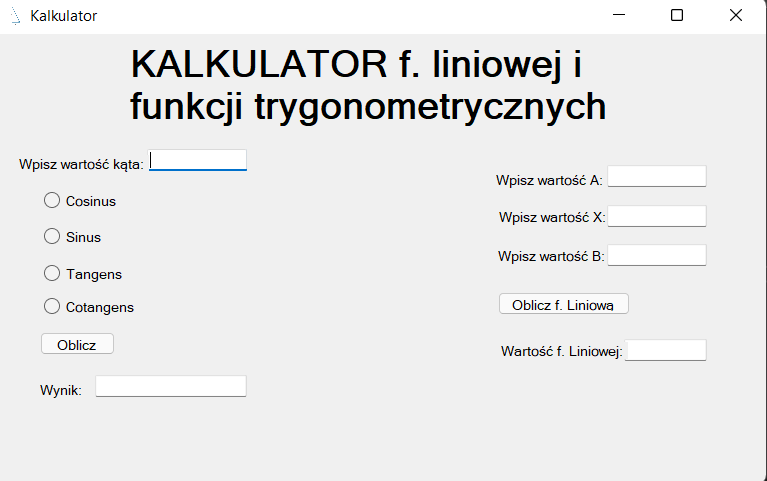
1. Wstęp

Aplikacja ma za zadanie wyliczać wartości funkcji trygonometrycznych (sinus, cosinus, tangens, cotangens) dla podanego przez użytkownika kąta. Program oblicza również wartość funkcji liniowej dla wzoru podanych przez użytkownika. Użytkownik podaje argument A, B oraz X. Aplikacja zwraca wyliczoną wartość.

1. Budowa (interfejs)



1. **Funkcje trygonometryczne**

Część ta zawiera:

- Label z wartością „Wpisz wartość kąta: ” oraz TextBoxa umożliwiającego wpisanie wartości kąta.

- 4 RadioButtony z których to wybiera opcję wybraną przez siebie

- Przycisk który wywołuję obliczenia

- Label o wartości „wynik” oraz TextBoxa do którego zwracana jest obliczona wartość.

1. **Funkcja liniowa**

Część ta zawiera:

- 4 labele opisujące TextBoxy zaimplementowane obok

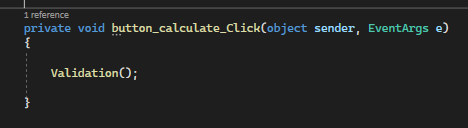
- 4 TextBoxy z czego do trzech użytkownik wpisuje wartości dla argumentów oraz jeden do którego wpisywany jest wynik

- Button wywołujący obliczenia

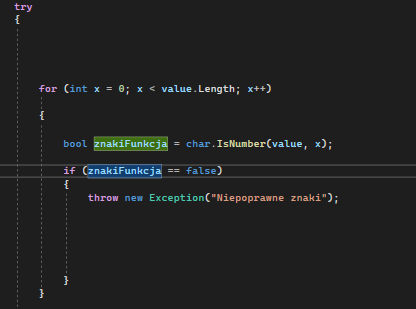
1. Budowa (kod)



**Konstruktor klasy przechowującej wyniki obliczeń**



**Przycisk liczący funkcję trygonometryczną w którym wywołujemy metodę która oblicza i waliduje.**



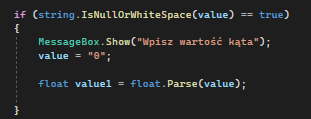
Przypisujemy wartość TextBoxa do zmiennej value

**Zmienna check** służąca do switcha który wybiera odpowiednią metodę liczącą odpowiednią funkcję trygonometryczną

**Walidacja** czy użytkownik wpisał do TextBoxa znaki które nie są liczbami.

Zbudowana jest z pętli która przechodzi przez każdy znak i sprawdza czy jest on liczbą przez użycie if z warunkiem.

**Zmienna typu bool znakiFunkcja** która zwraca wartość true jeśli jest liczbą. Na jej podstawie zbudowany jest IF który jeśli **zmienna bool** zwróci false wyrzuca ekscepcje która łapana jest przez **Try-Catch**.

