

System bankowy w C++	
Kurs Programowanie Obiektowe	Termin Środa 9:15
Temat System bankowy	Etap Dokumentacja końcowa
Skład grupy Natalia Nadolna, Damian Kociolek, Kamil Pokrywa, Michał Wesołowski	Nr grupy 5
Prowadzący Mgr inż. Konrad Kluwak	data 3 czerwca 2020

Ogólny opis projektu

Aplikacja jest symulacją rzeczywistego systemu bankowego, z którym mamy do czynienia na co dzień. Umożliwia ona przeprowadzenie najważniejszych działań takie jak zalogowanie się na własne konto gdzie możemy wykonywać takie operacje jak sprawdzenie danych swojego konta, dokonanie wpłaty oraz wypłaty pieniędzy czy oczywiście wykonanie przelewu. Aplikacja umożliwia także założenie nowego konta w systemie.

Interface programu

The figure consists of four screenshots of a C++ application window titled 'C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe'. The application is a text-based menu system for a bank simulation.

- Top-left screenshot:** Shows the main menu with options: 'Zaloguj sie do systemu', 'Zaloguj konto', 'Administracja <weryfikacja>', and 'Zamknij program'. The user has selected 'Zaloguj konto'.
- Top-right screenshot:** Shows the account creation process. It prompts for 'Podaj imie:' (Damian), 'Podaj nazwisko:' (Kociolek), 'Podaj pesel:' (99110410000), 'Podaj login:' (Tulipan), 'Podaj haslo:' (Kwiatek12345), and 'Potwierdz haslo:' (Kwiatek12345).
- Bottom-left screenshot:** Shows the login process. It prompts for 'Podaj login:' (Tulipan) and 'Podaj haslo:' (Kwiatek12345). After successful login, it says 'Zalogowano' and 'Press any key to continue . . . '.
- Bottom-right screenshot:** Shows the account verification process. It prompts for 'Podaj tajny login ADMINISTRATORA:' (admin) and 'Podaj tajne haslo ADMINISTRATORA:' (admin). It then offers options: 'Nacisnij 0 aby wylogowac' and 'Nacisnij 1 aby wyswietlic baze Klientow'. The user has selected '1'.

Rysunek 1: Zrzuty ekranu ukazujące przykładowy

Środowisko oraz użyte biblioteki

Głównym narzędziem pracy było środowisko programistyczne Microsoft Visual Studio. Udostępnianie kodu źródłowego było przeprowadzone przy pomocy platformy GitHub oraz prywatnego dysku w chmurze. Aplikacja może zostać uruchomiona pod kontrolą systemu operacyjnego Windows.

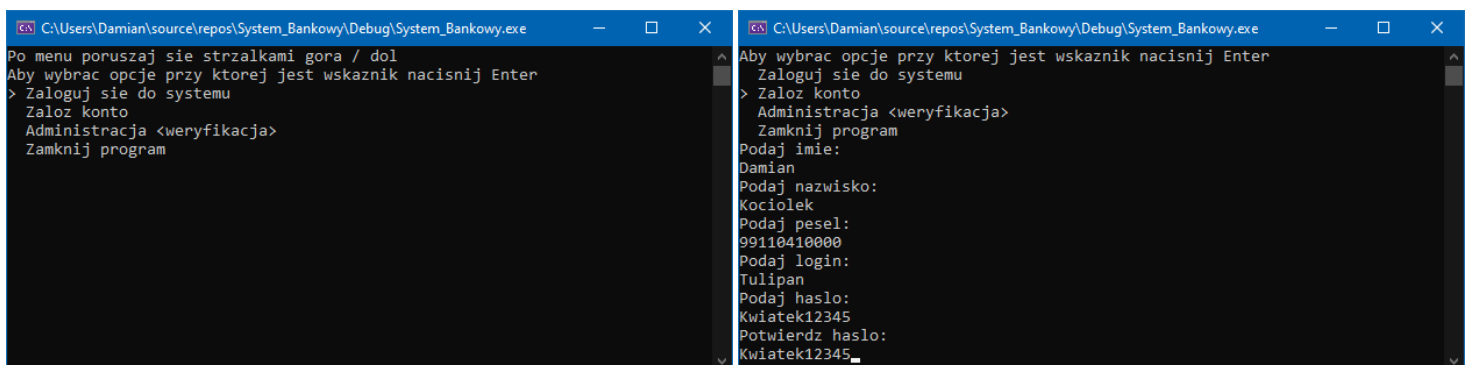
Program wykorzystuje następujące biblioteki:

