

# Kalkulator dla Elektroników I Informatyków

Autor: Jakub Stelmach

Akademia Górniczo-Hutnicza

## Spis treści

Lista oznaczeń.....	3
1 Wstęp .....	4
2. Funkcjonalność.....	4
3. Analiza problemu.....	5
Przeliczanie Systemów Liczbowych: .....	5
Podstawowe Funkcje Matematyczne:.....	5
Intuicyjny Interfejs Użytkownika: .....	5
Zabezpieczenia i Komunikaty Błędów: .....	5
4 Projekt Techniczny.....	6
4.1 Zabezpieczenia .....	11
4.2 Problemy projektowe .....	11
5 Opis realizacji.....	12
6 Testy jednostkowe.....	13
7 Podręcznik użytkownika .....	14
8 Bibliografia.....	18

## Lista oznaczeń

1. HEX – system heksadecymalny
2. BIN – system dwójkowy
3. OCT – system ósemkowy
4. DEC – system dziesiętny
5. NaN – Not a Number – dana wartość nie jest liczbą

## 1 Wstęp

Niniejszy raport stanowi dokumentację procesu tworzenia prostego, lecz funkcjonalnego kalkulatora, mającego na celu ułatwienie codziennych obliczeń dla elektroników oraz informatyków. Projekt nie tylko stawia sobie za zadanie dostarczenie przyjaznego narzędzia, ale również zdobywanie wiedzy praktycznej z zakresu projektowania aplikacji w języku C++, wykorzystując przy tym kluczowe narzędzia takie jak CMake, Google Test (gtest) oraz środowisko Visual Studio.

Podstawowe założenia projektu:

- Przygotowanie dokumentacji projektowej
- Opracowanie projektu kalkulatora w środowisku MFC Visual Studio, wykorzystanie narzędzi cmake oraz gTest

## 2. Funkcjonalność

Kalkulator, stworzony z myślą o elektronikach, umożliwia szybkie przeliczanie pomiędzy systemami liczbowymi, takimi jak dziesiętny, ósemkowy, szesnastkowy i binarny. Posiada również podstawowe funkcje matematyczne, takie jak dodawanie, odejmowanie, mnożenie i dzielenie, co ułatwia codzienne obliczenia w pracy elektronika. Zoptymalizowany interfejs użytkownika pozwala na łatwe przełączanie między funkcjami, co przekłada się na efektywność korzystania z narzędzia.

Aby zabezpieczyć użytkownika i poprawić interakcję z kalkulatorem, dodano różne zabezpieczenia. Wprowadzanie danych z klawiatury ekranowej ułatwia uniknięcie błędów związanych z wprowadzaniem danych. Komunikaty błędów informują o niemożliwości dzielenia przez 0, a wynik takiej operacji oznaczony jest jako "NaN" (not a number). Dodatkowo, przy przekształcaniu danych między systemami liczbowymi konieczne jest dodanie odpowiednich przedrostków, takich jak "0x" dla systemu szesnastkowego, "0o" dla systemu ósemkowego oraz "0b" dla systemu dwójkowego.

Projekt, mimo swej prostoty, skupia się na dostarczeniu efektywnego narzędzia, które spełni potrzeby elektroników czy informatyków w codziennej pracy, jednocześnie umożliwiając zdobycie przydatnych umiejętności programistycznych.

### 3. Analiza problemu

#### Przeliczanie Systemów Liczbowych:

- Kalkulator umożliwia szybkie przeliczanie między systemami liczbowymi: dziesiętnym, ósemkowym, szesnastkowym i binarnym.
- Efektywne narzędzie dla elektroników operujących na różnych systemach liczbowych.

#### Podstawowe Funkcje Matematyczne:

- Oferuje podstawowe operacje matematyczne: dodawanie, odejmowanie, mnożenie i dzielenie.
- Wsparcie dla szybkich obliczeń, istotnych w projektowaniu układów elektronicznych.

