

REV	DATA	ZMIANY
0.1	28.12.2023	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>
0.2	29.12.2023	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>
0.3	30.12.2023	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>
0.4	13.01.2024	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>
0.5	19.01.2024	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>
0.6	20.01.2024	<i>Michał Wilczak (mwilczak@student.agh.edu.pl)</i>

# KALKULATOR DLA ELEKTRONIKÓW/INFORMATYKÓW

Autor: Michał Wilczak  
Akademia Górniczo-Hutnicza

## **Spis treści**

---

1. Wstęp.....	str.4.
2. Wymagania systemowe (requirements).....	str.5.
3. Funkcjonalność (functionality).....	str.6.
4. Analiza problemu (problem analysis).....	str.7.
5. Projekt techniczny (technical design).....	str.8.
6. Klasa oraz jej najważniejsze funkcje.....	str.9.
7. Użyte biblioteki.....	str.10.
8. Wykorzystane przykłady.....	str.10.
9. Opis realizacji (implementation report).....	str.11.
10. Opis wykonanych testów (testing report).....	str.11.
11. Podręcznik użytkownika (user's manual).....	str.12.
Bibliografia.....	str.13.

## **Lista oznaczeń**

---

UML	Unified Modeling Language
-----	---------------------------

# 1. Wstęp

Dokument dotyczy projektu aplikacji kalkulatora dla elektroników/informatyków. Celem tego kalkulatora jest możliwość wykonania kilku podstawowych działań dla elektroników i informatyków wraz z możliwością przełączenia się na systemy: binarny, ósemkowy, szesnastkowy lub dziesiętny.

## **2. Wymagania systemowe (*requirements*)**

Podstawowe założenia projektu:

1. Przygotowanie dokumentacji projektu
2. Opracowanie architektury programu: potrzebne klasy i funkcje.
3. Napisanie kodu programu realizującego założone funkcjonalności.

### **3.Funkcjonalność (*functionality*)**

Kalkulator ma funkcjonalności takie, jak:

- wykonywanie działań matematycznych: dodawanie, odejmowanie, mnożenie i dzielenie,
- zamiana systemów liczbowych: dziesiętny, dwójkowy, ósemkowy i szesnastkowy – możliwość wyboru systemu liczbowego przed zamianą i po zamianie,

## 4. Analiza problemu (*problem analysis*)

Jesteśmy przyzwyczajeni do zapisywania liczb w systemie dziesiętnym, ale w elektronice i informatyce często wykorzystywane są systemy: dwójkowy, ósemkowy i szesnastkowy. Komputer jest urządzeniem cyfrowym, które do wykonywania operacji wykorzystuje dane zapisane w postaci dwóch stanów: stanu niskiego (0) oraz stanu wysokiego (1), co odpowiada dwóm symbolom wykorzystywanym w systemie dwójkowym. Systemy: ósemkowy i szesnastkowy również mają zastosowanie w tych dziedzinach. System ósemkowy wykorzystywany jest do przypisywania uprawnień w systemach operacyjnych Linux. Adresy IP w wersji 6 są zapisywane w postaci liczb w systemie szesnastkowym. Również adresy pamięci zapisywane są w systemie szesnastkowym.

Z tego powodu przydatnym narzędziem dla elektroników i informatyków jest kalkulator umożliwiający zamianę tych systemów liczbowych na naturalny dla człowieka system dziesiętny oraz na operacje w drugą stronę.

System dwójkowy na kolejnych pozycjach przyjmuje kolejne potęgi liczby 2 przemnożone przez cyfry 0 i 1, system ósemkowy przyjmuje kolejne potęgi liczby 8 przemnożone przez liczby z zakresu od 0 do 7, system dziesiętny kolejne potęgi liczby 10 przemnożone przez liczby z zakresu od 0 do 9, a system szesnastkowy kolejne potęgi liczby 16 przemnożone przez liczby z zakresu od 0 do 15 (10 jest reprezentowane przez A, 11 przez B, 12 przez C, 13 przez D, 14 przez E, a 15 przez F). Kolejne potęgi po przemnożeniu są dodawane i w wyniku tej operacji otrzymujemy liczbę zapisaną w danym systemie liczbowym.

