.1. Wymagania funkcjonalne (funkcje systemu).**
- **3.2. Wymagania niefunkcjonalne (ograniczenia).**

4. Dodatki.

- **4.1. Harmonogram prac nad projektem.**

1. Wprowadzenie.

1.1. Cel.

Dokument ten opisuje wymagania jakie ma spełnić oprogramowanie i jego funkcję.

Dokument przeznaczony dla programistów, testerów i użytkowników.

1.2. Zakres.

Produkt będzie szyfrował tekst w celu ochrony danych. Ze względu na liczne ataki hakerów i ataki na prywatność użytkownika, aplikacja okaże się dobrym rozwiązaniem.

Powstaje z myślą o osobach prywatnych chcących wykorzystać szyfrowanie w celach bezpieczeństwa lub też rozrywkowych.

1.3. Definicje, akronimy i skróty.

RSA - Algorytm Rivesta-Shamira-Adlemana – popularny algorytm kryptograficzny używający dwóch kluczy: klucz publiczny (szyfrujący) i klucz prywatny (deszyfrujący).

Plik txt - plik tekstowy.

Szyfrowanie / **Kodowanie** - zapisywanie informacji za pomocą szyfru / kodu.

Deszyfracja / **Odkodowanie** - odczytywanie informacji zapisanej za pomocą szyfru / kodu.

Tekst jawny - wiadomość, która nie została jeszcze zaszyfrowana (lub została odszyfrowana).

Szyfrogram - wiadomość, która została zaszyfrowana.

1.4. Referencje, odsyłacze do innych dokumentów.

Nie dotyczy.

1.5. Krótki przegląd.

W rozdziale **2.** został przedstawiony opis oprogramowania.

W rozdziale **3.** przedstawione są funkcje oprogramowania.

W dodatkach przedstawiony będzie harmonogram prac nad projektem.

2. Ogólny opis.

2.1. Walory użytkowe i przydatność projektowanego systemu.

Program ma być w stanie skutecznie zabezpieczyć tekst przed niepowołanymi odbiorcami. Szyfrując algorytmem RSA przy odpowiednik ukryciu przez użytkownika klucza prywatnego szyfr jest na obecne czasy nie do złamania. Ilość obliczeń potrzebnych do złamania kodu bez klucza zajmuje dziesiątki lat. Jest to więc skuteczna metoda wykorzystywana na przykład przez banki. Użytkownik będzie mógł być spokojny o swoje dane.

2.2. Ogólne możliwości projektowanego systemu.

- Wytworzenie pary kluczy (klucza prywatnego i publicznego).
- Szyfrowanie wprowadzonego teksu za pomocą klucza publicznego.
- Szyfrowanie pliku txt za pomocą klucza publicznego.
- Deszyfrowanie wprowadzonego teksu za pomocą klucza prywatnego.
- Deszyfrowanie pliku txt za pomocą klucza prywatnego.

2.3. Ogólne ograniczenia.

- Liczby pierwsze użyte do wytworzenia kluczy muszą być dla bezpieczeństwa usunięte.
- Klucz prywatny musi być zabezpieczony.
- Obliczenia wykonywane przy szyfrowaniu i deszyfrowaniu muszą być w zasięgu sprzętu na którym są wykonywane (dzisiejsze komputery).

2.4. Charakterystyka użytkowników.

Użytkownik – Osoba szyfrująca lub deszyfrująca tekst.

Przypadki użycia programu przez użytkownika:

- Tworzenie pary kluczy (prywatny i publiczny).
- Szyfrowanie danych.
- Odczytywanie zaszyfrowanych danych.

2.5. Środowisko operacyjne.

Do prawidłowego funkcjonowania programu wymagany będzie system operacyjny zgodny z rodziną systemów Windows.

2.6. Założenia i zależności.

Komputer użytkownika musi być wyposażony minimalnie w:

- System operacyjny zgodny z rodziną systemów Windows(min. Windows 7).
- Procesor Intel Atom® lub Intel® Core™ i3
- Miejsce na dysku: 1 GB
- Monitor.
- Mysz.
- Klawiatura.

Zależności: Brak zależności.

