

| REV | Data | Zmiany |
|-------|------|---|
| 1.0.0 | | <i>Michał Berdzik</i> (<i>michalberdzik97@gmail.com</i>) |
| | | |
| | | |

Kalkulator

Autor: Michał Berdzik
Akademia Górniczo-Hutnicza
Im. Stanisława Staszica



Kraków (C) 2018

| | |
|---|-----------|
| Lista oznaczeń | 2 |
| Wstęp | 3 |
| Wymagania systemowe | 3 |
| Funkcjonalność | 3 |
| Analiza problemu | 4 |
| Projekt techniczny | 5 |
| Opisy klas | 5 |
| Klasa Main Page | 5 |
| Klasa Calculator | 6 |
| Klasa Converter | 7 |
| Klasy Conversion | 8 |
| Klasa NumbSysInterpreter | 9 |
| Klasa Infix2Postfix | 10 |
| Klasa calcRPN | 10 |
| Algorytm działania klasy Infix2Postfix | 11 |
| Algorytm działania klasy calcRPN | 12 |
| Diagram klas | 13 |
| Opis realizacji | 13 |
| Podręcznik użytkownika | 14 |
| Opis aplikacji | 14 |
| Metodologia rozwoju i utrzymania systemu | 16 |

Lista oznaczeń

| | |
|-----------|---|
| RPN (ONP) | Reverse Polish Notation (Odwrotna Notacja Polska) |
| NSI | Number System Interpreter |

Wstęp

Dokument dotyczy opracowania aplikacji kalkulatora wraz z wbudowanym konwerterem jednostek. Program został wykonany jako Uniwersalna Aplikacja Windows, więc może być obsługiwana na różnych urządzeniach.

Wymagania systemowe

- Windows 10 lub nowszy
- Aplikacja jest przystosowana do użytku na różnych urządzeniach (PC, smartfon, XBOX)

Funkcjonalność

Do funkcjonalności programu możemy zaliczyć:

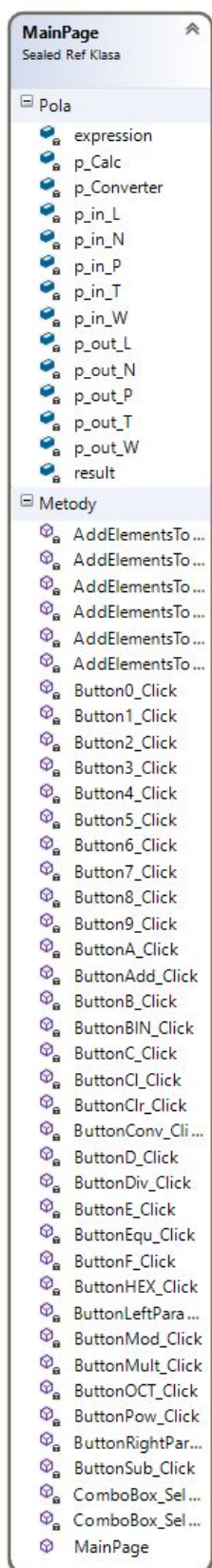
- Łatwy sposób wprowadzania wyrażenia
- Realizacja obliczeń z zachowaniem kolejności wykonywania działań
- Automatyczna konwersja systemów - możliwość wykonywania działań na liczbach z różnych systemów liczbowych
- Konwersja jednostek długości, wagi, wartości elektronicznych (mocy, wzmocnienia), temperatury oraz systemów liczbowych
- Zapamiętywanie wartości ostatnio obliczonego wyrażenia
- Możliwość przerywania obliczeń w dowolnym momencie

Analiza problemu

- **Implementacja algorytmu do RPN:** Algorytm potrzebny, aby kalkulator mógł zrealizować działania zachowując ich kolejność.
- **Stworzenie klasy do obliczenia RPN:** Program liczący wartość podanego wyrażenia RNP.
- **Stworzenie algorytmu do interpretacji NSI:** Algorytm analizuje podane mu wyrażenie, rozpoznając typ systemu liczbowego i odpowiednio przekształcając wyrażenie dopasowuje je aby móc je przekazać do algorytmu RPN.
- **Klasa konwertera:** Klasa pozwalająca na konwersję różnych typów jednostek.

