

| REV | DATA | ZMIANY |
|-----|------------|---|
| 1 | 22.12.2023 | <i>Paweł Harmata (pharmata@student.agh.edu.pl</i> |
| 2 | 16.02.2024 | <i>Paweł Harmata (pharmata@student.agh.edu.pl</i> |
| | | |
| | | |

KALKULATOR DLA ELEKTRONIKÓW/INFORMATYKÓW

Autor: Paweł Harmata
Akademia Górniczo-Hutnicza

Spis treści

| | |
|---|-----------|
| 1. WSTĘP | 4 |
| 2. FUNKCJONALNOŚĆ (<i>FUNCTIONALITY</i>) | 6 |
| 3. ANALIZA PROBLEMU (<i>PROBLEM ANALYSIS</i>) | 7 |
| 4. PROJEKT TECHNICZNY (<i>TECHNICAL DESIGN</i>) | 9 |
| 5. OPIS REALIZACJI (<i>IMPLEMENTATION REPORT</i>) | 10 |
| 6. OPIS WYKONANYCH TESTÓW (<i>TESTING REPORT</i>) - LISTA BUGGÓW, UZUPEŁNIENÍ, ITD. 11 | |
| 7. PODRĘCZNIK UŻYTKOWNIKA (<i>USER'S MANUAL</i>) | 14 |
| 7.1 TEST APLIKACJI NA WINDOWS | 14 |
| 8. METODOLOGIA ROZWOJU I UTRZYMANIA SYSTEMU (<i>SYSTEM MAINTENANCE AND DEPLOYMENT</i>) | 17 |
| BIBLIOGRAFIA | 18 |

Lista oznaczeń

| | |
|-----|-----------------------|
| BIN | System binarny |
| HEX | System heksadecymalny |
| DEC | System dziesiętny |
| OCT | System oktalny |

1. Wstęp

Dokument dotyczy opracowania kalkulatora dla elektroników/informatyków z prostym interfejsem użytkownika. Zadaniem tego kalkulatora jest konwersja liczb na różne systemy liczbowe takich jak system dwójkowy ósemkowy i szesnastkowy. Wykonywaniu prostych obliczeń na liczbach zmiennoprzecinkowych oraz powyższych systemach liczbowych.

1. Wymagania systemowe (*requirements*)

Podstawowe założenia projektu:

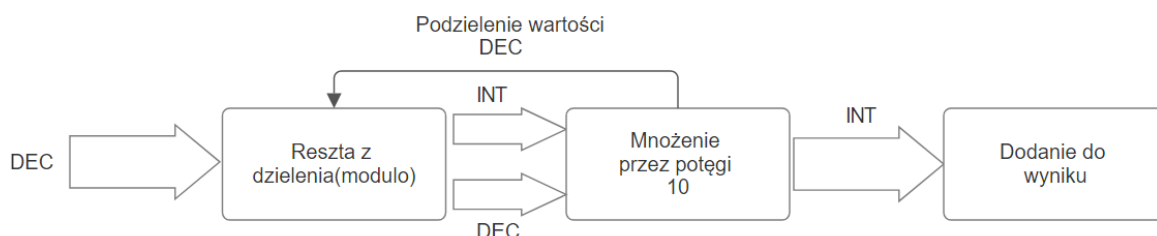
1. Stworzenie planu i opisu podstawowych elementów projektu.
2. Stworzenie systemu klas odpowiadających za konwersje liczb z DEC na BIN, OCT i HEX oraz konwersje w drugą stronę za pomocą odpowiednich metod.
3. Stworzenie głównej klasy która będzie odpowiadać za łączenie modułów oraz będzie odpowiadać za podstawowe funkcje kalkulatora
4. Wykorzystanie programu CMake
5. Stworzenie prostego interfejsu dialogowego w funkcji main
6. Przeprowadzanie testów Google Test
7. Stworzenie repozytorium github
8. Stworzenie finalnych wersji kodów oraz dokumentacji
9. Przesłanie wszystkich elementów na github

2. Funkcjonalność (*functionality*)

Kalkulator zostanie napisany w języku C++ 20. Za pomocą prostego dialogu Windows wybieramy interesującą nas funkcjonalność. Konwersje lub kalkulator dla konkretnego systemu liczbowego. Następnie wpisujemy wymagane dane w przypadku kalkulatorów liczby oraz interesujące nas działanie, a w przypadku konwertera wybieramy interesujący nas konwerter i wpisujemy liczbę do zmiany. Cały program ma działać w języku polskim. Systemy liczbowe obsługiwane przez program to BIN, OCT, HEX i DEC. Kalkulator domyślnie ma działać dla liczb zmiennoprzecinkowych.

