



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Technologie IoT - Internetu Rzeczy

Kacper Drabicki
Nr albumu studenta w68520

Aplikacja emulująca działanie konta bankowego

Prowadzący: mgr inż. Ewa Żesławska

PROJEKT - PROGRAMOWANIE OBIEKTOWE

Rzeszów 2025

Spis treści

Wstęp	4
1 Założenia oraz cele projektu	5
1.1 Cele projektu	5
1.2 Założenia projektu	5
1.3 Wymagania Funkcjonalne i Niefunkcjonalne	6
2 Opis techniczny projektu	8
2.1 Język i narzędzia	8
2.2 Baza danych	9
2.3 Zarządzanie danymi	9
2.4 Minimalne wymagania sprzętowe	9
2.5 Platforma testowa	9
3 Opis struktury projektu	10
3.1 Diagram klas	10
3.2 Opis klas	11
4 Harmonogram realizacji projektu	12
4.1 Diagram Gantta	12
4.2 Problemy i rozwiązania	13
4.3 Repozytorium i system kontroli wersji	13
5 Prezentacja warstwy użytkowej projektu	14
5.1 Opis aplikacji	14
5.2 Menu Główne	14
5.3 Dodawanie konta	15
5.4 Wpłata środków	15
5.5 Wypłata środków	16
5.6 Naliczanie odsetek	17
5.7 Zmiana oprocentowania	18
5.8 Wyświetlanie informacji o koncie	19
5.9 Usuwanie konta	19
6 Podsumowanie	20
Bibliografia	21

Wstęp

Aplikacja konta bankowego jest innowacyjnym narzędziem zaprojektowanym z myślą o ułatwieniu procesu bankowania i zarządzania finansami. Została zaprojektowana jako narzędzie umożliwiające zarządzanie finansami w sposób intuicyjny i elastyczny. Aplikacja oferuje podstawowe operacje na kontach bankowych, takie jak dodawanie kont, wpłaty, wypłaty, naliczanie odsetek, zmianę oprocentowania oraz usuwanie konta. Aplikacja umożliwia różne opcje, takie jak: dodawanie nowego konta, wpłatę pieniędzy, wypłatę pieniędzy, naliczenie odsetek, zmianę oprocentowania konta, usunięcia rachunku czy też wyświetlenia informacji o koncie.

Rozdział 1

Założenia oraz cele projektu

1.1 Cele projektu

Głównym celem tego projektu jest stworzenie aplikacji konsolowej dla systemu Windows, która w realistyczny sposób emuluje operacje związane z obsługą konta bankowego. Aplikacja ta ma za zadanie:

- Zapewnić bankierom narzędzie do zarządzania kontami klientów, umożliwiając wykonywanie podstawowych operacji, takich jak tworzenie, modyfikacja i usuwanie kont.
- Umożliwić szybkie i wygodne wykonywanie operacji finansowych, takich jak wpłaty, wypłaty i naliczanie odsetek.
- Zapewnić narzędzie, które minimalizuje możliwość popełnienia błędów przy zarządzaniu kontami poprzez odpowiednie zabezpieczenia i walidację wprowadzanych danych.
- Dostarczyć proste i intuicyjne środowisko konsolowe do wykonywania operacji bankowych.

1.2 Założenia projektu

Aplikacja została zaprojektowana z myślą o prostocie i funkcjonalności. Kluczowe założenia projektu obejmują:

- **Dodawanie konta:** Aplikacja umożliwia dodanie nowego konta z unikalnym numerem identyfikacyjnym, nazwiskiem właściciela, początkowym saldem, ustalonym debetem i oprocentowaniem.
- **Wpłata środków:** Bankier ma możliwość dokonywania wpłat na wybrane konto.
- **Wypłata środków:** Bankier ma możliwość wypłacenia środków z wybranego konta z uwzględnieniem dostępnego salda i limitu debetu.
- **Naliczanie odsetek:** Aplikacja pozwala na naliczenie odsetek od zgromadzonych środków na koncie według aktualnego oprocentowania.
- **Zmiana oprocentowania:** Bankier ma możliwość zmiany oprocentowania dla wybranego konta.
- **Usuwanie konta:** Aplikacja pozwala na zamknięcie wybranego konta po potwierdzeniu decyzji przez użytkownika.
- **Wyświetlanie informacji o koncie:** Bankier ma możliwość wyświetlenia szczegółowych informacji o koncie, w tym stan, nazwisko właściciela, oprocentowanie i debet.

