

Bank

Wygenerowano przez Doxygen 1.9.6

Rozdział 1

Indeks klas

1.1 Lista klas

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

konto	??
s_data	??
s_trans	??
s_zakres	??

Rozdział 2

Indeks plików

2.1 Lista plików

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

PPK pBANK/Bank/Bank/ funkcje.cpp	??
PPK pBANK/Bank/Bank/ Header.h	??
PPK pBANK/Bank/Bank/ main.cpp	??

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury konta

```
#include <Header.h>
```

Atrybuty publiczne

- int `id`
- std::string `imie`
- std::string `drugie_imie`
- std::string `nazwisko`
- std::string `adres`
- std::string `miasto`
- std::string `kod_pocztowy`
- std::string `nr_konta`
- std::string `pesel`
- std::string `nr_telefonu`
- double `saldo`
- double `debet`

3.1.1 Opis szczegółowy

konta - struktura która przechowuje dane z pliku uzytkownicy.txt

3.1.2 Dokumentacja atrybutów składowych

3.1.2.1 adres

```
konta::adres
```

zmienna przechowuje adres klienta

3.1.2.2 debet

`konta::debet`

zmienna przechowuje maksymalny debet na koncie klienta

3.1.2.3 drugie_imie

`konta::drugie_imie`

zmienna przechowuje drugie imie klienta

3.1.2.4 id

`konta::id`

zmienna przechowuje identyfikator konta

3.1.2.5 imie

`konta::imie`

zmienna przechowuje imie klienta

3.1.2.6 kod_pocztowy

`konta::kod_pocztowy`

zmienna przechowuje kod pocztowy klienta

3.1.2.7 miasto

`konta::miasto`

zmienna przechowuje miasto zamieszkania klienta

3.1.2.8 nazwisko

`konta::nazwisko`

zmienna przechowuje nazwisko klienta

3.1.2.9 nr_konta

`konta::nr_konta`

zmienna przechowuje numer konta klienta

3.1.2.10 nr_telefonu

```
konta::nr_telefonu
```

zmienna przechowuje numer telefonu klienta

3.1.2.11 pesel

```
konta::pesel
```

zmienna przechowuje pesel klienta

3.1.2.12 saldo

```
konta::saldo
```

zmienna przechowuje saldo konta klienta

Dokumentacja dla tej struktury została wygenerowana z pliku:

- PPK pBANK/Bank/Bank/[Header.h](#)

3.2 Dokumentacja struktury s_data

```
#include <Header.h>
```

Atrybuty publiczne

- int [dzien](#)
- int [miesiac](#)
- int [rok](#)

3.2.1 Opis szczegółowy

[s_data](#) - struktura która przechowuje rozłożoną na czynniki datę transakcji

3.2.2 Dokumentacja atrybutów składowych

3.2.2.1 dzien

```
s_data::dzien
```

zmienna przechowuje dzień dla konkretnej daty

3.2.2.2 miesiac

```
s_data::miesiac
```

zmienna przechowuje miesiąc dla konkretnej daty

3.2.2.3 rok

```
s_data::rok
```

zmienna przechowuje rok dla konkretnej daty

Dokumentacja dla tej struktury została wygenerowana z pliku:

- PPK pBANK/Bank/Bank/[Header.h](#)

3.3 Dokumentacja struktury s_trans

```
#include <Header.h>
```

Atrybuty publiczne

- int [id](#)
- int [numer](#)
- std::string [rodzaj](#)
- std::string [konto](#)
- std::string [data](#)
- double [kwota](#)