#### Intuicyjny Interfejs Użytkownika:

- Zoptymalizowany interfejs ułatwia szybkie przełączanie między funkcjami.
- Zwiększa efektywność użytkowania, zwłaszcza w dynamicznym środowisku projektowania.

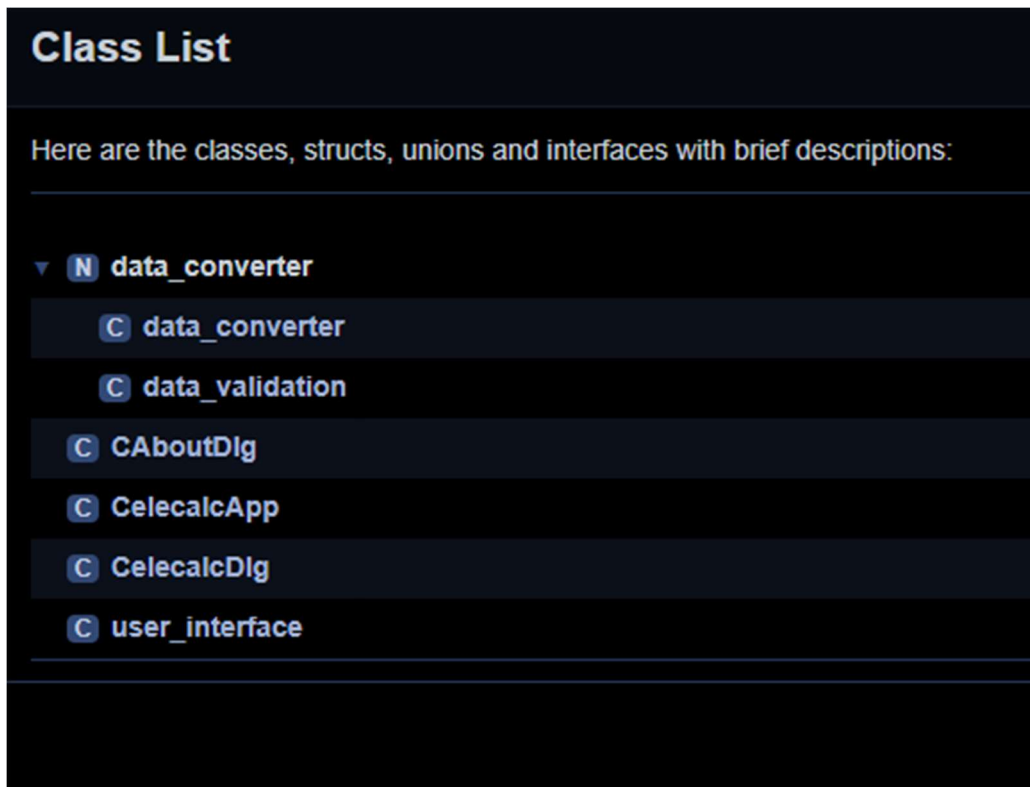
#### Zabezpieczenia i Komunikaty Błędów:

- Wprowadzanie danych z klawiatury ekranowej eliminuje błędy związane z wprowadzaniem danych.
- Komunikaty błędów, zwłaszcza przy dzieleniu przez zero (NaN), informują użytkownika o problemach.
- Przekształcanie danych z przedrostkami (0x, 0o, 0b) zabezpiecza przed niepoprawnymi wynikami przy zmianie systemów liczbowych.