- `iostream` - do obsługi wejścia/wyjścia
- `tring` - do operacji na łańcuchach znaków
- `vector` - dynamiczne tablice, wektory obiektów klas
- `fstream` - do obsługi plików tekstowych
- `conio.h` - w celu wykorzystania funkcji `getch()`
- `time.h` - obsługa czasu
- `windows.h` - do użycia funkcji `system("cls")` oraz `Sleep()`
- `winbase.h` - aktualny czas pobierany z systemu

Spis funkcjonalności udostępnionych w aplikacji

1 Założenie konta

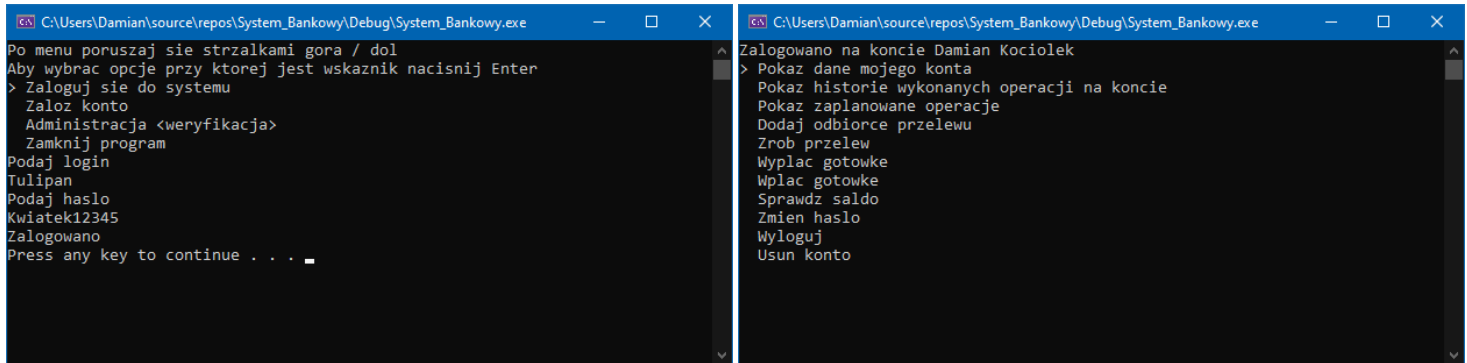
Użytkownik aplikacji ma możliwość założenia nowego konta w systemie. Aby założyć konto użytkownik proszony jest o podanie Imienia, Nazwiska, a następnie numeru PESEL. Aby konto zostało utworzone użytkownik musi podać poprawny numer PESEL oraz musi mieć ukończone 13 lat (jest to sprawdzane przy użyciu aktualnej daty w systemie). Następnie użytkownik proszony jest o podanie loginu, którym będzie logować się do systemu oraz hasła, które następnie musi powtórzyć (aby konto zostało utworzone oba hasła muszą się zgadzać). Indywidualny numer konta nadawany jest automatycznie.