## 5. Projekt techniczny (*technical design*)

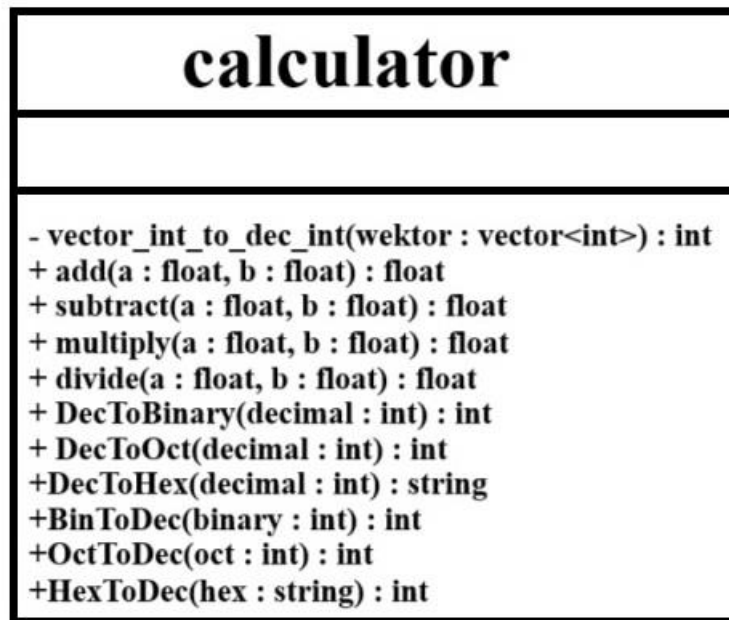


Diagram UML przedstawiający klasę `calculator` nie ma żadnych atrybutów, ale ma operacje: prywatną funkcję zamieniającą liczbę zapisaną jako `vector<int>` na liczbę typu `int` zapisaną w systemie dziesiętnym, publiczne funkcje realizujące działania matematyczne: dodawanie, odejmowanie, mnożenie i dzielenie oraz publiczne funkcje zamieniające liczby z systemu dziesiętnego na system binarny, ósemkowy i szesnastkowy oraz publiczne funkcje zamieniające liczby z systemu binarnego, ósemkowego i szesnastkowego na system dziesiętny.



## 6.Klasa oraz jej najważniejsze funkcje

```
17 //klasa zawierająca funkcje do wykonywania działań matematycznych oraz funkcje do zamiany systemów liczbowych
18
19 class calculator
20 {
21     float add(float a, float b) //funkcja realizująca dodawanie
22     float subtract(float a, float b) //funkcja realizująca odejmowanie
23     float multiply(float a, float b) //funkcja realizująca mnożenie
24     float divide(float a, float b) //funkcja realizująca dzielenie
25
26     int DecToBinary(int decimal) //wzorowano się na kodzie ze strony: https://www.geeksforgeeks.org/program-to-convert-decimal-to-binary/
27     { //funkcja zamieniająca lczbę z systemu dziesiętnego na system binarny
28
29     int DecToOct(int decimal) //wzorowano się na kodzie funkcji zamieniającej liczbę z systemu dziesiętnego
30     { //na system binarny
31
32     string DecToHex(int decimal) //wzorowano się na kodzie ze strony:https://www.geeksforgeeks.org/program-to-convert-decimal-to-hexadecimal/
33     { //funkcja zamieniająca liczbę z systemu dziesiętnego na system szesnastkowy
34
35     int BinToDec(int binary) //funkcja zamieniająca liczbę z systemu binarnego na system dziesiętny
36
37     int OctToDec(int oct) //funkcja zamieniająca liczbę z systemu ósemkowego na system dziesiętny
38
39     int HexToDec(string hex) //funkcja zamieniająca liczbę z systemu szesnastkowego na system dziesiętny
```

## 7. Użyte biblioteki

---

Do zaimplementowania kodu realizującego kalkulator wykorzystano następujące biblioteki:

- `iostream`,
- `cmath`,
- `limits`,
- `vector`,
- `string`,
- `algorithm`,

Wykorzystane funkcje pochodzące z wyżej wymienionych bibliotek:

- `std::numeric_limits` – pochodzi z biblioteki `<limits>`,
- `std::reverse` – pochodzi z biblioteki `<algorithm>`,
- `std::to_string` – pochodzi z biblioteki `<string>`,
- `pow` – pochodzi z biblioteki `<cmath>`,
- `std::vector` – pochodzi z biblioteki `<vector>`,
- `std::cin`, `std::cout` – pochodzą z biblioteki `<iostream>`,

## 8. Wykorzystane przykłady

---

Wykorzystano funkcje na maksymalną i minimalną wartość zmiennej typu `float` na podstawie filmu pod linkiem: <https://www.youtube.com/watch?v=7aZbYJ5UTC8&t=2480s> (minuta 41:20)