## 4 Projekt Techniczny

*Dane zostały przygotowane za pomocą narzędzia Doxygen*

*Hierarchia klas*



**Class List**

Here are the classes, structs, unions and interfaces with brief descriptions:

- ▼ **N** data\_converter
  - C** data\_converter
  - C** data\_validation
- C** CAboutDlg
- C** CelecalcApp
- C** CelecalcDlg
- C** user\_interface

*Klasy stworzone na potrzeby kalkulatora*

data\_converter > data\_converter >

## data\_converter::data\_converter Class Reference

### Public Member Functions

void	<b>number_systems_converter</b>	(string data, string data_type)
string	<b>BIN_handler</b>	()
string	<b>DEC_handler</b>	()
string	<b>OCT_handler</b>	()
string	<b>HEX_handler</b>	()

### Private Attributes

string	<b>DEC</b>
string	<b>BIN</b>
string	<b>OCT</b>
string	<b>HEX</b>

The documentation for this class was generated from the following files:

- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/data\_converter.h
- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/data\_converter.cpp

## data\_converter::data\_validation Class Reference

### Public Member Functions

bool	<b>validate_data</b>	(string data)
------	----------------------	---------------

### Public Attributes

data_converter	<b>converter</b>
----------------	------------------

### Private Attributes

string	<b>saved_data</b>
const string	<b>valid_BIN</b>
const string	<b>valid_OCT</b>
const string	<b>valid_DEC</b>
const string	<b>valid_HEX</b>

### Friends

class	<b>converter</b>
-------	------------------

The documentation for this class was generated from the following files:

- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/data\_converter.h
- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/data\_converter.cpp

## user\_interface Class Reference

### Public Member Functions

void	<b>set_output</b> (string data)
string	<b>get_output</b> ()
string	<b>get_history</b> ()
void	<b>clear</b> ()
void	<b>clear_last</b> ()
void	<b>simple_calculator</b> (string data)

### Private Attributes

string	<b>history</b>
string	<b>output</b>
string	<b>expression</b>
string	<b>previous_sign</b>
string	<b>actual_number</b>
double	<b>memory</b>
bool	<b>is_dot_added</b>
bool	<b>is_NaN</b>
bool	<b>is_negative</b>

The documentation for this class was generated from the following files:

- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/ele\_calcDlg.h
- F:/Pliki/Studia/cyganek/projekt github/Calculator\_for\_electronics/ele\_calcDlg.cpp

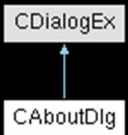


*Klasy automatycznie generowane poprzez środowisko Visual Studio*

Protected Member Functions | List of all members

## CAboutDlg Class Reference

Inheritance diagram for CAboutDlg:



```

classDiagram
    CDialogEx <|-- CAboutDlg
  
```

### Protected Member Functions

virtual void <b>DoDataExchange</b> (CDataExchange *pDX)
---

Public Member Functions | List of all members

## CelecalcApp Class Reference

Inheritance diagram for CelecalcApp:



```

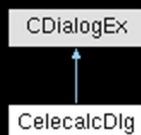
classDiagram
    CWinApp <|-- CelecalcApp
  
```

### Public Member Functions

virtual BOOL <b>InitInstance</b> ()
-------------------------------------

## CelecalcDlg Class Reference

Inheritance diagram for CelecalcDlg:



### Public Member Functions

	CelecalcDlg (CWnd *pParent=nullptr)
afx_msg void	<b>OnBnClickedButton1</b> ()
afx_msg void	<b>OnBnClickedButton2</b> ()
afx_msg void	<b>OnBnClickedButton3</b> ()
afx_msg void	<b>OnBnClickedButton4</b> ()
afx_msg void	<b>OnBnClickedButton5</b> ()
afx_msg void	<b>OnBnClickedButton6</b> ()
afx_msg void	<b>OnBnClickedButton7</b> ()
afx_msg void	<b>OnBnClickedButton8</b> ()
afx_msg void	<b>OnBnClickedButton9</b> ()
afx_msg void	<b>OnBnClickedButtonDot</b> ()
afx_msg void	<b>OnBnClickedButton0</b> ()
afx_msg void	<b>OnBnClickedButtonEquals</b> ()
afx_msg void	<b>OnBnClickedButtonBin</b> ()
afx_msg void	<b>OnBnClickedButtonOct</b> ()
afx_msg void	<b>OnBnClickedButtonDec</b> ()
afx_msg void	<b>OnBnClickedButtonHex</b> ()
afx_msg void	<b>OnBnClickedButtonMultiply</b> ()
afx_msg void	<b>OnBnClickedButtonDivide</b> ()
afx_msg void	<b>OnBnClickedButtonAdd</b> ()
afx_msg void	<b>OnBnClickedButtonSubtract</b> ()
afx_msg void	<b>OnEnChangeInputExpressions</b> ()
afx_msg void	<b>OnEnChangeInputHistory</b> ()
afx_msg void	<b>OnEnChangeOutput</b> ()
afx_msg void	<b>OnBnClickedButtonClear</b> ()
afx_msg void	<b>OnBnClickedButtonClearLast</b> ()
void	<b>on_init_settings</b> ()