Rysunek 2: Zrzuty ekranu ukazujące przykładowy proces zakładania konta

2 Zalogowanie się do systemu

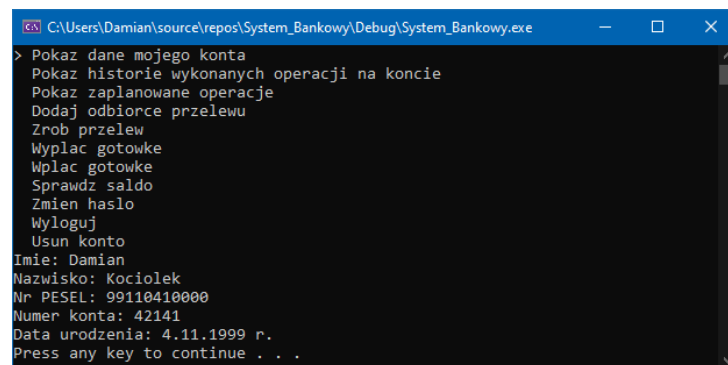
Po założeniu konta w systemie użytkownik może zalogować się na swoje własne konto, w tym celu proszony jest o podanie loginu i hasła które zostały ustawione podczas tworzenia konta.



Rysunek 3: Zrzuty ekranu ukazujące przykładowy proces logowania

2.1 Pokaż dane mojego konta

Użytkownik ma możliwość sprawdzić poprawność wprowadzonych danych podczas tworzenie konta takich jak imię, nazwisko, PESEL, data urodzenia (która została pobrana z numeru PESEL) oraz automatycznie wygenerowany numer konta.



Rysunek 4: Przykładowy podgląd danych użytkownika

2.2 Historia wykonanych operacji

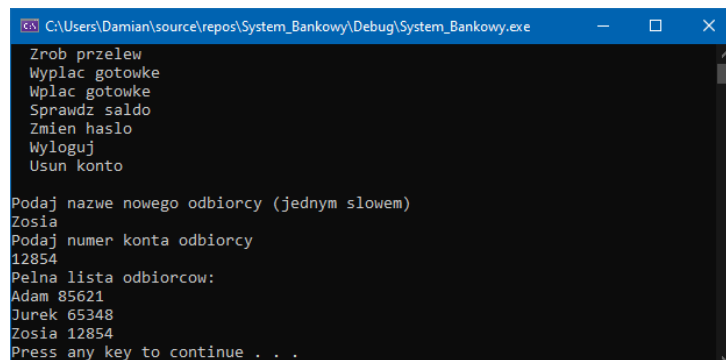
Użytkownik ma możliwość podejrzenia wszystkich operacji na koncie (rodzaj działania oraz data wykonania)

2.3 Zaplanowane operacje

Użytkownik ma możliwość sprawdzenia listy zaplanowanych przelewów.

2.4 Dodanie nowego odbiorcy

Użytkownik może dodać nowego odbiorcę do przelewu, w tym celu proszony jest o podanie nazwy odbiorcy oraz numeru konta.



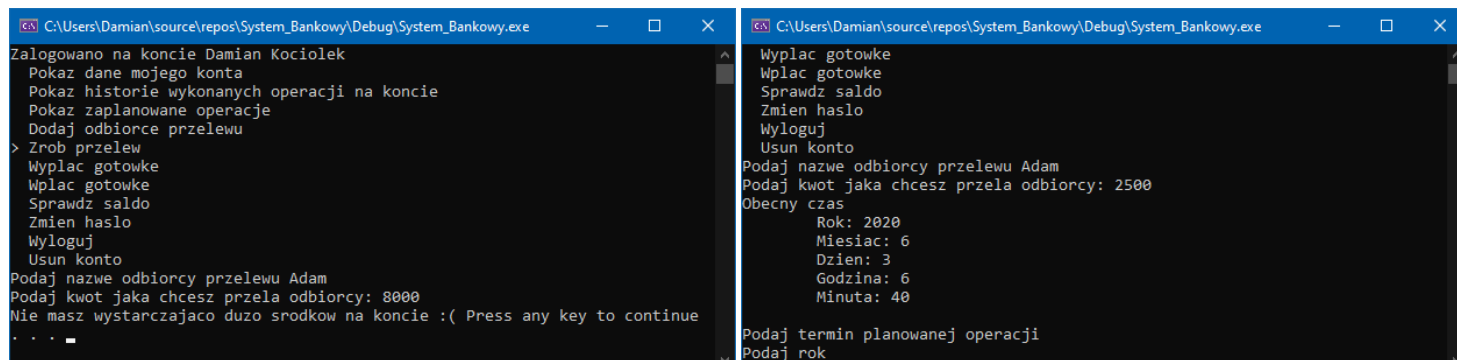
```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Zrob przelew
Wypłac gotówkę
Wpłac gotówkę
Sprawdź saldo
Zmien hasło
Wyloguj
Usun konto

Podaj nazwę nowego odbiorcy (jednym słowem)
Zosia
Podaj numer konta odbiorcy
12854
Pełna lista odbiorców:
Adam 85621
Jurek 65348
Zosia 12854
Press any key to continue . . .
```