3. Specyficzne wymagania.

3.1. Wymagania funkcjonalne (funkcje systemu).

3.1.1. Menu główne.

Waga duża.

Program ma posiadać czytelne menu umiejscowione na środku okna z następującymi opcjami:

- „Tworzenie pary kluczy”.
- „Szyfrowanie”.
- „Czytaj szyfrogram”.
- „Wyjście”.

Każda opcja ma powodować przejście ekranu aplikacji w inne okno posiadające swoje odrębne funkcję opisane niżej.

Opcje mają posiadać jednakowej wielkości przycisk.

Czcionka użyta na przyciskach ma być jednakowa i o jednakowej wielkości. Pierwsze słowa mają zaczynać się wielką literą.

Napisy mają być umiejscowione na środku, pionowo i poziomo, względem swojego przycisku.

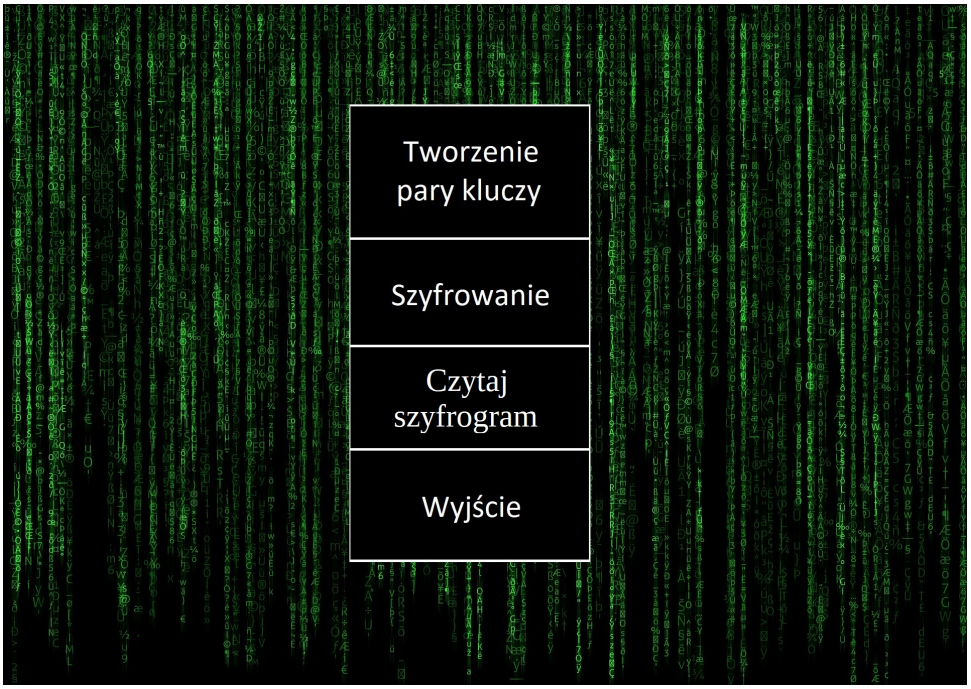
Waga średnia.

Menu ma posiadać tematyczne, związane z kodami tło.

Waga niska.

Kliknięcie klawisza „Esc” zostaje uruchomiony przycisk „Wyjście”.

Szkic menu główne:



3.1.2. Tworzenie pary kluczy.

Waga duża.

Po kliknięciu w menu głównym przycisku „Tworzenie pary kluczy”, okno menu ma się zmienić w okno przeznaczone do tworzenie pary kluczy.

Okno ma posiadać dwa przyciski „Stwórz parę kluczy” i „Powrót”, zaczynające się wielką literą.

Na górze ekrany wyświetlona jest informacja:

„W tym oknie tworzysz parę kluczy które służą do zabezpieczenia teksu. Klucz publiczny służy do szyfrowania Twoich plików lub przekazujesz osobie która ma zaszyfrować tekst do odczytu dla Ciebie.
Klucz prywatny służy do odczytania Twoich plików lub wiadomości od innych zaszyfrowanych kluczem publicznym. Zadbaj o jego bezpieczeństwo.”