3.3.1 Opis szczegółowy

[s_trans](#) - struktura która przechowuje dane z pliku transakcje.txt

3.3.2 Dokumentacja atrybutów składowych

3.3.2.1 data

```
s_trans::data
```

zmienna przechowuje datę wykonania transakcji

3.3.2.2 id

```
s_trans::id
```

zmienna przechowuje identyfikator transakcji

3.3.2.3 konto

```
s_trans::konto
```

zmienna przechowuje numer konta klienta

3.3.2.4 kwota

```
s_trans::kwota
```

zmienna przechowuje kwotę wykonanej transakcji

3.3.2.5 numer

```
s_trans::numer
```

zmienna przechowuje numer transakcji w banku

3.3.2.6 rodzaj

```
s_trans::rodzaj
```

zmienna przechowuje typ wykonanej transakcji (wpłata = "+"/wyplata = "-")

Dokumentacja dla tej struktury została wygenerowana z pliku:

- PPK pBANK/Bank/Bank/[Header.h](#)

3.4 Dokumentacja struktury s_zakres

```
#include <Header.h>
```

Atrybuty publiczne

- int [dzien1](#)
- int [miesiac1](#)
- int [rok1](#)
- int [dzien2](#)
- int [miesiac2](#)
- int [rok2](#)
- double [p_finan](#)
- double [k_finan](#)

3.4.1 Opis szczegółowy

`s_zakres` - struktura która przechowuje zdefiniowany przez użytkownika zakres finansowy i datowy do generowania raportów

3.4.2 Dokumentacja atrybutów składowych

3.4.2.1 dzien1

```
s_zakres::dzien1
```

zmienna przechowuje dzień dla daty zdefiniowanej jako dolny próg

3.4.2.2 dzien2

```
s_zakres::dzien2
```

zmienna przechowuje dzień dla daty zdefiniowanej jako górny próg

3.4.2.3 k_finan

```
s_zakres::k_finan
```

zmienna przechowuje górny próg finansowy

3.4.2.4 miesiac1

```
s_zakres::miesiac1
```

zmienna przechowuje miesiąc dla daty zdefiniowanej jako dolny próg

3.4.2.5 miesiac2

```
s_zakres::miesiac2
```

zmienna przechowuje miesiąc dla daty zdefiniowanej jako górny próg

3.4.2.6 p_finan

```
s_zakres::p_finan
```

zmienna przechowuje dolny próg finansowy

3.4.2.7 rok1

```
s_zakres::rok1
```

zmienna przechowuje rok dla daty zdefiniowanej jako dolny próg

3.4.2.8 rok2

```
s_zakres::rok2
```

zmienna przechowuje rok dla daty zdefiniowanej jako górny próg

Dokumentacja dla tej struktury została wygenerowana z pliku:

- PPK pBANK/Bank/Bank/[Header.h](#)

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku PPK pBANK/Bank/Bank/funkcje.cpp

```
#include "Header.h"
```

Funkcje

- `std::vector< std::vector< std::string > > wczytaj_baze` (const std::string &baza_danych)
- `std::vector< std::vector< std::string > > wydobywanie_danych` (const std::string &baza_danych)
- `std::vector< std::vector< std::string > > wczytaj_transakcje` (const std::string &baza_transakcji)
- `std::vector< std::vector< std::string > > wydobywanie_transakcji` (const std::string &baza_transakcji)
- void `pokaz_baze` (std::vector< `konta` > &uzytkownicy)
- void `pokaz_transakcje` (std::vector< `s_trans` > &transakcje)
- int `licznik_linii` (std::vector< std::vector< std::string > > &dane)
- int `licznik_linii2` (std::vector< std::vector< std::string > > &dane2)
- `std::vector< konta > tworzenie_struktur` (std::vector< std::vector< std::string > > &dane, int l_linii)
- `std::vector< s_trans > tworzenie_struktur2` (std::vector< std::vector< std::string > > &dane2, int l_linii2)
- void `sprawdz_saldo` (std::vector< `konta` > &uzytkownicy, int l_linii)
- void `wplata` (std::vector< `konta` > &uzytkownicy, std::vector< `s_trans` > &transakcje, `s_data` data)
- void `wypлата` (std::vector< `konta` > &uzytkownicy, std::vector< `s_trans` > &transakcje, `s_data` data)
- void `przelew` (std::vector< `konta` > &uzytkownicy, std::vector< `s_trans` > &transakcje, `s_data` data)
- void `zwrot` (std::vector< `konta` > &uzytkownicy, std::vector< `s_trans` > &transakcje, char &last)
- `s_zakres f_zakres` ()
- bool `zakres_dat` (std::vector< `s_trans` > &transakcje, `s_zakres` zakres, int &licznik)
- void `genRaport` (std::vector< `konta` > uzytkownicy, std::vector< `s_trans` > transakcje)
- void `zapis_struktury` (std::vector< `konta` > &uzytkownicy, std::vector< `s_trans` > &transakcje)
- bool `wyjście` (bool &re_menu)

