Vigenere

Wygenerowano przez Doxygen 1.9.3

Rozdział 1

Indeks klas

1.1 Lista klas

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

Rozdział 2

Indeks plików

2.1 Lista plików

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

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury Params

`#include <funkcje.h>`

Atrybuty publiczne

- std::string flaga_dzialania
- std::string inputFile
- std::string kluczFile
- std::string outputFile

3.1.1 Opis szczegółowy

Struktura zawierająca dane wejściowe

3.1.2 Dokumentacja atrybutów składowych

3.1.2.1 flaga_dzialania

`std::string Params::flaga_dzialania`

flaga mówiąca o operacji, którą będziemy wykonywać (szyfrowanie, deszyfrowanie, łamanie kodu)

3.1.2.2 inputFile

`std::string Params::inputFile`

nazwa pliku wejściowego

3.1.2.3 kluczFile

```
std::string Params::kluczFile
```

nazwa pliku zawierającego klucz

3.1.2.4 outputFile

```
std::string Params::outputFile
```

nazwa pliku wyjściowego

Dokumentacja dla tej struktury została wygenerowana z pliku:

- C:/Users/user/source/repos/polsl-aei-ppk-katowice/a682f497-gr22-repo/projekt/Projekt/funkcje.h

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku C:/Users/user/source/repos/polsl-aei-ppk-katowice/a682f497-gr22-repo/projekt/Projekt/funkcje.cpp

```
#include <iostream>
#include <sstream>
#include <string>
#include <fstream>
#include <vector>
#include "funkcje.h"
#include <map>
#include <unordered_map>
#include <cctype>
#include <algorithm>
```

Funkcje

- bool czytaj_Parametry (int argc, char **argv, Params &params)
- void pomoc (const std::string &progName)
- void szyfrowanie_znaku (char &znak_tekstu, const std::string &klucz, int &licznik_klucza)
- void deszyfrowanie_znaku (std::string klucz, char &znak_tekstu, int &licznik_klucza)
- std::string szyfrowanie (const Params &params)
- std::string wczytaj_plik (std::string nazwa_pliku)
- std::string deszyfrowanie (Params &params)
- std::vector< std::string > wyznaczanie_ciagow (std::string &tekst)
- int NWD (int a, int b)
- bool komparator_klucza (const std::pair< int, int > &a, const std::pair< int, int > &b)
- bool komparator_wartosci (const std::pair< int, int > &a, const std::pair< int, int > &b)
- std::vector< std::pair< int, int > > sortuj (std::map< int, int > &mapa)
- int NWDv2 (std::vector< int > &liczby)
- std::vector< int > wyznaczanie_roznic (std::unordered_map< std::string, std::vector< int > > &slowa)
- int wyznaczanie_dlugosci_klucza (std::vector< std::string > &ciagi)
- std::string filtrowanie_tekstu (std::string tekst)
- char wyznacz_znak_klucza (std::string &tekst, int dlugosc_klucza, int poz_klucza)
- std::string wyznaczanie_klucza (std::string &tekst, int dlugosc_klucza)
- std::string lamanie_kodu (Params &params)
- bool decyzjator (Params &params)

4.1.1 Dokumentacja funkcji

4.1.1.1 czytaj_Parametry()

```
bool czytaj_Parametry (
            int argc,
            char ** argv,
            Params & params )
```

Funkcja zczytująca parametry do programu i zapisująca je do struktury

Parametry

argc	wartości parametrów z którymi został wywołany program
argv	tablica tych wartości
params	struktura do której zapisywane są parametry

Zwraca

funkcja zwraca true, jeżeli program został wywołany z odpowiednimi parametrami. W przeciwnym razie zwraca false

4.1.1.2 decyzjator()

```
bool decyzjator (
            Params & params )
```

Główna funkcja sterująca. Decyduje, którą z operacji będziemy wykonywać na podstawie parametrów wejściowych