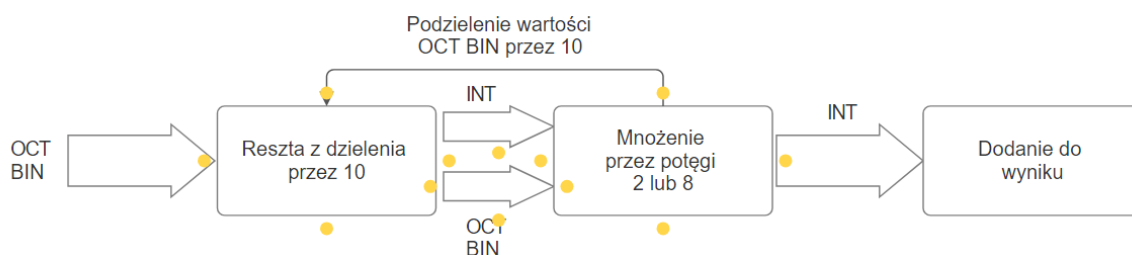
3. Analiza problemu (*problem analysis*)

Pierwszym istotnym problemem jest konwersja. W celu zmiany z DEC na HEX, BIN oraz OCT. Żeby to osiągnąć wykorzystamy pętlę w której użyjemy funkcji modulo odpowiednio z 2 dla BIN, 8 dla OCT i 16 dla HEX na wartości wejściowej. Wartość otrzymana z modulo będzie mnożona przez potęgę 10 (stopień potęgi powinien być równy numerowi powtórzenia pętli) i dodawana do wyniku. Wartość wejściowa będzie dzielona przez liczbę równą liczbie modulo. Proces powinien trwać dopóki wartość wejściowa się nie wyzeruje. Taka metoda odzwierciedla najprostszą metodę matematyczną do konwersji. Dzięki modulo określamy cyfrę na danej „pozycji” w liczbie a mnożąc ją razy potęgę 10 „wybieramy” tą pozycję.



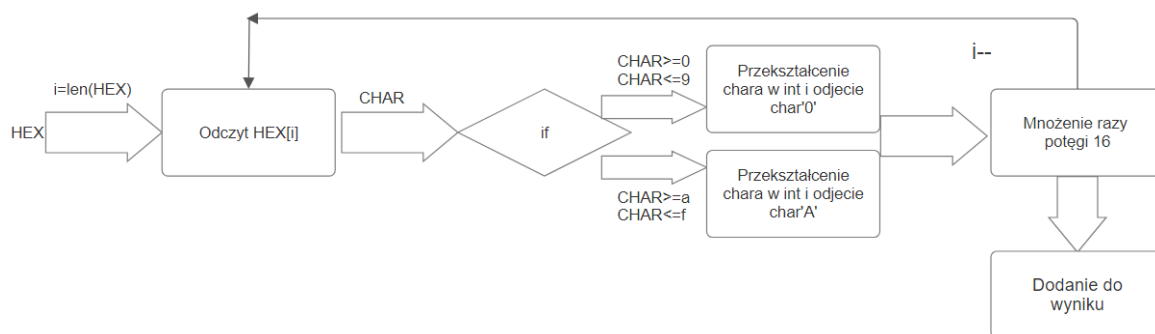
Rysunek 3.1 Pokazuje uproszczony schemat działania konwertera DEC do HEX, BIN oraz OCT

Drugim problemem jest konwersja w drugą stronę. Zmiana z BIN ,OCT oraz HEX na DEC. W tym przypadku trzeba zastosować dwa różne schematy osobny dla BIN oraz OCT oraz osobny dla HEX. BIN oraz OCT wykorzystuje tylko cyfry arabskie a HEX wykorzystuje również litery. Dla BIN oraz OCT znowu wykorzystamy pętlę oraz funkcję modulo z cyfrą 10. Ma to na celu pozyskiwanie wartości ostatniej cyfry , następnie pozyskaną liczbę mnożymy razy potęgę liczby odpowiadającej systemowi tj. 2 dla BIN oraz 8 dla OCT. Stopień potęgi jest równy pozycji cyfry w liczbie co jest równo znaczne z numerem pętli. Otrzymaną wartość dodamy do wyniku i za pomocą dzielenia przez 10 usuniemy ostatnią cyfrę. Pętlę powinna być wykonywana do wyzerowania się wartości wejściowej.



