

Projekt zaliczeniowy z PPK-SSI

Wygenerowano przez Doxygen 1.9.3

1 Indeks klas	1
1.1 Lista klas	1
2 Indeks plików	3
2.1 Lista plików	3
3 Dokumentacja klas	5
3.1 Dokumentacja struktury Ustawienia	5
3.1.1 Opis szczegółowy	5
4 Dokumentacja plików	7
4.1 Dokumentacja pliku vigenere.h	7
4.1.1 Opis szczegółowy	8
4.1.2 Dokumentacja funkcji	8
4.1.2.1 czytajUstawienia()	8
4.1.2.2 deszyfruj()	8
4.1.2.3 odgadnijKlucz()	9
4.1.2.4 pobierzZKlucza()	9
4.1.2.5 pobierzZPliku()	10
4.1.2.6 pozycja_w_alfabecie()	10
4.1.2.7 przerobPlik()	11
4.1.2.8 rozmiarKlucza()	11
4.1.2.9 szyfruj()	12
4.2 vigenere.h	12
Indeks	15

Rozdział 1

Indeks klas

1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Ustawienia

Struktura zawiera nazwy plików: wejściowego, wyjściowego i pliku z kluczem oraz wybrany tryb. Automatycznie ustawiany jest dodatkowy tryb inne, przypisywany, gdy żaden inny nie zostanie wybrany

5

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

vigenere.h	7
--------------------------------------	---

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury Ustawienia

Struktura zawiera nazwy plików: wejściowego, wyjściowego i pliku z kluczem oraz wybrany tryb. Automatycznie ustawiany jest dodatkowy tryb inne, przypisywany, gdy żaden inny nie zostanie wybrany.

```
#include <vigenere.h>
```

Atrybuty publiczne

- std::string **sciezkaPlikuWejsciowego**
- std::string **sciezkaPlikuWyjsciowego**
- std::string **sciezkaPlikuZKluczem**
- **Flaga tryb** =Flaga::inne

3.1.1 Opis szczegółowy

Struktura zawiera nazwy plików: wejściowego, wyjściowego i pliku z kluczem oraz wybrany tryb. Automatycznie ustawiany jest dodatkowy tryb inne, przypisywany, gdy żaden inny nie zostanie wybrany.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [vigenere.h](#)

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku vigenere.h

Komponenty

- struct [Ustawienia](#)

Struktura zawiera nazwy plików: wejściowego, wyjściowego i pliku z kluczem oraz wybrany tryb. Automatycznie ustawiany jest dodatkowy tryb inne, przypisywany, gdy żaden inny nie zostanie wybrany.

Wyliczenia

- enum class [Flaga](#) { **kodowanie** , **dekodowanie** , **lamanie** , **inne** }

Unia, która przechowuje możliwe do wybrania tryby. Trybami, jakie można wybrać, są: tryb szyfrowania, dekodowania lub lamania, istnieje także jeden dodatkowy tryb inne.

Funkcje

- bool [czytajUstawienia](#) (int argc, char **argv, [Ustawienia](#) &ustawienia)

Funkcja czyta parametry podane przez użytkownika.

- void **pomoc** (const std::string &nazwaProgramu)

Funkcja wypisuje komunikat o braku lub niepoprawności wpisanych parametrów.

- std::string [pobierzZKlucza](#) (const [Ustawienia](#) &ustawienia)

Funkcja wczytuje tekst z podanego pliku wejściowego z kluczem.

- bool [przerobPlik](#) (const [Ustawienia](#) &ustawienia, std::string &klucz)

Funkcja odpowiada za wysłanie odpowiednich znaków do odpowiednich funkcji. Funkcja otwiera plik wejściowy i pobiera z niego jeden znak, jeśli nie jest on literą, zostaje od razu wczytany do pliku wyjściowego. W przeciwnym przypadku znak zostaje wysłany, razem z odpowiednim znakiem klucza do odpowiedniej funkcji: szyfrowania lub deszyfrowania, w zależności od wybranego trybu.

- int [pozycja_w_alfabecie](#) (char znak)

Funkcja znajduje pozycję w alfabecie przekazanego znaku, jest wykorzystywana w funkcji: szyfruj oraz deszyfruj.

- char [szyfruj](#) (char znak, char znak_klucza)

Funkcja znajduje wartości liczbowe danych liter i szyfruje przekazany znak.

