

# **KALKULATOR PROGRAMISTY**

Autor: Adam Zielina  
Akademia Górniczo-Hutnicza

Kraków (C) 2022

## 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> ).....	8
5. OPIS REALIZACJI ( <i>IMPLEMENTATION REPORT</i> ).....	9
7. OPIS WYKONANYCH TESTÓW ( <i>TESTING REPORT</i> ) - LISTA BUGGÓW, UZUPEŁNIENI, ITD.	12

## **Lista oznaczeń**

---

API	Application Programming Interface
OOD	Object-Oriented Design
OOP	Object-Oriented Programming
STL	Standard Template Library
BIN	Binary system
HEX	Hexadecimal system
DEC	Decimal system

# 1. Wstęp

Dokument zawiera dokumentację techniczną programu kalkulatora programisty. Kalkulator programisty jest kalkulatorem umożliwiającym wykonywanie konwersji między różnymi systemami liczbowymi, jak również wykonywanie prostych operacji arytmetycznych i logicznych dla danych systemów. Kalkulator taki może być przydatny dla programistów wykonujących operacje bitowe w szczególności dla programistów systemów wbudowanych gdzie tego typu działania mają często miejsce.

## 1a. Wymagania systemowe (*requirements*)

Podstawowe założenia projektu:

1. Stworzeniu hierarchii zależności
2. Utworzenie klasy odpowiedzialnej za konwersje, której zmienne reprezentują wyniki konwersji liczbowej dla danych systemów
3. Utworzenie pliku nagłówkowego z deklaracjami funkcji oraz pliku źródłowego z definicjami tych funkcji.
4. Stworzenie algorytmów konwersji oraz obliczeń arytmetyczno-logicznych w definicjach
5. Stworzenie funkcji odpowiadających za dostarczenie odpowiednich danych do funkcji stricte algorytmicznych
6. Stworzenie prostej struktury komunikacji z użytkownikiem

## **2. Funkcjonalność (*functionality*)**

### 1. Podstawowe działania arytmetyczne:

- dodawanie
- odejmowanie
- mnożenie
- dzielenie

dostępne dla 2 dowolnych liczb podanych w dowolnym systemie liczbowym

### 2. Podstawowe działania logiczne:

- AND
- OR
- XOR
- NOT

dostępne dla 2 dowolnych liczb podanych w dowolnym systemie liczbowym z wyjątkiem NOT, który działa dla jednej liczby.

### 3. konwersja liczb na inne systemy liczbowe:

- BIN
- HEX
- DEC

Możliwość konwersji liczby z jednego z podanych systemów liczbowych na inne

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

Problem konwersji można podzielić na 3 zasadnicze części

1. konwersja na DEC
2. konwersja z DEC
3. pozostałe konwersje

1. Konwersja na DEC polega na przemnażaniu każdej cyfry przez podstawę danego systemu do potęgi równej pozycji na której cyfra się znajduje w liczbie, a następnie zsumowaniu wszystkich wyników.

Np.

$$53a_{(16)} = 5 \cdot 16^2 + 3 \cdot 16^1 + 10 \cdot 16^0 = 1338_{(10)}$$

$$1001_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^0 = 9_{(10)}$$

2. konwersja z DEC polega na dzieleniu naszej liczby przez podstawę systemu na który chcemy dokonać konwersji, otrzymana z tego działania reszta jest jedną z cyfr docelowej liczby poczynając od najmniej znaczącej.