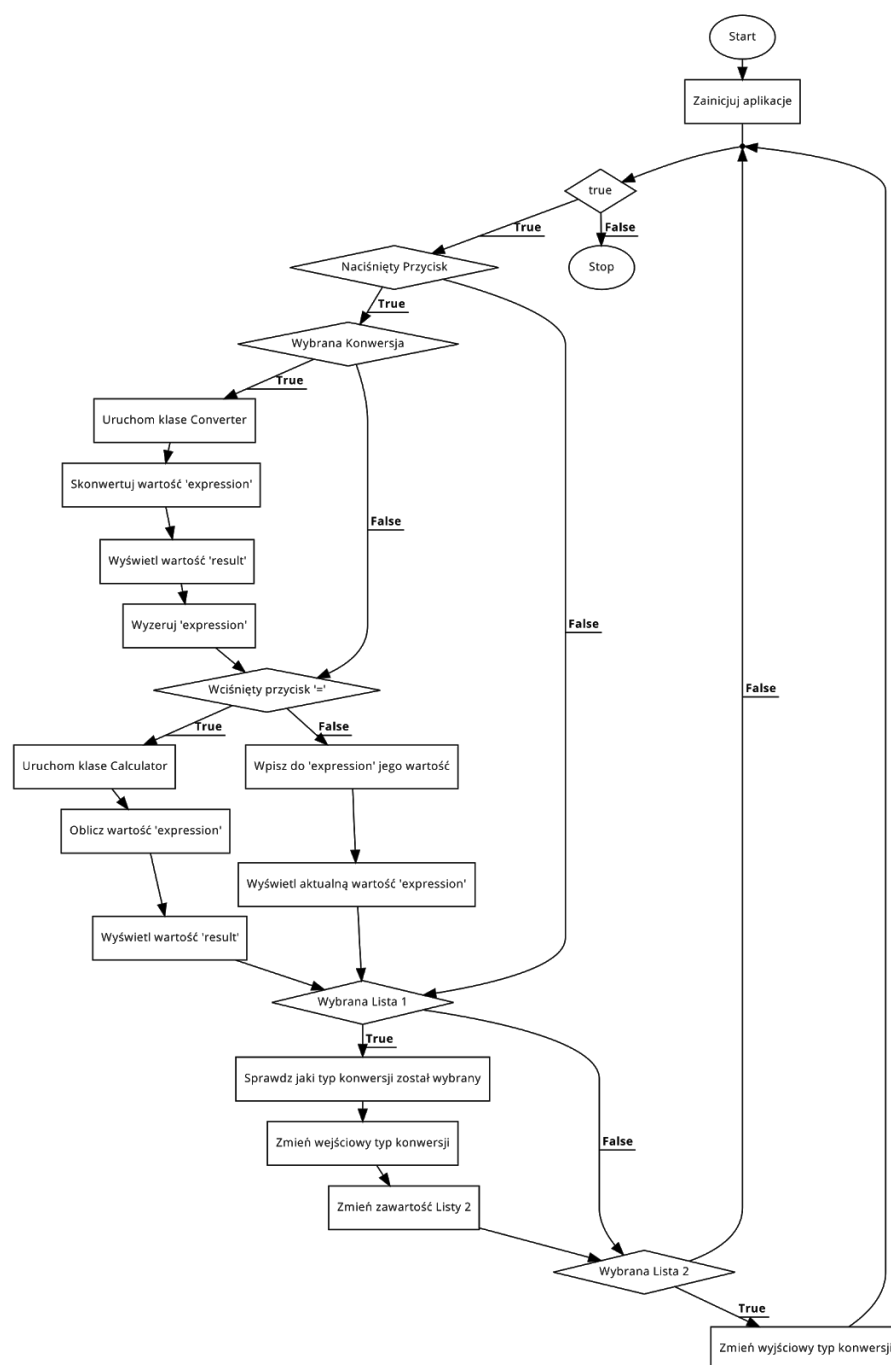
Projekt techniczny

Opisy klas

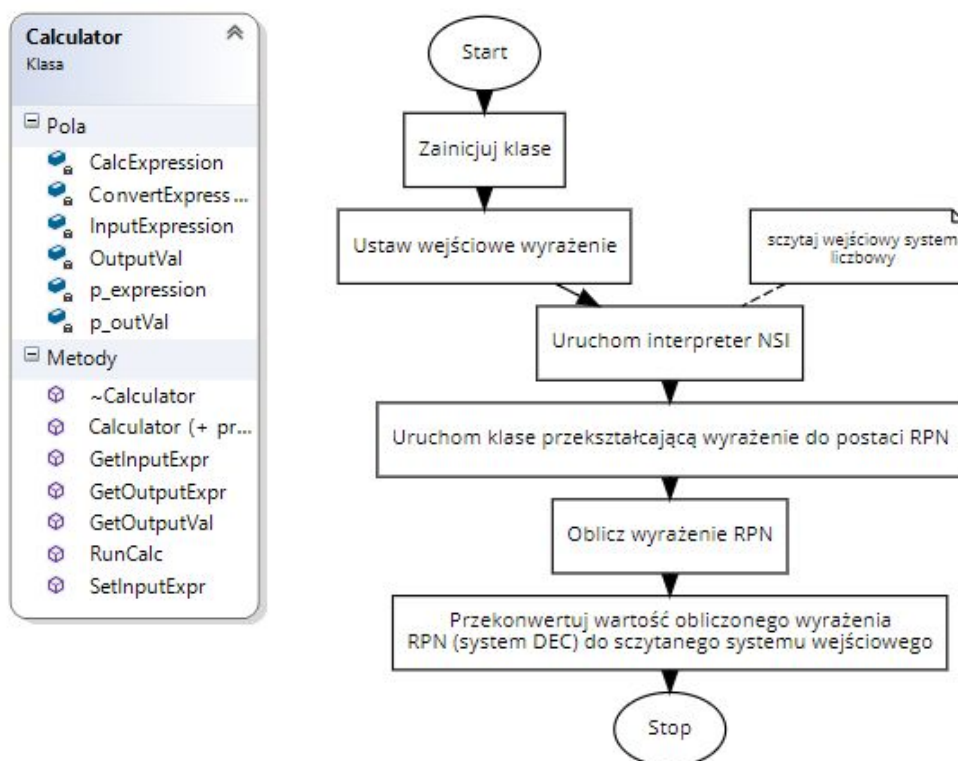


Klasa Main Page

- Klasa odpowiedzialna za działanie całej aplikacji
- Zawiera metody do obsługi przycisków użytych w aplikacji takich jak przyciski do wyboru cyfr oraz przyciski specjalne.

