

REV	DATA	ZMIANY	OSOBA
0.0	18.12.2021	<i>Konceptja projektu, hierarchia, system klas</i>	<i>Jakub Guza (guzajakub@student.agh.edu.pl)</i>
0.1	25.12.2021	<i>Napisanie pierwszej klasy E_Calc</i>	<i>Jakub Guza (guzajakub@student.agh.edu.pl)</i>
0.2	28.12.2021	<i>Obsługa menu, wybory dla użytkownika</i>	<i>Jakub Guza (guzajakub@student.agh.edu.pl)</i>
0.3	03.01.2021	<i>Gotowe wszystkie funkcje do konwersji między systemami liczbowymi</i>	<i>Jakub Guza (guzajakub@student.agh.edu.pl)</i>
0.4	07.01.2021	<i>Gotowy kalkulator z podstawowymi działaniami matematycznymi</i> <i>LAST VERSION</i>	<i>Jakub Guza (guzajakub@student.agh.edu.pl)</i>

KALKULATOR DLA ELEKTRONIKÓW W C++ PRZY UŻYCIU OOP.

Autor: Jakub Guza

Akademia Górniczo-Hutnicza

Kraków (C) 2021

SPIS TREŚCI

1. WSTĘP	4
2. FUNKcjONALNOŚĆ (<i>FUNCTIONALITY</i>)	5
3. PROJEKT TECHNICZNY (<i>TECHNICAL DESIGN</i>)	6
DIAGRAM UML	6
KLASY ORAZ FUNKCJE ZAWARTE W PROJEKCIE	7
4. OPIS REALIZACJI (<i>IMPLEMENTATION REPORT</i>)	9
5. PODRĘCZNIK UŻYTKOWNIKA (<i>USER'S MANUAL</i>)	10
6. OPIS WYKONANYCH TESTÓW (<i>TESTING REPORT</i>)	13

LISTA OZNACZEŃ

BIN	System binarny
HEX	System heksadecymalny
DEC	System decymalny
OOP	Object-Oriented Programming
CL	Command line

1. WSTĘP

Dokument dotyczy opracowania programu, który będzie służył jako kalkulator dla elektroników. Celem tego programu jest możliwość szybkiej kalkulacji potrzebnych nam wartości, przełączanie systemów, działanie na liczbach zespolonych.

Możliwe jest również wykonywanie działań, jako wynik dostajemy wartości w każdym systemie tj. hex bin oraz dec.

Podstawowe założenia projektu:

1. Przygotowanie klasy kalkulatora oraz metod konwersji między systemami liczbowymi.
2. Przygotowanie funkcji oraz obsługi „zadań” użytkownika, czyli menu oraz interakcja poprzez CL.
3. Napisanie klasy Complex i zintegrowanie jej z kalkulatorem, by móc obliczać również liczby zespolone.
4. Opracowanie obsługi błędów (tj. złych inputów wprowadzanych przez użytkowników)
5. Nauka i poznawanie języka C++.

Dzięki temu projektowi, możemy nauczyć się konwersji między systemami liczbowymi, pisania kodu w języku C++, zapoznanie się z technikami programowania obiektowego jako osoba która pisała projekt, oraz dać możliwość początkującym osobom, które poznają dopiero systemy liczbowe czy inne inżynierskie obliczenia, do poznania ich i pobawienia się możliwościami tego kalkulatora.

2. FUNKCJONALNOŚĆ (*FUNCTIONALITY*)

Głównym założeniem projektu, jest łatwa i szybka możliwość przeliczania systemów liczbowych jak i podstawowe działania matematyczne.

