

Projekt poczty elektronicznej

Wygenerowano przez Doxygen 1.9.1

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Dokumentacja klas	5
3.1 Dokumentacja klasy Files	5
3.1.1 Opis szczegółowy	5
3.1.2 Dokumentacja konstruktora i destruktora	5
3.1.2.1 Files()	5
3.1.3 Dokumentacja funkcji składowych	6
3.1.3.1 ifFileExists()	6
3.1.3.2 reWriteContentFile()	6
3.2 Dokumentacja klasy Projekt::Login	6
3.2.1 Opis szczegółowy	7
3.2.2 Dokumentacja konstruktora i destruktora	7
3.2.2.1 Login()	7
3.2.2.2 ~Login()	7
3.3 Dokumentacja struktury ReadMails::Mail	7
3.3.1 Opis szczegółowy	8
3.4 Dokumentacja klasy Projekt::POP3	8
3.4.1 Opis szczegółowy	9
3.4.2 Dokumentacja konstruktora i destruktora	9
3.4.2.1 POP3()	9
3.4.2.2 ~POP3()	9
3.4.3 Dokumentacja funkcji składowych	9
3.4.3.1 convertBtokB()	9
3.4.3.2 ReplaceAll()	10
3.4.3.3 toPolishWords()	10
3.5 Dokumentacja klasy POP3Login	10
3.5.1 Opis szczegółowy	11
3.5.2 Dokumentacja konstruktora i destruktora	11
3.5.2.1 POP3Login()	11
3.5.3 Dokumentacja funkcji składowych	11
3.5.3.1 loginPOP3()	11
3.6 Dokumentacja klasy ReadMails	12
3.6.1 Opis szczegółowy	12
3.6.2 Dokumentacja konstruktora i destruktora	12
3.6.2.1 ReadMails()	12
3.6.3 Dokumentacja funkcji składowych	12
3.6.3.1 downloadLastMails()	12
3.6.3.2 getMailInfo()	13

3.7 Dokumentacja klasy Projekt::SMTP	13
3.7.1 Opis szczegółowy	14
3.7.2 Dokumentacja konstruktora i destruktora	14
3.7.2.1 SMTP()	14
3.7.2.2 ~SMTP()	14
3.8 Dokumentacja klasy SMTPLogin	15
3.8.1 Opis szczegółowy	15
3.8.2 Dokumentacja konstruktora i destruktora	15
3.8.2.1 SMTPLogin()	15
3.8.3 Dokumentacja funkcji składowych	15
3.8.3.1 loginSMTP()	16
3.9 Dokumentacja klasy WriteMail	16
3.9.1 Opis szczegółowy	16
3.9.2 Dokumentacja konstruktora i destruktora	16
3.9.2.1 WriteMail()	17
3.9.3 Dokumentacja funkcji składowych	17
3.9.3.1 writeMail()	17
Indeks	19

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Files	5
System::Windows::Forms::Form	
Projekt::Login	6
Projekt::POP3	8
Projekt::SMTP	13
ReadMails::Mail	7
POP3Login	10
ReadMails	12
SMTPLogin	15
WriteMail	16

Rozdział 2

Indeks klas

2.1 Lista klas

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

Files	5
Projekt::Login	6
ReadMails::Mail	7
Projekt::POP3	8
POP3Login	10
ReadMails	12
Projekt::SMTP	13
SMTPLogin	15
WriteMail	16

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy Files

```
#include <Files.h>
```

Metody publiczne

- `Files` (string `fileName`)
- bool `ifFileExists` ()
- void `reWriteContentFile` (const string &content)

Atrybuty publiczne

- string `fileName`
Nazwa pliku.

3.1.1 Opis szczegółowy

Klasa obsługująca strumienie plików

W tej klasie możemy sprawdzić czy istnieje plik o danej nazwie metoda (`ifFileExists()`) oraz możemy wkonać zapis do pliku metoda (`reWriteContentFile()`).

