

REV	DATA	ZMIANY
0.1	10.01.2025	<i>Jakub Marchewczyk</i> <i>marchewczyk@student.agh.edu.pl</i>
0.2	19.01.2025	<i>Jakub Marchewczyk</i> <i>marchewczyk@student.agh.edu.pl</i>
0.3	25.01.2025	<i>Jakub Marchewczyk</i> <i>marchewczyk@student.agh.edu.pl</i>

# KALKULATOR DLA ELEKTRONIKÓW/INFORMATYKÓW

Autor: Jakub Marchewczyk  
Akademia Górniczo-Hutnicza

Kraków (C) 2025

## Spis treści

1.	WSTĘP.....	4
2.	FUNKCJONALNOŚĆ ( <i>FUNCTIONALITY</i> ) .....	5
3.	ANALIZA PROBLEMU ( <i>PROBLEM ANALYSIS</i> ) .....	6
4.	PROJEKT TECHNICZNY ( <i>TECHNICAL DESIGN</i> ) .....	8
4.1	DIAGRAMY UML.....	8
4.2	ROLE KLAS W PROJEKCIE .....	10
5.	OPIS REALIZACJI ( <i>IMPLEMENTATION REPORT</i> ) .....	11
6.	OPIS WYKONANYCH TESTÓW ( <i>TESTING REPORT</i> ). .....	12
7.	PODRĘCZNIK UŻYTKOWNIKA ( <i>USER'S MANUAL</i> ).....	13

## Lista oznaczeń

DEC	Dziesiętny system liczbowy
BIN	Dwójkowy system liczbowy
OCT	Ósemkowy system liczbowy
HEX	Heksadecymalny system liczbowy
UML	Unified Modeling Language

# 1. Wstęp

Dokument dotyczy opracowania kalkulatora dla elektroników/programistów wykonującego podstawowe operacje matematyczne oraz logiczne w systemie dziesiętnym, binarnym, ósemkowym i heksadecymalnym oraz konwersje pomiędzy tymi systemami.

## 2. Funkcjonalność (*functionality*)

### 2.1 Możliwości kalkulatora

Kalkulator posiada możliwość wykonywania podstawowych operacji matematycznych na liczbach w systemie dziesiętnym (w tym możliwość obliczeń na liczbach zmiennoprzecinkowych), binarnym, ósemkowym oraz heksadecymalnym. Możliwe jest również wykonywanie operacji logicznych bitowo, takich jak: OR, AND, NOR, NAND, XOR, XNOR.

### 2.2 Działanie przycisków kalkulatora

Przycisk	Działanie
Cyfry 0-9, litery A-F, znak ‘.’	Wprowadzenie danych do wyświetlacza
Przycisk +, -, *, /	Wybór działania matematycznego do wykonania
Przycisk OR, AND, NOR, NAND, XOR, XNOR	Wybór funkcji logicznej
Przycisk =	Wykonanie wcześniej wybranego równania, wyświetlenie wyniku.
Przycisk Backspace	Usunięcie ostatniego znaku z wyświetlacza
Przycisk +/-	Zmiana znaku wyświetlanej liczby
Przycisk DEC, BIN, OCT, HEX	Wybór systemu liczbowego
Przycisk CE	Wyczyszczenie wyświetlacza, rozpoczęcie nowego równania
Przycisk ANS	Wyświetlenie wyniku ostatniego równania

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

Konwersja pomiędzy systemami liczbowymi. Przedstawienie liczby jest pojęciem abstrakcyjnym, gdyż zależy od systemu, z którego korzystamy. Działania matematyczne odbywają się w taki sam sposób dla różnych systemów, różnią się one tylko przedstawieniem liczby.

System dwójkowy (binarny) – system o podstawie 2, zapis liczb przy użyciu dwóch cyfr: 0 i 1 wskazujących kolejne potęgi liczby 2 (system pozycyjny).

Przykładowy zapis liczby 22 w systemie binarnym prezentuje się następująco: 10110. Można to sprawdzić przemnażając cyfry liczby w BIN przez odpowiadające im potęgi cyfry 2 i sumując.