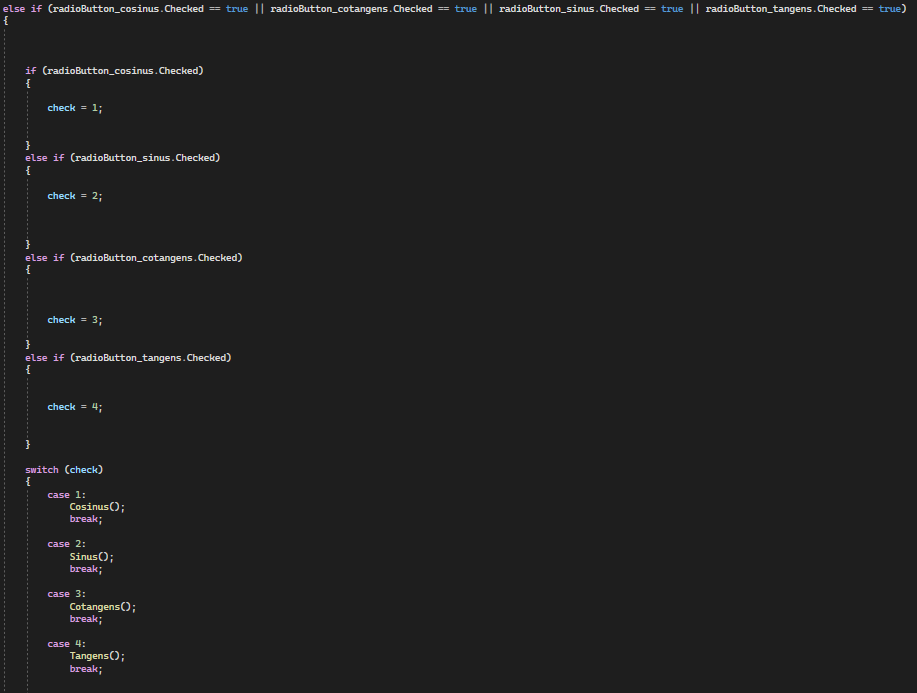


If który sprawdza czy wartość TextBoxa z wartością kąta podaną przez użytkownika jest równa true wtedy wyrzuca MessageBoxa z treścią żeby użytkownik wpisał kąt.

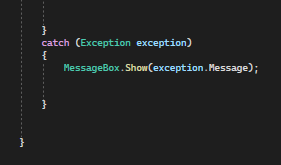
Przypisujemy **zmiennej value** wartość „0”

**Zmienna Value1** o wartości parsowanej zmiennej value do typu float

Jeśli warunek będzie inny niż true program przechodzi do sprawdzenia który przycisk został zaznaczony, jeśli dany radioButton jest zaznaczony program przypisuje odpowiednią wartość zmiennej check w celu użycia jej później w switch wybierającej odpowiednią metodę liczącą wartość odpowiedniej funkcji trygonometrycznej

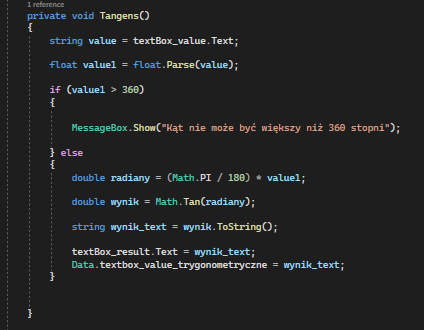


**Switch** który wybiera odpowiednią metodę za pomocą zmiennej check.



Część catch Try-catch wyrzucającą **MessageBox** z wiadomością podaną w części try z wiadomością o wpisaniu wartości kąta.

1. **Metoda licząca Tangens**



**Zmienna typu string value** pobierająca wartość z **TextBox\_value**

**Zmienna typu float value1** przechowująca wartość parsowanej zmiennej value na typ float

Warunek jeśli **value1** jest większe niż 360 program wyrzuca MessageBoxa z wiadomością o tym, że kąt nie może być większy od 360. Jeśli warunek nie zostanie spełniony program przechodzi do liczenia tangensa.