Parametry

params	parametry wejściowe

Zwraca

funkcja zwraca true jeśli była odpowiednia flaga działania i udało się wejść do odpowiedniego segmentu funkcji związanego z tą flagą. W przeciwnym razie zwraca false

4.1.1.3 deszyfrowanie()

```
std::string deszyfrowanie (
            Params & params )
```

Główna funkcja deszyfrująca

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca odszyfrowany tekst

4.1.1.4 deszyfrowanie_znaku()

```
void deszyfrowanie_znaku (
            std::string klucz,
            char & znak_tekstu,
            int & licznik_klucza )
```

Funkcja deszyfruje znak tekstu kolejnym znakiem klucza i jeśli znak dało się zdeszyfrować (tzn. a-z, A-z), to przypisuje go wartości oryginału

Parametry

klucz	klucz którym deszyfrujemy
znak_tekstu	kolejny znak tekstu do zdeszyfrowania
licznik_klucza	kolejny znak klucza, a gdy dojdziemy do jego końca, bierze się go od początku

4.1.1.5 filtrowanie_tekstu()

```
std::string filtrowanie_tekstu (
            std::string tekst )
```

Funkcja wyznacza tekst składający się z samych znaków A-Z. Duże litery zostawia, małe litery zamienia na duże, a resztę znaków pomija

Parametry

tekst	tekst do przefiltrowania

Zwraca

funkcja zwraca przefiltrowany tekst

4.1.1.6 komparator_klucza()

```
bool komparator_klucza (
            const std::pair< int, int > & a,
            const std::pair< int, int > & b )
```

Funkcja porównująca pary na podstawie klucza

Parametry

a	
b	

Zwraca

funkcja zwraca true jeśli a.first > b.first, w przeciwnym razie false

4.1.1.7 komparator_wartosci()

```
bool komparator_wartosci (
            const std::pair< int, int > & a,
            const std::pair< int, int > & b )
```

Funkcja porównująca pary na podstawie wartości

Parametry

a	
b	

Zwraca

funkcja zwraca true jeśli a.second > b.second, w przeciwnym razie false

4.1.1.8 lamanie_kodu()

```
std::string lamanie_kodu (
            Params & params )
```

Główna funkcja do łamania szyfru

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca odszyfrowany tekst

4.1.1.9 NWD()

```
int NWD (
            int a,
            int b )
```

Funkcja wyznacza nwd rekurencyjnie dwóch liczb metodą euklidesa

Parametry

a	pierwsza liczba
b	druga liczba

Zwraca

funkcja zwraca nwd tych dwóch liczb

4.1.1.10 NWDv2()

```
int NWDv2 (
            std::vector< int > & liczby )
```

Funkcja wyznacza statystecznie największą liczbę, dla której ilość powstałych nwd jest największa

Parametry

liczby	wyniki operacji nwd dla wszystkich ciągów

Zwraca

funkcja zwraca jedno statystycznie najczęstsze i największe nwd

4.1.1.11 pomoc()

```
void pomoc (
            const std::string & progName )
```

Funkcja wyświetla okno pomocy, gdy czytaj_Parametry zwróci false

Parametry

progName	nazwa programu

4.1.1.12 sortuj()

```
std::vector< std::pair< int, int > > sortuj (
            std::map< int, int > & mapa )
```

Funkcja zamienia mapę na wektor i sortuje go

Parametry

mapa	mapa do posortowania

Zwraca

funkcja zwraca posortowanego wektora, utworzonego z mapy

4.1.1.13 szyfrowanie()

```
std::string szyfrowanie (
            const Params & params )
```

Główna funkcja szyfrująca

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca zaszyfrowany tekst

4.1.1.14 szyfrowanie_znaku()

```
void szyfrowanie_znaku (
            char & znak_tekstu,
            const std::string & klucz,
            int & licznik_klucza )
```