1.3 Wymagania Funkcjonalne i Niefunkcjonalne

Wymagania Funkcjonalne:

1. Dodawanie konta:

- Aplikacja musi umożliwiać dodanie nowego konta bankowego.
- Nowe konto musi mieć unikalny numer identyfikacyjny.
- Konto musi zawierać dane właściciela (imię i nazwisko).
- Podczas tworzenia konta można podać początkowe saldo, debet oraz oprocentowanie.
- Dane konta muszą być zapisane w bazie danych.

2. Wpłata środków:

- Aplikacja musi umożliwiać wpłacanie środków na konto.
- Należy podać numer konta, na które ma zostać dokonana wpłata.
- Należy podać kwotę wpłaty.
- Aktualne saldo konta musi zostać zaktualizowane.

3. Wypłata środków:

- Aplikacja musi umożliwiać wypłacanie środków z konta.
- Należy podać numer konta, z którego ma zostać dokonana wypłata.
- Należy podać kwotę wypłaty.
- Aplikacja musi sprawdzać, czy na koncie znajdują się wystarczające środki lub dostępny debet.
- Aktualne saldo konta musi zostać zaktualizowane po wypłacie.
- W przypadku braku środków, wyświetlany jest komunikat o braku wystarczających środków.

4. Naliczanie odsetek:

- Aplikacja musi umożliwiać naliczenie odsetek od środków zgromadzonych na koncie.
- Należy podać numer konta, dla którego mają zostać naliczone odsetki.
- Należy wziąć pod uwagę aktualne oprocentowanie konta.
- Aktualne saldo konta musi zostać zaktualizowane o naliczone odsetki.

5. Zmiana oprocentowania:

- Aplikacja musi umożliwiać zmianę oprocentowania konta.
- Należy podać numer konta, dla którego ma zostać zmienione oprocentowanie.
- Należy podać nowe oprocentowanie.
- Nowe oprocentowanie musi być zapisane w bazie danych.

6. Usuwanie konta:

- Aplikacja musi umożliwiać usuwanie kont.
- Należy podać numer konta, które ma zostać usunięte.
- Aplikacja powinna wymagać potwierdzenia decyzji o usunięciu konta.
- Dane konta muszą zostać usunięte z bazy danych.

7. Wyświetlanie informacji o koncie:

- Aplikacja musi umożliwiać wyświetlenie informacji o koncie.
- Należy podać numer konta, o którym mają zostać wyświetlone informacje.
- Wyświetlone informacje muszą obejmować: numer konta, nazwisko właściciela, aktualne saldo, oprocentowanie oraz debet.

Wymagania Niefunkcjonalne:

1. Wydajność:

- Aplikacja musi działać szybko i sprawnie, nawet przy dużym obciążeniu (duża liczba kont).
- Czas reakcji na polecenia użytkownika musi być minimalny.

2. Bezpieczeństwo:

- Aplikacja musi chronić dane kont użytkowników przed nieautoryzowanym dostępem.
- Hasła nie są przechowywane w jawnej postaci.

3. Łatwość użytkowania:

- Interfejs konsolowy musi być intuicyjny i czytelny dla użytkownika.
- Obsługa aplikacji ma być prosta i zrozumiała.
- Komunikaty wyświetlane przez aplikację powinny być jasne i informatywne.

4. Modułowość:

- Aplikacja musi być zaprojektowana w sposób modułowy, co umożliwi łatwe dodawanie nowych funkcji w przyszłości.