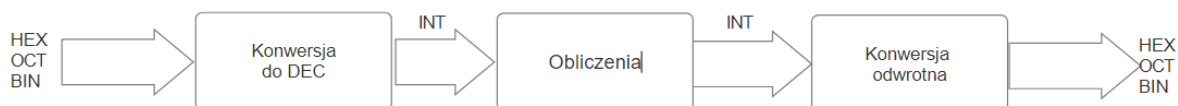
Rysunek 3.2 Pokazuje uproszczony schemat działania konwertera BIN, OCT do DEC

W przypadku konwersji z HEX do DEC w pętli odczytamy elementy stringa po indeksach. Sprawdzimy czy element jest cyfra czy literą a następnie odejmiemy wartość „0” lub „a” w kodzie ASCII. Mnożymy otrzymaną wartość przez potęgę 16 i dodajemy do wyniku. Pętla powinna trwać do przejścia przez wszystkie elementy



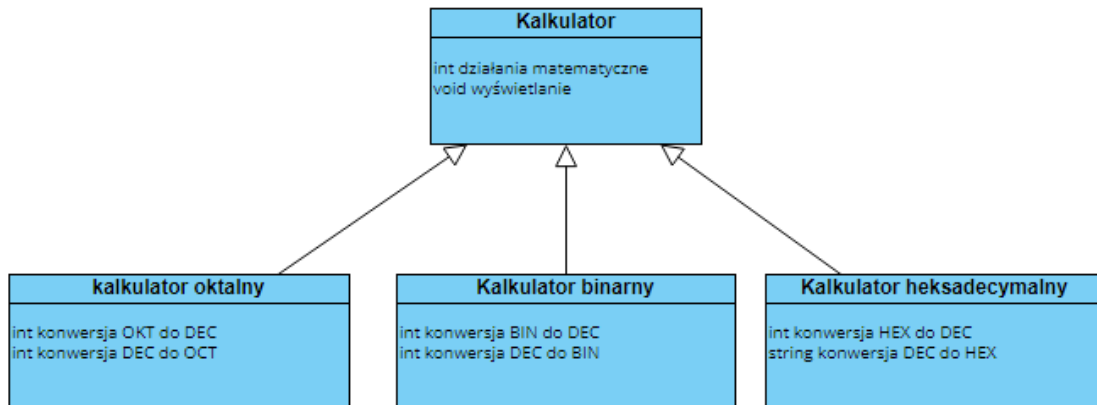
Rysunek 3.3 Pokazuje uproszczony schemat działania konwertera HEX do DEC

Ostatnim elementem jest kalkulator w celu przeprowadzenia prostych obliczeń algebraicznych na różnych systemach liczbowych w tym celu przeprowadzimy konwersje z systemów BIN, OCT lub HEX na DEC. Następnie przeprowadzimy interesujące nas obliczenia. Na koniec konwertując na system pierwotny.



Rysunek 3.4 Pokazuje uproszczony schemat działania kalkulatora

4. Projekt techniczny (*technical design*)



Projekt ma składać się 3 podrzędnych klas oraz jednej nadrzędnej. Klasa kalkulator oktalny ma być odpowiedzialna za konwersje z systemu OCT do DEC i DEC do OCT. Klasa kalkulator binarny ma być odpowiedzialna za konwersje z systemu BIN do DEC i DEC do BIN. Klasa kalkulator heksadecymalny ma być odpowiedzialna za konwersje z systemu HEX do DEC i DEC do HEX. Te 3 klasy należą do klas podrzędnych.