Funkcja szyfruje znak tekstu kolejnym znakiem klucza i jeśli znak dało się zaszyfrować (tzn. a-z, A-Z), to przypisuje go wartości oryginału

Parametry

znak_tekstu	kolejny znak tekstu do zaszyfrowania
klucz	klucz którym szyfrujemy
licznik_klucza	kolejny znak klucza, a gdy dojdziemy do jego końca, bierze się go od początku

4.1.1.15 wczytaj_plik()

```
std::string wczytaj_plik (
            std::string nazwa_pliku )
```

Funkcja wczytuje plik o nazwie z parametrów wejsciowych

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca tekst z tego pliku

4.1.1.16 wyznacz_znak_klucza()

```
char wyznacz_znak_klucza (
            std::string & tekst,
            int dlugosc_klucza,
            int poz_klucza )
```

Funkcja wyznacza literę klucza na podstawie co n-tej litery (n-długość klucza) i wyznacza najczęściej powtarzającą się. W języku polskim jest to "a", więc na tej podstawie obliczamy przesunięcie

Parametry

tekst	tekst głowny z którego zczytujemy litery
dlugosc_klucza	długość klucza
poz_klucza	zmienna opisująca którego znaku klucza szukamy, co którą literę wczytujemy

Zwraca

funkcja zwraca literę klucza

4.1.1.17 wyznaczanie_ciagow()

```
std::vector< std::string > wyznaczanie_ciagow (
            std::string & tekst )
```

Funkcja wyznacza trzyliterowe ciągi

Parametry

tekst	tekst, który będziemy analizować

Zwraca

funkcja zwraca te ciągi

4.1.1.18 wyznaczanie_dlugosci_klucza()

```
int wyznaczanie_dlugosci_klucza (
            std::vector< std::string > & ciagi )
```

Funkcja wyznacza długość klucza na podstawie największej ilości powtórzeń nwd dla danego ułożenia różnic odległości miedzy powtarzającymi się ciągami, oraz zmienia tą kolejność dla dokładniejszego wyniku

Parametry

ciagi	3 literowe ciągi ułożone z kolejnych liter zaszyfrowanego tekstu

Zwraca

funkcja zwraca długość klucza

4.1.1.19 wyznaczanie_klucza()

```
std::string wyznaczanie_klucza (
            std::string & tekst,
            int dlugosc_klucza )
```

Funkcja wyznacza klucz na podstawie jego długości. Sprawdza co n-tą literę (n-długość klucza) i wyznacza najczęściej powtarzającą się. W języku polskim jest to "a", więc na tej podstawie obliczamy przesunięcie

Parametry

tekst	tekst głowny z którego zczytujemy litery
dlugosc_klucza	długość klucza

Zwraca

funkcja zwraca klucz

4.1.1.20 wyznaczanie_roznic()

```
std::vector< int > wyznaczanie_roznic (
            std::unordered_map< std::string, std::vector< int > > & slowa )
```

Funkcja wyznacza różnice między pierwszym i każdym kolejnym wystąpieniem poszczególnego 3-literowego ciągu

Parametry

slowa	mapa tych 3-literowych ciągów (<ciąg>, <pozycje wystąpienia>)

Zwraca

funkcja zwraca różnice pomiędzy pierwszym i każdym kolejnym wystąpieniem poszczególnego 3-literowego ciągu

4.2 Dokumentacja pliku C:/Users/user/source/repos/polsl-aei-ppk-katowice/a682f497-gr22-repo/projekt/Projekt/funkcje.h

```
#include <iostream>
#include <string>
#include <vector>
#include <cctype>
#include <unordered_map>
#include <map>
```

