|  |  |  |  |
| --- | --- | --- | --- |
|  | REV | DATA | ZMIANY |
|  | 0.1 | 10.01.2025 | *Jakub Marchewczyk*  *marchewczyk@student.agh.edu.pl* |
|  | 0.2 | 19.01.2025 | *Jakub Marchewczyk*  *marchewczyk@student.agh.edu.pl* |
|  | 0.3 | 25.01.2025 | *Jakub Marchewczyk*  *marchewczyk@student.agh.edu.pl* |

**Kalkulator dla elektroników/informatyków**

Autor: Jakub Marchewczyk

Akademia Górniczo-Hutnicza

Kraków (C) 2025

Spis treści

[1. Wstęp 4](#_Toc188618780)

[2. Funkcjonalność (*functionality*) 5](#_Toc188618781)

[3. Analiza problemu (*problem analysis*) 6](#_Toc188618782)

[4. Projekt techniczny (*technical design*) 8](#_Toc188618783)

[4.1 Diagramy UML 8](#_Toc188618784)

[4.2 Role klas w projekcie 10](#_Toc188618785)

[5. Opis realizacji (*implementation report*) 11](#_Toc188618786)

[6. Opis wykonanych testów (*testing report*). 12](#_Toc188618787)

[7. Podręcznik użytkownika (*user's manual*) 13](#_Toc188618788)

Lista oznaczeń

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

# 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.

# 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 |

# 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 przemnażają cyfry liczby w BIN przez odpowiadające im potęgi cyfry 2 i sumując.

**Przykład 1.**

1∙24 + 0∙23 + 1∙22 + 1∙21 + 0∙20 = 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 przemnażają cyfry liczby w OCT przez odpowiadające im potęgi cyfry 8 i sumując.

**Przykład 2.**

2∙81 + 6∙80 = 22

Konwersję liczby z systemu dwójkowego do systemu ósemkowego dokonujemy dzieląc ciąg binarny w „trójki” z czego każda trójka reprezentuję 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∙22 + 1∙21 + 0∙20 = 2

1∙22 + 1∙21 + 0∙20 = 4

0101102 = 248

System szesnastkowy (heksadecymalny) – pozycyjny system o podstawie 16, zapis liczb przy użyciu cyfr: 0-9 oraz liter A-F, system ten zatem wykorzystuje 15 znaków wskazujących kolejne potęgi liczby 16 .

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

**Przykład 4.**

1∙161 + 6∙160 = 22

Konwersję liczby z systemu dwójkowego do systemu szesnastkowego dokonujemy dzieląc ciąg binarny w „czwórki” z czego każda czwórka reprezentuję 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∙24 + 0∙22 + 0∙21 + 1∙20 = 1

0∙24 + 1∙22 + 1∙21 + 0∙20 = 6

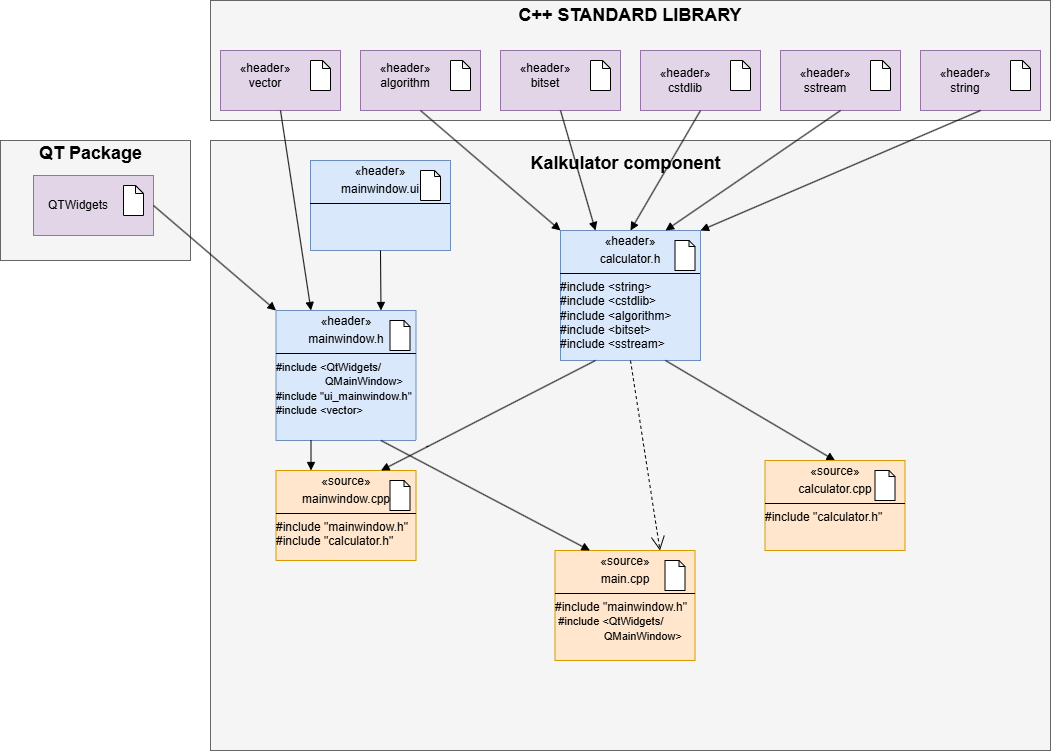
000101102 = 1616

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.**

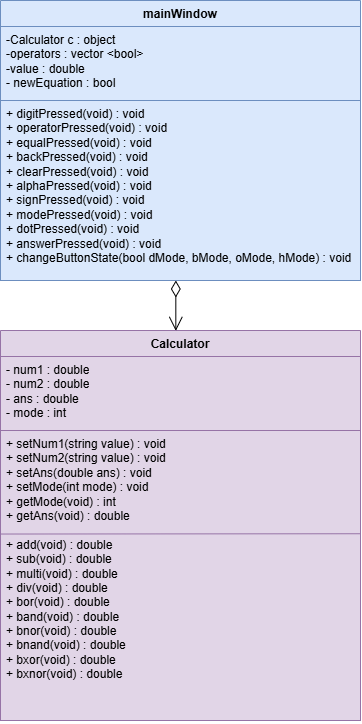
|  |
| --- |
|  |

# Projekt techniczny (*technical design*)

## Diagramy



**Rysunek 4-1.** UML artifact diagramprzestawiający 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 wskazuję 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).

## 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.

# Opis realizacji (*implementation report*)

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

Oprogramowanie:

- Visual Studio 2022

- Pakiet QT

- CMake

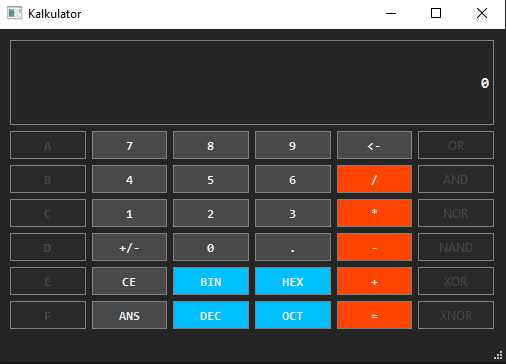
- Kompilator MVSC2022 64 bit

# 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.

# 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 jaka operacja 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 znajduję 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/>