Zmienne

- `s_data f_data` (struct `s_data`)

4.1.1 Opis szczegółowy

Plik zawierający definicje funkcji(ciała) projektu Bank

4.1.2 Dokumentacja funkcji

4.1.2.1 f_zakres()

```
s_zakres f_zakres ( )
```

Funkcja pobiera od użytkownika i zapisuje zakres finansowy i datowy do generowania raportów.

Parametry

funkcja	nie przyjmuje żadnych parametrów
---------	----------------------------------

Zwraca

funkcja zwraca zakres - element struktury `s_zakres` przechowujący bieżący zakres dla aktualnie generowanego raportu

4.1.2.2 gen_raport()

```
void gen_raport (
    std::vector< konta > uzytkownicy,
    std::vector< s_trans > transakcje )
```

Funkcjonalność nr8 - umożliwia generowanie posortowanych, wedle preferencji raportów

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.3 licznik_linii()

```
int licznik_linii (
    std::vector< std::vector< std::string > > & dane )
```

Funkcja zlicza linie z pliku `uzytkownicy.txt`

Parametry

<i>dane</i>	- wektor wektorów dla użytkowników
-------------	------------------------------------

Zwraca

funkcja zwraca licznik_linii - liczbe linni w pliku uzytkownicy.txt

4.1.2.4 licznik_linii2()

```
int licznik_linii2 (
    std::vector< std::vector< std::string > > & dane2 )
```

Funkcja zlicza linie z pliku transakcje.txt

Parametry

<i>dane2</i>	- wektor wektorów dla transakcji
--------------	----------------------------------

Zwraca

funkcja zwraca licznik_linii2 - liczbe linni w pliku transakcje.txt

4.1.2.5 pokaz_baze()

```
void pokaz_baze (
    std::vector< konta > & uzytkownicy )
```

Funkcjonalność nr1 - pokazuje baze użytkowników banku;

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
--------------------	---

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.6 pokaz_transakcje()

```
void pokaz_transakcje (
    std::vector< s_trans > & transakcje )
```

Funkcjonalność nr1 - pokazuje baze transakcji w banku;

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
-------------------	---

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.7 przelew()

```
void przelew (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    s_data data )
```

Funkcjonalność nr6 - przelew, wykonanie przelewu z jednego konta na drugie.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury <i>s_data</i> , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.8 sprawdz_saldo()

```
void sprawdz_saldo (
    std::vector< konta > & uzytkownicy,
    int l_linii )
```

Funkcjonalność nr3 - sprawdź saldo. Sprawdza saldo o podanym numerze konta

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>l_linii</i>	- liczbe linni w pliku uzytkownicy.txt

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.9 tworzenie_struktur()

```
std::vector< konta > tworzenie_struktur (
    std::vector< std::vector< std::string > > & dane,
    int l_linii )
```

Funkcja uzupełnia stuktury danymi z wektora, wektorów dla struktury użytkowników.

Parametry

<i>dane</i>	- wektor wektorów dla użytkowników
<i>l_linii</i>	- liczba linii w pliku uzytkownicy.txt

Zwraca

strukture konta, która przechowuje dane z pliku uzytkownicy.txt

4.1.2.10 tworzenie_struktur2()

```
std::vector< s_trans > tworzenie_struktur2 (
    std::vector< std::vector< std::string > > & dane2,
    int l_linii2 )
```

Funkcja uzupełnia stuktury danymi z wektora, wektorów dla struktury transakcji.