**Zmienna double radiany** konwertująca kąty na radiany w celu użycia jest w funkcji Math.Tan

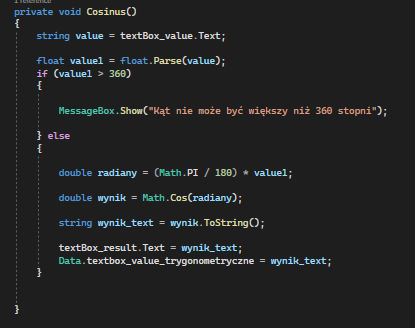
**Zmienna double wynik** przechowująca wynik funkcji Math.Tan od radianów

**Zmienna string wynik\_text** przechowująca wynik skonwertowanej zmiennej wynik do typu string.

Przypisywana jest wartość zmiennej **wynik\_text** do TextBoxa **textBox\_result**

Przypisywana jest wartość zmiennej **wynik\_text** do zmiennej **textbox\_value\_trygonometryczna** z klasy Data

1. **Metoda licząca Cosinus**



**Zmienna typu string value** pobierająca wartość z TextBoxa

**Zmienna typu float value1** przechowująca wartość parsowanej zmiennej value na typ float

Warunek jeśli value1 jest większe niż 360 program wyrzuca MessageBoxa z wiadomością o tym, że kąt nie może być większy od 360. Jeśli warunek nie zostanie spełniony program przechodzi do liczenia tangensa.

**Zmienna double radiany** konwertująca kąty na radiany w celu użycia jest w funkcji **Math.Cos**

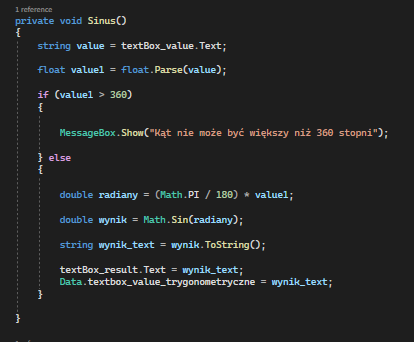
**Zmienna double wynik** przechowująca wynik funkcji **Math.Cos** od radianów

**Zmienna string wynik\_tex**t przechowująca wynik skonwertowanej zmiennej wynik do typu string.

Przypisywana jest wartość zmiennej **wynik\_text** do **TextBoxa textBox\_result**

Przypisywana jest wartość zmiennej **wynik\_text** do zmiennej **textbox\_value\_trygonometryczna** z klasy Data

1. **Metoda licząca Sinus**



**Zmienna typu string value** pobierająca wartość z TextBoxa

**Zmienna typu float value1** przechowująca wartość parsowanej zmiennej value na typ float

Warunek jeśli **value1** jest większe niż 360 program wyrzuca **MessageBox** z wiadomością o tym, że kąt nie może być większy od 360. Jeśli warunek nie zostanie spełniony program przechodzi do liczenia tangensa.

**Zmienna double radiany** konwertująca kąty na radiany w celu użycia jest w funkcji **Math.Sin**

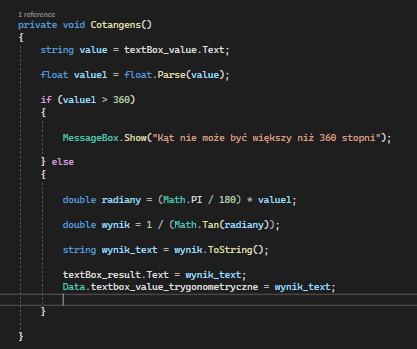
**Zmienna double wynik** przechowująca wynik funkcji **Math.Sin** od radianów

**Zmienna string wynik\_text** przechowująca wynik skonwertowanej zmiennej wynik do typu string.

Przypisywana jest wartość zmiennej **wynik\_text** do **TextBoxa textBox\_result**

Przypisywana jest wartość zmiennej **wynik\_text** do zmiennej **textbox\_value\_trygonometryczna** z klasy **Data**

1. **Metoda licząca Cotangens**



**Zmienna typu string value** pobierająca wartość z TextBoxa

**Zmienna typu float value1** przechowująca wartość parsowanej zmiennej value na typ float

Warunek jeśli **value1** jest większe niż 360 program wyrzuca MessageBox z wiadomością o tym, że kąt nie może być większy od 360. Jeśli warunek nie zostanie spełniony program przechodzi do liczenia tangensa.