Komponenty

- struct Params

Funkcje

- bool czytaj_Parametry (int argc, char **argv, Params &params)
- void pomoc (const std::string &progName)
- void szyfrowanie_znaku (char &znak_tekstu, const std::string &klucz, int &licznik_klucza)
- void deszyfrowanie_znaku (std::string klucz, char &znak_tekstu, int &licznik_klucza)
- std::string szyfrowanie (const Params &params)
- std::string wczytaj_plik (std::string nazwa_pliku)
- std::string deszyfrowanie (Params &params)
- std::vector< std::string > wyznaczanie_ciagow (std::string &tekst)
- int NWD (int a, int b)

- int NWDv2 (std::vector< int > &liczby)
- int wyznaczanie_dlugosci_klucza (std::vector< std::string > &ciagi)
- std::string filtrowanie_tekstu (std::string tekst)
- std::string lamanie_kodu (Params &params)
- bool decyzjator (Params &params)
- std::vector< int > wyznaczanie_roznic (std::unordered_map< std::string, std::vector< int > > &slowa)
- std::string wyznaczanie_klucza (std::string &tekst, int dlugosc_klucza)
- char wyznacz_znak_klucza (std::string &tekst, int dlugosc_klucza, int poz_klucza)
- bool komparator_wartosci (const std::pair< int, int > &a, const std::pair< int, int > &b)
- bool komparator_klucza (const std::pair< int, int > &a, const std::pair< int, int > &b)
- std::vector< std::pair< int, int > > sortuj (std::map< int, int > &mapa)

4.2.1 Dokumentacja funkcji

4.2.1.1 czytaj_Parametry()

```
bool czytaj_Parametry (
            int argc,
            char ** argv,
            Params & params )
```

Funkcja zczytująca parametry do programu i zapisująca je do struktury

Parametry

argc	wartości parametrów z którymi został wywołany program
argv	tablica tych wartości
params	struktura do której zapisywane są parametry

Zwraca

funkcja zwraca true, jeżeli program został wywołany z odpowiednimi parametrami. W przeciwnym razie zwraca false

4.2.1.2 decyzjator()

```
bool decyzjator (
            Params & params )
```

Główna funkcja sterująca. Decyduje, którą z operacji będziemy wykonywać na podstawie parametrów wejściowych

Parametry

params	parametry wejściowe

Zwraca

funkcja zwraca true jeśli była odpowiednia flaga działania i udało się wejść do odpowiedniego segmentu funkcji związanego z tą flagą. W przeciwnym razie zwraca false

4.2.1.3 deszyfrowanie()

```
std::string deszyfrowanie (
            Params & params )
```

Główna funkcja deszyfrująca

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca odszyfrowany tekst

4.2.1.4 deszyfrowanie_znaku()

```
void deszyfrowanie_znaku (
            std::string klucz,
            char & znak_tekstu,
            int & licznik_klucza )
```

Funkcja deszyfruje znak tekstu kolejnym znakiem klucza i jeśli znak dało się zdeszyfrować (tzn. a-z, A-z), to przypisuje go wartości oryginału

Parametry

klucz	klucz którym deszyfrujemy
znak_tekstu	kolejny znak tekstu do zdeszyfrowania
licznik_klucza	kolejny znak klucza, a gdy dojdziemy do jego końca, bierze się go od początku

4.2.1.5 filtrowanie_tekstu()

```
std::string filtrowanie_tekstu (
            std::string tekst )
```

Funkcja wyznacza tekst składający się z samych znaków A-Z. Duże litery zostawia, małe litery zamienia na duże, a resztę znaków pomija

Parametry

tekst	tekst do przefiltrowania

Zwraca

funkcja zwraca przefiltrowany tekst

4.2.1.6 komparator_klucza()

```
bool komparator_klucza (
            const std::pair< int, int > & a,
            const std::pair< int, int > & b )
```

Funkcja porównująca pary na podstawie klucza

Parametry

a	
b	

Zwraca

funkcja zwraca true jeśli a.first > b.first, w przeciwnym razie false

4.2.1.7 komparator_wartosci()

```
bool komparator_wartosci (
            const std::pair< int, int > & a,
            const std::pair< int, int > & b )
```