**Przykład 1.**

$$1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 22$$

System ósemkowy – podstawie 8, zapis liczb przy użyciu cyfr 0-7. Podobnie jak system binarny jest to system pozycyjny więc ciąg cyfr w liczbie jest ciągiem mnożników odpowiednich potęg podstawy w tym przypadku cyfry 8. Przykładowy zapis liczby 22 w systemie ósemkowym prezentuje się następująco: 26. Można to sprawdzić przemnażając cyfry liczby w OCT przez odpowiadające im potęgi cyfry 8 i sumując.

**Przykład 2.**

$$2 \cdot 8^1 + 6 \cdot 8^0 = 22$$

Konwersję liczby z systemu dwójkowego do systemu ósemkowego dokonujemy dzieląc ciąg binarny w „trójki” z czego każda trójka reprezentuje jedną cyfrę w systemie ósemkowym.

**Przykład 3.**

Aby przekształcić z BIN na OCT należy uzupełnić ilość cyfr tak aby otrzymać wielokrotność 3.

$$10110 = 010\ 110$$

Dzieląc powyższy ciąg binarny na dwie liczby po 3 cyfry otrzymamy

$$0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

$$1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 4$$

$$010110_2 = 24_8$$

System szesnastkowy (heksadecymalny) – pozycyjny system o podstawie 16, zapis liczb przy użyciu cyfr: 0-9 oraz liter A-F, system ten zatem wykorzystuje 16 znaków wskazujących kolejne potęgi liczby 16.

Przykładowy zapis liczby 22 w HEX prezentuje się następująco: 16. Można to sprawdzić przemnażając cyfry liczby w HEX przez odpowiadające im potęgi cyfry 16 i sumując.

**Przykład 4.**

$$1 \cdot 16^1 + 6 \cdot 16^0 = 22$$

Konwersję liczby z systemu dwójkowego do systemu szesnastkowego dokonujemy dzieląc ciąg binarny w „czwórki” z czego każda czwórka reprezentuje jeden symbol w systemie szesnastkowym.

**Przykład 5.**

Aby przekształcić z BIN na HEX należy uzupełnić ilość cyfr tak aby otrzymać wielokrotność 4.