5. Dostępność:

- Aplikacja ma działać w systemach Windows 10 i 11.

Rozdział 2

Opis techniczny projektu

2.1 Język i narzędzia

Aplikacja została napisana w języku programowania **C#** przy użyciu **Microsoft Visual Studio Community 2022** w wersji 17.12.3 (64-bitowej). W ramach **Visual Studio** zostały wykorzystane:

- **Język C#:** Wersja narzędzia języka C# to v4.12.0-3.24572.7.
- **Microsoft .NET Framework:** Aplikacja korzysta z .NET Framework w wersji v4.8.09037.
- **Menedżer pakietów NuGet:** Wersja v6.12.2, użyty do zarządzania pakietami.
- **Microsoft JVM Debugger** Wersja v1.0, użyty do debugowania aplikacji.
- **SQL Server Data Tools:** Wersja v17.12.83.3, użyte do integracji z bazą danych SQL.

2.2 Baza danych

Do zarządzania bazą danych wykorzystano **Microsoft SQL Server 2019**. Narzędziem do zarządzania bazą danych jest **SQL Server Management Studio v19.1.56.0**. Została zaprojektowana w celu przechowywania danych o kontach bankowych. Składa się z jednej tabeli:

- **Konto:**

- *numer_konta* (INT, IDENTITY(1,1), PRIMARY KEY): Unikalny numer konta bankowego. Jest to klucz główny tabeli i jest automatycznie inkrementowany przy dodawaniu nowego rekordu.
- *wlasciciel* (VARCHAR(100)): Nazwisko właściciela konta.
- *stan* (DECIMAL(15, 2)): Aktualne saldo konta.
- *oprocentowanie* (DECIMAL(5, 2)): Oprocentowanie konta.
- *debet* (DECIMAL(15, 2)): Maksymalny debet na koncie.

Opis relacji: Brak relacji z innymi tabelami, ze względu na prostotę aplikacji.

Typy danych:

- INT (Liczby całkowite) – do przechowywania numerów kont. Użyto też w opcji Identity do auto-incrementu numeru konta.
- VARCHAR (Tekst o zmiennej długości) - do przechowywania nazwisk właścicieli kont.
- DECIMAL (Liczby zmiennoprzecinkowe) - do przechowywania salda, debetu i oprocentowania.

2.3 Zarządzanie danymi

Dane są zarządzane za pomocą klasy *DatabaseManager*. Odpowiedzialna jest ona za połączenie z bazą danych i wykonywanie operacji CRUD (Create, Read, Update, Delete) na danych dotyczących kont. Połączenie z bazą danych jest realizowane za pomocą klasy *Connection*.

2.4 Minimalne wymagania sprzętowe

- **Procesor:** Intel Core i3 lub równoważny.
- **Pamięć RAM:** 2 GB.
- **Przestrzeń dyskowa:** 50 MB.
- **System operacyjny:** Windows 10 lub 11 (64-bit).

