

REV	DATA	ZMIANY
0.1	07.05.2016	<i>Bogusław Cyganek (cyganek@agh.edu.pl)</i>

# KALKULATOR DLA ELEKTRONIKÓW I INFORMTYKÓW

Autor: Jakub Puch

## Akademia Górniczo-Hutnicza

2024

## Spis treści

---

[1.](#) 4

[2.](#) 7

[3.](#) 7

[4.](#) 9

[5.](#) Błąd! Nie zdefiniowano zakładki.

[6.](#) 12

[7.](#) 13

[BIBLIOGRAFIA](#)

19

## **Lista oznaczeń**

---

[tu dodajemy swoje oznaczenia użyte w tym dokumencie]

bin	System binarny
oct	System ósemkowy
hex	System szesnastkowym
float	Liczby zmiennoprzecinkowe

## **1. Wstęp**

Ten projekt to implementacja kalkulatora w języku C++, który obsługuje różne operacje matematyczne, takie jak dodawanie, odejmowanie, mnożenie, dzielenie oraz potęgowanie. Kalkulator umożliwia również konwersję liczb między różnymi systemami liczbowymi, takimi jak binarny, ósemkowy, dziesiętny i szesnastkowy. Dodatkowo, projekt zawiera mechanizmy obsługi błędów, takie jak sprawdzanie dzielenia przez zero, niepoprawnych operacji matematycznych czy nieprawidłowych danych wejściowych

# **1. Wymagania projektowe**

a) Celem projektu jest stworzenie aplikacji – kalkulatora wykonującego kilka podstawowych działań dla elektroników, czyli z możliwością przełączenia się na systemy: bin, oct, hex, albo float.

b) Aplikacja składa się zasadniczo z dwóch modułów:

- systemu klas, hierarchii, design patterns realizujących działania kalkulatora (jak najbardziej niezależne od platformy i języka)
- interfejsu użytkownika (prosty dialog pod Windows lub Linux)

c) Należy zwrócić szczególną uwagę na przejrzystość projektu oraz możliwość zastosowania poznanych technik projektowania i programowania obiektowego.

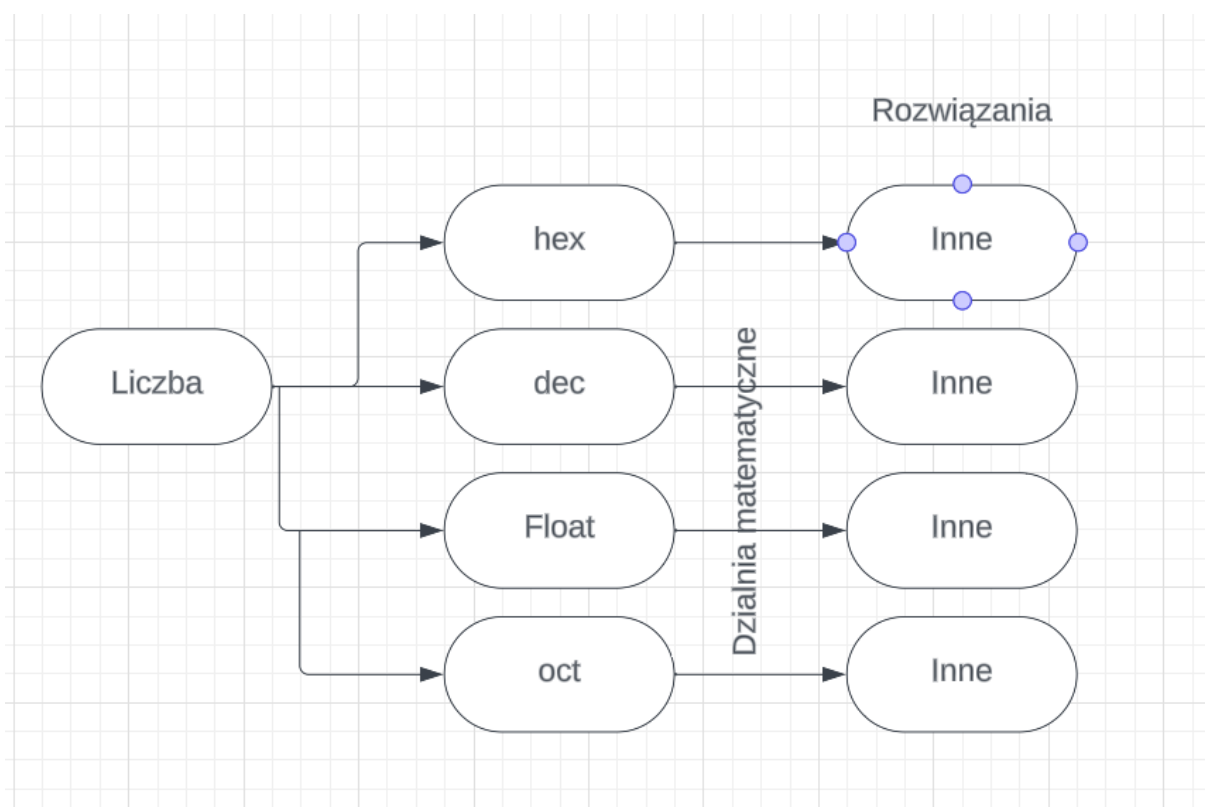
d) Obowiązkowe jest korzystanie z narzędzi: debugger, CMake, git oraz bazy GitHub (osobiste konto); Ważne jest podłączenie biblioteki do testowania oprogramowania Google Test GTest (opis m.in. znajduje się z appendixie książki o C++).