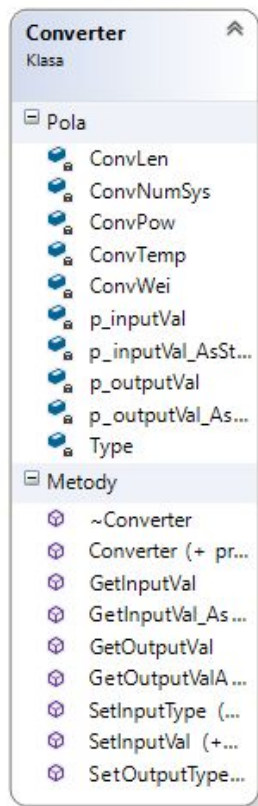


Klasa Calculator



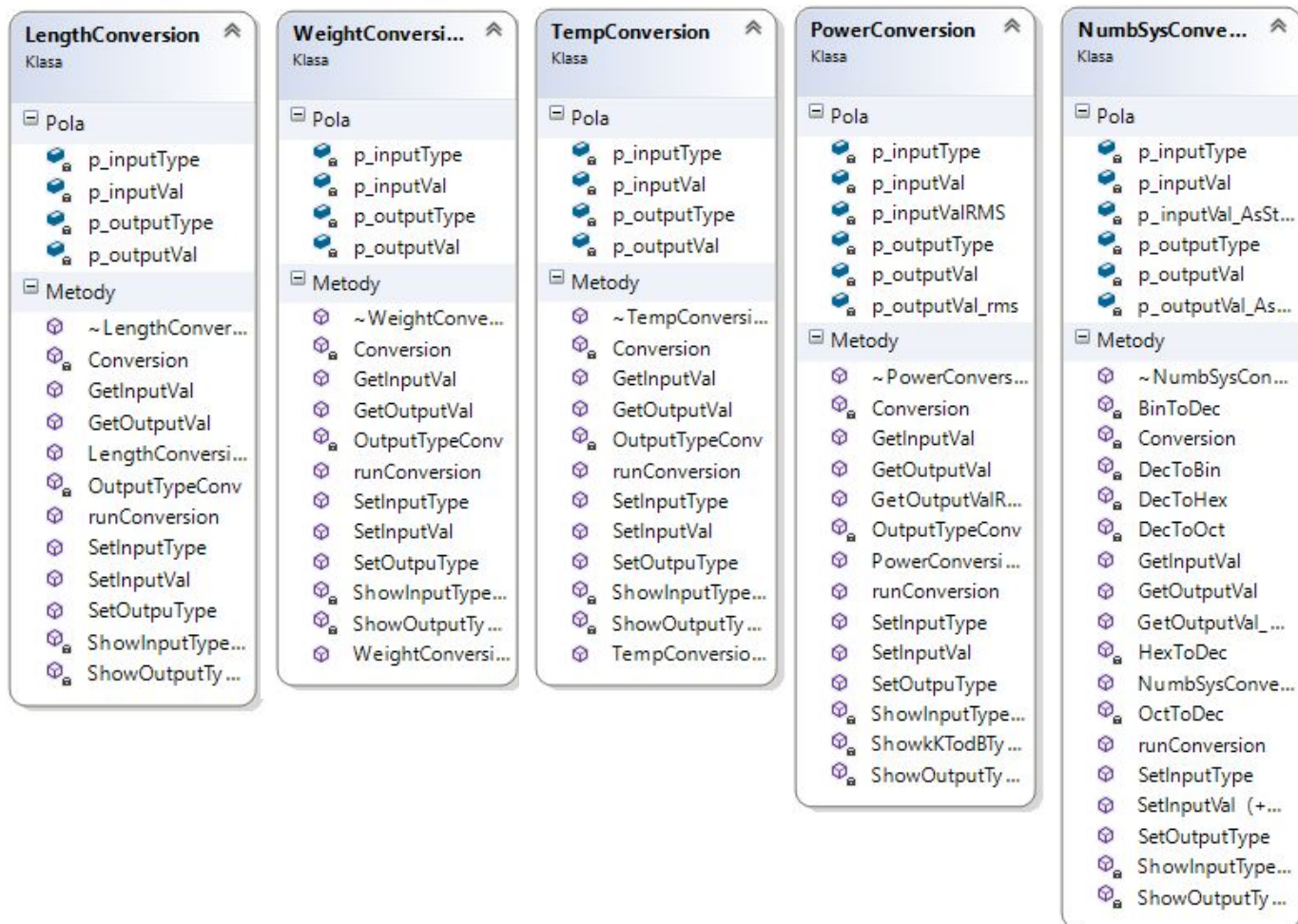
- Klasa Calculator korzysta z klas calcRPN, Infix2Postfix, NumbSysInterpreter oraz NumbSysConverter
- Stanowi uproszczenie wykonywania obliczeń przez hermetyzację całego procesu obliczeń do jednej klasy.

