

Kalkulator dla elektroników/ informatyków

Autor: Jakub Wojtycza
Akademia Górniczo-Hutnicza

Spis treści

1.	WSTĘP.....	3
2.	FUNKcjONALNOŚĆ (<i>FUNCTIONALITY</i>).....	4
3.	ANALIZA PROBLEMU (<i>PROBLEM ANALYSIS</i>).....	5
4.	PROJEKT TECHNICZNY (<i>TECHNICAL DESIGN</i>).....	6
5.	OPIS REALIZACJI (<i>IMPLEMENTATION REPORT</i>)	9
6.	OPIS WYKONANYCH TESTÓW (<i>TESTING REPORT</i>) - LISTA BUGGÓW, UZUPEŁNIEŃ, ITD.....	10
7.	PODRĘCZNIK UŻYTKOWNIKA (<i>USER'S MANUAL</i>).....	11
8.	BIBLIOGRAFIA.....	12

1. Wstęp

Celem projektu jest przygotowanie narzędzia, które będzie użyteczne dla elektroników/informatyków. Kalkulator ma przyspieszyć konwertowanie liczb z określonych systemów na inne. W programie skupiam się na czterech głównych systemach liczbowych, które są powszechnie używane w środowisku elektronicznym i informatycznym: binarny, ósemkowy, dziesiętny oraz szesnastkowy. Projekt ma na celu eliminację potencjalnych błędów wynikających z ręcznego przeliczania liczb między systemami.

Poniżej znajduje się tabela z odpowiednimi znakami w każdym systemie liczbowym:

System liczbowy	Poprawne znaki
Binarny (Bin)	0 oraz 1
Ósemkowy (Oct)	0-7
Dziesiętny (Dec)	0-9
Szesnastkowy (Hex)	0-9 oraz „A” – „F”

Tabela nr 1 Przedstawia poprawne znaki w odpowiednich systemach liczbowych

2. Funkcjonalność

Kalkulator został zaprojektowany z myślą o intuicyjnym interfejsie graficznym, umożliwiającym szybką i łatwą konwersję liczb pomiędzy różnymi systemami liczbowymi. Po lewej stronie interfejsu znajduje się jedno pole, które pozwala użytkownikowi wprowadzić liczbę w jednym z czterech systemów liczbowych: binarnym, ósemkowym, dziesiętnym lub szesnastkowym. Po wprowadzeniu liczby, użytkownik może wybrać docelowy system liczbowy poprzez intuicyjne menu rozwijane. Kalkulator natychmiastowo przelicza wprowadzoną wartość na liczbę odpowiadającą wybranemu systemowi, prezentując wynik w tym samym interfejsie. Kalkulator operuje na liczbach całkowitych.

Należy przestrzegać odpowiednich wytycznych każdego z systemów liczbowych, inaczej program wyświetli nam błąd. Nie należy również przekształcać z tego samego systemu na ten sam (np. bin – bin). Poniżej tabela z możliwymi błędami:

Przykład złego użycia	Błędy (okienko pop-out)
Konwersja na ten sam system (Bin-Bin)	„To nie ma sensu”
Dziesiętkowy (Dec) np. „10a”	„Wprowadź poprawną liczbę dziesiętną”
Binarny (Bin) np. 0123”	„Wprowadź poprawną liczbę binarną”
Ósemkowy (Oct) np. „98”	„Wprowadź poprawną liczbę Ósemkową”
Szesnastkowy (Hex) np. „16h”	„Wprowadź poprawną liczbę Szesnastkową. Proszę używaj dużych liter”
Zostawione buty pola przy wyborze systemu	„Parametr jest niepoprawny”

Tabela nr 2 Przedstawia możliwe błędy przy wprowadzaniu danych

3. Analiza problemu

- a) **Obsługa Różnych Systemów Liczbowych:** Konwersja liczb między systemami szesnastkowym, binarnym, dziesiętnym i ósemkowym wymagała zrozumienia różnic między nimi oraz opracowania algorytmów umożliwiających dokładne przeliczenia.
- b) **Intuicyjność Interfejsu Graficznego:** Wprowadzenie jednego pola do wprowadzania liczby i menu rozwijanego do wyboru systemu liczbowego miało zapewnić prostotę obsługi. Konieczne było staranne dostosowanie interfejsu, aby zminimalizować ryzyko błędów użytkownika i zapewnić czytelność wyników konwersji.
- c) **Zabezpieczenie Przed Błędami:** Ważnym aspektem było uwzględnienie zabezpieczeń przed niepoprawnymi danymi wejściowymi, takimi jak wprowadzenie nieprawidłowego formatu liczby.