Funkcja porównująca pary na podstawie wartości

Parametry

a	
b	

Zwraca

funkcja zwraca true jeśli a.second > b.second, w przeciwnym razie false

4.2.1.8 lamanie_kodu()

```
std::string lamanie_kodu (
            Params & params )
```

Główna funkcja do łamania szyfru

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca odszyfrowany tekst

4.2.1.9 NWD()

```
int NWD (
            int a,
            int b )
```

Funkcja wyznacza nwd rekurencyjnie dwóch liczb metodą euklidesa

Parametry

a	pierwsza liczba
b	druga liczba

Zwraca

funkcja zwraca nwd tych dwóch liczb

4.2.1.10 NWDv2()

```
int NWDv2 (
            std::vector< int > & liczby )
```

Funkcja wyznacza statystecznie największą liczbę, dla której ilość powstałych nwd jest największa

Parametry

liczby	wyniki operacji nwd dla wszystkich ciągów

Zwraca

funkcja zwraca jedno statystycznie najczęstsze i największe nwd

4.2.1.11 pomoc()

```
void pomoc (
            const std::string & progName )
```

Funkcja wyświetla okno pomocy, gdy czytaj_Parametry zwróci false

Parametry

progName	nazwa programu

4.2.1.12 sortuj()

```
std::vector< std::pair< int, int > > sortuj (
            std::map< int, int > & mapa )
```

Funkcja zamienia mapę na wektor i sortuje go

Parametry

mapa	mapa do posortowania

Zwraca

funkcja zwraca posortowanego wektora, utworzonego z mapy

4.2.1.13 szyfrowanie()

```
std::string szyfrowanie (
            const Params & params )
```

Główna funkcja szyfrująca

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca zaszyfrowany tekst

4.2.1.14 szyfrowanie_znaku()

```
void szyfrowanie_znaku (
            char & znak_tekstu,
            const std::string & klucz,
            int & licznik_klucza )
```

Funkcja szyfruje znak tekstu kolejnym znakiem klucza i jeśli znak dało się zaszyfrować (tzn. a-z, A-Z), to przypisuje go wartości oryginału

Parametry

znak_tekstu	kolejny znak tekstu do zaszyfrowania
klucz	klucz którym szyfrujemy
licznik_klucza	kolejny znak klucza, a gdy dojdziemy do jego końca, bierze się go od początku

4.2.1.15 wczytaj_plik()

```
std::string wczytaj_plik (
            std::string nazwa_pliku )
```

Funkcja wczytuje plik o nazwie z parametrów wejsciowych

Parametry

params	parametry wejsciowe

Zwraca

funkcja zwraca tekst z tego pliku

4.2.1.16 wyznacz_znak_klucza()

```
char wyznacz_znak_klucza (
            std::string & tekst,
            int dlugosc_klucza,
            int poz_klucza )
```

Funkcja wyznacza literę klucza na podstawie co n-tej litery (n-długość klucza) i wyznacza najczęściej powtarzającą się. W języku polskim jest to "a", więc na tej podstawie obliczamy przesunięcie

Parametry

tekst	tekst głowny z którego zczytujemy litery
dlugosc_klucza	długość klucza
poz_klucza	zmienna opisująca którego znaku klucza szukamy, co którą literę wczytujemy

Zwraca

funkcja zwraca literę klucza

4.2.1.17 wyznaczanie_ciagow()

```
std::vector< std::string > wyznaczanie_ciagow (
            std::string & tekst )
```

Funkcja wyznacza trzyliterowe ciągi

Parametry

tekst	tekst, który będziemy analizować

Zwraca

funkcja zwraca te ciągi

4.2.1.18 wyznaczanie_dlugosci_klucza()

```
int wyznaczanie_dlugosci_klucza (
            std::vector< std::string > & ciagi )
```