Rysunek 5: Przykładowy podgląd listy dodanych odbiorców

2.5 Wykonanie przelewu

Użytkownik ma możliwość wykonania przelewu. Aby wykonać przelew użytkownik proszony jest o podanie nazwy odbiorcy, numeru konta, kwotę którą chce przelać oraz datę wykonania przelewu. Ad jest możliwość wykonania przelewu, do odbiorcy, który nie znajduje się na liście odbiorców oraz na numer konta osoby spoza banku. I że jeszcze podaje się dokładny czas przelewu (datę, godzinę i minutę). I jeszcze tam jakaś praca z plikami była, to o tym też warto wspomnieć



```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Zalogowano na koncie Damian Kociolek
Pokaż dane mojego konta
Pokaż historie wykonanych operacji na koncie
Pokaż zaplanowane operacje
Dodaj odbiorcę przelewu
> Zrob przelew
Wypłac gotówkę
Wpłac gotówkę
Sprawdź saldo
Zmien hasło
Wyloguj
Usun konto
Podaj nazwę odbiorcy przelewu Adam
Podaj kwotę jaką chcesz przelać odbiorcy: 2500
Obecny czas
Rok: 2020
Miesiąc: 6
Dzień: 3
Godzina: 6
Minuta: 40
Podaj termin planowanej operacji
Podaj rok
```

Rysunek 6: Zrzuty ekranu ukazujące przykładowy proces logowania

2.6 Wypłacanie gotówki

Użytkownik ma możliwość wypłacenia gotówki, w tym celu jest proszony o podanie i hasła oraz kwoty jaką chce wypłacić. Aby wypłata została wykonana pomyślnie hasło musi być wpisane poprawnie oraz kwota wypłaty nie może być większa niż aktualny stan konta.

2.7 Wpłata gotówki

Użytkownik ma możliwość wpłacenia gotówki, w tym celu jest proszony o podanie i hasła oraz kwoty jaką chce wpłacić. Aby wypłata została wykonana pomyślnie hasło musi być wpisane poprawnie.

```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Pokaż dane mojego konta
Pokaż historie wykonanych operacji na koncie
Pokaż zaplanowane operacje
Dodaj odbiorcę przelewu
Zrob przelew
Wyplac gotówkę
> Wpłac gotówkę
Sprawdź saldo
Zmien hasło
Wyloguj
Usuń konto
W celu potwierdzenia wpisz hasło: Tulipan12345
Podane hasło jest błędne, spróbuj ponownie: Kwiatek12345
Podaj kwotę jaka chcesz wpłacić
5000
Pomyślnie wpłacono pieniądze.
Press any key to continue . . .
```

```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Pokaż dane mojego konta
Pokaż historie wykonanych operacji na koncie
Pokaż zaplanowane operacje
Dodaj odbiorcę przelewu
Zrob przelew
> Wyplac gotówkę
Wpłac gotówkę
Sprawdź saldo
Zmien hasło
Wyloguj
Usuń konto
W celu potwierdzenia wpisz hasło: Kwiatek12345
Podaj kwotę jaka chcesz wypłacić
2500
Pomyślnie wypłacono pieniądze.
Press any key to continue . . .
```