Klasa Converter

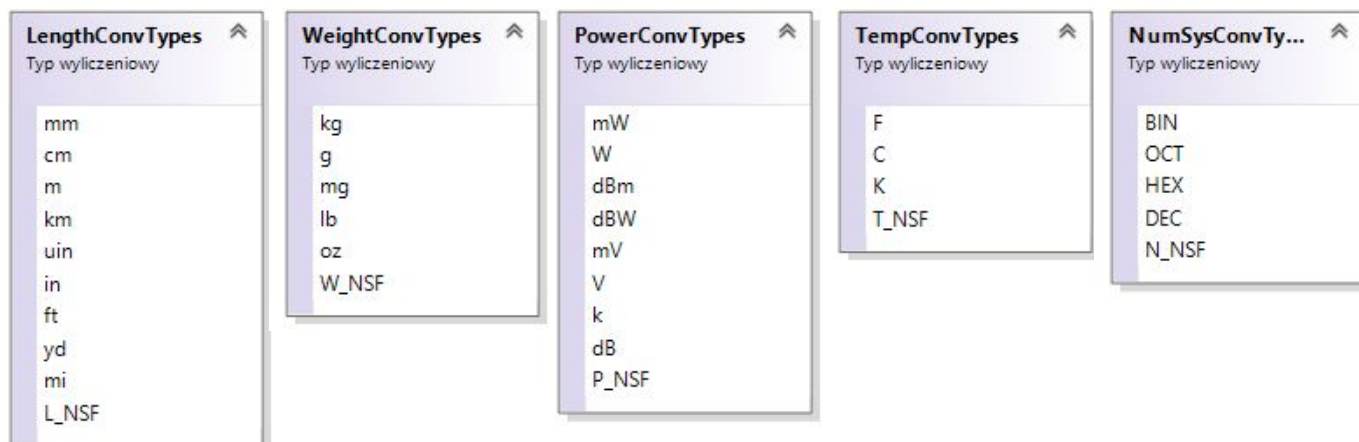


- Klasa Converter łączy w sobie funkcjonalności klas do konwersji długości, wagi, mocy, temperatury oraz systemów liczbowych
- Aby zapewnić bezproblemowe działanie bez błędów konwersji (typu konwersja kg -> mV) zastosowałem różne wersje konstruktorów, aby zapewnić dopasowanie wejściowego i wyjściowego typu konwersji. Ponadto zastosowałem tryb wyliczeniowy *enum*, aby móc rozpoznawać jaki typ konwersji jest ustalony, dzięki czemu przy ustawianiu wartości *SetInputVal()* program wie jaką konwersję zrealizować.

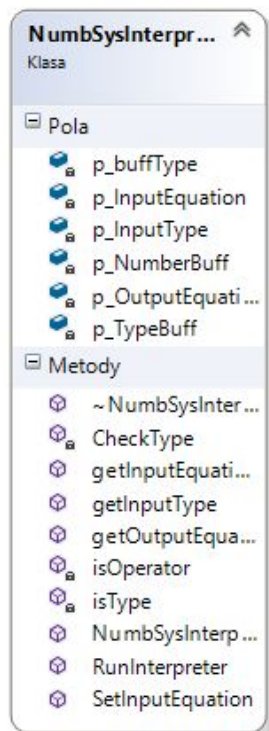
Klasy Conversion



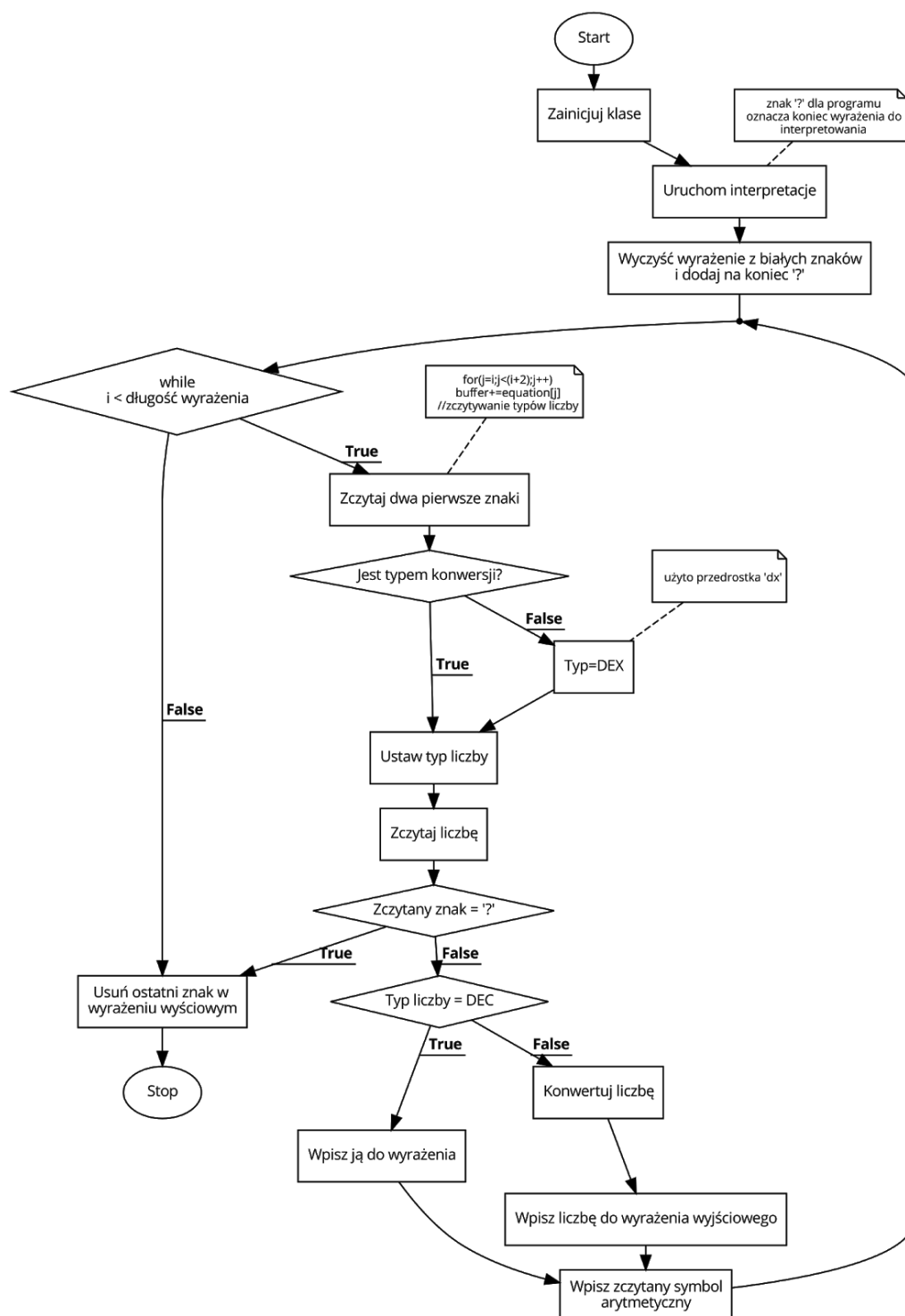
- Zasada działania tych klas jest prosta i opiera się na rozpoznaniu typu wejściowego i wyjściowego a następnie wartość wejściowa jest odpowiednio mnożona lub dzielona, aby uzyskać skonwertowaną wartość.
- W każdej z klas jest zastosowany tryb wyliczeniowy *enum* do określania typów konwersji
- W każdym typie *enum* występuje element NSF oznaczający typ nie określony (Not Specified)