Niżej po lewej znajduje się przycisk „Stwórz parę kluczy” uruchamia funkcję która odpowiednim algorytmem losowania i działań na liczbach pierwszych tworzy dwa kluczę: prywatny i publiczny.
Klucze te są wyświetlane w oknach opisanych odpowiednio „klucz publiczny” i „klucz prywatny” w wyznaczonym miejscu poniżej przycisku „Stwórz parę kluczy”.

Na dole po lewej znajduje się przycisk „Powrót” który ma wracać do menu głównego. Umiejscowiony analogicznie tak jak w oknach „Szyfrowanie” i „Czytaj szyfrogram”.

Waga średnia.

W tle okna ma się znajdować identyczne tło jak w oknie menu głównego.

Waga niska.

Kliknięcie klawisza „Esc” zostaje uruchomiony przycisk „Powrót”.

Szkic okna tworzenia kluczy:

3.1.3. Szyfrowanie.

Waga duża.

Po kliknięciu w menu głównym przycisku „Szyfrowanie”, okno menu ma się zmienić w okno przeznaczone do szyfrowania tekstu i plików .txt.

Okno ma posiadać przyciski:

- „Dodaj plik tekstowy” - przycisk umożliwia wybranie pliku tekstowego (.txt) który ma zostać zaszyfrowany po użyciu przycisku „Szyfruj plik”.
- „Szyfruj plik” - przycisk ma uruchamiać funkcję szyfrujący tekst zawarty w pliku i zwracać plik z szyfrogramem.
- „Szyfruj tekst” - przycisk ma uruchamiać funkcję szyfrującą tekst zawarty w polu na tekst jawny i zwracać tekst zaszyfrowany do pola z szyfrogramem.
- „Powrót” - przycisk który ma wracać do menu głównego. Ma znajdować się na dole po lewej tak jak w oknach „Tworzenie pary kluczy” i „Czytaj szyfrogram”.

Napisy na przyciskach ma zaczynać się wielką literą.

Okno ma posiadać dwa miejsca na tekst:

- Pole na tekst jawny, ma mieć możliwość dodania tekstu. Szyfrowanie następuje po kliknięciu przycisku „Szyfruj tekst”.
- Pole z szyfrogramem ma mieć możliwość odczytania i skopiowania szyfrogramu. Możliwość edycji ma być zablokowana.

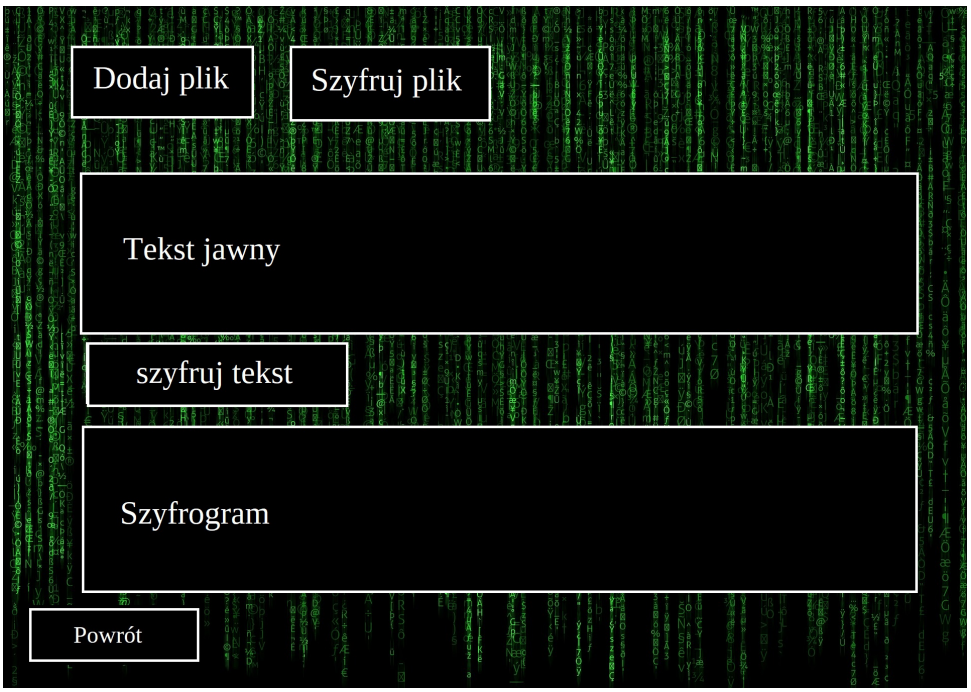
Waga średnia.