3.1.2 Dokumentacja konstruktora i destruktor

3.1.2.1 Files()

```
Files::Files (
    string fileName )
```

Konstruktor dla klasy `Files`

Parametry

<code>fileName</code>	- Nazwa pliku podawana do konstruktora
-----------------------	--

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 `ifFileExists()`

```
bool Files::ifFileExists ( )
```

Metoda `ifFileExists()` sprawdza czy istnieje plik o danej nazwie, który został wysłany wcześniej do konstruktora

Zwraca

- Funkcja zwraca 1 jeśli plik już istnieje, 0 jeśli plik nie istnieje.

3.1.3.2 `reWriteContentFile()`

```
void Files::reWriteContentFile (
    const string & content )
```

Metoda `reWriteContentFile()` zapisuje do pliku o danej nazwie kontent, który przyjmuje jako argument

Parametry

<code>content</code>	- Kontent który będzie zapisywany do pliku
----------------------	--

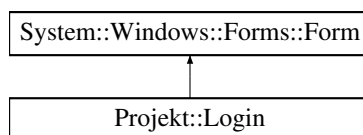
Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Files.↔h
- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Files.↔cpp

3.2 Dokumentacja klasy `Projekt::Login`

```
#include <Login.h>
```

Diagram dziedziczenia dla `Projekt::Login`



Metody publiczne

- [Login](#) (void)

Metody chronione

- [~Login](#) ()

3.2.1 Opis szczegółowy

Klasa główna odpowiedzialna za logowanie do danego protokołu [POP3](#) / [SMTP](#)

3.2.2 Dokumentacja konstruktora i destruktora

3.2.2.1 Login()

```
Projekt::Login::Login (
    void ) [inline]
```

Konstruktor dla klasy [Login](#), który inicjuje wszystkie niezbędne komponenty

3.2.2.2 ~Login()

```
Projekt::Login::~~Login ( ) [inline], [protected]
```

Destruktor dla klasy [Login](#), który usuwa wszystkie utworzone wcześniej komponenty

Dokumentacja dla tej klasy została wygenerowana z pliku:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Login.↔h

3.3 Dokumentacja struktury ReadMails::Mail

```
#include <ReadMails.h>
```


Atrybuty publiczne

- time_t `receivedTime`
Data otrzymania wiadomosci.
- string `sender`
Nastepujaco: osoba wysylajaca, temat, wiadomosc, typ wiadomosci.
- string `subject`
- string `content`
- string `getContentType`
- int `sizeInBytes`
Rozmiar w bajtach [B].
- int `mailId`
Id otrzymanej wiadomosci.

3.3.1 Opis szczegółowy

Struktura `Mail` przechowujące odpowiednie zmienne w celu uzyskania listy wiadomosci

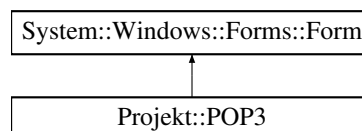
Dokumentacja dla tej struktury została wygenerowana z pliku:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Read↔ Mails.h

3.4 Dokumentacja klasy Projekt::POP3

```
#include <POP3.h>
```

Diagram dziedziczenia dla Projekt::POP3



Metody publiczne

- void `ReplaceAll` (string &s, const string &search, const string &replace)
- string `toPolishWords` (string myWord)
- int `convertBtokB` (const int &value)
- `POP3` (TextBox^ `serverAddress`, TextBox^ `userName`, TextBox^ `password`, int `port`)

Atrybuty publiczne

- TextBox `serverAddress`
Textbox adresu serwera POP3
- TextBox `userName`
Textbox nazwy uzytkownika
- TextBox `password`
Textbox hasla uzytkownika
- int `port`
Port serwera POP3.