Parametry

<i>dane2</i>	- wektor wektorów dla transakcji
<i>l_linii2</i>	- liczba linii w pliku transakcje.txt

Zwraca

strukture [s_trans](#), która przechowuje dane z pliku transakcje.txt

4.1.2.11 wczytaj_baze()

```
std::vector< std::vector< std::string > > wczytaj_baze (
    const std::string & baza_danych )
```

Funkcja wczytuje plik uzytkownicy.txt zawierający bazę danych użytkowników banku

Parametry

<i>baza_danych</i>	- plik uzytkownicy.txt
--------------------	------------------------

Zwraca

baza - wektor zawierający wektory

4.1.2.12 wczytaj_transakcje()

```
std::vector< std::vector< std::string > > wczytaj_transakcje (
    const std::string & baza_transakcji )
```

Funkcja wczytuje plik transakcje.txt zawierający bazę transakcji wykonanych w banku

Parametry

<i>baza_transakcji</i>	- plik transakcje.txt
------------------------	-----------------------

Zwraca

trans - wektor zawierający wektory

4.1.2.13 wpłata()

```
void wpłata (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    s_data data )
```

Funkcjonalność nr4 - wpłata pieniędzy. Wpłaca podana kwotę na wybrane konto.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury <i>s_data</i> , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.14 wydobywanie_danych()

```
std::vector< std::vector< std::string > > wydobywanie_danych (
    const std::string & baza_danych )
```

Funkcja wydobywa z vectora vectorów zawierającego stringi, pojedyncze stringi dla bazy użytkowników[kolumna][wiersz]

Parametry

<i>baza_danych</i>	- plik uzytkownicy.txt
--------------------	------------------------

Zwraca

dane - wektor wektorów dla użytkowników

4.1.2.15 wydobycie_transakcji()

```
std::vector< std::vector< std::string > > wydobycie_transakcji (
    const std::string & baza_transakcji )
```

Funkcja wydobywa z vectora vectorów zawierającego stringi, pojedyncze stringi dla bazy transakcji[kolumna][wiersz]

Parametry

<i>baza_transakcji</i>	- plik transakcje.txt
------------------------	-----------------------

Zwraca

dane2 - wektor wektorów dla transakcji

4.1.2.16 wyjscie()

```
bool wyjscie (
    bool & re_menu )
```

Funkcjonalność nr 9 - wyjście z aplikacji i załadowanie plików.

Parametry

<i>re_menu</i>	- kiedy re_menu=true powtórz menu, kiedy re_menu=false zakończ program
----------------	--

Zwraca

funkcja zwraca re_menu - true albo false

4.1.2.17 wypłata()

```
void wypłata (
    std::vector< konta > & uzytkownicy,
```

```
std::vector< s_trans > & transakcje,  
s_data data )
```

Funkcjonalność nr5 - wypłata pieniędzy. Wypłaca podaną kwotę z wybranego konta.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury <i>s_data</i> , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.18 zakres_dat()

```
bool zakres_dat (  
    std::vector< s_trans > & transakcje,  
    s_zakres zakres,  
    int & licznik )
```

Funkcja sprawdza zadeklarowany zakres dat i weryfikuje czy dana data ma się wliczać do raportu, czy nie.

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>zakres</i>	- element struktury <i>s_zakres</i> przechowujący bieżący zakres dla aktualnie generowanego raportu
<i>licznik</i>	- zmienna służąca do przekazania funkcji, która transakcja jest sprawdzana z funkcji generującej raporty

Zwraca

funkcja zwraca true jeśli ma się liczyć do raportu lub false jeśli nie ma się liczyć

4.1.2.19 zapis_struktury()

```
void zapis_struktury (  
    std::vector< konta > & uzytkownicy,  
    std::vector< s_trans > & transakcje )
```