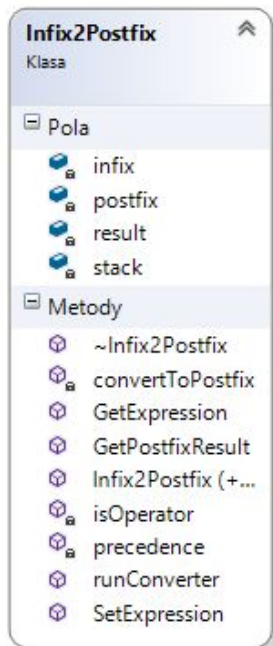
Klasa NumbSysInterpreter



- Klasa zrealizowana jako wzorzec projektowy 'Interpreter' służąca do interpretacji wyrażenia matematycznego, gdzie elementy są podane w różnych systemach liczbowych
- Klasa dokonuje interpretacji wszystkich elementów wyrażenia i konwertuje je do typu DEC

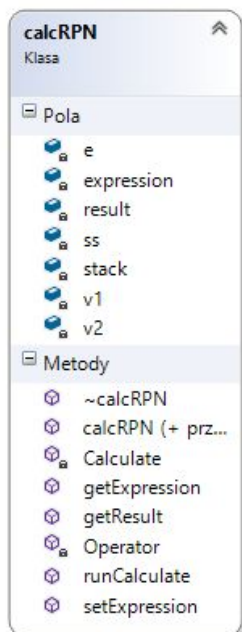


Klasa Infix2Postfix



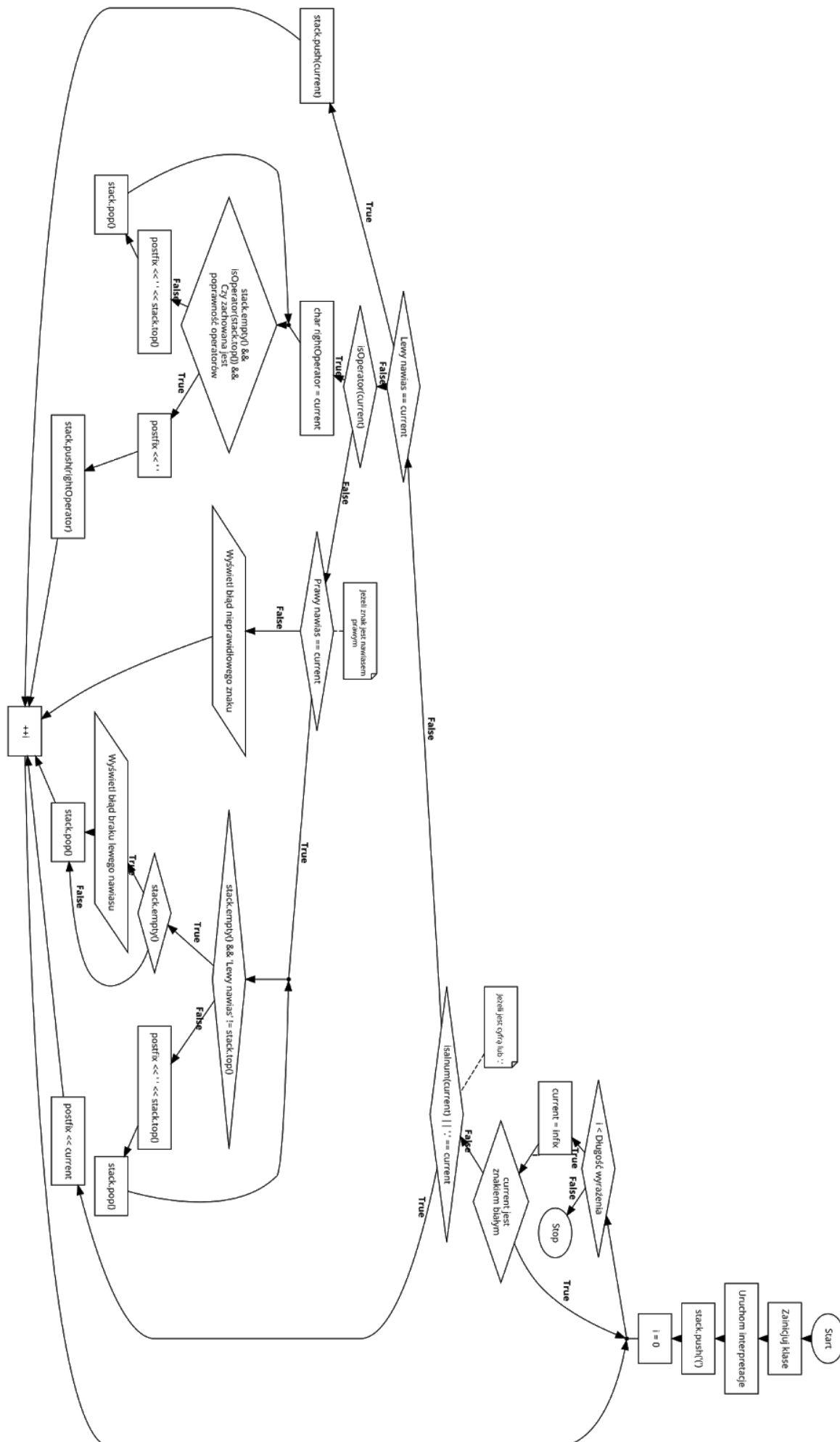
- Klasa zrealizowana jako wzorzec projektowy 'Interpreter' służąca do interpretacji wyrażenia matematycznego. Klasa przekształca wyrażenie podane jako typ Infix do wyrażenia typu Postfix (przekształca wyrażenie matematyczne do postaci RPN)
- Schemat algorytmu działania klasy jest dostępny na stronie 12