Metody chronione

- `~POP3()`

3.4.1 Opis szczegółowy

Klasa `POP3` odpowiedzialna za wyswietlanie wiadomosci, dzialajaca za pomoca protokolu `POP3`

3.4.2 Dokumentacja konstruktora i destruktora

3.4.2.1 POP3()

```
Projekt::POP3::POP3 (
    TextBox^ serverAddress,
    TextBox^ userName,
    TextBox^ password,
    int port ) [inline]
```

Konstruktor dla klasy `POP3`, który inicjuje wszystkie niezbędne komponenty oraz zapisuje odczytanie wiadomosci do listy

3.4.2.2 ~POP3()

```
Projekt::POP3::~~POP3 ( ) [inline], [protected]
```

Destruktor dla klasy `POP3`, który usuwa wszystkie utworzone wcześniej komponenty

3.4.3 Dokumentacja funkcji składowych

3.4.3.1 convertBtokB()

```
int Projekt::POP3::convertBtokB (
    const int & value ) [inline]
```

Funkcja `convertBtokB` zamienia wartosci podane w bajtach [B] na kilobajty [kB]

Parametry

<code>value</code>	- Wartosc podana w bajtach [B]
--------------------	--------------------------------

Zwraca

- Funkcja zwraca wartosc przekonwertowana na kilobajty [kB]

3.4.3.2 ReplaceAll()

```
void Projekt::POP3::ReplaceAll (
    string & s,
    const string & search,
    const string & replace ) [inline]
```

Funkcja `Replace_All` zamienia dana litere na inna

Parametry

<i>s</i>	- Cale slowo
<i>search</i>	- Szukana litera
<i>replace</i>	- Litera na ktora ma zostac zmieniona powyzsza litera <i>serach</i>

Zwraca

- Funkcja zwraca zamienione slowo, uwzgledniajac powyzsze warunki poszczegolnych liter.

3.4.3.3 toPolishWords()

```
string Projekt::POP3::toPolishWords (
    string myWord ) [inline]
```

Funkcja `toPolishWords` zamienia wartosci ktore sa nie czytelne do programu na polskie znaki

Parametry

<i>myWord</i>	- Slowo ktore bedzie zmieniane
---------------	--------------------------------

Zwraca

- Funkcja zwraca zamienione slowo

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/POP3.↵
h

3.5 Dokumentacja klasy POP3Login

```
#include <POP3Login.h>
```

Metody publiczne

- `POP3Login` (string `serverAddress`, string `userName`, string `password`, int `port=110`)
- string `loginPOP3` ()

Atrybuty publiczne

- string `serverAddress`
Adresu serwera POP3.
- string `userName`
Nazwa użytkownika.
- string `password`
Hasło użytkownika.
- int `port`
Port serwera POP3.

3.5.1 Opis szczegółowy

Klasa `POP3Login` odpowiedzialna za probe logowania do protokołu POP3

3.5.2 Dokumentacja konstruktora i destruktora

3.5.2.1 POP3Login()

```
POP3Login::POP3Login (
    string serverAddress,
    string userName,
    string password,
    int port = 110 )
```

Konstruktor `POP3Login` przechowujące dane do logowania

3.5.3 Dokumentacja funkcji składowych

3.5.3.1 loginPOP3()

```
string POP3Login::loginPOP3 ( )
```

Metoda `loginPOP3()` odpowiedzialna za probe logowania do protokołu POP3

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/POP3↔Login.h
- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/POP3↔Login.cpp

3.6 Dokumentacja klasy ReadMails

```
#include <ReadMails.h>
```

Komponenty

- struct [Mail](#)

Metody publiczne

- [ReadMails](#) (string [serverAddress](#), string [userName](#), string [password](#), int [port](#)=110)
- [Mail](#) [getMailInfo](#) (POP3ClientSession &session, int messageId)
- vector< [Mail](#) > [downloadLastMails](#) (int numberOfMails=3)

Atrybuty publiczne

- string [serverAddress](#)
Adresu serwera POP3.
- string [userName](#)
Nazwa uzytkownika.
- string [password](#)
Haslo uzytkownika.
- int [port](#)
Port serwera POP3.