Główna klasa kalkulator ma za zadanie łączyć i wykorzystywać 3 podrzędne klasy do przeprowadzania obliczeń na różnych systemach liczbowych. Sama ma posiadać dodatkowe metody służące do zwiększenia funkcjonalności kalkulatora.

Klasy będą używane w mainie i każda metoda będzie możliwa do użytku osobno.

5. Opis realizacji (*implementation report*)

Projekt zostanie zrealizowany na komputerze z systemem Windows 11 w programie Visual Studio 2022. W celu przetestowania kodu użyty zostanie debbuger oraz Google Test. W celach udostępnienia projektu do oceny zostanie wykorzystany git oraz platforma github. Ostatnim narzędziem użytym w projekcie będzie CMake.

Specyfikacja maszyny testowej:

Windows 11 wersja 23H2
Procesor: AMD Ryzen 7 5800X 3.8Ghz
RAM: 32 GB DDR4
Dysk: 1 TB M2

Wersje programów:

Visual Studio 2022 wersja 17.8.3
CMake wersja 3.28.1

Za pomocą Google Test przetestowane zostaną wszystkie konwertery. Przeprowadzone zostaną po 3 testy dla każdego konwertera które sprawdzą poprawność działania konwertera oraz dodatkowe testy sprawdzające działanie zabezpieczeń przed nie poprawnymi znakami lub wartościami

W Visual Studio ustawiono wersje języka na C++ 20.

6. Opis wykonanych testów (*testing report*) - lista buggów, uzupełnień, itd.

| Kod usterki | Data | Autor | Opis | Stan |
|-------------|-------|---------------|---|------------|
| 1 | 22.12 | Paweł Harmata | Nie poprawnie przeprowadzona implementacja funkcji hex_dec Test funkcji powoduje crash | Naprawiona |
| 2 | 01.01 | Paweł Harmata | Brak ochrony przed niepoprawnymi inputami | Naprawiona |
| 3 | 15.02 | Paweł Harmata | Nie poprawny format funkcji dec_bin maksymalna wartość 1023 potem przestaje poprawnie działać | Naprawiona |
| 4 | 16.02 | Paweł Harmata | Błąd gtest dla TEST(KalkulatorHexTest, DecToHexInvalidInput) nie działa poprawnie | Naprawiona |
| | | | | |
| | | | | |
| | | | | |

TESTY GOOGLE TEST

Przeprowadzone podstawowe testy dla funkcji algebraicznych:

```

//Przetestowanie dodanych funkcji z operacjami algebraicznymi
TEST(KalkulatorTest, Dzielenie) {
    kalkulator kalk;

    EXPECT_EQ(2, kalk.dzielenie(6, 3));
    EXPECT_EQ(0, kalk.dzielenie(0, 5));
    EXPECT_EQ(-100, kalk.dzielenie(5, 0));
}

TEST(KalkulatorTest, Silnia) {
    kalkulator kalk;

    EXPECT_EQ(1, kalk.silnia(0));
    EXPECT_EQ(1, kalk.silnia(1));
    EXPECT_EQ(120, kalk.silnia(5));
}

TEST(KalkulatorTest, Potega) {
    kalkulator kalk;

    EXPECT_DOUBLE_EQ(8, kalk.potega(2, 3));
    EXPECT_DOUBLE_EQ(1.0 / 8, kalk.potega(2, -3));
}

TEST(KalkulatorTest, Pierwiastek) {
    kalkulator kalk;

    EXPECT_DOUBLE_EQ(4, kalk.pierwiastek(16, 2));
    EXPECT_TRUE(std::isnan(kalk.pierwiastek(-16, 2)));
    EXPECT_TRUE(std::isnan(kalk.pierwiastek(16, 0)));
    EXPECT_DOUBLE_EQ(3, kalk.pierwiastek(27, 3));
}