- char [deszyfruj](#) (char znak, char znak_klucza)

Funkcja znajduje wartości liczbowe danych liter i deszyfruje przekazany znak.

- std::string [pobierzZPliku](#) (const [Ustawienia](#) &ustawienia)

Funkcja wczytuje tekst z pliku wejściowego i przygotowuje go do użycia w funkcjach potrzebnych do złamania zaszyfrowanego tekstu. Funkcja usuwa z wczytanego tekstu wszystkie znaki, które nie są literami i wczytuje je do jednego napisu.

- int `rozmiarKlucza` (std::string &zaszyfrowany)

Funkcja znajduje jaką długość ma klucz wykorzystany do zaszyfrowania tekstu. Funkcja wykorzystuje metodę Friedmana, sprawdzając przy jakim przesunięciu tekstu liczba koincydencji jest największa.

- std::string `odgadnijKlucz` (std::string &zaszyfrowany, int dlugosc_klucza)

Funkcja odgaduje jakie słowo zostało użyte jako klucz do zaszyfrowania tekstu. Funkcja wykorzystuje metodę częstości pojawiania się liter w języku angielskim - zakłada, że najczęściej powtarzająca się litera w tekście to "e".

4.1.1 Opis szczegółowy

[vigenere.h](#)

4.1.2 Dokumentacja funkcji

4.1.2.1 czytajUstawienia()

```
bool czytajUstawienia (
    int argc,
    char ** argv,
    Ustawienia & ustawienia )
```

Funkcja czyta parametry podane przez użytkownika.

Parametry

<code>argc</code>	- liczba podanych wyrazów w wierszu poleceń
<code>argv</code>	- tablica wpisanych wyrazów w wierszu poleceń
<code>ustawienia</code>	- zmienna typu <code>Ustawienia</code> , gdzie wpisywane są podawane nazwy plików

Zwraca

bool

4.1.2.2 deszyfruj()

```
char deszyfruj (
    char znak,
    char znak_klucza )
```

Funkcja znajduje wartości liczbowe danych liter i deszyfruje przekazany znak.

Parametry

<i>znak</i>	- przekazana litera tekstu zaszyfrowanego
<i>znak_klucza</i>	- przekazana litera klucza

Zwraca

char

Oto graf wywołań dla tej funkcji:



4.1.2.3 odgadnijKlucz()

```
std::string odgadnijKlucz (
    std::string & zaszyfrowany,
    int dlugosc_klucza )
```

Funkcja odgaduje jakie słowo zostało użyte jako klucz do zaszyfrowania tekstu. Funkcja wykorzystuje metodę częstości pojawiania się liter w języku angielskim - zakłada, że najczęściej powtarzająca się litera w tekście to "e".

Parametry

<i>zaszyfrowany</i>	- zaszyfrowany tekst, składający się jedynie z liter
<i>dlugosc_klucza</i>	- znaleziona długość klucza szyfrującego tekst

Zwraca

std::string

4.1.2.4 pobierzZKlucza()

```
std::string pobierzZKlucza (
    const Ustawienia & ustawienia )
```

Funkcja wczytuje tekst z podanego pliku wejściowego z kluczem.

Parametry

<i>ustawienia</i>	- zmienna przechowuje nazwe pliku wejsciowego z kluczem
-------------------	---

Zwraca

std::string

4.1.2.5 pobierzZPliku()

```
std::string pobierzZPliku (
    const Ustawienia & ustawienia )
```

Funkcja wczytuje tekst z pliku wejsciowego i przygotowuje go do uzycia w funkcjach potrzebnych do zlamania zaszyfrowanego tekstu. Funkcja usuwa z wczytanego tekstu wszystkie znaki, ktore nie sa literami i wczytuje je do jednego napisu.

Parametry

<i>ustawienia</i>	- zmienna przechowuje nazwe pliku wejsciowego
-------------------	---

Zwraca

std::string

4.1.2.6 pozycja_w_alfabecie()

```
int pozycja_w_alfabecie (
    char znak )
```

Funkcja znajduje pozycje w alfabecie przekazanego znaku, jest wykorzystywana w funkcji: szyfruj oraz deszyfruj.