### Public Attributes

CFont	<b>button_font</b>
CFont	<b>text_font</b>
CFont	<b>system_exchange_font</b>
CButton	<b>button_2</b>
CButton	<b>button_1</b>
CButton	<b>button_3</b>
CButton	<b>button_4</b>
CButton	<b>button_5</b>
CButton	<b>button_6</b>
CButton	<b>button_7</b>
CButton	<b>button_8</b>
CButton	<b>button_9</b>
CButton	<b>button_dot</b>
CButton	<b>button_0</b>
CButton	<b>button_equals</b>
CButton	<b>button_bin</b>
CButton	<b>button_oct</b>
CButton	<b>button_dec</b>
CButton	<b>button_hex</b>
CButton	<b>button_multiply</b>
CButton	<b>button_divide</b>
CButton	<b>button_add</b>
CButton	<b>button_subtract</b>
CEdit	<b>c_expressions</b>
CEdit	<b>c_history</b>
CEdit	<b>c_output</b>
CString	<b>c_output_text</b>
CString	<b>c_history_text</b>
CString	<b>c_expressions_text</b>
data_converter::data_validation	<b>converter</b>
user_interface	<b>user_interface</b>
CButton	<b>button_eval_exp</b>
CButton	<b>button_clear</b>
CButton	<b>button_clear_last</b>

### Protected Member Functions

virtual void	<b>DoDataExchange</b> (CDataExchange *pDX)
virtual BOOL	<b>OnInitDialog</b> ()
afx_msg void	<b>OnSysCommand</b> (UINT nID, LPARAM lParam)
afx_msg void	<b>OnPaint</b> ()
afx_msg HCURSOR	<b>OnQueryDragIcon</b> ()

### Protected Attributes

HICON	<b>m_hIcon</b>
-------	----------------

## 4.1 Zabezpieczenia

Aby skutecznie zabezpieczyć aplikację przed potencjalnymi problemami, w wielu kluczowych miejscach zastosowano dyrektywy języka, takie jak try-catch. Dzięki temu mechanizmowi obsługi błędów możliwe jest łatwe i kontrolowane reagowanie na sytuacje wyjątkowe, zapobiegając awariom aplikacji.

Mechanizm try-catch został szczególnie wykorzystany przy operacjach matematycznych oraz konwersji między systemami liczbowymi, co umożliwia unikanie krytycznych błędów i zapewnia stabilność działania programu.

Większość potencjalnych błędów jest niwelowana poprzez analizę danych wprowadzanych przez użytkownika i ograniczenie ilości możliwych operacji, co dodatkowo zwiększa bezpieczeństwo korzystania z kalkulatora.