```

Przeprowadzone podstawowe testy dla Konwerterów:

| | |
|--|---|
| <pre> //Test konwerterów //binarny do decymalnego TEST(KalkulatorBinTest, BinToDecValidInput) { kalkulator_bin kalk_bin; EXPECT_EQ(5, kalk_bin.bin_dec(101)); EXPECT_EQ(12, kalk_bin.bin_dec(1100)); } TEST(KalkulatorBinTest, BinToDecInvalidInput) { kalkulator_bin kalk_bin; EXPECT_EQ(-3, kalk_bin.bin_dec(1234)); } //decymalnego do binarny TEST(KalkulatorBinTest, DecToBinValidInput) { kalkulator_bin kalk_bin; EXPECT_EQ(101, kalk_bin.dec_bin(5)); EXPECT_EQ(1100, kalk_bin.dec_bin(12)); } TEST(KalkulatorBinTest, DecToBinInvalidInput) { kalkulator_bin kalk_bin; EXPECT_EQ(-2, kalk_bin.dec_bin(-5)); } </pre> | <pre> //oktalny do decymalny TEST(KalkulatorOCTTest, OctToDecValidInput) { kalkulator_octa kalk_oct; EXPECT_EQ(45, kalk_oct.oct_dec(55)); EXPECT_EQ(10, kalk_oct.oct_dec(12)); } TEST(KalkulatorOCTTest, OctToDecInvalidInput) { kalkulator_octa kalk_oct; EXPECT_EQ(-3, kalk_oct.oct_dec(999)); } //decymalnego do oktalny TEST(KalkulatorOCTTest, DecToOctValidInput) { kalkulator_octa kalk_oct; EXPECT_EQ(177, kalk_oct.dec_oct(127)); EXPECT_EQ(21, kalk_oct.dec_oct(17)); } TEST(KalkulatorOCTTest, DecToOctInvalidInput) { kalkulator_octa kalk_oct; EXPECT_EQ(-2, kalk_oct.dec_oct(-5)); } </pre> |
|--|---|

```

//heksadecymalny do decymalny
TEST(KalkulatorHexTest, HexToDecValidInput) {
    kalkulator_hex kalk_hex;
    EXPECT_EQ(13, kalk_hex.hex_dec("D"));
    EXPECT_EQ(3871, kalk_hex.hex_dec("F1F"));
}

TEST(KalkulatorHexTest, HexToDecInvalidInput) {
    kalkulator_hex kalk_hex;
    EXPECT_EQ(-3, kalk_hex.hex_dec("J"));
}

//decymalnego do heksadecymalny
TEST(KalkulatorHexTest, DecToHexValidInput) {
    kalkulator_hex kalk_hex;
    EXPECT_EQ("3E7", kalk_hex.dec_hex(999));
    EXPECT_EQ("4D", kalk_hex.dec_hex(77));
}

TEST(KalkulatorHexTest, DecToHexInvalidInput) {
    kalkulator_hex kalk_hex;
    EXPECT_EQ("-2", kalk_hex.dec_hex(-5));
}