## **2. Funkcjonalność (*functionality*)**

Kalkulator został zaprojektowany z myślą o intuicyjnym interfejsie, który umożliwia łatwe korzystanie z różnych funkcji oraz w dowolnym momencie zmianę systemu liczbowego. W programie jest też zawarta kolejność działań co jest bardzo ciekawym dodatkiem do całego programu. Program także „chroni użytkownika przed nim samym”: W przypadku dzielenia przez zero, program zgłasza wyjątek "Division by zero". Również obsługuje błędy, takie jak próba konwersji liczby ujemnej na system binarny, ósemkowy czy szesnastkowy.

### 3. Analiza problemu (*problem analysis*)

Problemem jest różnorodność sposobów w jaki odbywają się działania arytmetyczne w różnych systemach liczbowych każde działanie generuje ten sam wynik w innym systemie liczbowym.









# 4. Projekt techniczny (*technical design*)

Calculator Class Reference

Static Public Member Functions | List of all members

Static Public Member Functions

static std::wstring

convertBinaryToDecimalExpression (const std::wstring &binaryExpr)

static std::wstring

decimalToBinary (double decimalValue)

static double

binaryToDecimal (const std::wstring &binaryNumber)

static std::wstring

convertOctalToDecimalExpression (const std::wstring &octalExpr)

static std::wstring

decimalToOctal (double decimalValue)

static double

octalToDecimal (const std::wstring &octalNumber)

static std::wstring

convertHexToDecimalExpression (const std::wstring &hexExpr)

static std::wstring

decimalToHex (double decimalValue)

static double

hexToDecimal (const std::wstring &hexNumber)

static void

switchMode (CalculatorMode newMode)

static CalculatorMode

getMode ()

static std::unique\_ptr< Expression >

parse (const std::wstring &expr)

static std::unique\_ptr< Expression >

parsePrimary (std::wstring::const\_iterator &it, const std::wstring::const\_iterator &end)

Expression Class Reference

abstract

Public Member Functions | List of all members

Inheritance diagram for Expression:

Expression

BinaryExpression

ExponentiationExpression

NumberExpression

Public Member Functions

virtual double

evaluate () const =0

BinaryExpression Class Reference

Public Member Functions | List of all members

Inheritance diagram for BinaryExpression:

Expression

BinaryExpression

Public Member Functions

BinaryExpression (wchar\_t operation, std::unique\_ptr< Expression > left, std::unique\_ptr< Expression > right)

double

evaluate () const override

Member Function Documentation

◆ evaluate()


double BinaryExpression::evaluate ( ) const

Implements Expression.

### ExponentiationExpression Class Reference

Public Member Functions | List of all members

Inheritance diagram for ExponentiationExpression:



```
graph TD; Expression --> ExponentiationExpression
```

Public Member Functions

**ExponentiationExpression** (std::unique\_ptr< Expression > baseExpression, std::unique\_ptr< Expression > exponentExpression)

double `evaluate` () const override

Member Function Documentation

**• evaluate()**

double ExponentiationExpression::evaluate ( ) const


Implements Expression.

[Source](#) [Definition](#)

### NumberExpression Class Reference

Public Member Functions | List of all members

Inheritance diagram for NumberExpression:



```
graph TD; Expression --> NumberExpression
```

Public Member Functions

**NumberExpression** (double value)

double `evaluate` () const override

Member Function Documentation

**• evaluate()**

double NumberExpression::evaluate ( ) const

Implements Expression.

[Source](#) [Definition](#)

Wykorzystano program doxygen.

