

KALKULATOR DLA ELEKTRONIKÓW/INFORMATYKÓW

Autor: Gabriel Kraj

Akademia Górniczo-Hutnicza

Kraków © 2024

Spis treści

1. WSTĘP.....	3
2. FUNKCJONALNOŚĆ.....	4
3. ANALIZA PROBLEMU.....	5
4. PROJEKT TECHNICZNY.....	6
5. OPIS REALIZACJI.....	9
6. WYKONANE TESTY.....	10
7. PODRĘCZNIK UŻYTKOWNIKA.....	11
8. BIBLIOGRAFIA	13

1.Wstęp

Założeniem projektu jest stworzenie narzędzia, które znacznie ułatwi i przyspieszy pracę elektronikom i informatykom poprzez dostarczenie im intuicyjnego kalkulatora dla ich specyficznych potrzeb. Często używają oni liczb w różnych systemach, a ręczne ich przeliczanie powoduje nie tylko stratę czasu ale również często prowadzi do błędów. Projekt kalkulatora będzie zawierał dowolną możliwość przeliczania liczb z systemów: binarny, ósemkowy, dziesiętny, szesnastkowy od razu na wszystkie pozostałe systemy, a w razie wpisania nieodpowiedniej liczby dla danego systemu poinformuje użytkownika o jego błędzie.

Raport obejmuje funkcjonalność narzędzia, analizę problemu, projekt techniczny, opis realizacji, wykonane testy oraz podręcznik użytkownika.

2. Funkcjonalność

Projekt kalkulatora został wyposażony w czytelny interfejs użytkownika, składający się z rozwijanego pola wyboru systemu w jakim użytkownik poda daną wejściową do przeliczenia, pola do wpisania tej danej, oraz czterech odpowiednio podpisanych pól, zablokowanych dla użytkownika (nie można tam nic wpisać ręcznie), gdzie po wybraniu odpowiedniego systemu, wpisaniu poprawnej liczby oraz naciśnięciu przycisku „calculate”, pojawią się wyniki konwersji liczb. Kalkulator został zaprojektowany z myślą o przeliczaniu liczb całkowitych. Wyposażony jest również w przycisk „Reset” który usuwa znaki z wszystkich pól, aby ułatwić użytkownikowi przygotowanie przed przeliczeniem kolejnej liczby przez kalkulator, jednak jego użycie nie jest konieczne.

W kalkulatorze zostały również zawarte zasady przy wprowadzaniu liczb do przeliczenia dla odpowiednich systemów:

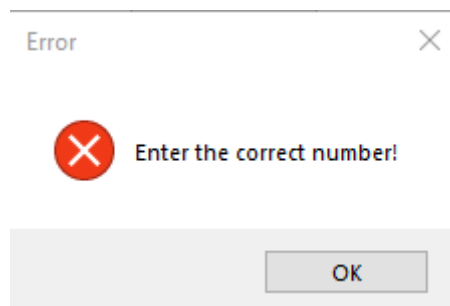
System konwersji:	Przyjmowane znaki
binarny	0 - 1
ósemkowy	0 - 7
dziesiętny	0 - 9
szesnastkowy	0-9 i A-F (duże oraz małe litery)

Takie zasady znacznie zmniejszają ryzyko pomyłki użytkownika przy wyborze systemów, a ich ewentualne nieprzestrzeganie powoduje pojawienie się odpowiednich komunikatów:

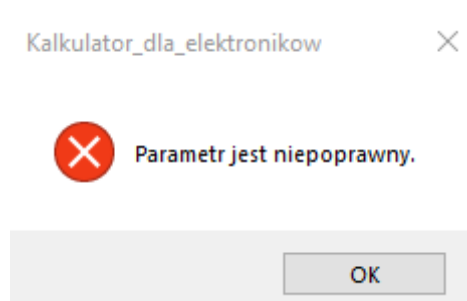
Złamana zasada:	Komunikat
Brak wyboru systemu	„Parametr jest niepoprawny”
Podanie niepoprawnego znaku	„Enter the correct number”

3. Analiza problemu

- a) Zrozumienie różnic pomiędzy systemami a następnie **zastosowanie odpowiednich algorytmów i funkcji**, dzięki którym nawet przy dużych liczbach konwersja jest dokładna.
- b) **Sprawdzenie danych wejściowych** w zależności od wybranego systemu w celu uniknięcia błędów podczas konwersji.
- c) **Odpowiedni interfejs graficzny** powinien być czytelny, przejrzysty oraz intuicyjny aby nawet użytkownik, który pierwszy raz z niego korzysta nie miał problemów z jego obsługą.