```

Wyniki testów:

```

[ OK ] KalkulatorTest.Dzielenie (0 ms)
[ RUN ] KalkulatorTest.Silnia
[ OK ] KalkulatorTest.Silnia (0 ms)
[ RUN ] KalkulatorTest.Potega
[ OK ] KalkulatorTest.Potega (0 ms)
[ RUN ] KalkulatorTest.Pierwiastek
Dla stopnia parzystego wartość musi być dodatniaStopień nie może być równy zero[ OK ] KalkulatorTest.Pierwiastek (0 ms)
[-----] 4 tests from KalkulatorTest (1 ms total)

[-----] 4 tests from KalkulatorBinTest
[ RUN ] KalkulatorBinTest.BinToDecValidInput
[ OK ] KalkulatorBinTest.BinToDecValidInput (0 ms)
[ RUN ] KalkulatorBinTest.BinToDecInvalidInput
Error: Błędna liczba binarna.
[ OK ] KalkulatorBinTest.BinToDecInvalidInput (0 ms)
[ RUN ] KalkulatorBinTest.DecToBinValidInput
[ OK ] KalkulatorBinTest.DecToBinValidInput (0 ms)
[ RUN ] KalkulatorBinTest.DecToBinInvalidInput
Error: Liczba nie może być ujemna.
[ OK ] KalkulatorBinTest.DecToBinInvalidInput (0 ms)
[-----] 4 tests from KalkulatorBinTest (1 ms total)

[-----] 4 tests from KalkulatorOCTTest
[ RUN ] KalkulatorOCTTest.OctToDecValidInput
[ OK ] KalkulatorOCTTest.OctToDecValidInput (0 ms)
[ RUN ] KalkulatorOCTTest.OctToDecInvalidInput
Error: Błędna liczba oktalna.
[ OK ] KalkulatorOCTTest.OctToDecInvalidInput (0 ms)
[ RUN ] KalkulatorOCTTest.DecToOctValidInput
[ OK ] KalkulatorOCTTest.DecToOctValidInput (0 ms)
[ RUN ] KalkulatorOCTTest.DecToOctInvalidInput
Error: Liczba nie może być ujemna..
[ OK ] KalkulatorOCTTest.DecToOctInvalidInput (0 ms)
[-----] 4 tests from KalkulatorOCTTest (1 ms total)

[-----] 4 tests from KalkulatorHexTest
[ RUN ] KalkulatorHexTest.HexToDecValidInput
[ OK ] KalkulatorHexTest.HexToDecValidInput (0 ms)
[ RUN ] KalkulatorHexTest.HexToDecInvalidInput
Error: To nie jest liczba hex 'J'
[ OK ] KalkulatorHexTest.HexToDecInvalidInput (0 ms)
[ RUN ] KalkulatorHexTest.DecToHexValidInput
[ OK ] KalkulatorHexTest.DecToHexValidInput (0 ms)
[ RUN ] KalkulatorHexTest.DecToHexInvalidInput
Error: Liczba nie może być ujemna.
[ OK ] KalkulatorHexTest.DecToHexInvalidInput (0 ms)
[-----] 4 tests from KalkulatorHexTest (2 ms total)

[-----] Global test environment tear-down
[-----] 16 tests from 4 test cases ran. (6 ms total)
[ PASSED ] 16 tests.

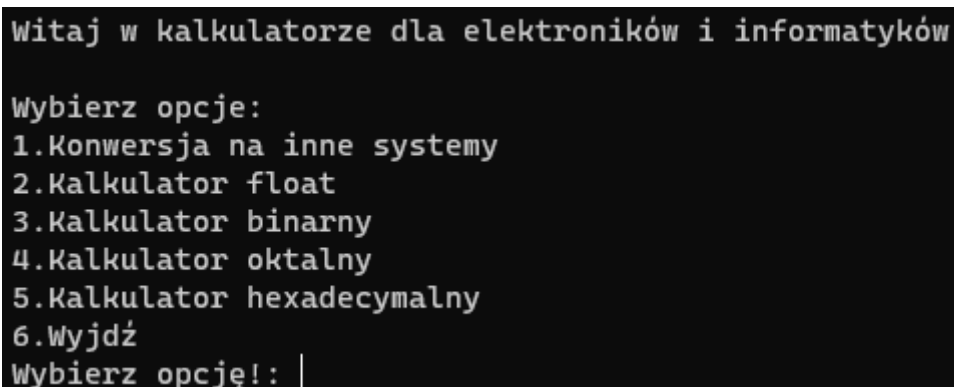
```

Wszystkie testy przeszły pomyślnie !

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

7.1 Test aplikacji na windows

Po uruchomieniu programu pojawi się proste okno dialogowe które wyświetli główne menu. Wybierając opcję od 1 do 5 wybieramy interesującą nas funkcjonalność, a wybierając 6 wychodzimy z programu

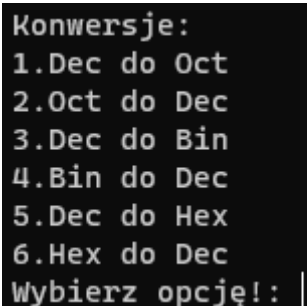


```
Witaj w kalkulatorze dla elektroników i informatyków

Wybierz opcje:
1.Konwersja na inne systemy
2.Kalkulator float
3.Kalkulator binarny
4.Kalkulator oktalny
5.Kalkulator hexadecymalny
6.Wyjdź
Wybierz opcję!: |
```

Rysunek 7-1. Główne menu programu

Jeśli wybierzemy konwersję pojawi się nowe menu służące do wyboru interesujących nas systemów liczbowych. Wybierając opcję od 1 do 6 wybieramy odpowiednią konwersję.



```
Konwersje:
1.Dec do Oct
2.Oct do Dec
3.Dec do Bin
4.Bin do Dec
5.Dec do Hex
6.Hex do Dec
Wybierz opcję!: |
```