## 5. Opis realizacji (*implementation report*)

Projekt wykonano wykorzystując:

- Microsoft Visual Studio 2022
- Kompilator MSCV 2022
- CMake (3.28.1)
- Google test

- **Doxygen (do tworzenia diagramu hierarchii klas)**

Rozwiązania:

- **Utworzono interfejs użytkownika**
- **Zastosowanie narzędzi: debugger, CMake**
- **Utworzenie kalkulatora wykonującego podstawowe działania dla wszystkich systemów liczbowych**
- **Wykonano przejrzysty kod stosując poznane techniki projektowania oraz programowania obiektowego**

## 6. Opis wykonanych testów

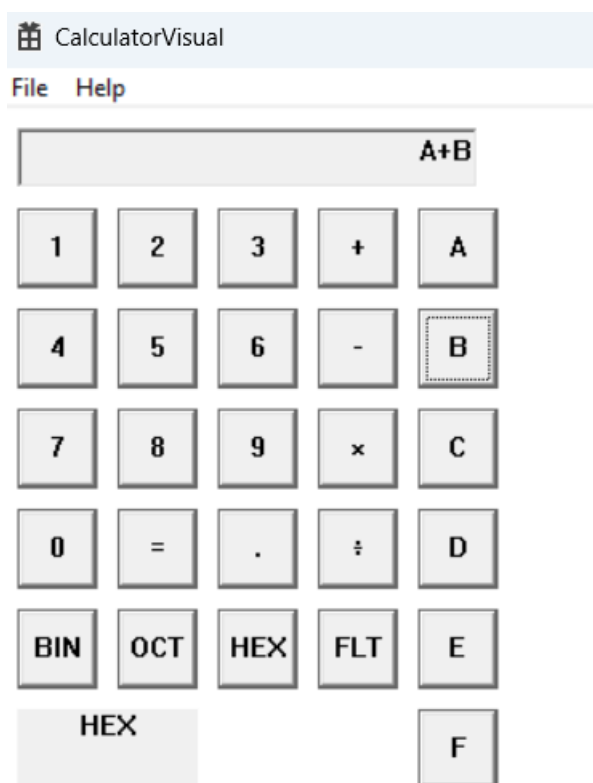
Testy były przeprowadzane za pomocą GTestu, sprawdzone zostały wszystkie funkcje.

```
[=====] Running 5 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 5 tests from CalculatorTests
[ RUN    ] CalculatorTests.CanAddNumbers
[ OK     ] CalculatorTests.CanAddNumbers (0 ms)
[ RUN    ] CalculatorTests.CanAddMoreNumbers
[ OK     ] CalculatorTests.CanAddMoreNumbers (0 ms)
[ RUN    ] CalculatorTests.OrderOfOperations_AdditionAndMultiplication
[ OK     ] CalculatorTests.OrderOfOperations_AdditionAndMultiplication (0 ms)
[ RUN    ] CalculatorTests.OrderOfOperations_MultiplicationAndAddition
[ OK     ] CalculatorTests.OrderOfOperations_MultiplicationAndAddition (0 ms)
[ RUN    ] CalculatorTests.OrderOfOperations_SubtractionAndDivision
[ OK     ] CalculatorTests.OrderOfOperations_SubtractionAndDivision (0 ms)
[-----] 5 tests from CalculatorTests (3 ms total)

[-----] Global test environment tear-down
[=====] 5 tests from 1 test suite ran. (6 ms total)
[ PASSED ] 5 tests.
```

Zdjęcie Gtestu dla kilku funkcji reszta została przeprowadzona w ten sam sposób

## 7. Podręcznik użytkownika (*user's manual*)



Program startowo posługuje się systemem zmiennoprzecinkowym ma również zablokowaną możliwość wyboru opcji nie dostępnych dla float. Aby wyświetlić wynik po wypisaniu wszystkich operacji należy wcisnąć przycisk =, aby wyzerować trzeba wybrać system jakiego będziemy chcieli stosować następnie.

Aby zmienić wybieramy jeden z dostępnych sytemów: bin, oct, hex, flt

W lewej dolnej części ekranu wyświetla się stosowany przez nas system liczbowy.

## **Bibliografia**

1. Cyganek b.: Programowanie w języku C++  
Wprowadzenie dla inżynierów. PWN, 2023.
2. Strona internetowa: [cplusplus.com](http://cplusplus.com)
3. Strona internetowa: <https://cpp0x.pl>
5. Strona internetowa: <https://www.geeksforgeeks.org/>
6. Strona internetowa: <https://chat.openai.com>