

KALKULATOR DLA INFORMATYKÓW/ ELEKTRONIKÓW

Autor: Paweł Oleszczak
Akademia Górniczo-Hutnicza

Spis treści

1.	WSTĘP.....	3
2.	FUNKCJONALNOŚĆ (<i>FUNCTIONALITY</i>)	4
3.	ANALIZA PROBLEMU (<i>PROBLEM ANALYSIS</i>)	6
4.	PROJEKT TECHNICZNY (<i>TECHNICAL DESIGN</i>)	8
5.	OPIS REALIZACJI (<i>IMPLEMENTATION REPORT</i>)	11
6.	OPIS WYKONANYCH TESTÓW (<i>TESTING REPORT</i>) - LISTA BUGGÓW, UZUPEŁNIENI, ITD. 13	

1. Wstęp

1.1 Cel Projektu

Raport ten ma na celu przedstawienie projektu tak zwanego ‘‘Konwertera Systemów Liczbowych’’, który został stworzony w celu umożliwienia użytkownikowi konwersji liczb między różnymi systemami liczbowymi.

1.2 Zakres Raportu

Raport obejmuje opis funkcjonalności programu, analizę problemu, szczegóły implementacji, przeprowadzone testy, projekt techniczny zawierający diagramy na temat klas i funkcji. Zestawienie tych wszystkich aspektów daje kompleksowy obraz stworzonego narzędzi

2. Funkcjonalność (*functionality*)

2.1 Wybór Systemu Źródłowego i Docelowego

Użytkownik ma możliwość intuicyjnego wyboru systemu liczbowego, z którego chce przekonwertować liczbę, oraz systemu, na który chce dokonać konwersji. Interfejs użytkownika jest zoptymalizowany pod kątem prostoty obsługi, prezentując czytelne opcje do wyboru.

2.2 Walidacja Danych Wejściowych

System Źródłowy	Zasady Walidacji
Binarny	Akceptuje tylko 0 i 1.
Ósemkowy	Akceptuje cyfry 0-7.
Dziesiętny	Akceptuje cyfry 0-9.
Szesnastkowy	Akceptuje cyfry 0-9 oraz A-F.

2.3 Konwersja Liczby

Po wyborze systemu źródłowego i docelowego, użytkownik wprowadza liczbę do przekonwertowania. Program dokonuje konwersji z jednego systemu liczbowego na drugi, korzystając z odpowiedniego algorytmu. Wynik jest prezentowany w czytelny sposób, uwzględniając format danego systemu liczbowego.

2.4 Interfejs Użytkownika

Interfejs użytkownika jest zaprojektowany w sposób przyjazny dla użytkownika, oferując czytelne komunikaty oraz jasne instrukcje.

2.5 Ostrzeżenia i Komunikaty

Aby ułatwić zrozumienie i obsługę programu, używane są jasne komunikaty oraz ostrzeżenia.

2.6 Prostota i Język

Wszystkie elementy interfejsu oraz komunikaty są sformułowane w prosty i zrozumiały sposób. Język użyty w programie jest dostosowany do odbiorcy, eliminując skomplikowane terminy tam, gdzie to możliwe.

3. Analiza problemu (*problem analysis*)

3.1 Wprowadzenie

Projekt "Konwerter Systemów Liczbowych" skupia się na potrzebach użytkowników związanych z konwersją liczb pomiędzy różnymi systemami liczbowymi. Analiza problemu obejmuje zrozumienie kontekstu, wymagań użytkownika oraz identyfikację głównych problemów, które projekt ma rozwiązać.

3.2 Kontekst Projektu

- **Konwersja Liczb:** Użytkownicy potrzebują narzędzia umożliwiającego konwersję liczb pomiędzy systemami binarnym, ósemkowym, dziesiętnym i szesnastkowym.
- **Prostota Obsługi:** Interfejs użytkownika powinien być prosty i intuicyjny, aby umożliwić użytkownikom łatwe korzystanie z programu bez specjalistycznej wiedzy.
- **Walidacja Danych:** Istotnym elementem jest zapewnienie poprawności wprowadzonych danych, aby uniknąć błędnych konwersji.

3.3 Cel Projektu

Celem projektu jest stworzenie programu, który spełnia powyższe wymagania i rozwiązuje identyfikowane problemy. Program ma być prosty w obsłudze, precyzyjny, a jednocześnie skuteczny w konwersji liczb pomiędzy systemami liczbowymi, eliminując błędy popełniane przy ręcznych obliczeniach.