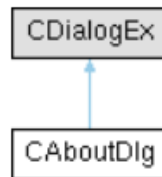
4. Projekt techniczny

Hierarchia klas:



Specyfikacja każdej z klas:

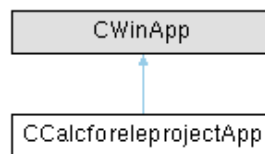
CAboutDlg:



Protected Member Functions

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

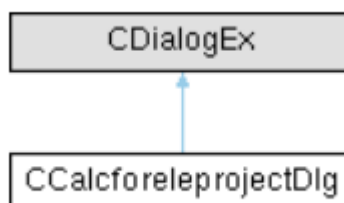
CCalcforeleprojectApp:



Public Member Functions

```
virtual BOOL InitInstance ()
```

CCalcforeleprojectDlg:



Public Member Functions

	CCalcforeleprojectDlg (CWnd *pParent=nullptr)
afx_msg void	OnEnChangeEdit2 ()
afx_msg void	OnBnClickedButton1 ()

Public Attributes

CComboBox	system_conversion_to
CComboBox	system_conversion_from
CString	cstring_value_from
CString	result

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 ()

Conversions:

Public Member Functions

string	DecimalToBinary	(string &dec_bin)
string	DecimalToOctal	(string &dec_oct)
string	DecimalToHexadecimal	(string &dec_num)
string	BinaryToDecimal	(string &bin_dec)
string	OctalToDecimal	(string &oct_dec)
string	HexadecimalToDecimal	(const string &hex_dec)

Doxygen w sposób przejrzysty przedstawił hierarchię klas oraz ich funkcje.

5.Opis Realizacji

- A) Środowisko Programistyczne:** Microsoft Visual Studio 2022 jako główne środowisko programistyczne do tworzenia aplikacji.
- B) CMake do Budowy Projektu:** Zastosowano narzędzie CMake (version 3.28.1) do konfiguracji i budowy projektu. CMake umożliwia tworzenie platformowo-niezależnych skryptów konfiguracyjnych, co ułatwia przenośność kodu na różne platformy.
- C) Google Test:** W celu zapewnienia wysokiej jakości kodu i funkcjonalności, zaimplementowano testy jednostkowe przy użyciu Google Test. Każda kluczowa funkcja, tak jak np. "DecimalToHexadecimal", była poddawana testom jednostkowym, co pozwalało na szybkie wykrywanie i poprawę ewentualnych błędów.
- D) Kompilator:** MSVC 2022, wchodzący w skład pakietu Visual Studio 2022, posłużył do kompilacji kodu źródłowego napisanego w języku C++. Jego integracja z Visual Studio usprawniła procesy debugowania
- E) Doxygen:** Doxygen został użyty do diagramów UML.