2.5 Platforma testowa

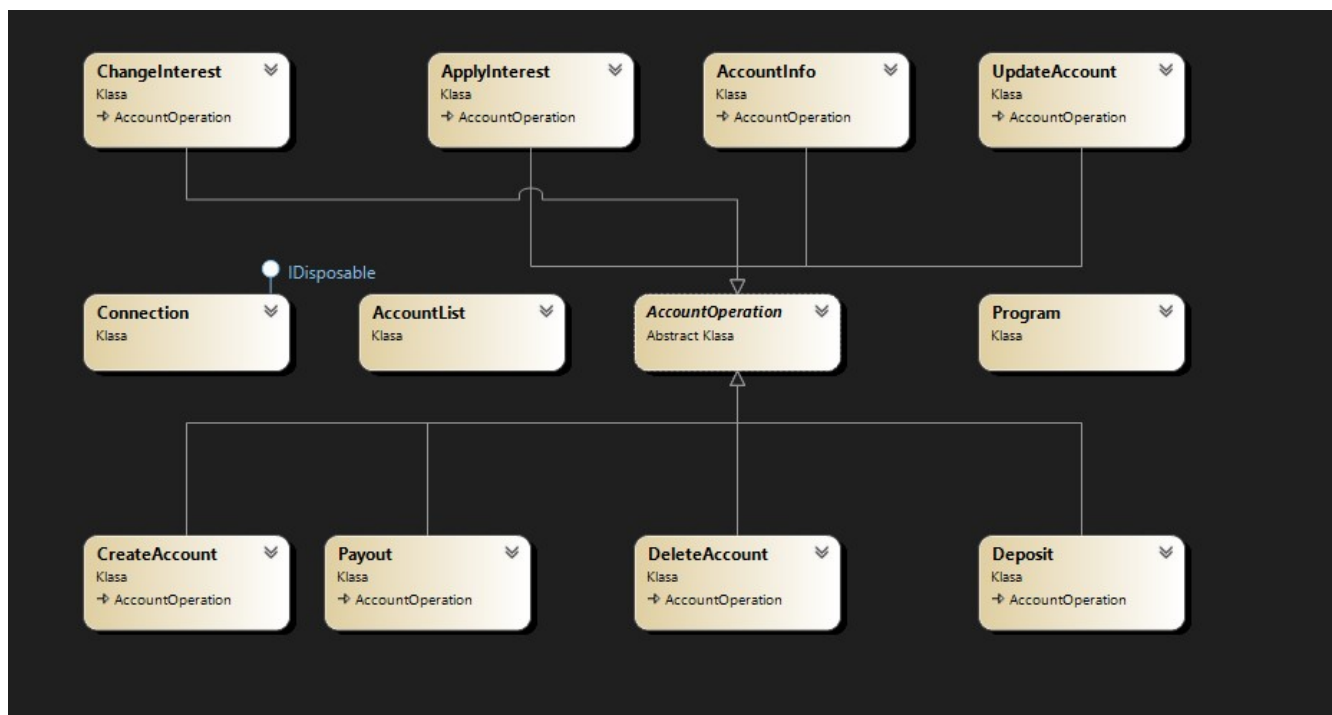
- **Procesor:** AMD Ryzen 7 5700X3D.
- **Karta graficzna:** AMD RX 6700XT.
- **Pamięć RAM:** 32 GB.
- **Przestrzeń dyskowa:** SSD 1 TB.
- **System operacyjny:** Windows 10 v22H2 (64-bit).

Rozdział 3

Opis struktury projektu

3.1 Diagram klas

Poniżej przedstawiam diagram klas, który obrazuje strukturę aplikacji, relacje między poszczególnymi klasami i ich atrybuty:



Rysunek 3.1: Diagram klas Aplikacji Bankowej

3.2 Opis klas

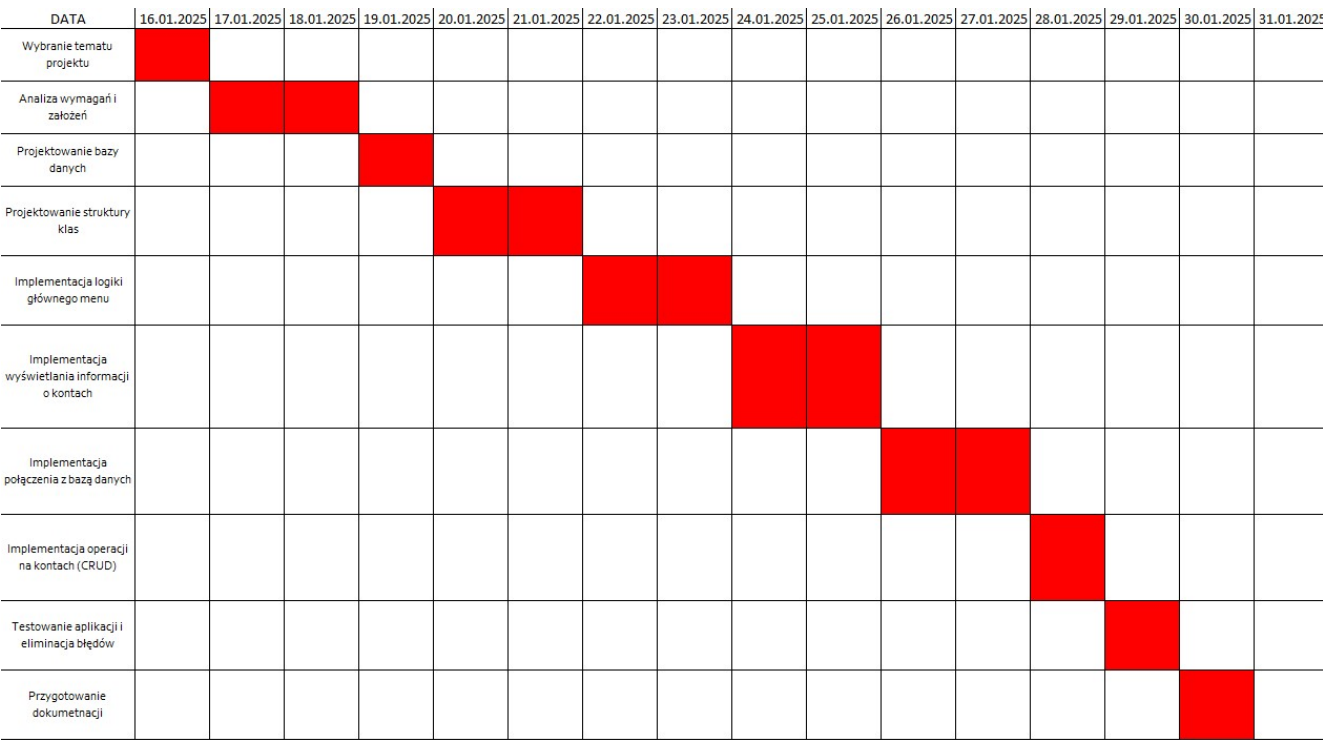
- **Program:** Główna klasa aplikacji. Zawiera metodę 'Main', która jest punktem startowym programu i zarządza działaniem aplikacji. Wyświetlaniem menu i wykonywaniem operacji.
- **AccountOperation:** Jest to klasa abstrakcyjna, która definiuje bazową strukturę dla operacji na kontach, posiadająca połączenie z bazą danych. Klasy wykonujące konkretne operacje (dodawanie, usuwanie, aktualizacja itd.) dziedziczą po tej klasie.
- **CreateAccount:** Klasa odpowiedzialna za tworzenie nowego konta w bazie danych, pobiera dane właściciela, stan, oprocentowanie i debet i wykonuje zapytanie INSERT.
- **AccountList:** Klasa statyczna udostępniająca funkcję listowania dostępnych kont z bazy danych, wyświetla numery kont i nazwiska właścicieli.
- **UpdateAccount:** Klasa odpowiedzialna za aktualizowanie istniejącego konta w bazie danych. Klasa ta pozwala na zmianę danych właściciela, stanu konta, oprocentowania i debetu.
- **DeleteAccount:** Klasa odpowiedzialna za usuwanie konta z bazy danych na podstawie przekazanego numeru konta.
- **Deposit:** Klasa odpowiedzialna za wpłacanie środków na konto, aktualizuje saldo konta po wpłacie podanej kwoty.
- **Payout:** Klasa odpowiedzialna za wypłacanie środków z konta, aktualizuje saldo konta po wypłacie podanej kwoty, sprawdza czy dostępne są wystarczające środki.
- **ApplyInterest:** Klasa odpowiedzialna za naliczanie odsetek od istniejącego konta, aktualizuje stan konta na podstawie oprocentowania.
- **ChangeInterest:** Klasa odpowiedzialna za zmianę oprocentowania konta. Pobiera numer konta i nowe oprocentowanie a następnie wykonuje zapytanie aktualizujące w bazie danych.
- **AccountInfo:** Klasa odpowiedzialna za wyświetlanie informacji o koncie z bazy danych. Pobiera z bazy danych i wyświetla dane o właścicielu, stanie, oprocentowaniu i debecie konta.
- **Connection:** Klasa odpowiedzialna za tworzenie i zarządzanie połączeniem z bazą danych, dostarcza połączenie dla operacji na kontach, otwiera i zamyka połączenie.