1. Komunikat po wprowadzeniu błędnej danej wejściowej



2. Komunikat przy braku wybrania systemu danej wejściowej

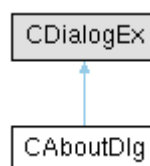
4. Projekt techniczny

Aby przedstawić następującą część projektu skorzystano z programu doxygen:

Class Hierarchy:



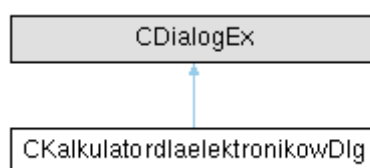
CAboutDlg:



Protected Member Functions

virtual void **DoDataExchange** (CDataExchange *pDX)

CKalkulatorDlaElektronikowDlg:



Public Member Functions

	CKalkulatorDlaElektronikowDlg (CWnd *pParent=nullptr)
afx_msg void	OnBnClickedButton1 ()
afx_msg void	OnBnClickedButton2 ()

Public Attributes

CComboBox	conversion_type
CString	value_dec
CString	value_bin
CString	value_oct
CString	value_hex
CString	initial_value
CString	float_value

Protected Member Functions

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

Protected Attributes

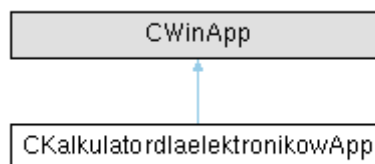
HICON	m_hicon
-------	----------------

Conversions Class Reference

Public Member Functions

std::string	binary2decimal	(const std::string &bin)
std::string	oct2decimal	(const std::string &oct)
std::string	hex2decimal	(const std::string &hex)
std::string	decimal2binary	(const std::string &dec)
std::string	decimal2oct	(const std::string &dec)
std::string	decimal2hex	(const std::string &dec)

CKalkulatorDlaElektronikowApp:



Hierarchia klas oraz ich funkcje zostały pokazane przejrzysto dzięki zastosowaniu programu doxygen.

5.Opis realizacji

Projekt zrealizowano korzystając z:

- **Microsoft Visual Studio 2022:** Zostało głównym środowiskiem programistycznym podczas tworzenia aplikacji ze względu na swoją przejrzystość oraz szeroką funkcjonalność.
- **Kompilator MSVC 2022:** Wchodzi w skład pakietu Visual Studio 2022, jest z nim zintegrowany co znacząco ułatwiło proces kompilacji i debugowania kodu źródłowego napisanego w języku C++.
- **CMake:** Zastosowano CMake w wersji 3.28.1 aby zbudować projekt. CMake jest narzędziem tworzącym skrypty konfiguracyjne umożliwiające niezależność od platformy projektów programistycznych.
- **Google test:** Aby sprawdzić poprawność zastosowanych w kodzie funkcji przeprowadzono testy przy użyciu Google test, sprawdzając każdą funkcję dla różnych przypadków przeliczania. Pozwoliło to na szybkie wykrycie i poprawę błędów.
- **Doxygen:** Posłużył do wygenerowania przejrzystego diagramu hierarchii klas.