**Zmienna double radiany** konwertująca kąty na radiany w celu użycia jest w funkcji Math.Tan

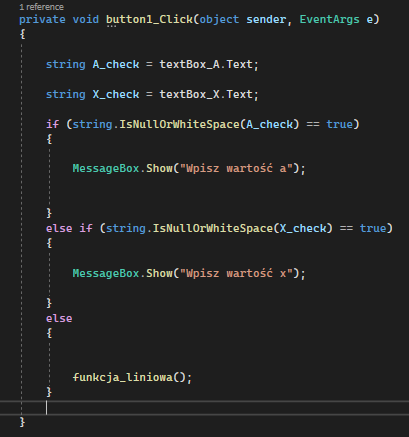
**Zmienna double wynik** przechowująca odwrócony wynik funkcji Math.Tan od radianów, dlatego że Cotangens jest odwrotnością Tangensa.

**Zmienna string wynik\_text** przechowująca wynik skonwertowanej zmiennej wynik do typu string.

Przypisywana jest wartość **zmiennej wynik\_text** do **TextBox textBox\_result**

Przypisywana jest wartość **zmiennej wynik\_text** do zmiennej **textbox\_value\_trygonometryczna** z klasy Data

Funkcja wywoływana po kliknięciu w przycisk **button\_liniowa**



Zmienne pobierające wartości z **TextBoxa textBox\_A** oraz **textBox\_B** w celu sprawdzenia czy użytkownik wpisał wartość.

Warunek jeśli **string A\_check** jest Null albo WhiteSpace jest równe true wyrzucany jest **MessageBox** z wiadomością żeby użytkownik wpisał wartość A.

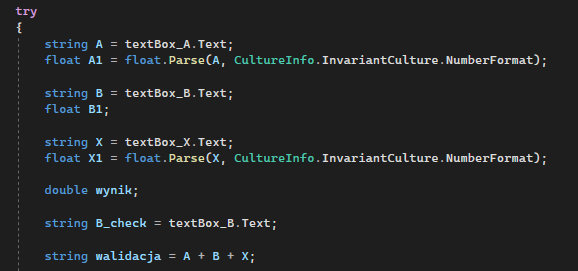
Jeśli warunek niespełniony przechodzi do

Warunku jeśli string **B\_check** jest Null albo WhiteSpace jest równe true wyrzucany jest MessageBox z wiadomością żeby użytkownik wpisał wartość B.

Jeśli warunek niespełniony wywoływana jest metoda liczenia wartości funkcji liniowej

1. **Metoda liczenia funkcji liniowej**

**Zmienna typu string** A przechowująca wartość **textBox\_A**

****

**Zmienna A typu float** przechowująca wartość parsowanej zmiennej A do typu float

Zmienna A1 przechowująca wartość parsowanej zmiennej A do typu float

**Zmienna B typu string** B przechowująca wartość **textBox\_B**

**Zmienna B1** służąca do przypisania jej wartości w dalszej części programu

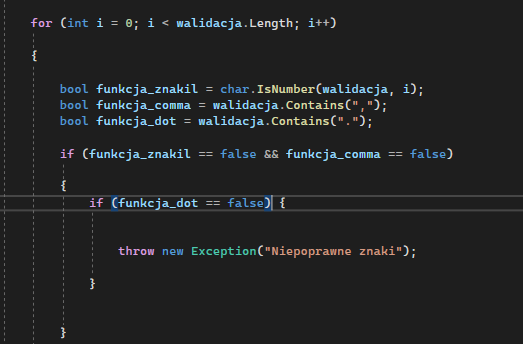
**Zmienna X typu string X** przechowująca wartość **textBox\_X**

**Zmienna typu float X1** przechowująca wartość parsowanej zmiennej X do typu float

**Zmienna wynikowa typu double**

**Zmienna typu string B\_check** przechowująca wartość **textBox\_B**

**Zmienna walidacja typu string** łącząca wartości z wszystkich textBoxów w celu późniejszej walidacji czy użytkownik wpisał coś innego niż liczbę