$$100_{(10)}$$

$$30_{(10)}$$

$$100 / 2 = 50 R0$$

$$30 / 16 = 1 R E$$

$$50 / 2 = 25 R0$$

$$1 / 16 = 0 R1$$

$$25 / 2 = 12 R1$$

$$12 / 2 = 6 R0$$

$$6 / 2 = 3 R0$$

$$3 / 2 = 1 R1$$

$$1 / 2 = 0 R1$$

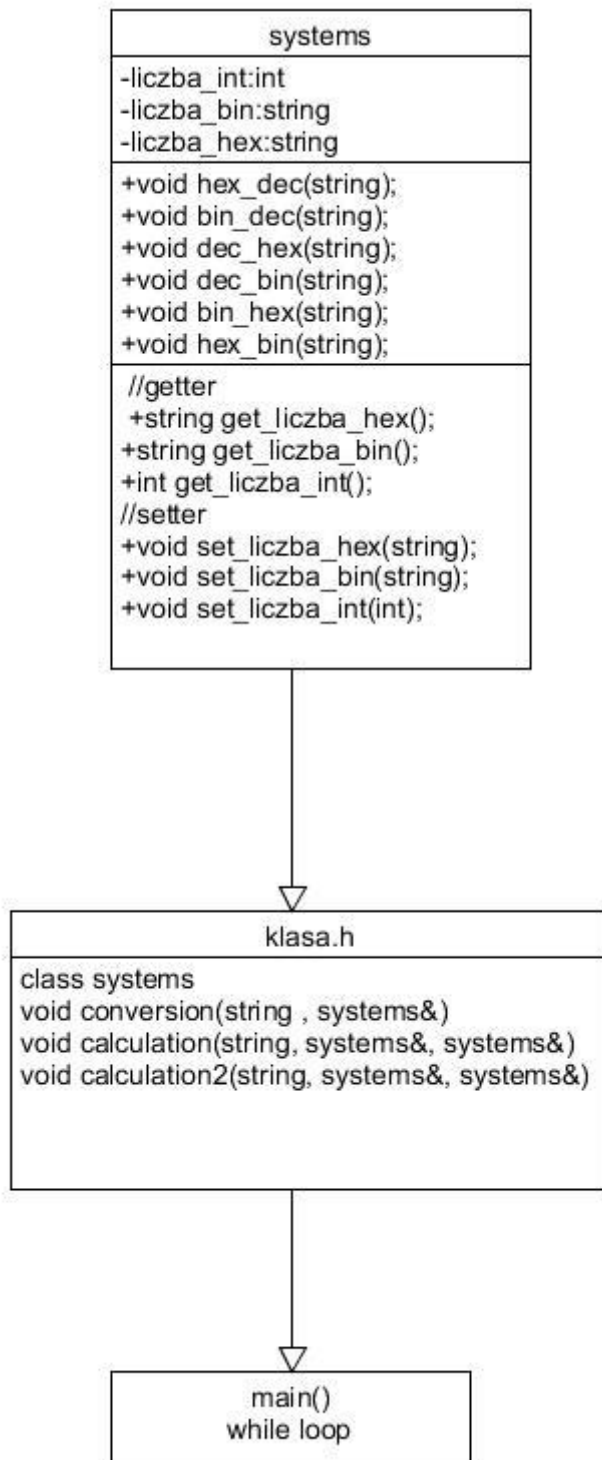
$$1100100_{(2)}$$

$$1E_{(16)}$$

3. Przy konwersjach z HEX na BIN i na odwrót można wykorzystać właściwość że pojedyncza cyfra HEX jest kodowana jako pojedyncza liczba 4 bitowa o wartości od 0 do 255.

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

Diagram UML





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

IDE: Microsoft visual studio 2019 w wersji 16.11

Język: C++

Nazwa systemu operacyjnego: Microsoft Windows 10 Home 10.0.19042 Kompilacja 19042

Procesor Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, 2594 MHz, Rdzenie: 4, Procesory logiczne: 8

Typ systemu x64-based PC

Model systemu ASUS X550JX

## 6. Instrukcja obsługi (manual)

Uruchamiając program przebywamy w głównym menu, które pozwala nam wybrać jedną z trzech funkcjonalności kalkulatora.

```
C:\Users\asus\Desktop\jpo\projekt\kalkulator\Debug\kalkulator.exe
wybierz tryb pracy
obliczenia arytmetyczne -> "calc" konwersja systemow -> "conv" lub obliczenia logiczne -> "logic"
```

### 1. tryb conv

Jest to tryb który pozwala nam na konwersję liczb z jednego systemu liczbowego na inne

Jako input należy wpisać liczbę wraz z przedrostkiem określającym jej system.

