**Kalkulator dla elektroników/informatyków**

**Z przedmiotu:** Języki programowania obiektowego

**Autor:** Kacper Janicki

**Kierunek:** Elektronika i Telekomunikacja

**Spis treści:**

1. WSTĘP...........................................................................................................................3

2. FUNKCJONALNOŚĆ (*FUNCTIONALITY*)...........................................................................4

3. ANALIZA PROBLEMU (*PROBLEM ANALYSIS*)..................................................................5

4. PROJEKT TECHNICZNY (*TECHNICAL DESIGN*).................................................................6

5. OPIS REALIZACJI (*IMPLEMENTATION REPORT*)............................................................10

6. OPIS WYKONANYCH TESTÓW (*TESTING REPORT*) .......................................................11

7. PODRĘCZNIK UŻYTKOWNIKA (*USER'S MANUAL*).........................................................12

8. BIBLIOGRAFIA..............................................................................................................14

**Lista oznaczeń:**

|  |  |
| --- | --- |
| MFC | Microsoft Foundation Class Library |

**1.Wstęp:**

Dokument dotyczy projektu kalkulatora dla elektroników/informatyków. Kalkulator ten został zaprojektowany z myślą o łatwej i szybkiej konwersji między różnymi systemami liczbowymi, takimi jak dziesiętny (decimal), binarny (binary), ósemkowy (octal) i szesnastkowy (hexadecimal). W projekcie został zaimplementowany prosty interfejs użytkownika za pomocą biblioteki MFC. Całość została wykonana w języku programowania C++ w środowisku Visual Studio 2022.

**2.FUNKCJONALNOŚĆ (*FUNCTIONALITY*):**

Kalkulator oferuje możliwość przeliczania liczb między czterema głównymi systemami liczbowymi: dziesiętnym, binarnym, ósemkowym i szesnastkowym. Użytkownik może wprowadzić liczbę w jednym systemie, a kalkulator natychmiast przeliczy ją na pozostałe trzy systemy. Interfejs jest prosty i intuicyjny, co pozwala na szybkie i łatwe przeliczanie liczb, obsługuje duże liczby. W razie błędnie wprowadzonych danych wyskoczy prosty i zrozumiały komunikat, dzięki któremu użytkownik będzie mógł zmienić dane na poprawne. Więcej informacji odnośnie komunikatów znajduje się w punkcie numer 7, czyli w podręczniku użytkownika (user’s manual) na stronie numer 12 .

**3.ANALIZA PROBLEMU (*PROBLEM ANALYSIS*):**

Praca z różnymi systemami liczbowymi jest nieodłącznym elementem pracy elektroników i informatyków. Często muszą oni przeliczać liczby z jednego systemu na inny, co może być czasochłonne i narażone na błędy, zwłaszcza przy ręcznym przeliczaniu.

## **Problem 1: Czasochłonność**

Ręczne przeliczanie liczb między różnymi systemami liczbowymi może być czasochłonne. Dla dużych liczb lub skomplikowanych konwersji, proces ten może zająć dużo czasu, który mógłby być lepiej wykorzystany na inne zadania.

## **Problem 2: Możliwość błędów**

Ręczne przeliczanie liczb jest narażone na błędy. Nawet drobne pomyłki mogą prowadzić do znacznych błędów w wynikach.

Kalkulator został zaprojektowany, aby rozwiązać te problemy, oferując szybką, dokładną i łatwą w użyciu platformę do przeliczania liczb między różnymi systemami liczbowymi. Dzięki temu, użytkownicy mogą skupić się na swojej pracy, zamiast spędzać czas na ręcznym przeliczaniu liczb.

**4.PROJEKT TECHNICZNY (*TECHNICAL DESIGN*):**

Za pomocą programu doxygen, wygenerowano listę wszystkich występujących klas w programie oraz występujących w klasach funkcji. Program doxygen ułatwił zrozumienie struktury projektu kalkulatora.

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Rysunek 1. Lista klas w projekcie

Diagram dla poszczególnych klas:**Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Opis wygenerowany automatycznie**

Rysunek Diagram dla CAboutDlg.

**Obraz zawierający tekst, Czcionka, linia, numer

Opis wygenerowany automatycznie**

Rysunek Diagram dla CCalcForEleApp.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Rysunek Diagram dla CCalcForEleDlg.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek Lista funkcji w klasie ZamianaLicz.