6. Wykonane testy

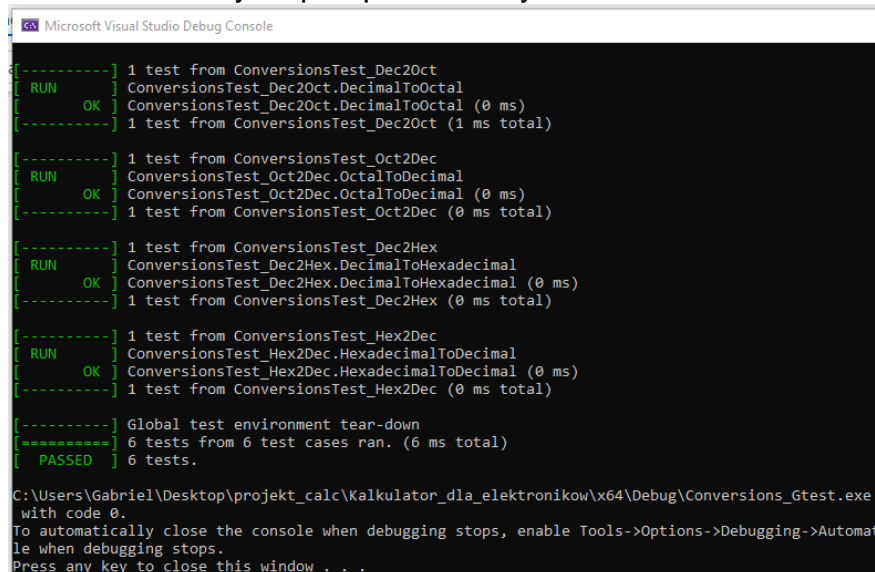
Przeprowadzono testy przy użyciu Google test, sprawdzając każdą funkcję dla różnych przypadków konwersji. Aby jak najlepiej sprawdzić działanie tych funkcji, przetestowano je kilka razy podając skrajnie różne dane wejściowe, co umożliwiło wykrycie błędu, który powodował że po podaniu do kalkulatora liczby 0, kalkulator nie wyświetlał wyników przeliczenia tej liczby na inne systemy. Błąd naprawiono.

Przykładowa lista przeprowadzonych testów (Reszta funkcji została przetestowana analogicznie:)

```
TEST(ConversionsTest_Dec2Hex, DecimalToHexadecimal) {
    conver dec_hex;
    EXPECT_EQ(dec_hex.decimal2hex("000"), "0");
    EXPECT_EQ(dec_hex.decimal2hex("123"), "7B");
    EXPECT_EQ(dec_hex.decimal2hex("255"), "FF");
    EXPECT_EQ(dec_hex.decimal2hex("1565"), "61D");
}

TEST(ConversionsTest_Hex2Dec, HexadecimalToDecimal) {
    conver hex_dec;
    EXPECT_EQ(hex_dec.hex2decimal("000"), "0");
    EXPECT_EQ(hex_dec.hex2decimal("7B"), "123");
    EXPECT_EQ(hex_dec.hex2decimal("FF"), "255");
    EXPECT_EQ(hex_dec.hex2decimal("61D"), "1565");
}
```