Klasa calcRPN



- Klasa przyjmuje wyrażenie RPN i na jego podstawie liczy jego wartość, którą zwraca jako *double* lub *string*. Klasa obsługuje takie operatory jak: +, -, *, /, ^, % (jako pierwiastek) i \$ (jako eksponenta).
- Możliwość dodania innych operatorów realizujących inne zaprogramowane im funkcje.
- Schemat algorytmu działania klasy jest dostępny na stronie 13

Algorytm działania klasy Infix2Postfix



Algorytm działania klasy calcRPN

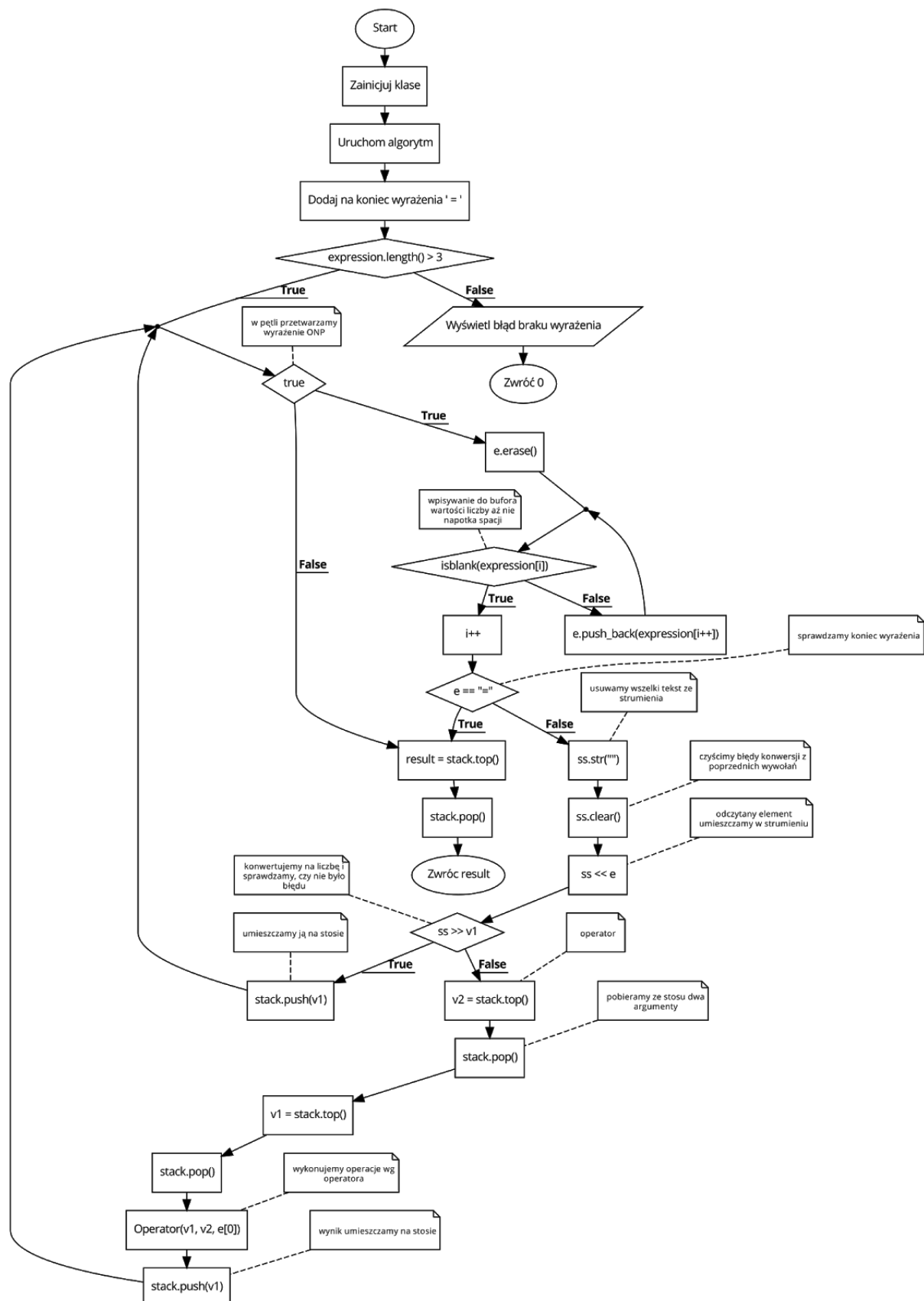
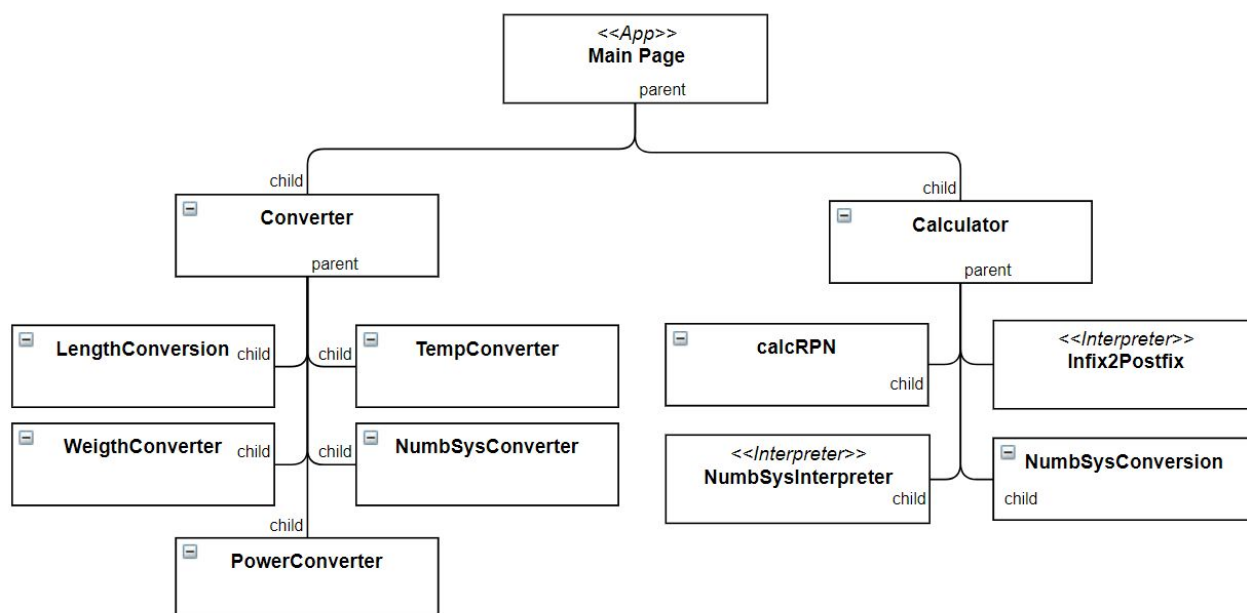