Rozdział 4

Harmonogram realizacji projektu

4.1 Diagram Gantta

Poniżej przedstawiam diagram Gantta, który szczegółowo opisuje harmonogram prac nad projektem:



Rysunek 4.1: Diagram Gantta

4.2 Problemy i rozwiązania

Podczas realizacji projektu napotkano kilka problemów, które zostały rozwiązane:

- **Problem z połączeniem z bazą danych:** Nieprawidłowa konfiguracja połączenia z bazą danych SQL.
- **Problem z implementacją operacji finansowych:** Nieprawidłowa implementacja metod do zarządzania danymi.
- **Problem z walidacją danych:** Brak pełnej walidacji danych wejściowych (np. ujemne kwoty wypłat).
- **Problem z wyświetlaniem informacji:** Informację w konsoli były wyświetlane w niewłaściwy sposób.

4.3 Repozytorium i system kontroli wersji

Wszystkie pliki związane z realizacją projektu są umieszczone w repozytorium GitHub:

<https://github.com/Szergonix/Konto>

Rozdział 5

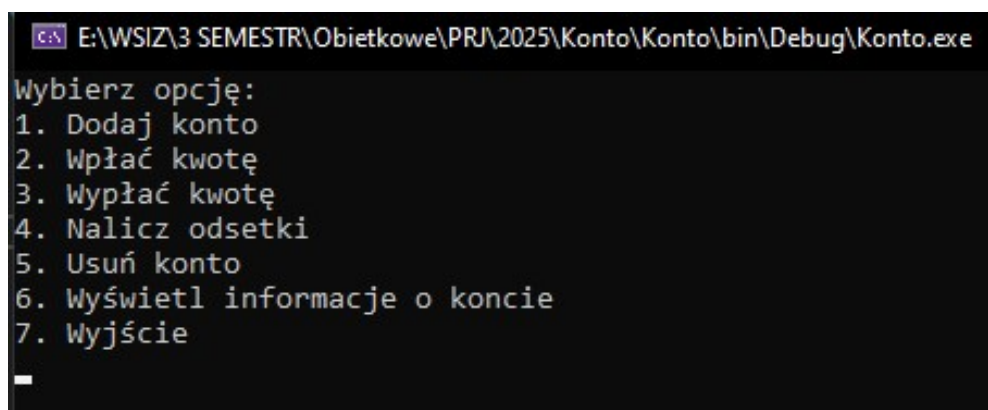
Prezentacja warstwy użytkowej projektu

5.1 Opis aplikacji

Aplikacja emulująca działanie konta bankowego jest aplikacją konsolową, która umożliwia bankierom zarządzanie kontami klientów. Została zaprojektowana z myślą o prostocie i intuicyjności. Użytkownik (bankier) po uruchomieniu aplikacji ma dostęp do menu z opcjami: dodawanie konta, wpłata środków, wypłata środków, naliczenie odsetek, zmiana oprocentowania, usunięcie konta, wyświetlanie informacji o koncie. Użytkownik komunikuje się z aplikacją za pomocą wpisywania odpowiedniej liczby, która odpowiada za określoną funkcjonalność. Użytkownik komunikuje się z aplikacją za pomocą wpisywania odpowiedniego numeru, który odpowiada za określoną funkcjonalność. Po wybraniu opcji aplikacja prosi o podanie numeru konta, a w niektórych przypadkach o dodatkowe dane, np. kwotę do wpłaty lub nowe oprocentowanie.