Wiele zabezpieczeń zostało wprowadzonych na etapie debugowania aplikacji, eliminując potencjalne problemy już na wstępnym etapie rozwoju

## 4.2 Problemy projektowe

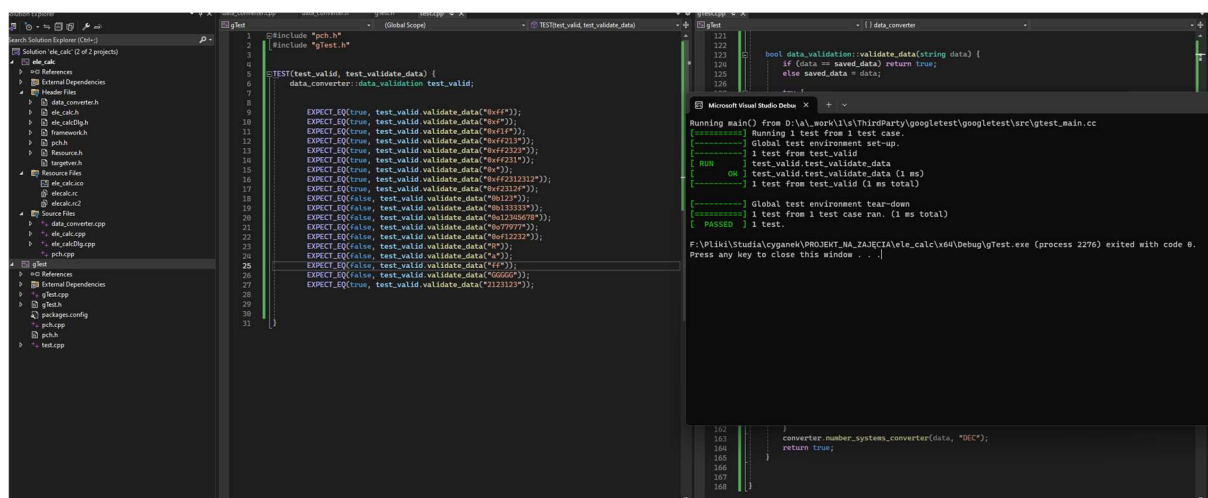
Podstawowym problemem był sposób przechowywania danych, w opisywanym tutaj kalkulatorze dane przechowywane są w postaci ciągów znaków co ułatwiło ich konwersję oraz walidację na etapie projektowania – dane przechowywane w postaci ciągu znaków mają też jedną dużą zaletę, przy dobrej implementacji nie występuje w nich problem ograniczonej dokładności jak w przypadku innych typów – a przy konwersji na systemy możliwe jest łatwe dodanie przedrostka oznaczającego zadany typ liczbowy

Jednym z dodatkowych problemów była optymalizacja środowiska - bardzo możliwe że środowisko na którym była projektowana aplikacja było po prostu uszkodzone – ale powodowało to wielorakie problemy poczynając od ciągle znikających plików z bibliotek, zanik skrótów klawiszowych, tworzenie procesów zombie blokujących dostęp do plików – z tego względu bardzo problematyczna była implementacja testów gcode, więc większość testów przeprowadzałem ręcznie w trakcie pracy nad programem.

## 5 Opis realizacji

1. **Środowisko programistyczne:** Wykorzystano Visual Studio 2022 do tworzenia, debugowania i optymalizacji kodu.
2. **System budowania:** Do konfiguracji projektu i zarządzania procesem budowy użyto CMake w wersji 3.28, co zapewniło niezależność od platformy i kompilatora.
3. **Testowanie jednostkowe:** Implementacja testów jednostkowych została zrealizowana przy użyciu frameworka Google Test, pomimo pewnych trudności.
4. **Dokumentacja kodu:** Doxygen posłużył do automatycznego generowania czytelnej dokumentacji kodu źródłowego, w tym diagramów klas.
5. **Środowisko aplikacji okienkowych:** Projekt oparto na środowisku MFC, umożliwiając szybkie tworzenie aplikacji okienkowych w systemie Windows.
6. **Podsumowanie:** Zastosowanie tych narzędzi umożliwiło efektywną realizację projektu, zapewniając jednocześnie czytelność kodu, poprawność implementacji i klarowną dokumentację, w trakcie projektowania pliki były zachowane na maszynie lokalnej.

