



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do obsługi macierzy

z przedmiotu

Języki Programowania Obiektowego

Elektronika i Telekomunikacja, III rok

Mikołaj Gruszka

Piątek 15:00

prowadzący: mgr. inż. Jakub Zimnol

10.01.2025r.

1. Cel

Projekt miał na celu implementację biblioteki do obsługi macierzy w języku C++. W skład zaprojektowanej przeze mnie biblioteki wchodziły dwie klasy: **Matrix** oraz **SquareMatrix**. Biblioteka pozwala na tworzenie macierzy tych klas oraz operacje na nich. Obiekty powyższych klas przyjmują różne typy danych jako argument.

2. Opis biblioteki

Biblioteka obsługuje trzy podstawowe sposoby tworzenia macierzy:

- a) **podstawowy konstruktor** (zostaje wtedy utworzona pusta macierz)
- b) **podanie wymiarów oraz wartości początkowej** (zostaje wtedy utworzona macierz o wartościach równych wartości początkowej)
- c) **podanie wektora wektorów** (argument jakim jest wektor wektorów staje się naszą macierzą)

Na macierzach można wykonywać następujące operacje:

- a) Dodawanie (+, +=)
- b) Odejmowanie (-, -=)
- c) Mnożenie (*, *=) - można pomnożyć przez inną macierz lub przez skalar

Dla rozwinięcia funkcjonalności biblioteki dodałem również ważne funkcje takie jak:

- a) Porównywanie (==, !=)
- b) Przypisanie wartości innej macierzy (=)
- c) Operator () - pozwala na dostęp do elementu macierzy o konkretnych współrzędnych
- d) Operator << - pozwala na wyświetlenie macierzy

Kolejne interesujące funkcje pomagające w obsłudze macierzy i ich modyfikacji to:

- a) `getRow()`, `getCol()` - zwracają odpowiednio wiersz lub kolumnę w formie wektora
- b) `getRows()`, `getCols()` - zwracają odpowiednio ilość wierszy lub kolumn macierzy
- c) `addRow()`, `addCol()` - dodają odpowiednio wiersz lub kolumnę na danej pozycji
- d) `removeRow()`, `removeCol()` - usuwają odpowiednio wiersz lub kolumnę na danej pozycji
- e) `setValues()`, `setRandomValues()`, `setOnes()`, `setZeroes()` - zmieniają elementy macierzy na konkretne wartości (wybrane przez użytkownika, losowe w zakresie, jedynki, zera)
- f) `transpose()` - wykonuje operację transpozycji macierzy

Dwie funkcje zdefiniowane jedynie dla klasy `SquareMatrix`:

- a) `identity()` - tworzy macierz jednostkową o konkretnych wymiarach
- b) `determinant()` - oblicza wyznacznik macierzy

Funkcje `addRow()`, `addCol()`, `removeRow()`, `removeCol()` zostały ograniczone dla klasy `SquareMatrix`, ponieważ nie chcemy zmieniać wymiarów obiektu tej klasy.

3. Kompilacja

Aby skompilować i uruchomić plik demonstracyjny należy przejść do foldera /src i użyć poniższych komend:

```
g++ -o example main.cpp
```

```
./example
```