$$10110 = 0001\ 0110$$

Dzieląc powyższy ciąg binarny na dwie liczby po 4 cyfry otrzymamy

$$0 \cdot 2^4 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1$$

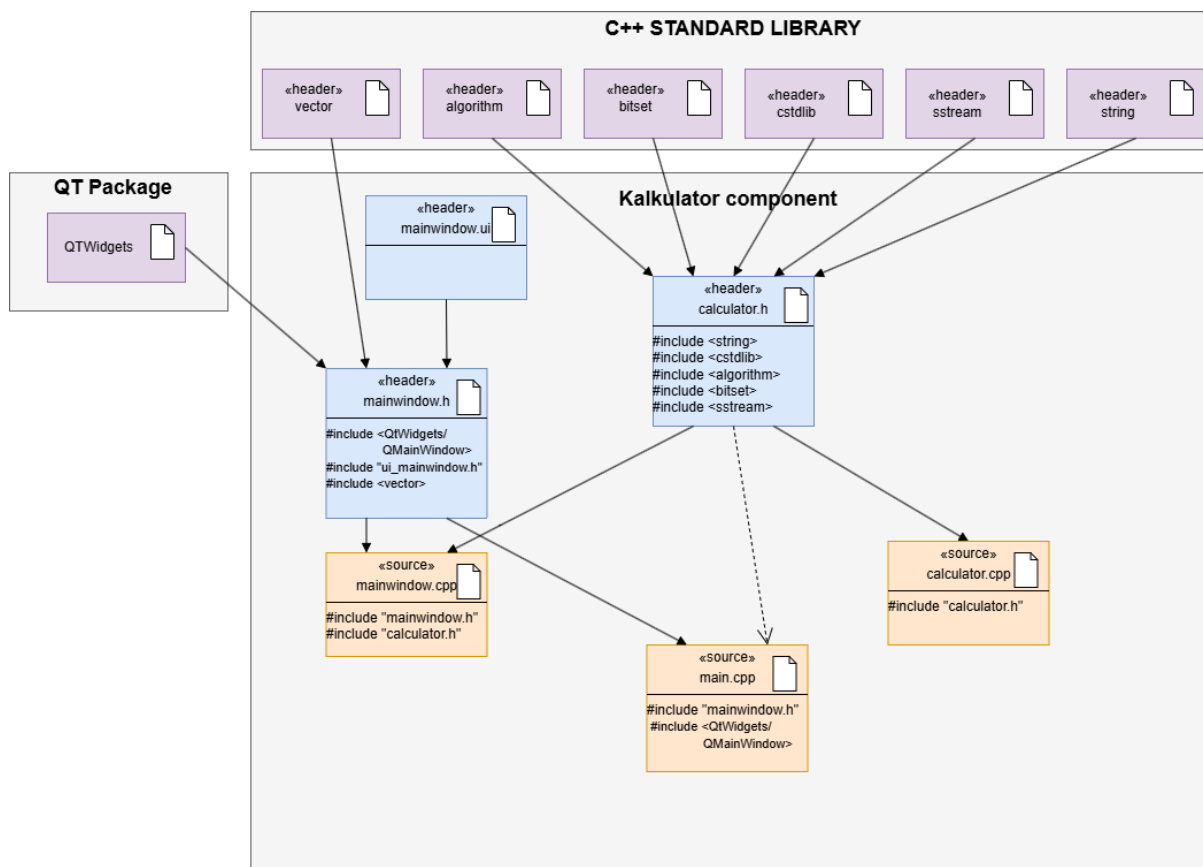
$$0 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$$

$$00010110_2 = 16_{16}$$

W celu wykonania powyższych konwersji za pomocą aplikacji kalkulatora, wprowadzane dane są typu **string**, a następnie odpowiednio konwertowane za pomocą funkcji **stod()** jeśli wprowadzane dane są w DEC, bądź **stoll()** jeśli wprowadzane dane są w innym systemie liczbowym. Dane wejściowe są następnie zapisywane do zmiennych typu **double**, na których wykonywane są wszystkie operacje przez kalkulator. Uzyskany wynik następnie jest ponownie konwertowany do **string** w odpowiednim systemie liczbowym, za pomocą funkcji **std::bitset**, **std::oct** i **std::hex**.