W tle okna ma się znajdować identyczne tło jak w oknie menu głównego.

Waga niska.

Kliknięcie klawisza „Esc” zostaje uruchomiony przycisk „Powrót”.

Szkic szyfrowanie:



3.1.4. Czytanie szyfrogramu.

Waga duża.

Po kliknięciu w menu głównym przycisku „Czytaj szyfrogram”, okno menu ma się zmienić w okno przeznaczone do czytania zaszyfrowanego tekstu i plików .txt.

Okno ma posiadać przyciski:

- „Dodaj plik tekstowy” - przycisk umożliwia wybranie pliku tekstowego (.txt) który ma zostać zaszyfrowany po użyciu przycisku „Szyfruj plik”.

- „Deszyfruj plik” - przycisk ma uruchamiać funkcję deszyfrującą tekst zawarty w pliku i zwracać plik z tekstem odszyfrowanym.

- „Deszyfruj tekst” - przycisk ma uruchamiać funkcję deszyfrującą tekst zawarty w polu na szyfrogram do odczytania i zwracać tekst do pola z tekstem jawnym.

- „Powrót” - przycisk który ma wracać do menu głównego. Ma znajdować się na dole po lewej tak jak w oknach „Tworzenie pary kluczy” i „Szyfrowanie”.

Napisy na przyciskach ma zaczynać się wielką literą.

Okno ma posiadać dwa pola na tekst:

- Ple na szyfrogram, ma mieć możliwość dodania tekstu. Szyfrowanie następuje po kliknięciu przycisku „Szyfruj tekst”.

- Pole z tekstem jawnym ma mieć możliwość odczytania i skopiowania tekstu zaszyfrowanego. Możliwość edycji ma być zablokowana.

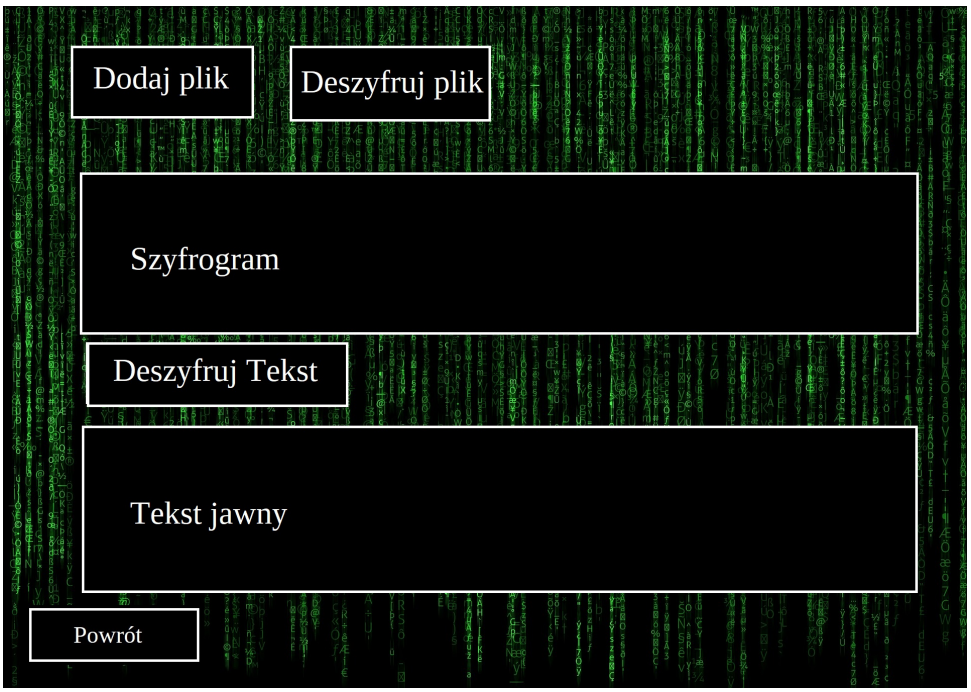
Waga średnia.