Funkcja wyznacza długość klucza na podstawie największej ilości powtórzeń nwd dla danego ułożenia różnic odległości miedzy powtarzającymi się ciągami, oraz zmienia tą kolejność dla dokładniejszego wyniku

Parametry

ciagi	3 literowe ciągi ułożone z kolejnych liter zaszyfrowanego tekstu

Zwraca

funkcja zwraca długość klucza

4.2.1.19 wyznaczanie_klucza()

```
std::string wyznaczanie_klucza (
            std::string & tekst,
            int dlugosc_klucza )
```

Funkcja wyznacza klucz na podstawie jego długości. Sprawdza co n-tą literę (n-długość klucza) i wyznacza najczęściej powtarzającą się. W języku polskim jest to "a", więc na tej podstawie obliczamy przesunięcie

Parametry

tekst	tekst głowny z którego zczytujemy litery
dlugosc_klucza	długość klucza

Zwraca

funkcja zwraca klucz

4.2.1.20 wyznaczanie_roznic()

```
std::vector< int > wyznaczanie_roznic (
            std::unordered_map< std::string, std::vector< int > > & slowa )
```

Funkcja wyznacza różnice między pierwszym i każdym kolejnym wystąpieniem poszczególnego 3-literowego ciągu

Parametry

slowa	mapa tych 3-literowych ciągów (<ciąg>, <pozycje wystąpienia>)

Zwraca

funkcja zwraca różnice pomiędzy pierwszym i każdym kolejnym wystąpieniem poszczególnego 3-literowego ciągu

4.3 funkcje.h

Idź do dokumentacji tego pliku.

```
1
2 #pragma once
3
4 #include<iostream>
5 #include<string>
6 #include<vector>
7 #include <cctype>
8 #include <unordered_map>
9 #include <map>
10
12 struct Params
13 {
14     std::string flaga_dzialania;
15     std::string inputFile;
16     std::string kluczFile;
```

```
17     std::string outputFile;
18 };
19
26 bool czytaj_Parametry(int argc, char** argv, Params& params);
27
31 void pomoc(const std::string& progName);
32
38 void szyfrowanie_znaku(char &znak_tekstu, const std::string &klucz, int& licznik_klucza);
39
45 void deszyfrowanie_znaku(std::string klucz, char& znak_tekstu, int& licznik_klucza);
46
51 std::string szyfrowanie(const Params& params);
52
57 std::string wczytaj_plik(std::string nazwa_pliku);
58
63 std::string deszyfrowanie(Params &params);
64
69 std::vector<std::string> wyznaczanie_ciagow(std::string &tekst);
70
76 int NWD(int a, int b);
77
82 int NWDv2(std::vector<int>& liczby);
83
88 int wyznaczanie_dlugosci_klucza(std::vector<std::string>&ciagi);
89
94 std::string filtrowanie_tekstu(std::string tekst);
95
100 std::string lamanie_kodu(Params &params);
101
106 bool decyzjator( Params& params);
107
112 std::vector<int>wyznaczanie_roznic(std::unordered_map<std::string, std::vector<int» &slowa);
113
119 std::string wyznaczanie_klucza(std::string &tekst, int dlugosc_klucza);
120
127 char wyznacz_znak_klucza(std::string &tekst, int dlugosc_klucza, int poz_klucza);
128
134 bool komparator_wartosci(const std::pair<int, int>& a, const std::pair<int, int>& b);
135
141 bool komparator_klucza(const std::pair<int, int>& a, const std::pair<int, int>& b);
142
147 std::vector<std::pair<int, int»sortuj(std::map<int, int> &mapa);
```

4.4 Dokumentacja pliku C:/Users/user/source/repos/polsl-aei-ppk-katowice/a682f497-gr22-repo/projekt/Projekt/main.cpp

```
#include <iostream>
#include <string>
#include <fstream>
#include "funkcje.h"
```

Funkcje

- int **main** (int argc, char ∗∗argv)

Indeks