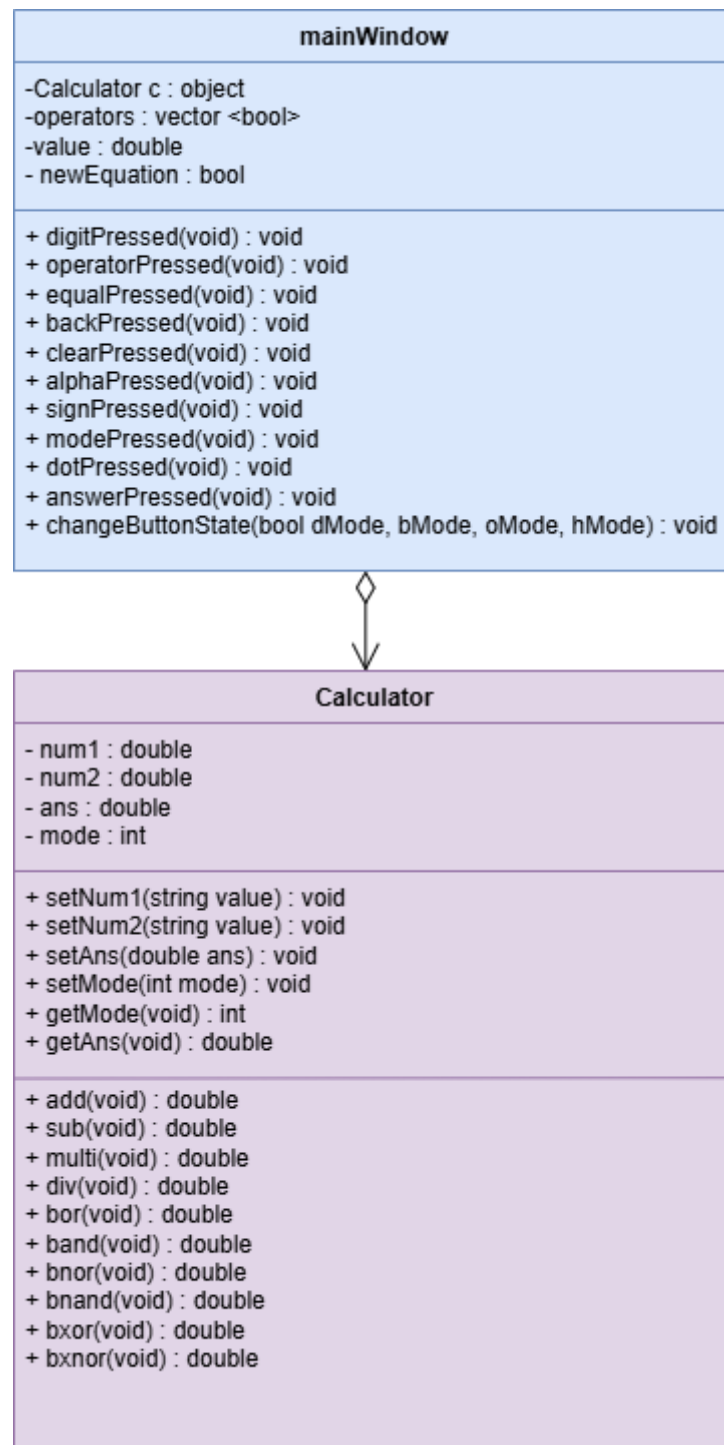
## 4. Projekt techniczny (*technical design*)

### 4.1 Diagramy



**Rysunek 4-1.** UML artifact diagram przedstawiający relacje i zależności między plikami w projekcie Kalkulator. Kolorem fioletowym oznaczono pliki z biblioteki standardowej c++ oraz pakiet QT odpowiedzialny za interfejs graficzny. Niebieskie pliki to pliki nagłówkowe zawierające deklaracje klas oraz funkcji. Pełne definicje wraz z komentarzami opisującymi kod znajdują się w plikach źródłowych oznaczonych kolorem pomarańczowym, do których również należy plik main. Linia ciągła oznacza, że plik jest załączony w innym pliku, do którego skierowany jest grot strzałki. Linia przerywana wskazując na pośrednie wykorzystywanie calculator.h przez main.cpp poprzez załączenie pliku mainWindow.h, do którego calculator.h został bezpośrednio załączony.





**Rysunek 4-2.** Diagram klas UML przedstawiający zależności pomiędzy klasami ich zmienne, metody oraz prawa dostępu (znak „-” private, znak „+” public). Klasa **mainWindow** ma w sobie klasę **Calculator** (relacja agregacji).

## **4.2 Role klas w projekcie**

---

Klasa Calculator odpowiedzialna jest za wykonywanie wszelkich operacji matematycznych, logicznych, przechowywanie ostatniego wyniku, konwersji pomiędzy systemami oraz przekształceniem ciągu binarnego do czytelnej dla użytkownika postaci (dzielenie ciągu binarnego po 4 znaki).

Klasa mainWindow odpowiada za wyświetlanie znaków, obsługę przycisków (w tym wywoływanie odpowiednich metod klasy Calculator) oraz całe okienko aplikacji.

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

Platforma testowa – komputer PC z windows 10 64 bit, procesor x64.