5.2 Menu Główne

Po uruchomieniu aplikacji, użytkownikowi zostaje wyświetlone menu z dostępnymi opcjami, gdzie użytkownik wybiera opcję poprzez wpisanie odpowiedniej cyfry:



```
E:\WSIZ\3 SEMESTR\Obietkowe\PRJ\2025\Konto\Konto\bin\Debug\Konto.exe
Wybierz opcję:
1. Dodaj konto
2. Wpłać kwotę
3. Wypłać kwotę
4. Nalicz odsetki
5. Usuń konto
6. Wyświetl informacje o koncie
7. Wyjście
_
```

Rysunek 5.1: Ekran główny aplikacji

5.3 Dodawanie konta

Po wybraniu opcji dodawania konta, aplikacja prosi o wprowadzenie danych nowego konta. Użytkownik wpisuje kolejno imię, nazwisko, stan początkowy, oprocentowanie i debet. Aplikacja po utworzeniu konta wyświetla wygenerowany numer konta:

```
1
Podaj właściciela: Adam Małysz
Podaj stan: 1000,00
Podaj oprocentowanie: 12,00
Podaj debet: -500,00
Konto utworzone z numerem: 14
```

Rysunek 5.2: Dodawanie konta

5.4 Wpłata środków

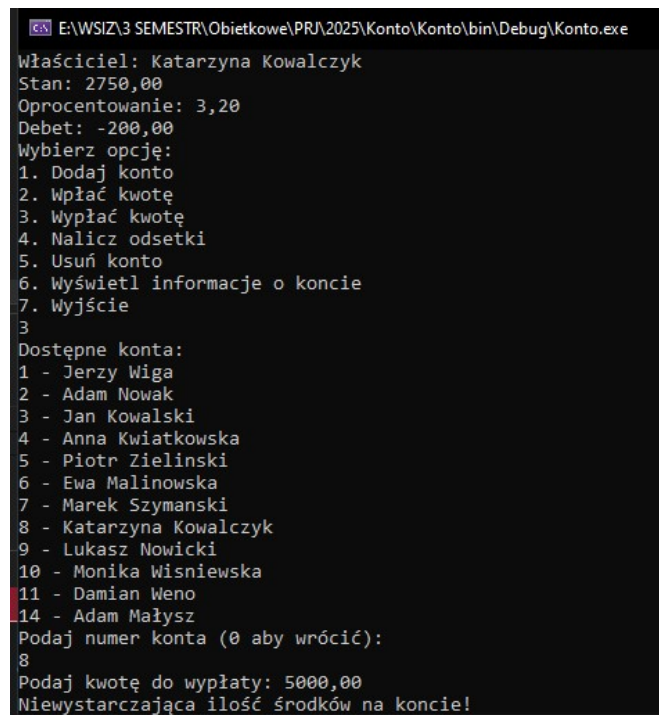
Po wybraniu opcji wpłaty środków, aplikacja wyświetla dostępne konta. Użytkownik wybiera numer konta i podaje kwotę wpłaty. Aplikacja aktualizuje saldo na koncie o podaną wartość:

```
2
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
14 - Adam Małysz
Podaj numer konta (0 aby wrócić):
1
Podaj kwotę do wpłaty: 1000,00
Dokonano wpłaty.
```

Rysunek 5.3: Wpłata środków

5.5 Wypłata środków

Podczas próby wypłaty środków, użytkownik jest informowany o dostępnych kontach i może wybrać numer konta. W sytuacji, gdy na koncie nie ma wystarczających środków do wypłaty, aplikacja informuje o tym użytkownika odpowiednim komunikatem:



```
E:\WSIZ\3 SEMESTR\Obietkowe\PRJ\2025\Konto\Konto\bin\Debug\Konto.exe
Właściciel: Katarzyna Kowalczyk
Stan: 2750,00
Oprocentowanie: 3,20
Debet: -200,00
Wybierz opcję:
1. Dodaj konto
2. Wpłać kwotę
3. Wypłać kwotę
4. Nalicz odsetki
5. Usuń konto
6. Wyświetl informacje o koncie
7. Wyjście
3
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
14 - Adam Małysz
Podaj numer konta (0 aby wrócić):
8
Podaj kwotę do wypłaty: 5000,00
Niewystarczająca ilość środków na koncie!
```