Hierarchia Klas w projekcie kalkulatora:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek Hierarchia klas.

Została również wygenerowana lista klas oraz funkcji w niej istniejących dla google testu, sprawdzającego poprawność działania funkcji dzięki któremu kalkulator zamienia naszą liczbę w liczby w innych systemach liczbowych.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek Lista klas w Gtest.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek Lista funkcji w klasie z Googletestu.

**5.OPIS REALIZACJI (*IMPLEMENTATION REPORT*):**

Wykorzystane narzędzia w projekcie:

**- Visual Studio 2022**: Wybrano to środowisko ze względu na jego wszechstronność i bogactwo funkcji, w tym bibliotekę MFC, co czyni go optymalnym wyborem dla tego projektu.

**- Debugger w Visual Studio 2022**: Umożliwił analizę, monitorowanie i rozwiązywanie problemów w kodzie podczas debugowania. Dzięki wyświetlanym błędom, możliwe było szybkie znalezienie i naprawienie błędów w kodzie.

**- CMake**: Utworzono plik CMakeLists, który umożliwia generowanie projektu na dowolnym komputerze. CMake jest szeroko stosowany w dużych projektach do budowania projektów na różnych systemach/platformach.

**- GitHub**: Wszystkie pliki projektu zostały umieszczone na tej platformie.

**- Google Test**: Za pomocą tego narzędzia przeprowadzono testy funkcji przeliczających liczby na inne systemy liczbowe. Pozwoliło to upewnić się, że kalkulator działa poprawnie we wszystkich przypadkach.

**- Doxygen**: Ten program umożliwił przedstawienie listy, hierarchii klas oraz funkcji występujących w tych klasach.

**6.OPIS WYKONANYCH TESTÓW (*TESTING REPORT*):**

Zostały przeprowadzone testy GTest, które sprawdzają poprawność działania poszczególnych funkcji konwercji liczb. Poniżej przedstawiono zrzut ekranu, na którym można zaobserwować kod wspomnianego GTestu oraz po prawej stronie wynik uruchomienia tego kodu w środowisku Visual Studio 2022. Przeprowadzone testy przebiegły w sposób pomyślny bez większych problemów. Dlatego można założyć, że funkcje przemiany liczb działają poprawnie.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek Zrzut ekranu kodu oraz wyniku Google test.

**7.Podręcznik użytkownika (*user's manual*):**

Poniżej znajduje się zrzut ekranu interfejsu użytkownika wykonanego za pomocą biblioteki MFC. Można na nim zauważyć cztery miejsca na wpisanie liczby, którą chcemy przeliczyć na pozostałe systemy liczbowe. Wystarczy wpisać znaną nam liczbę w odpowiednim miejscu, następnie kliknąć przycisk. „Convert”, a kalkulator w bardzo szybkim tempie przeliczy naszą liczbę na wszystkie inne systemy. Po przeliczeniu wystarczy nacisnąć przycisk „Clear”, który ma za zadanie wyczyścić cały kalkulator, aby móc wpisać następną liczbę do przeliczenia. W razie błędnego wpisania wartości danej liczby wyskoczą komunikaty, dzięki którym będzie można poprawić naszą liczbę.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek Interfejs użytkownika.

Komunikaty oraz błędy, które mogą wyskoczyć w razie złego wpisania liczby:

-Error numer 1 występuję gdy albo żadna wartość nie została wpisana albo gdy wpisano wartość do więcej niż jednego okienka.

Obraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

Rysunek Error numer 1.

-Error numer 2 wyskakuje gdy, została wpisana zła wartość w jakieś okienko.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek Error numer 2

-Error numer 3 wyskakuje gdy w okienku dla hexadecimal jest wpisana zła wartość, albo zostały użyte małe literki.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek Error numer 3.

**Bibliografia:**

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

[2] <https://www.geeksforgeeks.org/> - Strona Geeks for geeks

[3] https://stackoverflow.com/ - Strona Stackoverflow

[4] <https://github.com/> - GitHub

[5] <https://gitlab.kitware.com/cmake/community/-/wikis/FAQ#how-to-use-mfc-with-cmake>

[6] <https://chat.openai.com> – Chat GPT