Funkcja zapisuje zawartości wektorów do plików na zakończenie programu

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników

Zwraca

funkcja nie zwraca żadnej wartości

4.1.2.20 zwrot()

```
void zwrot (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    char & last )
```

Funkcjonalność nr7 - umożliwia cofnięcie ostatniej transakcji

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>last</i>	- zmienna znakowa, zapisująca informację o tym jaką transakcja była ostatnio wykonana - (wpłata/wypłata/przelew)

Zwraca

funkcja nie zwraca żadnej wartości

4.1.3 Dokumentacja zmiennych**4.1.3.1 f_data**

```
s_data f_data(struct s_data) (
    struct s_data )
```

Funkcja generuje bieżącą datę i rozpisuje ją na 3 zmienne (dzień, miesiąc, rok)

Parametry

<i>s_data</i>	- struktura która przechowuje rozłożoną na czynniki datę transakcji
---------------	---

Zwraca

funkcja zwraca element data, struktury [s_data](#)

4.2 Dokumentacja pliku PPK pBANK/Bank/Bank/Header.h

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <ctime>
#include <chrono>
#include <cstdio>
#include <algorithm>
```

Komponenty

- struct [konto](#)
- struct [s_trans](#)
- struct [s_data](#)
- struct [s_zakres](#)

Funkcje

- std::vector< std::vector< std::string > > [wczytaj_baze](#) (const std::string &baza_danych)
- std::vector< std::vector< std::string > > [wydobycie_danych](#) (const std::string &baza_danych)
- std::vector< std::vector< std::string > > [wczytaj_transakcje](#) (const std::string &baza_transakcji)
- std::vector< std::vector< std::string > > [wydobycie_transakcji](#) (const std::string &baza_transakcji)
- void [pokaz_baze](#) (std::vector< [konto](#) > &uzytkownicy)
- void [pokaz_transakcje](#) (std::vector< [s_trans](#) > &transakcje)
- std::vector< [konto](#) > [tworzenie_struktur](#) (std::vector< std::vector< std::string > > &dane, int l_linii)
- std::vector< [s_trans](#) > [tworzenie_struktur2](#) (std::vector< std::vector< std::string > > &dane2, int l_linii2)
- int [licznik_linii](#) (std::vector< std::vector< std::string > > &dane)
- int [licznik_linii2](#) (std::vector< std::vector< std::string > > &dane2)
- [s_data f_data](#) (struct [s_data](#))
- void [sprawdz_saldo](#) (std::vector< [konto](#) > &uzytkownicy, int l_linii)
- void [wpłata](#) (std::vector< [konto](#) > &uzytkownicy, std::vector< [s_trans](#) > &transakcje, [s_data](#) data)
- void [wypłata](#) (std::vector< [konto](#) > &uzytkownicy, std::vector< [s_trans](#) > &transakcje, [s_data](#) data)
- void [przelew](#) (std::vector< [konto](#) > &uzytkownicy, std::vector< [s_trans](#) > &transakcje, [s_data](#) data)
- void [zwrot](#) (std::vector< [konto](#) > &uzytkownicy, std::vector< [s_trans](#) > &transakcje, char &last)
- [s_zakres f_zakres](#) ()
- void [genRaport](#) (std::vector< [konto](#) > uzytkownicy, std::vector< [s_trans](#) > transakcje)
- bool [zakres_dat](#) (std::vector< [s_trans](#) > &transakcje, [s_zakres](#) zakres, int &licznik)
- void [zapis_struktury](#) (std::vector< [konto](#) > &uzytkownicy, std::vector< [s_trans](#) > &transakcje)
- bool [wyjście](#) (bool &re_menu)

4.2.1 Opis szczegółowy

Plik zawierający deklaracje funkcji(nagłówki) projektu Bank

4.2.2 Dokumentacja funkcji

4.2.2.1 f_data()