Rysunek 5.4: Wypłata środków (brak środków)

5.6 Naliczanie odsetek

Po wybraniu opcji naliczania odsetek aplikacja wyświetla dostępne konta. Po wybraniu konta, aplikacja nalicza odsetki:

```
4
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
14 - Adam Małysz
Podaj numer konta (0 aby wrócić):
7
Odsetki zostały naliczone.
Wybierz opcję:
1. Dodaj konto
2. Wpłać kwotę
3. Wypłać kwotę
4. Nalicz odsetki
5. Zmień oprocentowanie
6. Wyświetl informacje o koncie
7. Usuń konto
8. Wyjście
```

Rysunek 5.5: Naliczanie odsetek

5.7 Zmiana oprocentowania

Podczas zmiany oprocentowania, aplikacja prosi o wprowadzenie nowego oprocentowania. Użytkownik wpisuje nowy procent i zostaje poinformowany o dokonanej zmianie:

```
5
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
Podaj numer konta (0 aby wrócić):
9
Podaj nowe oprocentowanie: 13,00
Oprocentowanie zostało zmienione.
```

Rysunek 5.6: Zmiana oprocentowania

5.8 Wyświetlanie informacji o koncie

Aby wyświetlić informacje o koncie, należy wybrać odpowiednią opcję z menu. Aplikacja pobiera z bazy danych i wyświetla szczegółowe informacje o wybranym koncie:

```
6
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
Podaj numer konta (0 aby wrócić):
4
Właściciel: Anna Kwiatkowska
Stan: 2500,00
Oprocentowanie: 2,50
Debet: -150,00
```

Rysunek 5.7: Wyświetlanie informacji o koncie

5.9 Usuwanie konta

Podczas usuwania konta, aplikacja informuje o dostępnych kontach i umożliwia wybór konta do usunięcia. Po wybraniu numeru konta, konto jest usuwane z bazy danych.:

```
7
Dostępne konta:
1 - Jerzy Wiga
2 - Adam Nowak
3 - Jan Kowalski
4 - Anna Kwiatkowska
5 - Piotr Zielinski
6 - Ewa Malinowska
7 - Marek Szymanski
8 - Katarzyna Kowalczyk
9 - Lukasz Nowicki
10 - Monika Wisniewska
11 - Damian Weno
14 - Adam Małysz
Podaj numer konta (0 aby wrócić):
14
Czy na pewno chcesz usunąć konto? (tak/nie): tak
Konto zostało pomyślnie usunięte.
```

Rysunek 5.8: Usuwanie konta

Rozdział 6

Podsumowanie

Niniejsza aplikacja stanowi dopiero pierwszy etap jej rozwoju. Planowane kroki obejmują stworzenie intuicyjnego interfejsu użytkownika, który znacząco podniesie komfort i dostępność aplikacji. W tym celu, rozważane wprowadzenie graficznego interfejsu użytkownika (GUI), który znacząco podniesie jej atrakcyjność i intuicyjność lub technologii webowych, które umożliwią transformację aplikacji w responsywny serwis internetowy.

Bibliografia

- [1] <https://learn.microsoft.com/pl-pl/dotnet/csharp/>
- [2] <https://learn.microsoft.com/pl-pl/sql>
- [3] <https://learn.microsoft.com/pl-pl/sql/ssms/sql-server-management-studio-ssms>
- [4] <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient?view=netframework-4.8>
- [5] Język C. Mirosław J. Kubiak, Helion, 2019.
- [6] SQL w przykładach. Paweł Wimmer, Helion, 2022.
- [7] Programowanie w języku C. Marcin Lis, PWN, 2023.