3.6.1 Opis szczegółowy

Klasa [ReadMails](#) odpowiedzialna za odbieranie listy wiadomosci z poczty za pomoca protokolu POP3

3.6.2 Dokumentacja konstruktora i destruktor

3.6.2.1 ReadMails()

```
ReadMails::ReadMails (
    string serverAddress,
    string userName,
    string password,
    int port = 110 )
```

Konstruktor [ReadMails](#) przechowuje dane do logowania oraz odczytywania listy wiadomosci z poczty

3.6.3 Dokumentacja funkcji składowych

3.6.3.1 downloadLastMails()

```
vector< ReadMails::Mail > ReadMails::downloadLastMails (
    int numberOfMails = 3 )
```

Metoda [downloadLastMails\(\)](#) działająca na powyższej strukturze oraz na wektorze zwraca wektor wszystkich wiadomosci

Parametry

<i>numberOfMails</i>	- ilość wiadomości którą chcemy zobaczyć
----------------------	--

Zwraca

- zwraca wektor wiadomości

3.6.3.2 getMailInfo()

```
ReadMails::Mail ReadMails::getMailInfo (
    POP3ClientSession & session,
    int messageId )
```

Metoda `getMailInfo()` działająca na powyższej strukturze zwraca informacje o danej wiadomości

Parametry

<i>session</i>	- sesja na której działamy dla protokołu POP3
<i>messageId</i>	- id wiadomości z której chcemy otrzymać wszystkie niezbędne informacje

Zwraca

- zwraca nam wszystkie niezbędne informacje o danym mailu przekaza przez `messageId`

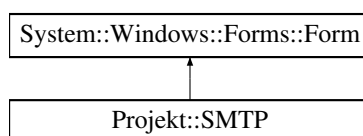
Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Read↔Mails.h
- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Read↔Mails.cpp

3.7 Dokumentacja klasy Projekt::SMTP

```
#include <SMTP.h>
```

Diagram dziedziczenia dla Projekt::SMTP



Metody publiczne

- [SMTP](#) (TextBox^ [serverAddress](#), TextBox^ [userName](#), TextBox^ [password](#), int [port](#))

Atrybuty publiczne

- TextBox [serverAddress](#)
Textbox adresu serwera POP3
- TextBox [userName](#)
Textbox nazwy uzytkownika
- TextBox [password](#)
Textbox hasla uzytkownika
- int [port](#)
Port serwera POP3.

Metody chronione

- [~SMTP](#) ()

3.7.1 Opis szczegółowy

Klasa [SMTP](#) odpowiedzialna za wysylanie wiadomosci, dzialajaca za pomoca protokolu [SMTP](#)

3.7.2 Dokumentacja konstruktora i destruktora

3.7.2.1 SMTP()

```
Projekt::SMTP::SMTP (
    TextBox^ serverAddress,
    TextBox^ userName,
    TextBox^ password,
    int port ) [inline]
```

Konstruktor dla klasy [SMTP](#), który inicjuje wszystkie niezbędne komponenty oraz zapisuje niezbędne dane do wysyłania wiadomości, zmienia tytuł dla formularza (Zaznacza na jakim emailu jesteśmy zalogowani)

3.7.2.2 ~SMTP()

```
Projekt::SMTP::~~SMTP ( ) [inline], [protected]
```

Destruktor dla klasy [SMTP](#), który usuwa wszystkie utworzone wcześniej komponenty

Dokumentacja dla tej klasy została wygenerowana z pliku:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/SMTP.↔h

3.8 Dokumentacja klasy SMTPLogin

```
#include <SMTPLogin.h>
```

Metody publiczne

- [SMTPLogin](#) (string [smtpserverAddress](#), string [userName](#), string [password](#), int [port](#)=25)
- string [loginSMTP](#) ()

Atrybuty publiczne

- string [smtpserverAddress](#)
Adresu serwera SMTP.
- string [userName](#)
Nazwa użytkownika.
- string [password](#)
Hasło użytkownika.
- int [port](#)
Port serwera SMTP.