W tle okna ma się znajdować identyczne tło jak w oknie menu głównego.

Waga niska.

Kliknięcie klawisza „Esc” zostaje uruchomiony przycisk „Powrót”.

Szkic Czytanie szyfrogramu:



3.2. Wymagania niefunkcjonalne (ograniczenia):

- Klucz prywatny i liczby pierwsze muszą być zabezpieczone, usunięte z pamięci programu by nie wpadły w niepowołane ręce.
- Program ma działać sprawnie i radzić sobie z obliczeniami które umożliwia.
- Pola tekstowe mają mieć ograniczenie ze względu na ilość znaków które pomieszczą.
- Grafiki użyte w oprogramowaniu muszą pochodzić z legalnego źródła.

4. Dodatki.

4.1. Harmonogram prac nad projektem:

03.12.2018: Odczytywanie tekstu z pliku tekstowego i zwracanie zmienionej wartości.

10.12.2018: Algorytm zmieniający tekst na ciąg cyfr. Funkcja tworząca losową liczbę pierwszą w danym zakresie.

17.12.2018: Algorytm znajdujący liczby tak by tworzyć parę licz względnie pierwszych (algorytm Euklidesa).

07.01.2019: Funkcja licząca odwrotność modulo (rozszerzony algorytm Euklidesa.)

Funkcja tworząca parę kluczy.

Funkcja szyfrująca tekst.

Aplikacja działająca w konsoli, testowanie całości algorytmu, dodanie możliwości odszyfrowania tekstu.

14.01.2019: Stworzenie okna aplikacji, dodanie do niego funkcji, dotowanie wyglądu.

4.2. Zmiany w specyfikacji:

08.12.2018:

W rozdziale 4.1.

Przed:

10.12.2018: Algorytm zmieniający tekst na ciąg cyfr.

Po:

10.12.2018: Algorytm zmieniający tekst na ciąg cyfr. Funkcja tworząca losową liczbę pierwszą w danym zakresie.

15.12.2018:

W rozdziale 4.1

Przed:

17.12.2018: Algorytm szyfrujący metodą RSA.

Po:

17.12.2018: Algorytm znajdujący liczby tak by tworzyć parę licz względnie pierwszych (algorytm Euklidesa).

31.12.2018:

W rozdziale 4.1.

Przed:

07.01.2019: Interfejs graficzny

14.01.2019: Łączenie elementów w całość, poprawki.

Po:

07.01.2019: Funkcja licząca odwrotność modulo (rozszerzony algorytm Euklidesa.)

Funkcja tworząca parę kluczy.

Funkcja szyfrująca tekst.

Aplikacja działająca w konsoli, testowanie całości algorytmu, dodanie możliwości odszyfrowania tekstu.

14.01.2019: Stworzenie okna aplikacji, dodanie do niego funkcji, dotowanie wyglądu.

W rozdziale 3.1.

Przed:

3.1.5. Menu wyjście.

Waga duża.

Po kliknięciu w menu głównym przycisku „Wyjście”, okno menu głównego ma zmienić się w okno menu wyjście.

Okno ma zawierać pytanie „Czy na pewno chcesz wyjść?” rozpoczęte wielką literą i zakończone znakiem zapytania.

Pod pytaniem mają być obok siebie dwa przyciski „TAK” i „NIE”.
Pisane wielkimi literami.

Przycisk „TAK” ma wyłączać aplikacje a przycisk „NIE” ma powracać do okna menu głównego.

Przyciski mają być jednakowej wielkości umiejscowione na środku względem boków okna (orientacja pozioma).

Przyciski wraz z pytaniem mają być umiejscowione na środku względem dołu i góry (orientacja pionowa).

Waga średnia.

W tle okna ma się znajdować identyczne tło jak w oknie menu głównego.

Waga niska.

Kliknięcie klawisza „Esc” zostaje uruchomiony przycisk „Tak”.

Szkic menu Wyjście:

Po:

Rezygnacja z „Menu Wyjście”.