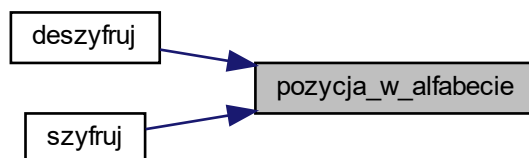
Parametry

<i>znak</i>	- przekazana litera
-------------	---------------------

Zwraca

int

Oto graf wywoływań tej funkcji:



4.1.2.7 przerobPlik()

```
bool przerobPlik (
    const Ustawienia & ustawienia,
    std::string & klucz )
```

Funkcja odpowiada za wysłanie odpowiednich znaków do odpowiednich funkcji. Funkcja otwiera plik wejściowy i pobiera z niego jeden znak, jeśli nie jest on litera, zostaje od razu wczytany do pliku wyjściowego. W przeciwnym przypadku znak zostaje wysłany, razem z odpowiednim znakiem klucza do odpowiedniej funkcji: szyfrowania lub deszyfrowania, w zależności od wybranego trybu.

Parametry

<i>ustawienia</i>	- zmienna przechowuje nazwy plików wejściowego oraz wyjściowego
<i>klucz</i>	- zmienna przechowuje napis służący za klucz do szyfrowania lub deszyfrowania tekstu

Zwraca

bool

4.1.2.8 rozmiarKlucza()

```
int rozmiarKlucza (
    std::string & zaszyfrowany )
```

Funkcja znajduje jaką długość ma klucz wykorzystany do zaszyfrowania tekstu. Funkcja wykorzystuje metodę Friedmana, sprawdzając przy jakim przesunięciu tekstu liczba koincydencji jest największa.

Parametry

<i>zaszyfrowany</i>	- zaszyfrowany tekst, składający się jedynie z liter
---------------------	--

Zwraca

int

4.1.2.9 szyfruj()

```
char szyfruj (
    char znak,
    char znak_klucza )
```

Funkcja znajduje wartości liczbowe danych liter i szyfruje przekazany znak.

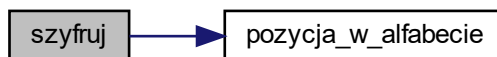
Parametry

<i>znak</i>	- przekazana litera tekstu jawnego
<i>znak_klucza</i>	- przekazana litera klucza

Zwraca

char

Oto graf wywołań dla tej funkcji:

**4.2 vigenere.h**

[Idź do dokumentacji tego pliku.](#)

```
1
2 #ifndef VIGENERE_H
3 #define VIGENERE_H
4
5
6
7
8
9 enum class Flaga
10 { kodowanie, dekodowanie, lamanie, inne};
11
12
13
14
15
16 struct Ustawienia
17 {
18     std::string sciezkaPlikuWejsciowego;
```



```
19     std::string sciezkaPlikuWyjsciowego;
20     std::string sciezkaPlikuZKluczem;
21     Flaga tryb=Flaga::inne;
22 };
23
32 bool czytajUstawienia(int argc, char** argv, Ustawienia& ustawienia);
33
37 void pomoc(const std::string& nazwaProgramu);
38
45 std::string pobierzZKlucza(const Ustawienia& ustawienia);
46
56 bool przerobPlik(const Ustawienia& ustawienia, std::string& klucz);
57
58
65 int pozycja_w_alfabecie(char znak);
66
74 char szyfruj(char znak, char znak_klucza);
75
83 char deszyfruj(char znak, char znak_klucza);
84
85
93 std::string pobierzZPliku(const Ustawienia& ustawienia);
94
102 int rozmiarKlucza(std::string& zaszyfrowany);
103
112 std::string odgadnijKlucz(std::string& zaszyfrowany, int dlugosc_klucza);
113
114 #endif
```


Indeks

czytajUstawienia
vigenere.h, [8](#)

deszyfruj
vigenere.h, [8](#)

odgadnijKlucz
vigenere.h, [9](#)

pobierzZKlucza
vigenere.h, [9](#)

pobierzZPliku
vigenere.h, [10](#)

pozycja_w_alfabecie
vigenere.h, [10](#)

przerobPlik
vigenere.h, [11](#)

rozmiarKlucza
vigenere.h, [11](#)

szyfruj
vigenere.h, [12](#)

Ustawienia, [5](#)

vigenere.h, [7](#)
 czytajUstawienia, [8](#)
 deszyfruj, [8](#)
 odgadnijKlucz, [9](#)
 pobierzZKlucza, [9](#)
 pobierzZPliku, [10](#)
 pozycja_w_alfabecie, [10](#)
 przerobPlik, [11](#)
 rozmiarKlucza, [11](#)
 szyfruj, [12](#)