Zmienna funkcja\_znakil typu bool sprawdzająca czy dany znak jest liczbą

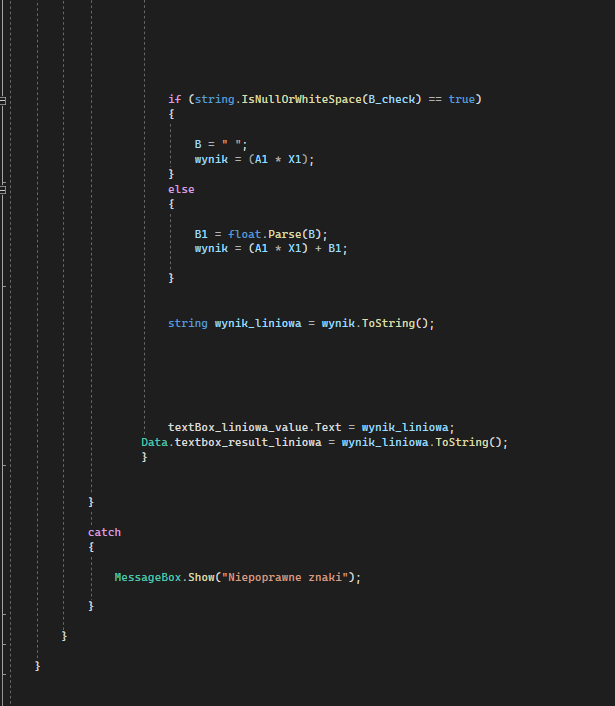
Zmienna funkcja\_comma sprawdzająca czy zmienna walidacja zawiera przecinek

Zmienna funkcja\_dot sprawdzająca czy zmienna walidacja zawiera kropkę

Walidacja czy użytkownik wpisał do któregoś z TextBoxa znaki które nie są liczbami.

Zbudowana jest z pętli która przechodzi przez każdy znak i sprawdza czy jest on liczbą przez użycie if z warunkiem.

Zmienna typu bool która zwraca wartość true jeśli jest liczbą. Na jej podstawie zbudowany jest IF który jeśli zmienna bool zwróci false oraz zmeinna funkcja\_comma zwróci false, przechodzi do kolejnego warunku czy funkcja\_dot zwróci false, jeśli tak to wyrzuca ekscepcje, jeśli wyrzuci true program przechodzi dalej bez wyświetlenia żadnego komunikatu.



Warunek sprawdzający czy wartość B jest wpisane, jeśli jest, przypisywana jest do niej wartość ‘ ‘

**Zmiennej wynik** przychowująca dodane do siebie zmienne A1 \* X1 gdyż B jest równe 0

Jeśli warunek nie jest spełniony program przechodzi do parsowania do typu float **zmiennej B**

Zmiennej wynik jest przypisywana wartość działania (A1 \* X1) + B1

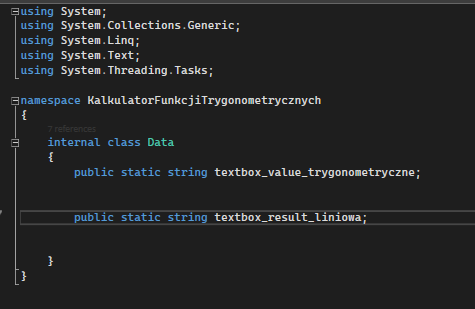
**Zmienna string wynik\_liniowa** przechowuje wartość zmiennej wynik która została przekonwertowana na string

**TextBoxow textBox\_liniowa\_value** jest przypisywana wartość zmiennej **wynik\_liniowa** w celu wyświetlenia jej na ekranie

**Zmiennej textbox\_result\_liniowa** jest przypisywana wartość przeparsowanej zmiennej **wynik\_liniowa**;

**Try-Catch** łapiący ekscepcje jeśli użytkownik wpisze coś innego niż liczby

Klasa Data



Klasa jest internal

Zmienna publiczna statyczna typu string textbox\_value\_trygonometryczne;

Zmienna publiczna statyczna typu string textbox\_value\_liniowa;

Zmienne przechowują wyniki obliczeń poszczególnych wyników.