Rysunek 7: Przykładowy operacja wpłaty i wypłaty gotówki

2.8 Sprawdzenie salda

Użytkownik ma możliwość sprawdzenia aktualnego stanu konta.

```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Zalogowano na koncie Damian Kociolek
Pokaż dane mojego konta
Pokaż historie wykonanych operacji na koncie
Pokaż zaplanowane operacje
Dodaj odbiorcę przelewu
Zrob przelew
Wyplac gotówkę
Wpłac gotówkę
> Sprawdź saldo
Zmien hasło
Wyloguj
Usuń konto
Twoje saldo wynosi: 2500 zł
Press any key to continue . . .
```

Rysunek 8: Przykładowy podgląd listy dodanych odbiorców

2.9 Zmiana hasła

Użytkownik ma możliwość zmienić swoje aktualne hasło na nowe. W celu zmiany hasła na nowe użytkownik jest proszony o podanie starego hasła oraz o dwukrotne powtórzenie nowego hasła.

```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Pokaż dane mojego konta
Pokaż historie wykonanych operacji na koncie
Pokaż zaplanowane operacje
Dodaj odbiorcę przelewu
Zrob przelew
Wyplac gotówkę
Wpłac gotówkę
Sprawdź saldo
> Zmien hasło
Wyloguj
Usuń konto
W celu potwierdzenia wpisz hasło: Kwiatek12345
Podaj obecne hasło: Kwiatek12345
Podaj nowe hasło: Drzewo12345
Podaj hasło ponownie: Drzewo12345
Hasło zostało zmienione.
Press any key to continue . . .
```