Oprogramowanie:

- Visual Studio 2022
- Pakiet QT
- CMake
- Kompilator MVSC2022 64 bit

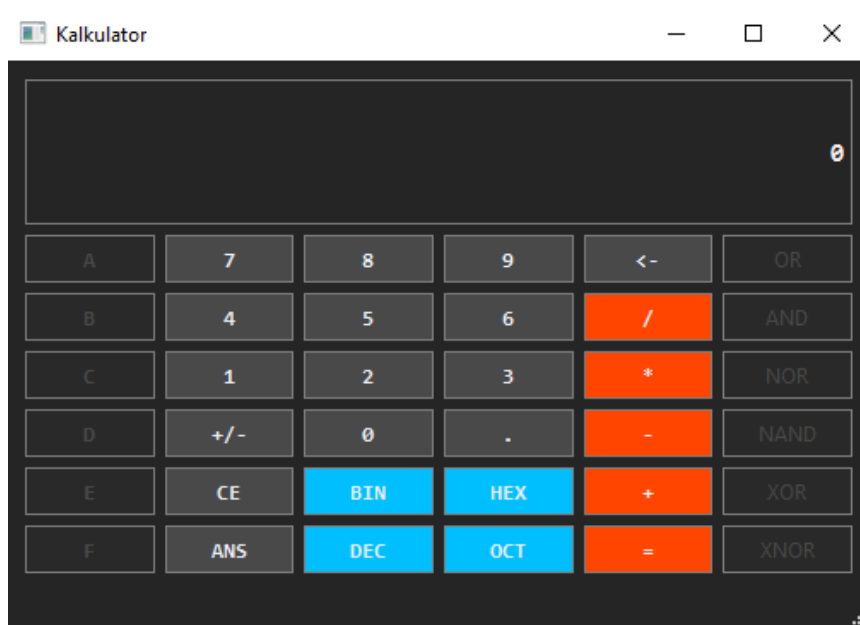
## 6. Opis wykonanych testów (*testing report*).

Kod usterki	Data	Autor	Opis	Stan
1	20.01.2025	Jakub Marchewczyk	Po wykonaniu operacji logicznej w HEX wynik zwracany jest w DEC	Naprawiona
2	21.01.2025	Jakub Marchewczyk	Przycisk ANS wyświetla zawsze odpowiedz w DEC	Naprawiona

Wykonane testy z wykorzystaniem GTest dla klasy Calculator – testy seterów, getterów, operacji dodawania, odejmowania, mnożenia, dzielenie dla danych wejściowych w różnych systemach liczbowych, konwersje pomiędzy wszystkimi z 4 zaimplementowanych systemów liczbowych.

Kod testów umieszczony w folderze tests w repozytorium projektu.

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



**Rysunek 7-1.** Główne okienko programu kalkulatora w trybie DEC (wyłączone operacje bitowe oraz znaki A-F).

Domyślnie kalkulator wyświetla 0 i czeka na wprowadzenie pierwszej liczby przez użytkownika za pomocą przycisków w okienku. Po wprowadzeniu pierwszej liczby użytkownik wybiera jaką operację ma zostać wykonana i wprowadza drugą liczbę w taki sam sposób jak pierwszą. Po wciśnięciu przycisku „=” na ekranie zostaje wyświetlony wynik.

Tryb DEC – wprowadzanie danych w DEC, możliwość wykonywania operacji na liczbach zmiennoprzecinkowych oraz ujemnych.

Tryb BIN – wprowadzanie danych w BIN możliwość wykonywania podstawowych działań matematycznych oraz operacji bitowych (włączone przyciski z ostatniej kolumny). Brak wsparcia dla liczb zmiennoprzecinkowych (cyfry po przecinku zostaną ucięte). W odróżnieniu do trybu DEC przyciski 2-9, ‘.’, +/- wyłączone.

Tryb OCT – w odróżnieniu do trybu DEC przyciski 8-9, ‘.’, +/- wyłączone, reszta funkcjonalności jak w trybie BIN.

Tryb HEX - w odróżnieniu do trybu DEC przyciski ‘.’, +/- wyłączone, przyciski A-F włączone, reszta funkcjonalności jak w trybie BIN.

Szczegółowy opis działania poszczególnych przycisków znajduje się w rozdziale 2 dokumentacji.

Bibliografia

- [1] Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
- [2] <https://stackoverflow.com/>
- [3] <https://en.cppreference.com/w/>