3.8.1 Opis szczegółowy

Klasa [SMTPLogin](#) odpowiedzialna za próbe logowania do protokołu SMTP

3.8.2 Dokumentacja konstruktora i destruktora

3.8.2.1 SMTPLogin()

```
SMTPLogin::SMTPLogin (  
    string smtpserverAddress,  
    string userName,  
    string password,  
    int port = 25 ) [inline]
```

Konstruktor [SMTPLogin](#) przechowujące dane do logowania

3.8.3 Dokumentacja funkcji składowych

3.8.3.1 loginSMTP()

```
string SMTPLogin::loginSMTP ( )
```

Metoda `loginSMTP()` odpowiedzialna za probe logowania do protokolu SMTP

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/SMTPLogin.↵
h
- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/SMTPLogin.↵
cpp

3.9 Dokumentacja klasy WriteMail

```
#include <WriteMail.h>
```

Metody publiczne

- `WriteMail` (string `smtpserverAddress`, string `userName`, string `password`, int `port`=25)
- string `writeMail` (string `toAdressEmailSend`, string `titleToSend`, string `messageToSend`)

Atrybuty publiczne

- string `smtpserverAddress`
Adresu serwera SMTP.
- string `userName`
Nazwa uzytkownika.
- string `password`
Haslo uzytkownika.
- int `port`
Port serwera SMTP.

3.9.1 Opis szczegółowy

Klasa `WriteMail` odpowiedzialna za probe wyslania wiadomosci za pomoca protokolu SMTP

3.9.2 Dokumentacja konstruktora i destruktora

3.9.2.1 WriteMail()

```
WriteMail::WriteMail (
    string smtpserverAddress,
    string userName,
    string password,
    int port = 25 )
```

Konstruktor `WriteMail` przechowujące dane do logowania

3.9.3 Dokumentacja funkcji składowych

3.9.3.1 writeMail()

```
string WriteMail::writeMail (
    string toAdressEmailSend,
    string titleToSend,
    string messageToSend )
```

Metoda `writeMail()` wysyła wiadomość email

Parametry

<i>toAdressEmailSend</i>	- adres email do kogo ma zostać wysłany email
<i>titleToSend</i>	- tytuł wiadomości email
<i>messageToSend</i>	- kontent wiadomości email (wiadomość)

Zwraca

- zwraca `ok` jeśli wiadomość została wysłana poprawnie, jeśli nie zwraca błąd (dlaczego wiadomość nie została wysłana)

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Write↔ Mail.h
- C:/Users/ArAs/Desktop/Studia/Sem.03/(PK) Programowanie komputerów/Projekt/Projekt/Projekt/Projekt/Write↔ Mail.cpp

Indeks

- ~Login
 - Projekt::Login, [7](#)
- ~POP3
 - Projekt::POP3, [9](#)
- ~SMTP
 - Projekt::SMTP, [14](#)
- convertBtokB
 - Projekt::POP3, [9](#)
- downloadLastMails
 - ReadMails, [12](#)
- Files, [5](#)
 - Files, [5](#)
 - ifFileExists, [6](#)
 - reWriteContentFile, [6](#)
- getMailInfo
 - ReadMails, [13](#)
- ifFileExists
 - Files, [6](#)
- Login
 - Projekt::Login, [7](#)
- loginPOP3
 - POP3Login, [11](#)
- loginSMTP
 - SMTPLogin, [15](#)
- POP3
 - Projekt::POP3, [9](#)
- POP3Login, [10](#)
 - loginPOP3, [11](#)
 - POP3Login, [11](#)
- Projekt::Login, [6](#)
 - ~Login, [7](#)
 - Login, [7](#)
- Projekt::POP3, [8](#)
 - ~POP3, [9](#)
 - convertBtokB, [9](#)
 - POP3, [9](#)
 - ReplaceAll, [10](#)
 - toPolishWords, [10](#)
- Projekt::SMTP, [13](#)
 - ~SMTP, [14](#)
 - SMTP, [14](#)
- ReadMails, [12](#)
 - downloadLastMails, [12](#)
- getMailInfo, [13](#)
- ReadMails, [12](#)
- ReadMails::Mail, [7](#)
- ReplaceAll
 - Projekt::POP3, [10](#)
- reWriteContentFile
 - Files, [6](#)
- SMTP
 - Projekt::SMTP, [14](#)
- SMTPLogin, [15](#)
 - loginSMTP, [15](#)
 - SMTPLogin, [15](#)
- toPolishWords
 - Projekt::POP3, [10](#)
- WriteMail, [16](#)
 - WriteMail, [16](#)
 - writeMail, [17](#)
- writeMail
 - WriteMail, [17](#)