

Dokumentacja projektu:
Biblioteka do obsługi macierzy



Autor projektu:
Sylwester Dąbrowski
Prowadzący:
mgr inż. Jakub Zimnol
Przedmiot:
Języki Programowania Obiektowego

9 stycznia 2026

1 Opis projektu

Projekt przedstawia implementację prostej biblioteki do obsługi macierzy matematycznych w języku C++, z wykorzystaniem mechanizmów programowania generycznego (szablonów). Biblioteka umożliwia tworzenie i wykonywanie operacji na macierzach prostokątnych, kwadratowych oraz trójkątnych, z zachowaniem poprawnych własności matematycznych i kontroli błędów.

Celem projektu było zaprojektowanie czytelnego i bezpiecznego interfejsu umożliwiającego wykonywanie podstawowych operacji algebraicznych na macierzach, takich jak dodawanie, odejmowanie, mnożenie, transpozycja, obliczanie wyznacznika oraz macierzy odwrotnej.

Biblioteka może znaleźć zastosowanie w prostych obliczeniach numerycznych, zadaniach dydaktycznych z algebry liniowej oraz jako baza do dalszej rozbudowy (np. o bardziej zaawansowane algorytmy numeryczne).

2 Struktura i opis klas

Projekt składa się z trzech głównych klas, zorganizowanych hierarchicznie:

- **Matrix<T>** – klasa bazowa reprezentująca ogólną macierz prostokątną o dowolnych wymiarach. Udostępnia podstawowe operacje na macierzach oraz mechanizmy dostępu do elementów.
- **SquareMatrix<T>** – klasa dziedzicząca po **Matrix<T>**, przeznaczona do obsługi macierzy kwadratowych. Rozszerza funkcjonalność o operacje charakterystyczne dla macierzy kwadratowych, takie jak obliczanie wyznacznika, macierzy dopełnień algebraicznych oraz macierzy odwrotnej.
- **TriangularMatrix<T>** – klasa dziedzicząca po **SquareMatrix<T>**, reprezentująca macierze trójkątne (dolne lub górne). Klasa wymusza zachowanie struktury trójkątnej poprzez kontrolę zapisu elementów oraz oferuje zoptymalizowane obliczanie wyznacznika.

Relacje dziedziczenia odzwierciedlają zależności matematyczne pomiędzy typami macierzy, co pozwala na ponowne wykorzystanie kodu i zachowanie spójności interfejsu.

3 Uruchomienie i kompilacja

Projekt wykorzystuje system budowania CMake. Do skompilowania aplikacji wymagany jest kompilator obsługujący standard C++20.

Sekwencja poleceń umożliwiająca poprawne zbudowanie projektu:

```
mkdir build
cd build
cmake ..
make
```

W wyniku kompilacji powstaje plik wykonywalny `matrixSD`, który zawiera program demonstracyjny prezentujący możliwości zaimplementowanej biblioteki.

4 Dodatkowe informacje

Kod projektu został podzielony na pliki nagłówkowe i źródłowe, a jego dokumentacja techniczna została przygotowana w formacie Doxygen bezpośrednio w plikach nagłówkowych.

Plik `main.cpp` pełni rolę programu testowego, demonstrującego poprawność działania zaimplementowanych klas, obsługę wyjątków oraz przykładowe obliczenia wykonane za pomocą funkcji z biblioteki.