```
s_data f_data (
    struct s_data )
```

Funkcja generuje bieżącą datę i rozpisuje ją na 3 zmienne (dzień, miesiąc, rok)

Parametry

<code>s_data</code>	- struktura która przechowuje rozłożoną na czynniki datę transakcji
---------------------	---

Zwraca

funkcja zwraca element data, struktury `s_data`

4.2.2.2 f_zakres()

```
s_zakres f_zakres ( )
```

Funkcja pobiera od użytkownika i zapisuje zakres finansowy i datowy do generowania raportów.

Parametry

<i>funkcja</i>	nie przyjmuje żadnych parametrów
----------------	----------------------------------

Zwraca

funkcja zwraca zakres - element struktury `s_zakres` przechowujący bieżący zakres dla aktualnie generowanego raportu

4.2.2.3 gen_raport()

```
void gen_raport (
    std::vector< konta > uzytkownicy,
    std::vector< s_trans > transakcje )
```

Funkcjonalność nr8 - umożliwia generowanie posortowanych, wedle preferencji raportów

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.4 licznik_linii()

```
int licznik_linii (
    std::vector< std::vector< std::string > > & dane )
```

Funkcja zlicza linie z pliku uzytkownicy.txt

Parametry

<i>dane</i>	- wektor wektorów dla użytkowników
-------------	------------------------------------

Zwraca

funkcja zwraca licznik_linii - liczbe linni w pliku uzytkownicy.txt

4.2.2.5 licznik_linii2()

```
int licznik_linii2 (
    std::vector< std::vector< std::string > > & dane2 )
```

Funkcja zlicza linie z pliku transakcje.txt

Parametry

<i>dane2</i>	- wektor wektorów dla transakcji
--------------	----------------------------------

Zwraca

funkcja zwraca licznik_linii2 - liczbe linni w pliku transakcje.txt

4.2.2.6 pokaz_baze()

```
void pokaz_baze (
    std::vector< konta > & uzytkownicy )
```

Funkcjonalność nr1 - pokazuje baze użytkowników banku;

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
--------------------	---

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.7 pokaz_transakcje()

```
void pokaz_transakcje (
    std::vector< s\_trans > & transakcje )
```

Funkcjonalność nr1 - pokazuje baze transakcji w banku;

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
-------------------	---

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.8 przelew()

```
void przelew (
    std::vector< konta > & uzytkownicy,
    std::vector< s\_trans > & transakcje,
    s\_data data )
```

Funkcjonalność nr6 - przelew, wykonanie przelewu z jednego konta na drugie.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury s_data , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.9 sprawdz_saldo()

```
void sprawdz_saldo (
    std::vector< konta > & uzytkownicy,
    int l_linii )
```

Funkcjonalność nr3 - sprawdź saldo. Sprawdza saldo o podanym numerze konta

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>l_linii</i>	- liczbe linni w pliku uzytkownicy.txt

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.10 tworzenie_struktur()

```
std::vector< konta > tworzenie_struktur (
    std::vector< std::vector< std::string > > & dane,
    int l_linii )
```

Funkcja uzupełnia stuktury danymi z wektora, wektorów dla struktury użytkowników.

Parametry

<i>dane</i>	- wektor wektorów dla użytkowników
<i>l_linii</i>	- liczba linii w pliku uzytkownicy.txt

Zwraca

strukture konta, która przechowuje dane z pliku uzytkownicy.txt

4.2.2.11 tworzenie_struktur2()

```
std::vector< s_trans > tworzenie_struktur2 (
    std::vector< std::vector< std::string > > & dane2,
    int l_linii2 )
```

Funkcja uzupełnia stuktury danymi z wektora, wektorów dla struktury transakcji.

Parametry

<i>dane2</i>	- wektor wektorów dla transakcji
<i>l_linii2</i>	- liczba linii w pliku transakcje.txt

Zwraca

strukture [s_trans](#), która przechowuje dane z pliku transakcje.txt

4.2.2.12 wczytaj_baze()