Rysunek 7-2. Menu konwersji

Następnie podajemy wartość którą chcemy skonwertować. Po pomyślnej konwersji program wyświetli wynik a my wrócimy do menu głównego.

```
Konwersje:
1.Dec do Oct
2.Oct do Dec
3.Dec do Bin
4.Bin do Dec
5.Dec do Hex
6.Hex do Dec
Wybierz opcję!: 1
Podaj liczbę do konwersji: 23
27
Wybierz opcję:
1.Konwersja na inne systemy
2.Kalkulator float
3.Kalkulator binarny
4.Kalkulator oktalny
5.Kalkulator hexadecymalny
6.Wyjdź
Wybierz opcję!: |
```

Rysunek 7-3. Wybór oraz efekt konwersji

Jeśli wybierzemy menu któregoś z kalkulatorów program zapyta nas o podanie liczby, tu wpisujemy pierwszą liczbę na której mają zostać wykonane obliczenia. Następnie po wciśnięciu enter zostaniemy zapytani o operator. Za pomocą operatora wybieramy operacje algebraiczną którą chcemy wykonać.

Dostępne operatory

+ - dodawanie

- - odejmowanie

* - mnożenie

/- dzielenie

! - silnia

^ - potęga

P – pierwiastek

Po wybraniu operatora i wciśnięciu enter należy podać drugą liczbę dla której ma zostać użyta w obliczeniach. Po wybraniu liczby program wyświetli wynik i pokaże główne menu.

```
Wybierz opcje:
1.Konwersja na inne systemy
2.Kalkulator float
3.Kalkulator binarny
4.Kalkulator oktalny
5.Kalkulator hexadecymalny
6.Wyjdź
Wybierz opcję!: 2
Podaj pierwszą liczbę: 123
Podaj operator (+,-,*,/,!,^,P(pierwiastek): +
Podaj pierwszą liczbę: 23
Twój wynik to: 146

Wybierz opcje:
1.Konwersja na inne systemy
2.Kalkulator float
3.Kalkulator binarny
4.Kalkulator oktalny
5.Kalkulator hexadecymalny
6.Wyjdź
Wybierz opcję!: |
```

Rysunek 7-3. Działanie kalkulatora

8. Metodologia rozwoju i utrzymania systemu (*system maintenance and deployment*)

W czasie tworzenia projektu stosowano metodologie Agile. Szczególną uwagę zwrócono na element projektowania oraz dynamiczne planowanie szukając nowy rozwiązań i pomysłów ciągle doskonaląc projekt.

W pierwszym kroku, przygotowano wstępne konspekty algorytmów i funkcji, które miały być zaimplementowane do naszego projektu kalkulatora z konwerterem. Ten etap pomógł w określeniu podstawowych celów oraz problemów projektu.

Ustalono również wymagania sprzętowe oraz określono kolejność tworzenia poszczególnych elementów funkcji. Utworzono także plik CMake, który zawierał podstawowe parametry i był sukcesywnie uzupełniany o nowo tworzone klasy.

Następnie przystąpiono do implementacji hierarchii klas w kodzie źródłowym, co pozwoliło na realizację postanowionych celów oraz rozwiązanie problemów. Kolejnym krokiem było stworzenie klas konwerterów oraz przeprowadzenie pierwszych testów, które pomogły w weryfikacji poprawności działania.

Implementacja pełnego kodu była kolejnym etapem, na który składało się tworzenie algorytmów, funkcji i integracja wszystkich elementów projektu. W trakcie kolejnych testów systematycznie wprowadzano usprawnienia oraz dodawano nowe funkcje, co przyczyniło się do doskonalenia projektu.

Całość procesu rozwijania projektu prowadzona była w myśl metodologii Agile , gdzie każdy etap był analizowany i ulepszany na podstawie zdobytych doświadczeń, błędów oraz pomysłów.

Projekt można rozwinąć o kolejne systemy liczbowe, działania algebraiczne oraz operatory.

Bibliografia

- [1] Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
- [2] <https://www.geeksforgeeks.org/program-for-hexadecimal-to-decimal/>