Wynik przeprowadzonych testów:



```
Microsoft Visual Studio Debug Console

[ RUN      ] 1 test from ConversionsTest_Dec2Oct
[ OK       ] ConversionsTest_Dec2Oct.DecimalToOctal
[ OK       ] ConversionsTest_Dec2Oct.DecimalToOctal (0 ms)
[ PASSED   ] 1 test from ConversionsTest_Dec2Oct (1 ms total)

[ RUN      ] 1 test from ConversionsTest_Oct2Dec
[ OK       ] ConversionsTest_Oct2Dec.OctalToDecimal
[ OK       ] ConversionsTest_Oct2Dec.OctalToDecimal (0 ms)
[ PASSED   ] 1 test from ConversionsTest_Oct2Dec (0 ms total)

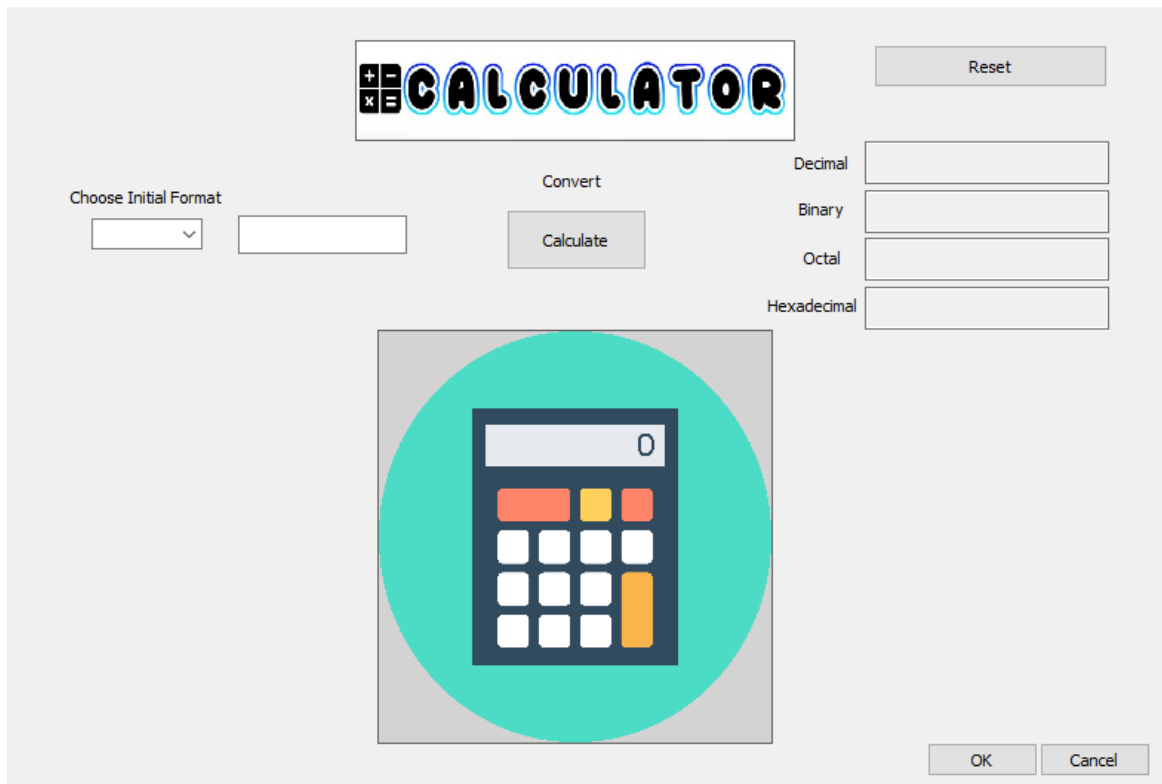
[ RUN      ] 1 test from ConversionsTest_Dec2Hex
[ OK       ] ConversionsTest_Dec2Hex.DecimalToHexadecimal
[ OK       ] ConversionsTest_Dec2Hex.DecimalToHexadecimal (0 ms)
[ PASSED   ] 1 test from ConversionsTest_Dec2Hex (0 ms total)

[ RUN      ] 1 test from ConversionsTest_Hex2Dec
[ OK       ] ConversionsTest_Hex2Dec.HexadecimalToDecimal
[ OK       ] ConversionsTest_Hex2Dec.HexadecimalToDecimal (0 ms)
[ PASSED   ] 1 test from ConversionsTest_Hex2Dec (0 ms total)

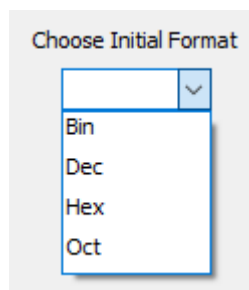
[ RUN      ] Global test environment tear-down
[ PASSED   ] 6 tests from 6 test cases ran. (6 ms total)
[ PASSED   ] 6 tests.

C:\Users\Gabriel\Desktop\projekt_calc\Kalkulator_dla_elektronikow\x64\Debug\Conversions_Gtest.exe
with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically
close when debugging stops.
Press any key to close this window . . .
```