Aby wpisać liczbę szesnastkową 0x np. 0xf

Aby wpisać liczbę binarną 0b np. 0b1

Aby wpisać liczbę dziesiętną brak przedrostka np. 1234

Aby wrócić do menu należy wpisać 'exit'

```
podaj liczbę w postaci 'xliczba'
przed wpisaniem liczby wybierz system '0x' '0b' lub 'NULL'
przykład: 0xa lub 0b1010 lub 1234
[wpisz 'exit' aby wyjść]
```

### 2. tryb calc

Tryb pozwalający na wykonanie podstawowych działań arytmetycznych na liczbach w różnych systemach.

Jako input należy wpisać liczbę wraz z przedrostkiem określającym jej system, znak działania , kolejną liczbę wraz z przedrostkiem.

Dostępne znaki to

Dodawanie – znak '+'

Odejmowanie – znak '-'

Mnożenie – znak '\*'

Dzielenie – znak '/'

Aby wrócić do menu należy wpisać 'exit'

```
prosze wpisac rownanie w postaci "liczba1znakliczba2"
przed wpisaniem liczby wybierz system '0x' '0b' lub 'NULL'
znak to '+' '-' '*' '/'
przyklad: 0xa+0b1111
[wpisz 'exit' aby wyjsc]
```

### 3. tryb logic

Tryb pozwalający na wykonanie podstawowych działań logicznych na liczbach w różnych systemach. Jako input należy wpisać liczbę wraz z przedrostkiem określającym jej system, znak działania , kolejną liczbę wraz z przedrostkiem.

Dostępne znaki to

AND – znak '&'

OR – znak '|'

XOR – znak '^'

Lub aby dokonać negacji liczby wpisać ją wraz ze znakiem tyldy na końcu

NOT – znak '~'

Aby wrócić do menu należy wpisać 'exit'

```
prosze wpisac rownanie w postaci "liczba1znakliczba2" lub "liczba1~" aby zanegowac
przed wpisaniem liczby wybierz system '0x' '0b' lub 'NULL'
znak to '&' '|' '^'
przyklad: 0xa|0b1111 lub 0xa~
[wpisz 'exit' aby wyjsc]
```

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

Testy:

Testowanie funkcjonalności dodawania:

Dodawanie działa gdy obie liczby są decymalne, dla innych systemów nie pojawia się wynik.

Rozwiązano:

Liczby składały się z formatu przedrostek systemu + liczba, w celu odpowiedniej konwersji a następnie dodawania należało usunąć przedrostek systemu.

Brak konwersji 0 dziesiętnego na inne systemy

Rozwiązano:

Dodano wyrażenie warunkowe na specjalny przypadek dla 0 przed pętlą konwersji

### Lista błędów

Ważność	Kod	Opis
Błąd	C7529	wiele deklaratów using wymaga co najmniej elementu „/std:c++17”
Błąd (aktywny)	E0413	brak odpowiedniej funkcji konwersji elementu "std::string" na "const char **"
Błąd	LNK2019	nierozpoznany symbol zewnętrzny "public: void __thiscall systems::bin_dec(class std::basic_string<char,struct std::char_traits<char>,class std::allocator<char> >) (?bin_dec@systems@@QAEXV?\$basic_string@DU?\$char_traits@D@std@@V?\$allocator@D@2@@std@@@Z) przywołany w funkcji "void __cdecl conversion(class std::basic_string<char,struct std::char_traits<char>,class std::allocator<char> >,class system &)" (?conversion@@YAXV?\$basic_string@DU?\$char_traits@D@std@@V?\$allocator@D@2@@std@@AAVsystems@@

Ważność	Kod	Opis	Projekt	Plik	Wiersz	Stan pominięcia
Ostrzeżenie	C4018	"<"; niezgodność typu ze znakiem/bez znaku	kalkulator	C:\Users\asus\Desktop\jpo\projekt\kalkulator\klasa.cpp	10	uwzględniono