Rysunek 9: Przykładowy proces zmiany hasła

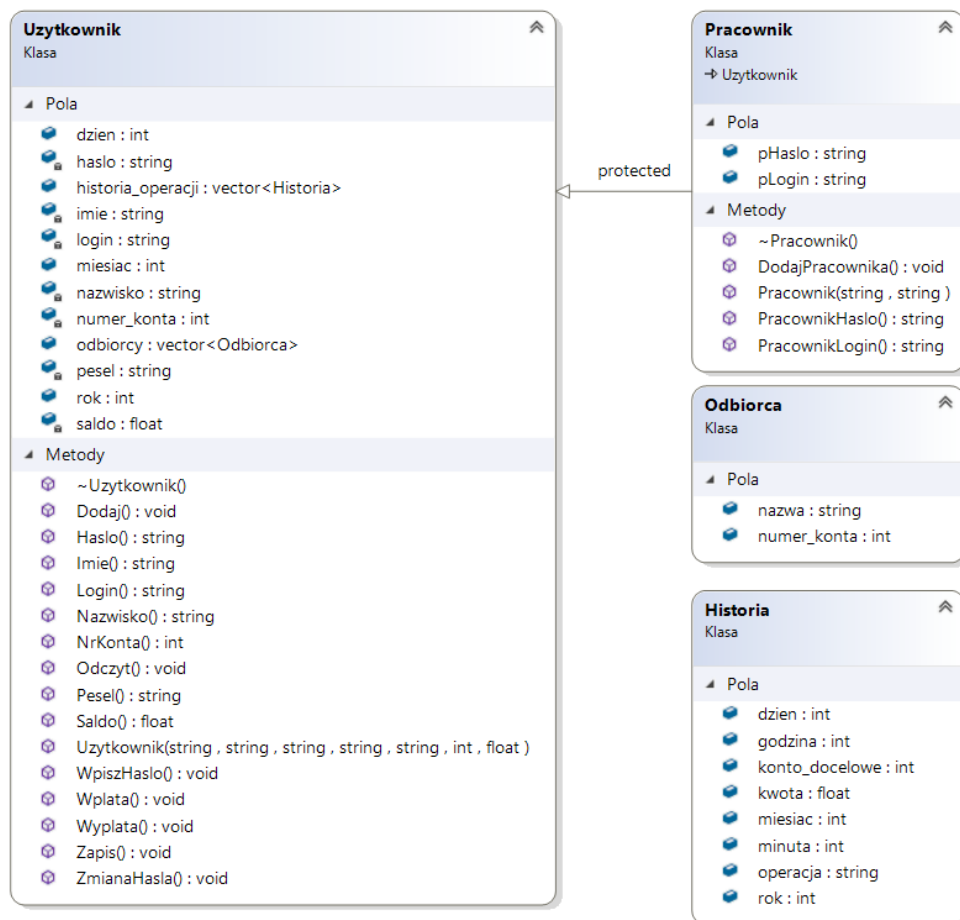
2.10 Wylogowanie

Użytkownik ma możliwość wylogować się z systemu. Wylogowanie nie zamyka aplikacji, umożliwia dalsze działania takie jak ponowne zalogowanie się na konto czy utworzenie innego nowego konta.

```
C:\Users\Damian\source\repos\System_Bankowy\Debug\System_Bankowy.exe
Zalogowano na koncie Damian Kociolek
Pokaz dane mojego konta
Pokaz historie wykonanych operacji na koncie
Pokaz zaplanowane operacje
Dodaj odbiorce przelewu
Zrob przelew
Wypłac gotowke
Wplac gotowke
Sprawdz saldo
Zmien haslo
> Wyloguj
  Usun konto
Trwa Wylogowywanie z konta, Prosimy Czekac...
Wylogowanie Poprawne
Dziekujemy i do zobaczenia
Press any key to continue . . .
```

Rysunek 10: Przykładowy proces zmiany hasła

Diagram klas



Rysunek 11: Diagram klas użytych w programie

Opis klas

Użytkownik

Klasa ta jest klasą główną obsługującą większość operacji związanych z obsługą konta. Jej atrybuty są wykorzystywane do logowania, zakładania konta i wszystkich innych operacji po zalogowaniu się.

Pola klasy:

- imię `string`
- nazwisko `string`
- PESEL `int`
- data urodzenia `struct (int, int, int)`
- login `string`
- hasło `string`
- `int rok;`
- `int miesiac;`
- `int dzien;`

Metody klasy:

- `vector <Odbiorca> odbiorcy;`
- `vector <Historia> historia operacji;`
- `KONSTRUKTOR Uzytkownik(...);`
- `DEKONSTRUKTOR ~ Uzytkownik();`
- `string Imie();`
- `string Nazwisko();`
- `string Pesel();`
- `string Login();`
- `string Haslo();`
- `string Urodzenie();`
- `void Dodaj();`
- `void ZmianaHasla();`
- `void WpiszHaslo();`
- `int NrKonta();`
- `float Saldo();`
- `void Wplata();`
- `void Wyplata();`
- `void Zapis();`
- `void Odczyt();`

Pracownik

Klasa posiada dostęp do bazy danych, oraz logowanie za pomocą loginu (admin) oraz hasła (admin)

Pola klasy:

- `string pLogin;`
- `string pHaslo;`

Metody klasy:

- `KONSTRUKTOR Pracownik(string = "admin", string = "admin");`
- `DEKONSTRUKTOR ~ Pracownik();`
- `string PracownikHaslo();`
- `string PracownikLogin();`
- `void DodajPracownika();`

Odbiorca

Klasa posiada atrybuty które umożliwiają dodanie nowego odbiorcy przelewu

- `string nazwa;`
- `int numer_konta;`

Historia

Klasa posiada i przechowuje atrybuty do zapisu/odczytu historii operacji na koncie

- `string operacja`
- `int rok, miesiac, dzien, godzina, minuta, konto_docelowe;`
- `float kwota;`