```
std::vector< std::vector< std::string > > wczytaj_baze (
    const std::string & baza_danych )
```

Funkcja wczytuje plik uzytkownicy.txt zawierający bazę danych użytkowników banku

Parametry

<i>baza_danych</i>	- plik uzytkownicy.txt
--------------------	------------------------

Zwraca

baza - wektor zawierający wektory

4.2.2.13 wczytaj_transakcje()

```
std::vector< std::vector< std::string > > wczytaj_transakcje (
    const std::string & baza_transakcji )
```

Funkcja wczytuje plik transakcje.txt zawierający bazę transakcji wykonanych w banku

Parametry

<i>baza_transakcji</i>	- plik transakcje.txt
------------------------	-----------------------

Zwraca

trans - wektor zawierający wektory

4.2.2.14 wpłata()

```
void wpłata (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    s_data data )
```

Funkcjonalność nr4 - wpłata pieniędzy. Wpłaca podana kwotę na wybrane konto.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury <i>s_data</i> , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.15 wydobywanie_danych()

```
std::vector< std::vector< std::string > > wydobywanie_danych (
    const std::string & baza_danych )
```

Funkcja wydobywa z wektora wektorów zawierającego stringi, pojedyncze stringi dla bazy użytkowników[kolumna][wiersz]

Parametry

<i>baza_danych</i>	- plik uzytkownicy.txt
--------------------	------------------------

Zwraca

dane - wektor wektorów dla użytkowników

4.2.2.16 wydobywanie_transakcji()

```
std::vector< std::vector< std::string > > wydobywanie_transakcji (
    const std::string & baza_transakcji )
```

Funkcja wydobywa z wektora wektorów zawierającego stringi, pojedyncze stringi dla bazy transakcji[kolumna][wiersz]

Parametry

<i>baza_transakcji</i>	- plik transakcje.txt
------------------------	-----------------------

Zwraca

dane2 - wektor wektorów dla transakcji

4.2.2.17 wyjście()

```
bool wyjście (
    bool & re_menu )
```

Funkcjonalność nr 9 - wyjście z aplikacji i załadowanie plików.

Parametry

<i>re_menu</i>	- kiedy re_menu=true powtórz menu, kiedy re_menu=false zakończ program
----------------	--

Zwraca

funkcja zwraca re_menu - true albo false

4.2.2.18 wypłata()

```
void wypłata (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    s_data data )
```

Funkcjonalność nr5 - wypłata pieniędzy. Wypłaca podaną kwotę z wybranego konta.

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>data</i>	- element struktury <i>s_data</i> , zawierająca strukturę z bieżącą datą

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.19 zakres_dat()

```
bool zakres_dat (
    std::vector< s_trans > & transakcje,
```



```
s_zakres zakres,
int & licznik )
```

Funkcja sprawdza zadeklarowany zakres dat i weryfikuje czy dana data ma się wliczać do raportu, czy nie.

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>zakres</i>	- element struktury <code>s_zakres</code> przechowujący bieżący zakres dla aktualnie generowanego raportu
<i>licznik</i>	- zmienna służąca do przekazania funkcji, która transakcja jest sprawdzana z funkcji generującej raporty

Zwraca

funkcja zwraca true jeśli ma się liczyć do raportu lub false jeśli nie ma się liczyć

4.2.2.20 zapis_struktury()

```
void zapis_struktury (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje )
```

Funkcja zapisuje zawartości wektorów do plików na zakończenie programu

Parametry

<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników

Zwraca

funkcja nie zwraca żadnej wartości

4.2.2.21 zwrot()

```
void zwrot (
    std::vector< konta > & uzytkownicy,
    std::vector< s_trans > & transakcje,
    char & last )
```

Funkcjonalność nr7 - umożliwia cofnięcie ostatniej transakcji

Parametry

<i>uzytkownicy</i>	- wektor zawierający struktury użytkowników
<i>transakcje</i>	- wektor zawierający struktury transakcji
<i>last</i>	- zmienna znakowa, zapisująca informację o tym jaką transakcja była ostatnio wykonana - (wpłata/wypłata/przelew)

Zwraca

funkcja nie zwraca żadnej wartości

4.3 Header.h

[Idź do dokumentacji tego pliku.](#)

