

Projekt zaliczeniowy



Programowanie obiektowe
Rok akademicki 2025/2026

Autorzy:

Alicja Gadula

System zarządzania bankiem

Projekt przedstawia prosty system zarządzania bankiem, w języku C# z wykorzystaniem programowania obiektowego oraz interfejsu WPF. Aplikacja umożliwia zarządzanie kontami bankowymi, wykonywanie podstawowych operacji finansowych (wpłacanie i wypłacanie) oraz zapis i odczyt danych z pliku XML.

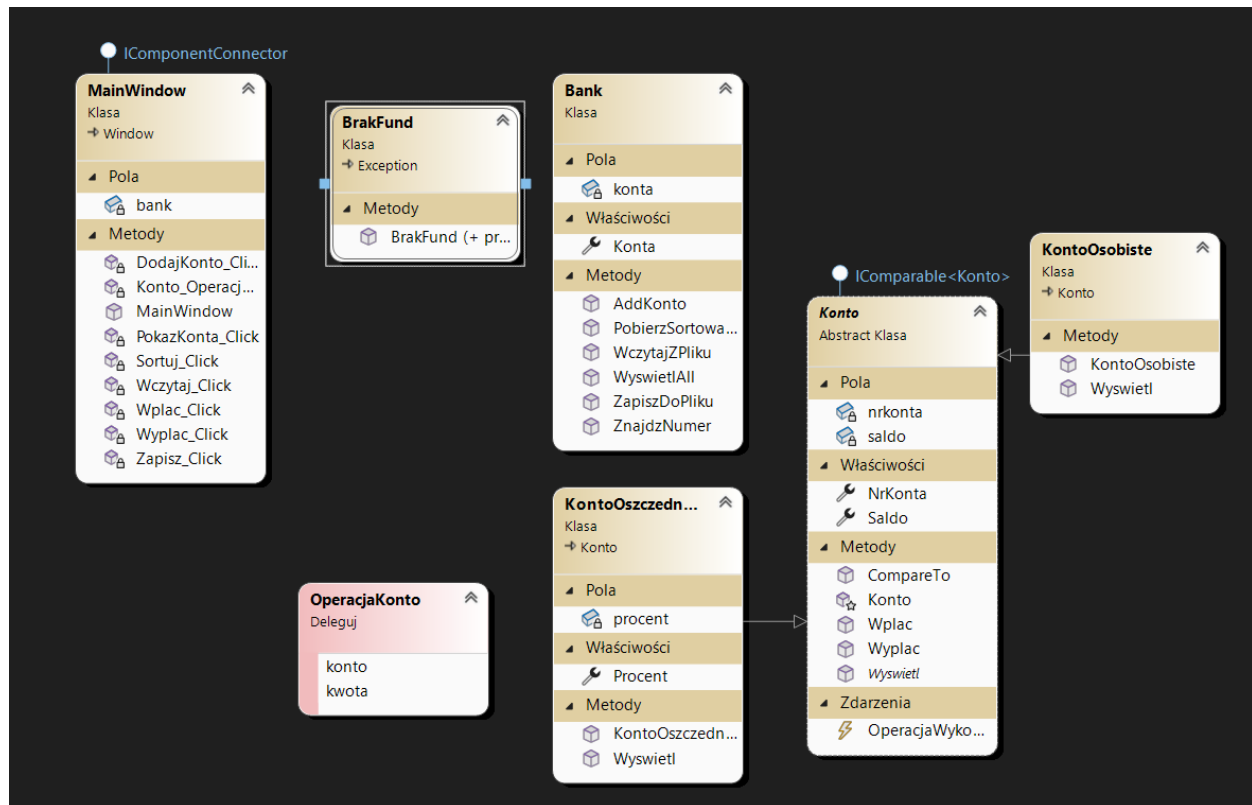
Podział ról

Autorka: Alicja Gadula

Projekt został wykonany indywidualnie

- implementacja klas (konta bankowych, Bank),
- implementacja interfejsu (WPF),
- obsługa zapisu i odczytu danych do pliku XML,
- przygotowanie sprawozdania

Diagram klas



Opis funkcjonalności

- dodawanie nowych kont bankowych
- wyświetlanie listy wszystkich kont
- wpłata środków na konto
- wypłata środków z konta
- sortowanie kont według salda
- zapis danych do pliku XML,
- odczyt danych z pliku XML

Funkcjonalności te realizowane są przez klasy:

- Bank
- Konto i klasy pochodne
- interfejs graficzny MainWindow

Opis klas

1. Klasa *Konto* (abstract)

Jest klasą bazową reprezentującą ogólne konto bankowe. Zawiera wspólne cechy i zachowania dla wszystkich kont.

Rola w projekcie:

- podstawa dla klas dziedziczących (*KontoOsobiste*, *KontoOszczednosc*),
- wymusza implementację metody `Wyswietl()` w klasach pochodnych,

Pola i właściwości:

- `NrKonta` – numer konta, chroniony przed pustą wartością,
- `Saldo` – aktualne saldo konta, zabezpieczone przed wartościami ujemnymi.

Zastosowano modyfikatory dostępu `protected` oraz `private`, aby uniemożliwić bezpośrednią modyfikację pól spoza klasy.

Klasa implementuje interfejs `IComparable<Konto>`, co umożliwia sortowanie kont według salda.

2. Klasa *KontoOsobiste*

Klasa dziedziczy po klasie *Konto* i reprezentuje standardowe konta bankowe.

Rola w projekcie:

- obsługa podstawowych operacji bankowych
- możliwość wpłaty i wypłaty środków

3. Klasa *KontoOszczednosc*

Klasa dziedziczy po klasie *Konto* i reprezentuje konta oszczędnościowe.

Rola w projekcie:

- przechowywanie środków wraz z oprocentowaniem
- rozszerzenie funkcjonalności konta o oprocentowanie

4. Klasa *Bank*

Klasa zarządza kolekcją kont bankowych.

Rola w projekcie:

- przechowywanie listy kont
- dodawanie nowych kont
- wyszukiwanie kont po numerze
- sortowanie kont według salda
- zapis i odczyt danych z pliku XML

Zastosowano `ICollection<Konto>`, aby uniemożliwić modyfikację listy kont z zewnątrz klasy.

5. Delegat i zdarzenie

W projekcie zastosowano delegat `OperacjaKonto` oraz zdarzenie `OperacjaWykonana`, które informuje o wykonaniu wpłaty lub wypłaty.

Rola w projekcie:

- wyświetlanie komunikatów po wykonaniu operacji finansowych.

Instrukcja obsługi

01. Uruchomić aplikację.
02. Wprowadzić numer konta oraz saldo początkowe.
03. Wybrać typ konta.
04. Kliknąć przycisk Dodaj konto.
05. Aby wykonać wpłatę / wypłatę, podać numer konta oraz kwotę i wybrać odpowiedni przycisk.
06. Aby zapisać dane użyć przycisku Zapisz do XML.
07. Aby odczytać dane z pliku, użyć przycisku Wczytaj z XML.

Testowanie

Aplikacja była testowana manualnie (poprzez dodanie użytkowników i interfejs).

Sprawdzono poprawność:

- dodawania kont
- wykonywania wpłat i wypłat
- sortowania kont
- zapisu i odczytu danych z pliku XML
- obsługi wyjątków

Podsumowanie

Projekt spełnia założenia programowania obiektowego oraz prezentuje praktyczne zastosowanie klas abstrakcyjnych, dziedziczenia, delegatów i zdarzeń. Aplikacja jest czytelna, funkcjonalna i może być dalej rozwijana.