Diagram klas



Opis realizacji

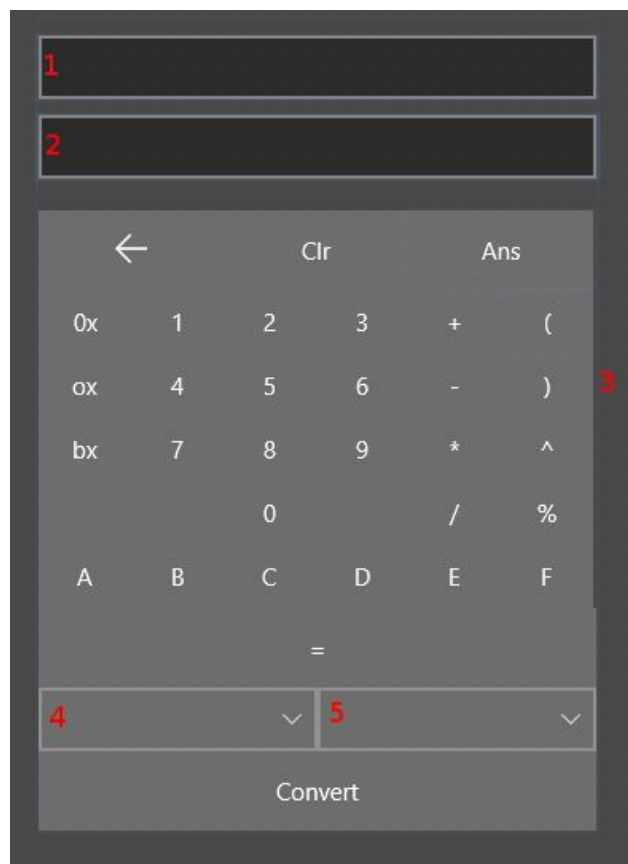
Program został napisany w Visual Studio 2015 wykorzystując szablon Uniwersalnej Aplikacji Windows. Szablon ten daje możliwość zaprojektowania działania aplikacji (w C++) oraz jej wyglądu i funkcji poszczególnych dodanych w niej elementów (w XAML).

Możliwości aplikacji:

- Wprowadzanie liczb - możliwość wprowadzenia liczb przez wbudowaną klawiaturę lub przez wpisanie wyrażenia do pierwszego okna programu
- Zapamiętanie wyniku - po kliknięciu przycisku "Ans" na ekranie zostaje wyświetlony wynik ostatniej operacji, który można użyć w kolejnych obliczeniach
- Kasowanie ekranu - kliknięcie przycisku "Clr" powoduje natychmiastowe zresetowanie wszystkich wartości użytych w programie
- Podstawowe operacje matematyczne
- Konwerter jednostek - po wybraniu typu wejściowego i wyjściowego wartość wpisana w okno wejściowe zostanie przekonwertowana i wyświetlona w drugim oknie