Dostępne funkcjonalności:

- zamiana systemów liczbowych

- * bin
- * hex
- * dec

- podstawowe operacje matematyczne

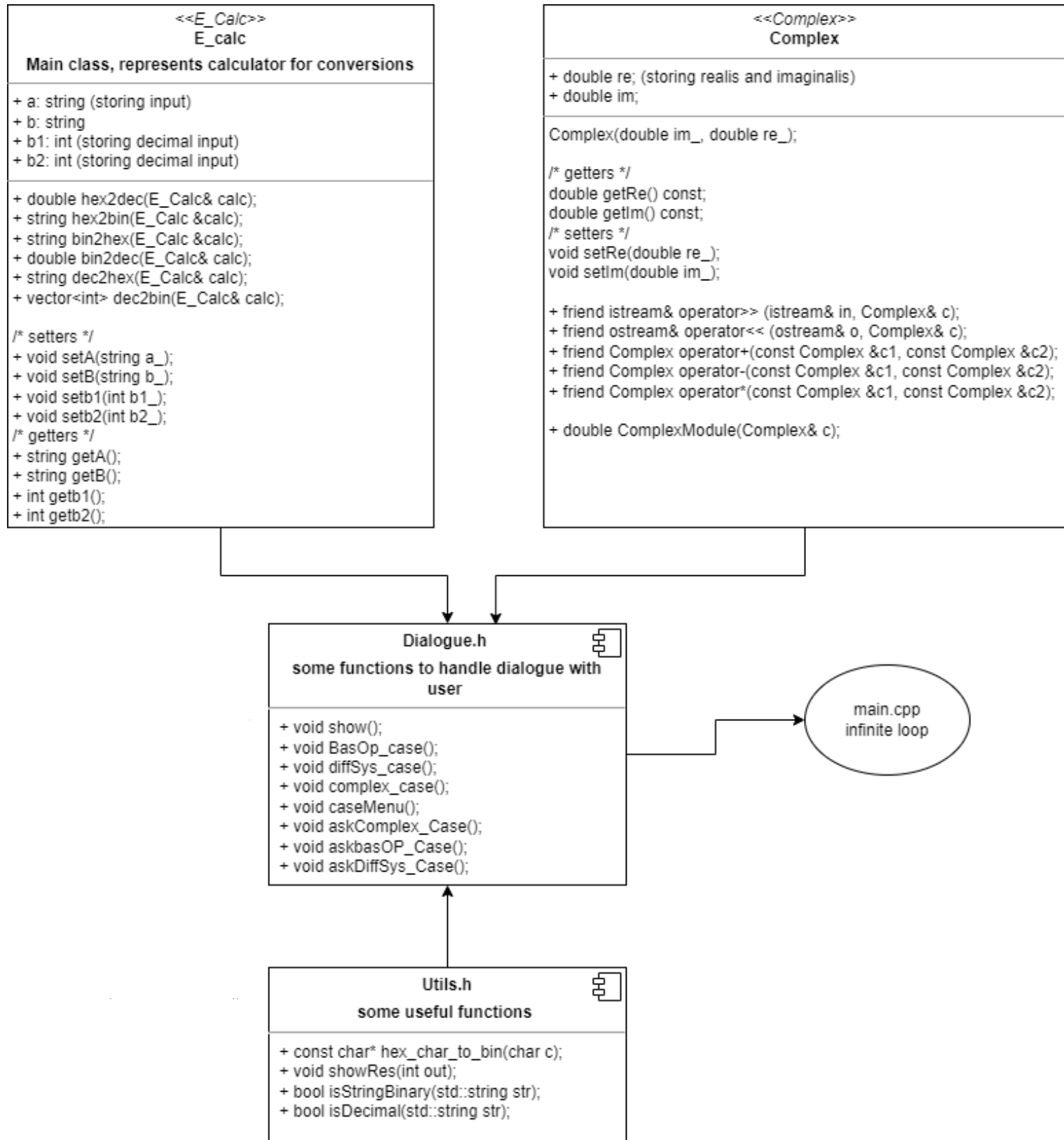
- * dodawanie
- * odejmowanie
- * dzielenie
- * mnożenie

- podstawowe operacje matematyczne na liczbach zespolonych

- * dodawanie
- * odejmowanie
- * obliczanie modułu
- * mnożenie

3. PROJEKT TECHNICZNY (*TECHNICAL DESIGN*)

DIAGRAM UML



KLASY ORAZ FUNKCJE ZAWARTE W PROJEKCIE

Klasa E_Calc

Zadaniem klasy *E_Calc* jest pobieranie wejściowych wartości do obiektu tejże klasy, oraz wykonywanie na nich obliczeń stricte związanych z systemami liczbowymi tj. konwersja na bin, hex, dec. Posiada ona cztery pola danych prywatnych, czyli wpisywane przez użytkownika dane wejściowe na których będą wykonywane obliczenia, setterów, getterów oraz funkcji zamieniających systemy liczbowe.