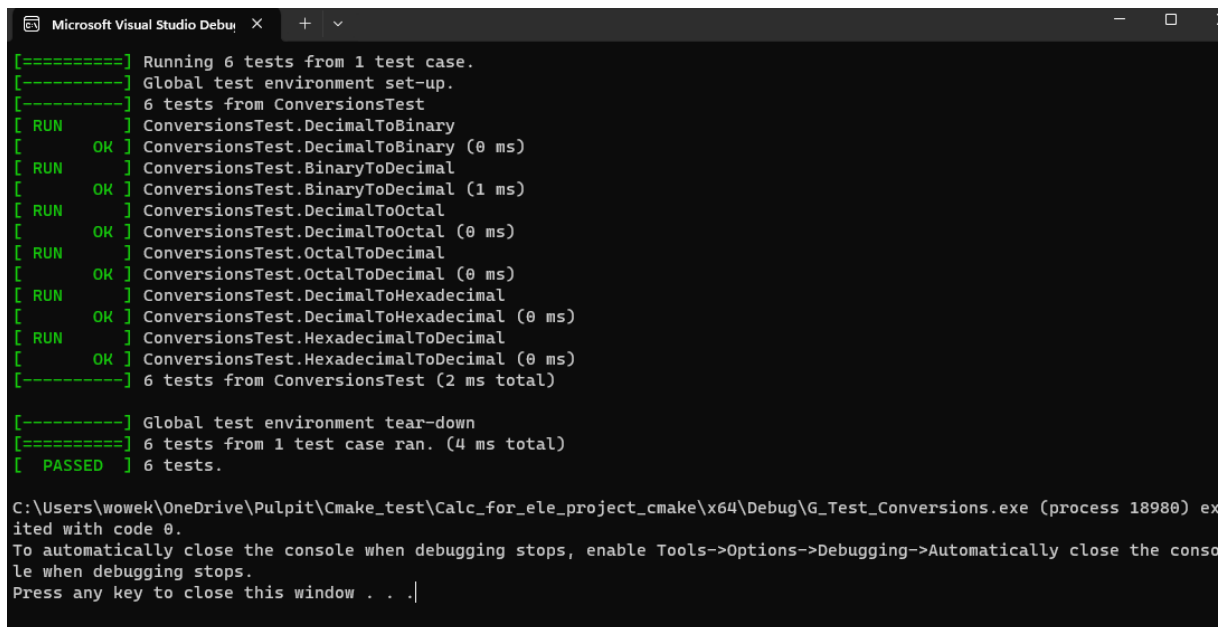
6.Opis wykonanych testów

Przeprowadzono testy każdej z funkcji odpowiedniego systemu liczbowego za pomocą GoogleTest (Gtest).

Przykładowa funkcja dla konwersji z systemu dziesiętkowego do szesnastkowego do przeprowadzenia testu. Wszystkie inne zostały wykonane analogicznie.

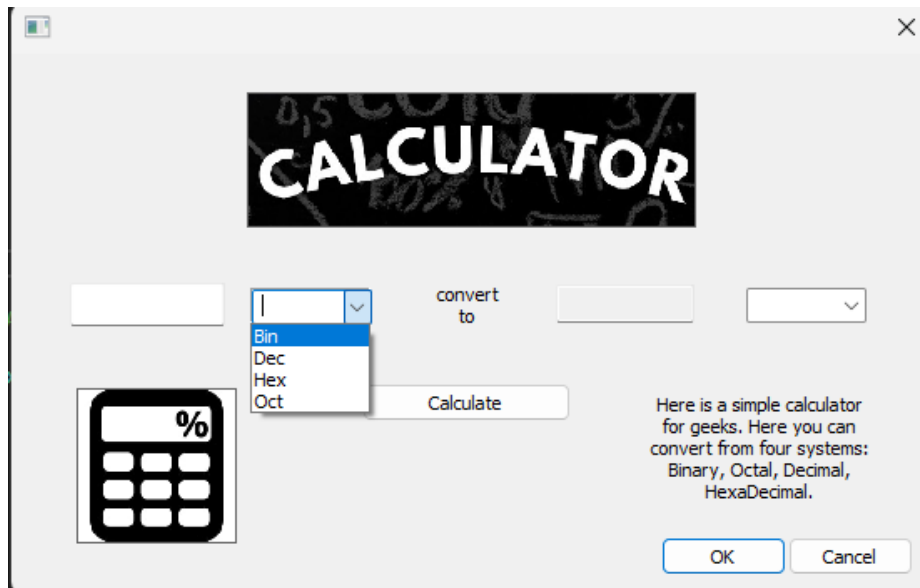
```
TEST(ConversionsTest, DecimalToHexadecimal) {  
    Test_Con System_Converter;  
    std::string x = "46";  
    std::string y = "156";  
    std::string z = "6843";  
  
    EXPECT_EQ(System_Converter.DecimalToHexadecimal(x), "2E");  
  
    EXPECT_EQ(System_Converter.DecimalToHexadecimal(y), "9C");  
  
    EXPECT_EQ(System_Converter.DecimalToHexadecimal(z), "1ABB");  
}
```

Potwierdzenie działania testów. Testy przebiegły pomyślnie.