Wykorzystano funkcję na epsilon dla zmiennej typu `float` na podstawie filmu pod linkiem: <https://www.youtube.com/watch?v=7aZbYJ5UTC8&t=1495s> (minuta 24:36)

Do zaimplementowania funkcji zamieniającej liczbę z systemu dziesiętnego na system binarny wzorowano się na kodzie dostępnym pod linkiem: <https://www.geeksforgeeks.org/program-decimal-binary-conversion/>

Funkcję zamieniającą liczbę z systemu dziesiętnego na system ósemkowy zaimplementowano przez analogię do funkcji zamieniającej liczbę z systemu dziesiętnego na system binarny.

Do zaimplementowania funkcji zamieniającej liczbę z systemu dziesiętnego na system szesnastkowy wzorowano się na kodzie dostępnym pod linkiem: <https://www.geeksforgeeks.org/program-decimal-hexadecimal-conversion/>

## 9.Opis realizacji (*implementation report*)

Platforma testowa: komputer PC z systemem operacyjnym Windows 11, zintegrowane środowisko programistyczne: Microsoft Visual Studio 2022, Platform Toolset: Microsoft Visual Studio 2022 (v143), standard języka C++: ISO C++ 20 Standard.

## 10.Opis wykonanych testów (*testing report*)

Kod usterki	Data	Autor	Opis	Stan
	13.01.2024	Michał Wilczak	Wykonano testy poprawności działania kodu za pomocą testów jednostkowych i biblioteki do testów GoogleTest	Test zakończył się pomyślnie

## 11. Podręcznik użytkownika (*user's manual*)

### Test aplikacji w systemie Windows

---

```
Funkcjonalnosc:  
1. Kalkulator  
2. Zamiana systemow liczbowych  
Wybierz funkcjonalnosc: 1  
Dzialania matematyczne:  
1. Dodawanie  
2. Odejmowanie  
3. Mnozenie  
4. Dzielenie  
Wybierz dzialanie matematyczne: 1  
Podaj pierwsza liczbe: 2.44  
Podaj druga liczbe: 1.55  
Wynik: 3.99
```

Rys. 1. Przykład działania programu dla funkcjonalności działań matematycznych

```
Funkcjonalnosc:  
1. Kalkulator  
2. Zamiana systemow liczbowych  
Wybierz funkcjonalnosc: 2  
Dostepne systemy liczbowe:  
1. Dziesietny  
2. Dwojkowy  
3. Osemkowy  
4. Szesnastkowy  
Podaj system liczbowy przed zamiana: 1  
Podaj system liczbowy po zamianie: 2  
Podaj liczbe:  
10  
Liczba w systemie dwojkowym: 1010
```

Rys. 2. Przykład działania programu dla funkcjonalności zamiany systemów liczbowych

Po uruchomieniu programu użytkownik jest proszony o wybranie funkcjonalności. W przypadku wybrania funkcji kalkulatora, użytkownik jest proszony o wybór działania matematycznego. Po wyborze jednego z czterech: dodawania, odejmowania, mnożenia lub dzielenia użytkownik może wprowadzić dwie liczby, każdą z nich osobno potwierdzając klawiszem „Enter”. Kalkulator obsługuje liczby zmiennoprzecinkowe, więc aby program prawidłowo je zinterpretował należy zamiast znaku , (przecinek) używać znaku . (kropka).

W przypadku wybrania funkcji zamiany systemów liczbowych, użytkownik jest proszony o wybór systemu liczbowego przed zamianą oraz po zamianie. Następnie użytkownik może podać liczbę, którą chce zamienić pomiędzy wybranymi systemami liczbowymi.

## Bibliografia

- [1] Cyganek B.: Programowanie w języku C++. Wprowadzenie dla inżynierów. PWN, 2023.
- [2] Cyganek B. *A refresher on the floating-point computations and the standard library*, cpponea, 22.07.2020, online: <https://www.youtube.com/watch?v=7aZbYJ5UTC8&t=1495s>,  
dostęp: 29.12.2023
- [3] *Program for Decimal to Binary Conversion*, GeeksforGeeks, 18.10.2023, online:  
<https://www.geeksforgeeks.org/program-decimal-binary-conversion/>, dostęp: 29.12.2023
- [4] *Program for decimal to hexadecimal conversion*, GeeksforGeeks, 22.12.2023, online:  
<https://www.geeksforgeeks.org/program-decimal-hexadecimal-conversion/>,  
dostęp: 29.12.2023