Klasa Complex

Zadaniem tej klasy, jest tworzenie obiektów typu Complex, czyli w skrócie liczb zespolonych. Posiada dwa pola danych prywatnych, czyli część rzeczywistą oraz urojoną. Posiada konstruktor, settery oraz gettery oraz przeciążone operatory wejścia, wyjścia, mnożenia, dodawania oraz odejmowania, tak aby poprzez klasyczne „strumienie” móc wyświetlać oraz modyfikować podane obiekty typu Complex.

Funkcje w headerze Dialogue.h

W tym pliku nagłówkowym umieściłem wszelakie potrzebne nam funkcje do interakcji z użytkownikiem, oraz do obsługi działań na obiektach klasy *E_Calc* oraz klasy *Complex*. Obsługuje ona odpowiednie switch-case'y, które wpisuje użytkownik, tak aby spełniały określoną rolę. Integrują one dwie klasy, tak aby mogły działać niezależnie od siebie.

Funkcje w headerze utils.h

W tym pliku nagłówkowym umieściłem kilka potrzebnych funkcji służących do integracji moich modułów. Przykładowo funkcja `hex_char_to_bin(char c)`, która przyjmuje znak i zamienia go na 4bitową postać binarna.

Kolejnym przykładem funkcji jest `showRes(int out)`, która przyjmuje decymalne wejście, i pokazuje wynik w trzech systemach, hex bin oraz dec. Kolejnym przykładem funkcji jest funkcja `isStringBinary(string str)` która pobiera string i sprawdza czy jest liczbą binarną. Jest potrzebna do obsługi poprawności wejścia użytkownika.

Reszta dokumentacji dostępna w kodach źródłowych plików jako komentarzy.

Zgodnie z dobrymi zasadami projektowania w języku C++, projekt został podzielony na pliki nagłówkowe jak i pliki źródłowe.

Pliki nagłówkowe:

calc.h

complex.h

dialogue.h

utils.h

Pliki źródłowe:

calc.cpp

complex.cpp

dialogue.cpp

utils.cpp

4. OPIS REALIZACJI (*IMPLEMENTATION REPORT*)

Projekt został napisany oraz kompilowany na systemie Windows 10 Home – wersja 10.0.19042.

Użyte środowisko programistyczne to Microsoft Visual Studio 2019 Wersja 16.11.5, wykorzystany język to C++17.

Działanie i kompilacja zostały przeprowadzone na laptopie Lenovo Legion Y540-15IRH.

Drugą jednostką na której zostały przeprowadzone testy był komputer PC znajomego:

PC

Procesor: Intel Core i5-3550, System: Windows 10 Home – wersja 10.0.19042.

Testy oraz zamierzone działanie programu zostało zrealizowane poprawnie i bezbłędnie.

5. PODRĘCZNIK UŻYTKOWNIKA (*USER'S MANUAL*)

```
-----  
Welcome to electronics calculator!  
-----  
  
Please choose what you wanna do:  
1. Switch between different systems [hex, bin, dec]  
2. Perform basic operations like [+ , - , : , /]  
3. Perform basic operations on complex numbers  
4. Exit  
Choose option and press enter:
```

Zostaniemy przywitani ekranem startowym, następnie możemy wybrać jedną z czterech opcji:

1. Zamiana systemów liczbowych
2. Wykonywanie podstawowych operacji matematycznych oraz uzyskanie wyniku w każdym z trzech systemów liczbowych
3. Wykonywanie podstawowych operacji matematycznych na liczbach zespolonych.
4. Wyjście z programu

```

-----
Welcome to electronics calculator!
-----

Please choose what you wanna do:
1. Switch between different systems [hex, bin, dec]
2. Perform basic operations like [+ , - , : , /]
3. Perform basic operations on complex numbers
4. Exit
Choose option and press enter: 1

Please choose which system you wanna enter:
1. Hex
2. Bin
3. Dec
2
Please enter binary value: 1101
Hex: D
Dec: 13

That's your result! Tell what you wanna do next:
1. Back to main menu
2. I wanna calculate same option again!
3. Exit
    
```

Po wybraniu pierwszej opcji, możemy następnie zadeklarować jaki system wpisujemy, sprawdzamy np. bin, wpisujemy wartość binarną np. „1101” tak jak wyżej, i otrzymujemy wynik w systemie heksadecymalnym oraz decymalnym. Następnie możemy wybrać jedną z trzech wartości:

1. Powrót do menu głównego
2. Wykonanie kolejny raz tej samej operacji
3. Wyjście z programu

Po wybraniu drugiej opcji, przechodzimy w tryb „kalkulatora” i wpisujemy dwie liczby, następnie jaką operację chcemy na nich wykonać i otrzymujemy wynik w każdym zdefiniowanym systemie liczbowym .

```

-----
Welcome to electronics calculator!
-----

Please choose what you wanna do:
1. Switch between different systems [hex, bin, dec]
2. Perform basic operations like [+ , - , : , /]
3. Perform basic operations on complex numbers
4. Exit
Choose option and press enter: 2
You've chosen basic operations
Please enter first decimal number: 7
Please enter second decimal number: 6

Now specify an operation, type
+ to perform addition
- for subtraction
* for multiplication
/ for division (INPUTS ARE INTS! be careful)
-> +
Dec: 13
Hex: D
Bin: 1101
That's your result! Tell what you wanna do next:
1. Back to main menu
2. I wanna calculate same option again!
3. Exit
    
```

Po wybraniu drugiej opcji, przechodzimy w tryb „kalkulatora dla liczb zespolonych” i wpisujemy dwie liczby zespolone, następnie jaką operacje chcemy na nich wykonać i otrzymujemy w postaci liczby zespolonej lub jej modułu, po wybraniu odpowiedniej opcji „m”.

```
Choose option and press enter: 3
You've chosen operations on complex numbers!

Please enter first complex number:
Please set real part: 2
Please set imaginary part: 3

Please enter second complex number:
Please set real part: 4
Please set imaginary part: 4
Your numbers: C1 = 2+3j
               C2 = 4+4j

Now specify an operation, type
+ to perform addition
- for subtraction
* for multiplication
m for module for both given numbers
+

2+3j + 4+4j = 6+7j
That's your result! Tell what you wanna do next:

1. Back to main menu
2. I wanna calculate same option again!
3. Exit
```

Po każdym wykonanym działaniu jakie tylko oferuje kalkulator, możemy zawsze wrócić do menu głównego, do tego samego działania bądź wyjście z programu.

Należy wystrzegać się poprawnego wpisywania wartości, lecz nie powinno to być problemem bo zaimplementowana została obsługa błędnych wartości wejściowych.

6. OPIS WYKONANYCH TESTÓW (*TESTING REPORT*)

Kod usterki	Data	Autor	Opis	Stan
Warning C264551	04.01	JG	Arithmetic overflow, należało użyć static_cast<>	Naprawione
Warning C26819	04.01	JG	Unannotated fallthrough between switch labels Należało switch-case'y zamknąć w klamry	Naprawione
Warning C26495	05.01	JG	E_Calc::b1 is uninitialized	Odnutowany
Errorv1	06.01	JG	Crash programu po wprowadzeniu wartości innej niż binarna, przy przeliczaniu liczby binarnej na inną	Naprawione, obsłużone wejście za pomocą sprawdzenia, czy wszystkie podane znaki są 0 lub 1
Errorv2	06.01	JG	Crash programu po wprowadzeniu wartości innej niż hexadecymalna, przy przeliczaniu liczby hex na inną	Naprawione, obsłużone wejście za pomocą sprawdzenia, czy wszystkie podane znaki są z zakresu 0-F