## 6 Testy jednostkowe



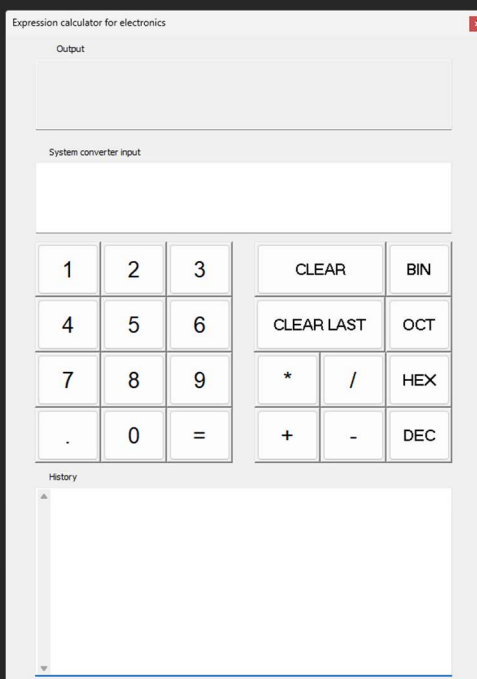
Testy przeprowadzone w zewnętrznym projekcie ze względu na problemy z implementacją gtest w MFC, funkcje natomiast są te same skopiowane do projektu gTest i przetestowane pod względem poprawności walidacji danych wejściowych do przetwarzania na inne systemy liczbowe