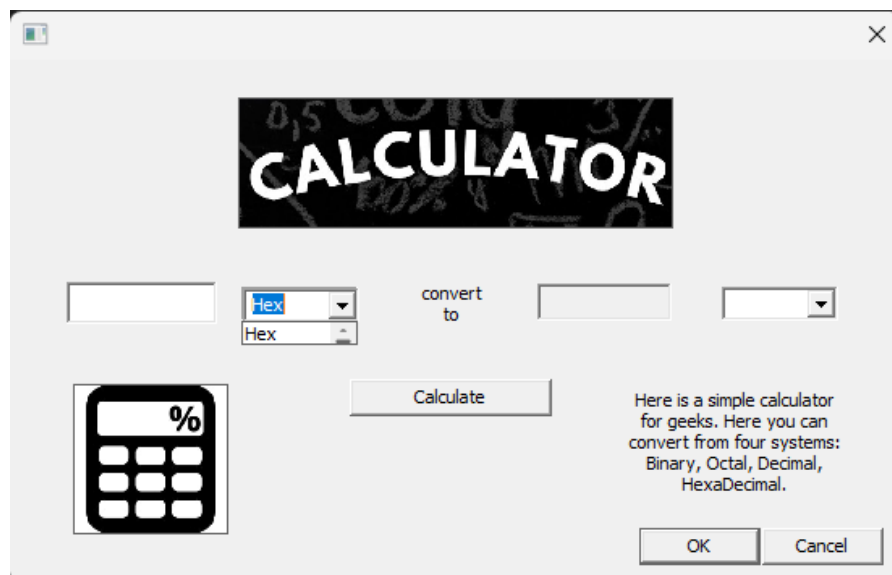


```
Microsoft Visual Studio Debug  X  +  -  
[=====] Running 6 tests from 1 test case.  
[-----] Global test environment set-up.  
[-----] 6 tests from ConversionsTest  
[ RUN      ] ConversionsTest.DecimalToBinary  
[ OK       ] ConversionsTest.DecimalToBinary (0 ms)  
[ RUN      ] ConversionsTest.BinaryToDecimal  
[ OK       ] ConversionsTest.BinaryToDecimal (1 ms)  
[ RUN      ] ConversionsTest.DecimalToOctal  
[ OK       ] ConversionsTest.DecimalToOctal (0 ms)  
[ RUN      ] ConversionsTest.OctalToDecimal  
[ OK       ] ConversionsTest.OctalToDecimal (0 ms)  
[ RUN      ] ConversionsTest.DecimalToHexadecimal  
[ OK       ] ConversionsTest.DecimalToHexadecimal (0 ms)  
[ RUN      ] ConversionsTest.HexadecimalToDecimal  
[ OK       ] ConversionsTest.HexadecimalToDecimal (0 ms)  
[-----] 6 tests from ConversionsTest (2 ms total)  
  
[-----] Global test environment tear-down  
[=====] 6 tests from 1 test case ran. (4 ms total)  
[ PASSED  ] 6 tests.  
  
C:\Users\wowek\OneDrive\Pulpit\Cmake_test\Calc_for_ele_project_cmake\x64\Debug\G_Test_Conversions.exe (process 18980) ex  
ited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso  
le when debugging stops.  
Press any key to close this window . . .|
```

7.Podręcznik użytkownika



W białym polu po lewej stronie wpisujemy odpowiednią wartość jaką chcemy przekonwertować i wybieramy odpowiedni system liczbowy odpowiadający tej wartości, wybieramy po prawej stronie system na jaki chcemy przekonwertować, klikamy przycisk „Calculate” i wynik pojawia nam się w polu po prawej stronie.



Po zbudowaniu projektu przez CMake, aby wybrać odpowiedni system należy zmieniać strzałką naszą listę, aby wybrać odpowiedni system liczbowy.

8. Bibliografia

1. Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
2. Strona Geeksforgeeks: <https://www.geeksforgeeks.org/>
3. ChatGPT: <https://chat.openai.com/>