```
00001 // #pragma once
00007 #ifndef Header_h
00008 #define Header_h
00009 #include <iostream>
00010 #include <vector>
00011 #include <string>
00012 #include <fstream>
00013 #include <sstream> // biblioteka dla stringstream
00014 #include <iomanip>
00015 #include <ctime>
00016 #include <chrono>
00017 #include <cstdio> // biblioteka zawiera funkcje sscanf - umożliwia rozłożenie daty w stringu na
    poszczególne składowe w intach. - raporty
00018 #include <algorithm> // biblioteka dla sortowania danych
00019
00047 struct konta {
00048
00049     int id;
00050     std::string imie, drugie_imie, nazwisko, adres, miasto, kod_pocztowy;
00051     std::string nr_konta, pesel, nr_telefonu;
00052     double saldo, debet;
00053 };
00054
00070 struct s_trans {
00071     int id, numer;
00072     std::string rodzaj, konto, data;
00073     double kwota;
00074 };
00075
00085 struct s_data { //struktura zawierająca date
00086     int dzien;
00087     int miesiac;
00088     int rok;
00089 };
00090
00110 struct s_zakres { // struktura zdefiniowanego zakresu do raportów
00111     int dzien1, miesiac1, rok1, dzien2, miesiac2, rok2;
00112     double p_finan, k_finan;
00113 };
00114
00120 std::vector<std::vector<std::string>> wczytaj_baze(const std::string& baza_danych);
00121
00127 std::vector<std::vector<std::string>> wydobywanie_danych(const std::string& baza_danych);
00128
00134 std::vector<std::vector<std::string>> wczytaj_transakcje(const std::string& baza_transakcji);
00135
00141 std::vector<std::vector<std::string>> wydobywanie_transakcji(const std::string& baza_transakcji);
00142
00148 void pokaz_baze(std::vector<konta>& uzytkownicy);
00149
00155 void pokaz_transakcje(std::vector<s_trans>& transakcje);
00156
00163 std::vector<konta> tworzenie_struktur(std::vector<std::vector<std::string>>& dane, int l_linii);
00164
00171 std::vector<s_trans> tworzenie_struktur2(std::vector<std::vector<std::string>>& dane2, int l_linii2);
00172
00178 int licznik_linii(std::vector<std::vector<std::string>>& dane);
00179
00185 int licznik_linii2(std::vector<std::vector<std::string>>& dane2);
00186
00192 s_data f_data(struct s_data);
00193
00200 void sprawdz_saldo(std::vector<konta>& uzytkownicy, int l_linii);
00201
00209 void wpłata(std::vector<konta>& uzytkownicy, std::vector<s_trans>& transakcje, s_data data);
00210
00218 void wypłata(std::vector<konta>& uzytkownicy, std::vector<s_trans>& transakcje, s_data data);
00219
00227 void przelew(std::vector<konta>& uzytkownicy, std::vector<s_trans>& transakcje, s_data data);
00228
00236 void zwrot(std::vector<konta>& uzytkownicy, std::vector<s_trans>& transakcje, char& last);
00237
00243 s_zakres f_zakres();
00244
```

```
00251 void gen_raport(std::vector<konta> uzytkownicy, std::vector<s_trans> transakcje);
00252
00260 bool zakres_dat(std::vector<s_trans>& transakcje, s_zakres zakres, int& licznik);
00261
00268 void zapis_struktury(std::vector<konta>& uzytkownicy, std::vector<s_trans>& transakcje);
00269
00275 bool wyjscie(bool& re_menu);
00276 #endif
```

4.4 Dokumentacja pliku PPK pBANK/Bank/Bank/main.cpp

```
#include "Header.h"
```

Funkcje

- int main (int argc, char *argv[])

4.4.1 Opis szczegółowy

Plik główny projektu Bank