## 7 Podręcznik użytkownika

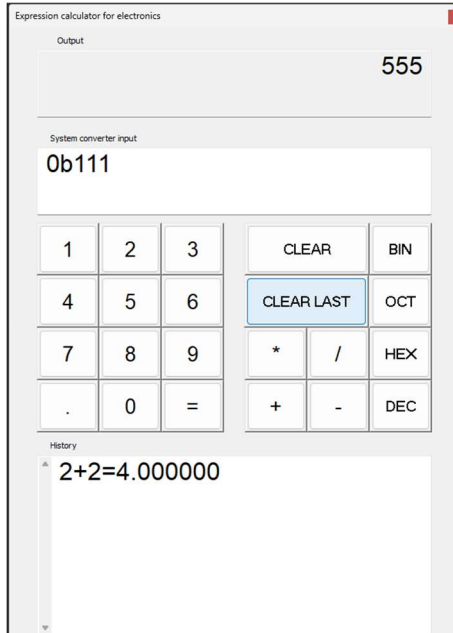
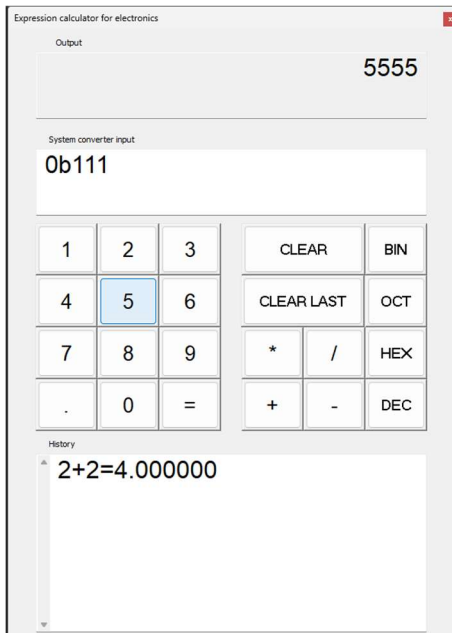
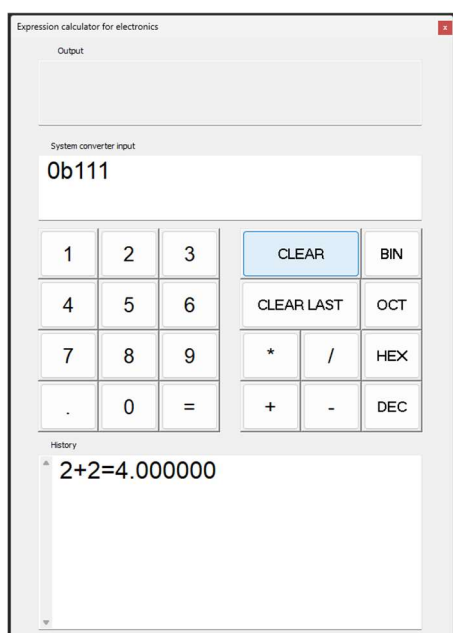
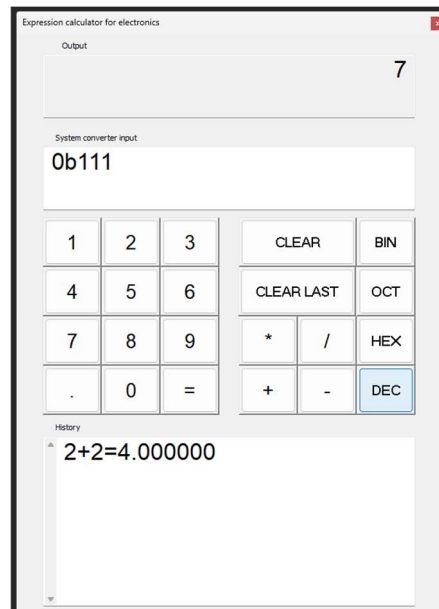
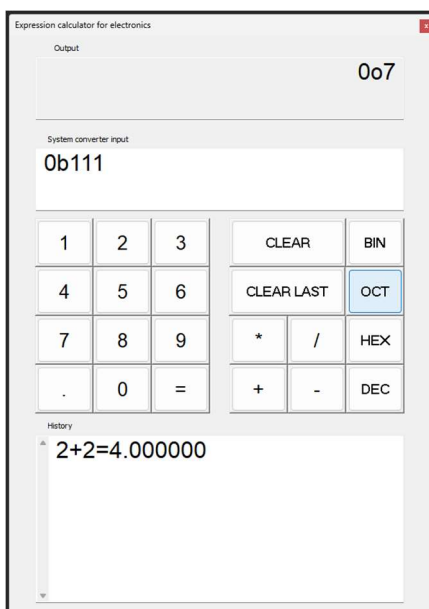
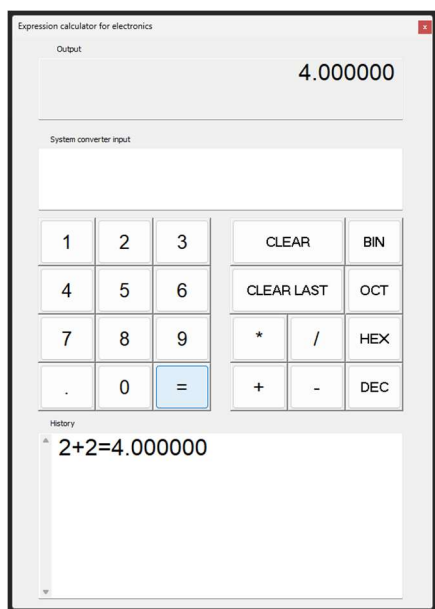
Panel kalkulatora składa się z:

1. Klawiatury numerycznej
2. Przycisków odpowiadających za usuwanie całego ekranu, oraz ostatniego znaku
3. Przycisków odpowiadających za konwersje na inne systemy liczbowe danych wprowadzonych w polu „system converter input”
4. Pola „output” na którym wyświetlane są wyniki i bieżące operacje
5. Pola „history” zawierającego uprzednio wprowadzane wyrażenia
6. Przycisku „zamknij” w prawym górnym rogu zamykającego aplikację

Przy obliczeniach matematycznych wykorzystywana jest klawiatura i przyciski, natomiast przy przekształcaniu między systemami wykorzystywane jest pole „system converter input”



Przykłady użycia kalkulatora, obliczanie sumy, przekształcenia liczby w systemie ósemkowym na liczbę w systemie ósemkowym oraz dziesiętnym, czyszczenie ekranu, oraz usuwanie ostatniej cyfry.