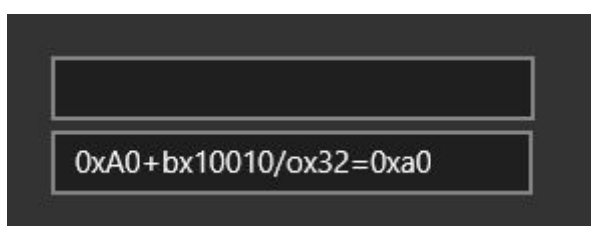
Podręcznik użytkownika

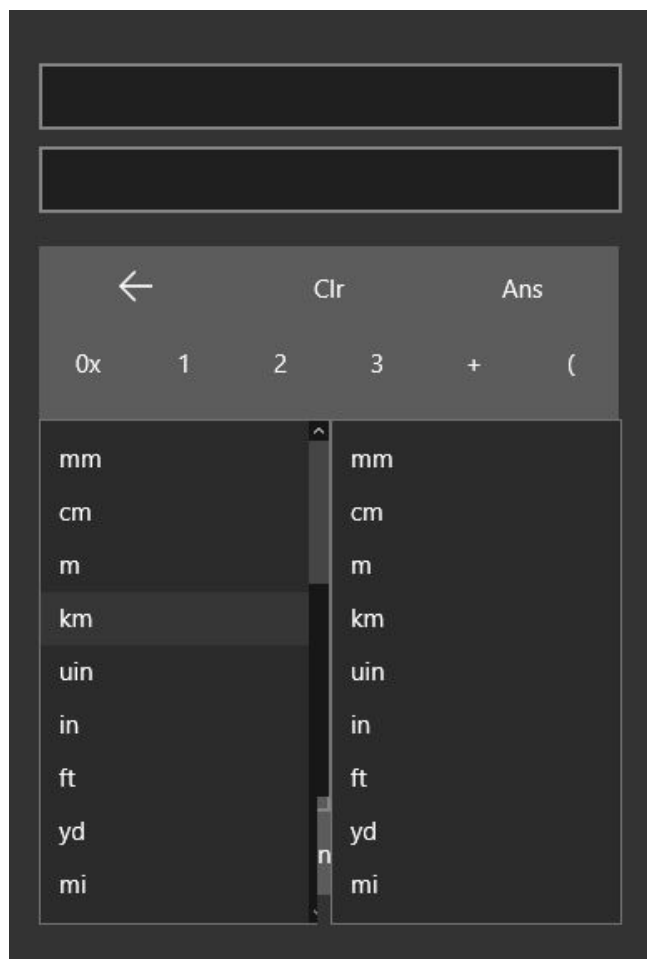
Opis aplikacji



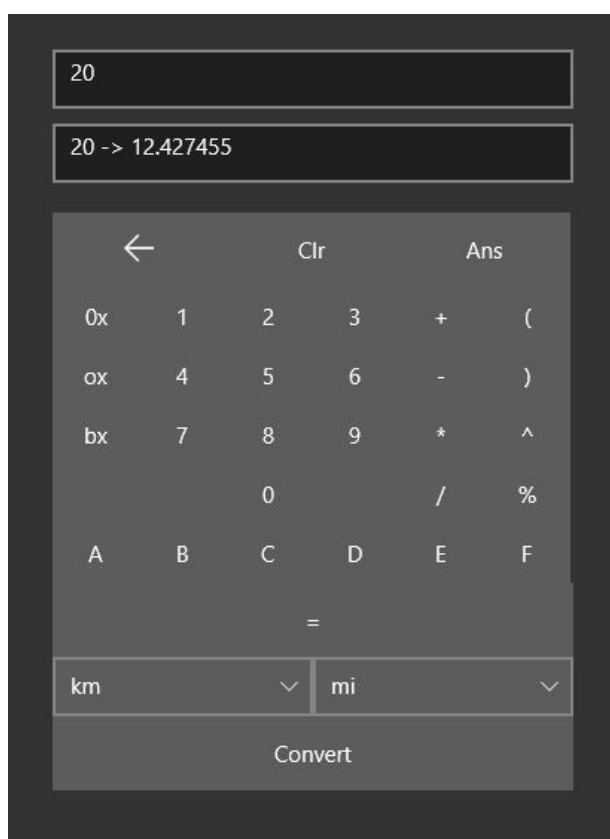
1. Okno wpisywania wyrażenia (wpisywanie na klawiaturze (3))
2. Okno obliczonej bądź skonwertowanej wartości.
3. Klawiatura numeryczna wraz z przyciskami specjalnymi
4. Lista wyboru typu wejściowego do konwersji jednostek
5. Lista wyboru typu wyjściowego konwersji
 - Przycisk % odpowiada za pierwiastek n-tego stopnia (np. 10%4 odpowiada $\sqrt[4]{10}$)
 - Przycisk <- odpowiada za usunięcie ostatniego znaku wpisanego do wyrażenia

Przyciski 0x, ox i bx służą do określania systemu w jakim liczba będzie zapisywana, dzięki temu możemy zapisać wyrażenie $0xA0 + bx10010 / ox32$, które będzie poprawnie obliczone, a wynik będzie podany w systemie wejściowym określonym przez typ pierwszej liczby wyrażenia (w podanym wcześniej przykładzie wynik zostanie podany w systemie HEX).





Lista 1 po wybraniu typu wejściowego konwersji automatycznie aktualizuje Listę 2 odpowiednimi typami konwersji zgodnymi z typem wybranym w Liście 1. (na rysunku obok został zaprezentowany przykład takiego automatycznego uzupełniania) Dzięki takiemu zabiegowi można zapobiec niewłaściwym typom konwersji, takim jak kg -> V.



Przykład konwersji 20 km do mil.

Metodologia rozwoju i utrzymania systemu

- Dodanie możliwości zapamiętywania historii operacji wykorzystując wzorzec projektowy *memorial*
- Poprawienie kontroli wprowadzanych liczb
- Dokładniejsza kontrola wprowadzanego wyrażenia - ulepszenie zabezpieczeń programu
- Wprowadzenie operacji na liczbach zespolonych
- Wykonywanie działań na macierzach
- Wprowadzenie obsługi wykonywania poleceń tekstowych - uruchamianie odpowiedniego programu przez rozpoznawanie poleceń wpisywanych w okno (np. `fft([1,2,3,4])`)