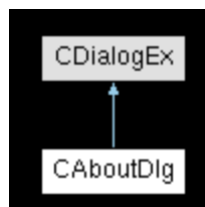
4. Projekt techniczny (*technical design*)

Uwagi wstępne

W celu pokazania użytych klas razem z metodami i atrybutami skorzystałem z programu doxygen, efekt był następujący:



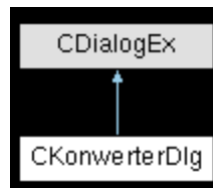
Wszystkie klasy zawarte w programie



Protected Member Functions

```

virtual void DoDataExchange (CDataExchange *pDX)
  
```

Public Member Functions

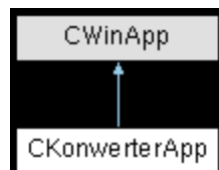
	CKonwerterDlg (CWnd *pParent=nullptr)
afx_msg void	OnCbnSelchangeCombo2 ()
afx_msg void	OnCbnSelchangeCombo3 ()
afx_msg void	OnEnChangeEdit1 ()
afx_msg void	OnBnClickedButton1 ()
CString	DecimalToBinary (int decimalValue)

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
int	m_nSelectedSystemCombo2
int	m_nSelectedSystemCombo3



Public Member Functions

virtual BOOL	InitInstance ()
--------------	------------------------

Podsumowanie

Program doxygen zobrazował klasy i klasy dziedziczące i ich funkcje w sposób czytelny i prosty.

5. Opis realizacji (*implementation report*)

5.1 Narzędzia

- **Visual Studio 2022:** zostało wybrane jako główne środowisko programistyczne ze względu na bogatą funkcjonalność.
- **Git:** nadal używany jako system kontroli wersji do zarządzania kodem źródłowym. Pozwala na skuteczne śledzenie zmian, zarządzanie gałęziami (branchami) oraz udostępnianie kodu w zdalnych repozytoriach.

5.2 Kompilator C++

Kompilator MSVC 2022, część pakietu Visual Studio 2022, został wykorzystany do kompilacji kodu źródłowego napisanego w języku C++. Współpracuje on z Visual Studio, co ułatwia debugowanie, analizę kodu oraz testowanie.

5.3 System Utrzymania Źródeł

Projekt nadal korzysta z systemu kontroli wersji Git do utrzymania źródeł. Struktura repozytorium obejmuje gałęzie dla różnych etapów rozwoju projektu, takie jak master dla stabilnej wersji, develop dla wersji w trakcie rozwoju, oraz gałęzie funkcjonalne do dodawania nowych funkcji.

5.4 Zarządzanie Zależnościami

Zależności projektu są nadal zarządzane za pomocą narzędzia CMake, które umożliwia skonfigurowanie, budowę i instalację projektu na różnych platformach. Dzięki temu osiągnięto większą elastyczność w zarządzaniu zależnościami.

5.5 Dokumentacja

Dokumentacja kodu jest nadal tworzona z wykorzystaniem komentarzy w formacie Doxygen, co pozwala generować czytelną dokumentację w formie HTML. Dostarcza ona szczegółowy opis funkcji, klas i interfejsów programistycznych.

5.6 Proces Testowania

Testowanie programu obejmuje różne przypadki, w tym testy jednostkowe dla klas konwersji przy użyciu Google Test.

6. Opis wykonanych testów (*testing report*)

```
#include "gtest/gtest.h"
#include "pch.h"
#include "Test.h"

TEST(DecimalToBinaryTest, PositiveValues) {
    EXPECT_EQ(CKonwerterFunctions::DecimalToBinary(10), "1010");
    EXPECT_EQ(CKonwerterFunctions::DecimalToBinary(255), "11111111");
}

TEST(DecimalToBinaryTest, Zero) {
    EXPECT_EQ(CKonwerterFunctions::DecimalToBinary(0), "0");
}

TEST(DecimalToBinaryTest, NegativeValues) {
    EXPECT_EQ(CKonwerterFunctions::DecimalToBinary(15), "101");
}

int main(int argc, char** argv) {
    ::testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

Kod Google Test (Testowanie kilku funkcji programu)

```
[=====] Running 3 tests from 1 test case.
[-----] Global test environment set-up.
[-----] 3 tests from DecimalToBinaryTest
[ RUN     ] DecimalToBinaryTest.PositiveValues
[ OK      ] DecimalToBinaryTest.PositiveValues (0 ms)
[ RUN     ] DecimalToBinaryTest.Zero
[ OK      ] DecimalToBinaryTest.Zero (0 ms)
[ RUN     ] DecimalToBinaryTest.NegativeValues
C:\Users\pawel\OneDrive\Pulpit\Projekt JP0\GTest\test.cpp(16): error: Expected equality of these values:
  CKonwerterFunctions::DecimalToBinary(15)
    Which is: "1111"
  "101"
[ FAILED  ] DecimalToBinaryTest.NegativeValues (2 ms)
[-----] 3 tests from DecimalToBinaryTest (2 ms total)

[-----] Global test environment tear-down
[=====] 3 tests from 1 test case ran. (3 ms total)
[ PASSED  ] 2 tests.
[ FAILED  ] 1 test, listed below:
[ FAILED  ] DecimalToBinaryTest.NegativeValues

1 FAILED TEST
```

Testy przebiegły tak jak powinny.

Bibliografia

1. **Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.**