7. Podręcznik użytkownika

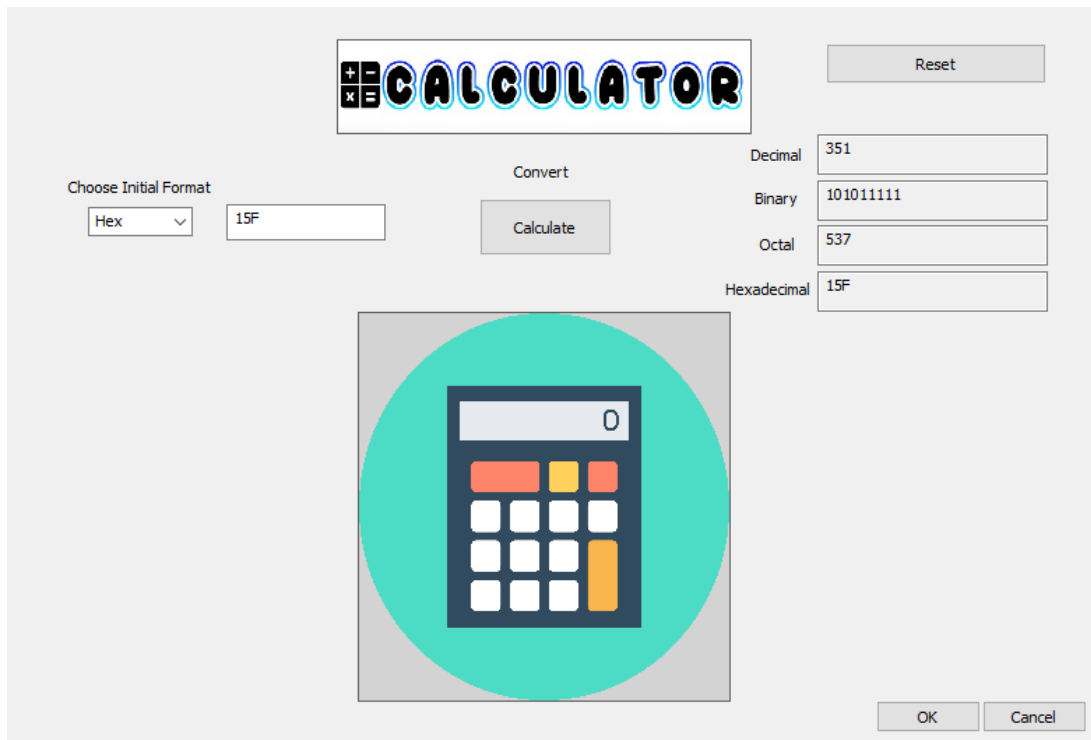


Aby skorzystać z kalkulatora po otwarciu interfejsu należy najpierw wybrać odpowiedni system konwersji z rozwijanego paska Choose Initial Format:



Następnie pozostało tylko wpisać w puste pole odpowiednią liczbę całkowitą którą chcemy przeliczyć oraz nacisnąć przycisk Calculate.

W odpowiednio podpisanych polach po prawej stronie wyświetlają się wyniki konwersji, a po naciśnięciu przycisku Reset możemy wyczyścić pola wynikowe. Aby rozpocząć następną konwersję wcale nie trzeba używać przycisku Reset wystarczy zmienić wpisaną przez nas liczbę lub system przeliczania a następnie po raz kolejny użyć przycisku Calculate. Kalkulator wyświetli wyniki konwersji.



The screenshot displays a web-based calculator application. At the top center is a logo with a calculator icon and the word "CALCULATOR" in a stylized, colorful font. Below the logo, on the left, is a "Choose Initial Format" section with a dropdown menu set to "Hex" and a text input field containing "15F". In the center, there is a "Convert" button and a "Calculate" button. On the right, there are four input fields for conversion results: "Decimal" (351), "Binary" (101011111), "Octal" (537), and "Hexadecimal" (15F). A "Reset" button is located above these fields. At the bottom center, there is a large circular icon of a calculator. In the bottom right corner, there are "OK" and "Cancel" buttons.

Przykład poprawnego użycia kalkulatora

8. Bibliografia'

1. Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
2. Strona Geeksforgeeks:
<https://www.geeksforgeeks.org/>
3. Strona Programiz: <https://www.programiz.com/>
4. Strona Cppreference: <https://en.cppreference.com/w/>
5. ChatGPT: <https://chat.openai.com/>
6. Strona Microsoft Tworzenia aplikacji MFC:
<https://learn.microsoft.com/pl-pl/cpp/mfc/reference/creating-an-mfc-application?view=msvc-170>