Kalkulator oprócz historii posiada również pamięć, więc poprzednio wyliczona wartość może zostać uzyskana w następnych obliczeniach

Expression calculator for electronics

Output: 372.000000

System converter input: 0b111

1	2	3	CLEAR	BIN
4	5	6	CLEAR LAST	OCT
7	8	9	*	/
.	0	=	+	-
			HEX	DEC

History:

- 2+2=4.000000
- 555\*2/3+2=372.000000

Expression calculator for electronics

Output: 372.000000+5

System converter input: 0b111

1	2	3	CLEAR	BIN
4	5	6	CLEAR LAST	OCT
7	8	9	*	/
.	0	=	+	-
			HEX	DEC

History:

- 2+2=4.000000
- 555\*2/3+2=372.000000

Expression calculator for electronics

Output: 377.000000

System converter input: 0b111

1	2	3	CLEAR	BIN
4	5	6	CLEAR LAST	OCT
7	8	9	*	/
.	0	=	+	-
			HEX	DEC

History:

- 2+2=4.000000
- 555\*2/3+2=372.000000
- 372.000000+5=377.000000

Przy próbie dzielenia przez zero z ekranu są usuwane dane, natomiast w historii występuje pozycja NaN – not a numer

Expression calculator for electronics

Output:

System converter input: 0b111

1	2	3	CLEAR	BIN
4	5	6	CLEAR LAST	OCT
7	8	9	*	/
.	0	=	+	-
			HEX	DEC

History:

- 2+2=4.000000
- 555\*2/3+2=372.000000
- 372.000000+5=377.000000
- NaN

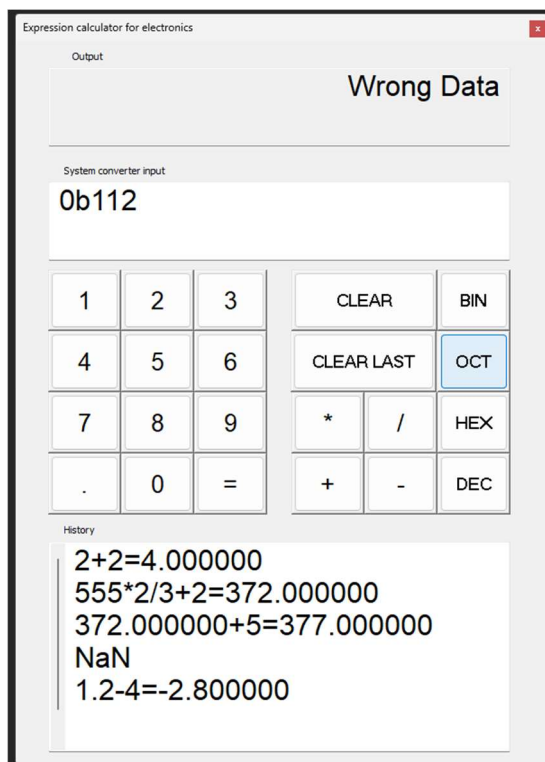
Umożliwia on również obliczenia na liczbach zmiennoprzecinkowych:

History:

- 2+2=4.000000
- 555\*2/3+2=372.000000
- 372.000000+5=377.000000
- NaN
- 1.2-4=-2.800000



Przy próbie przekształcenia liczby która nie jest z podanego systemu liczbowego na ekranie pojawia się napis „Wrong Data”



## 8 Bibliografia

1. Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
2. Dokumentacja MFC - <https://learn.microsoft.com/pl-pl/cpp/mfc/reference/mfc-classes?view=msvc-170>
3. Cpp0x - <https://cpp0x.pl/>
4. Stackoverflow - <https://stackoverflow.com/>
5. ChatGPT - <https://chat.openai.com/>