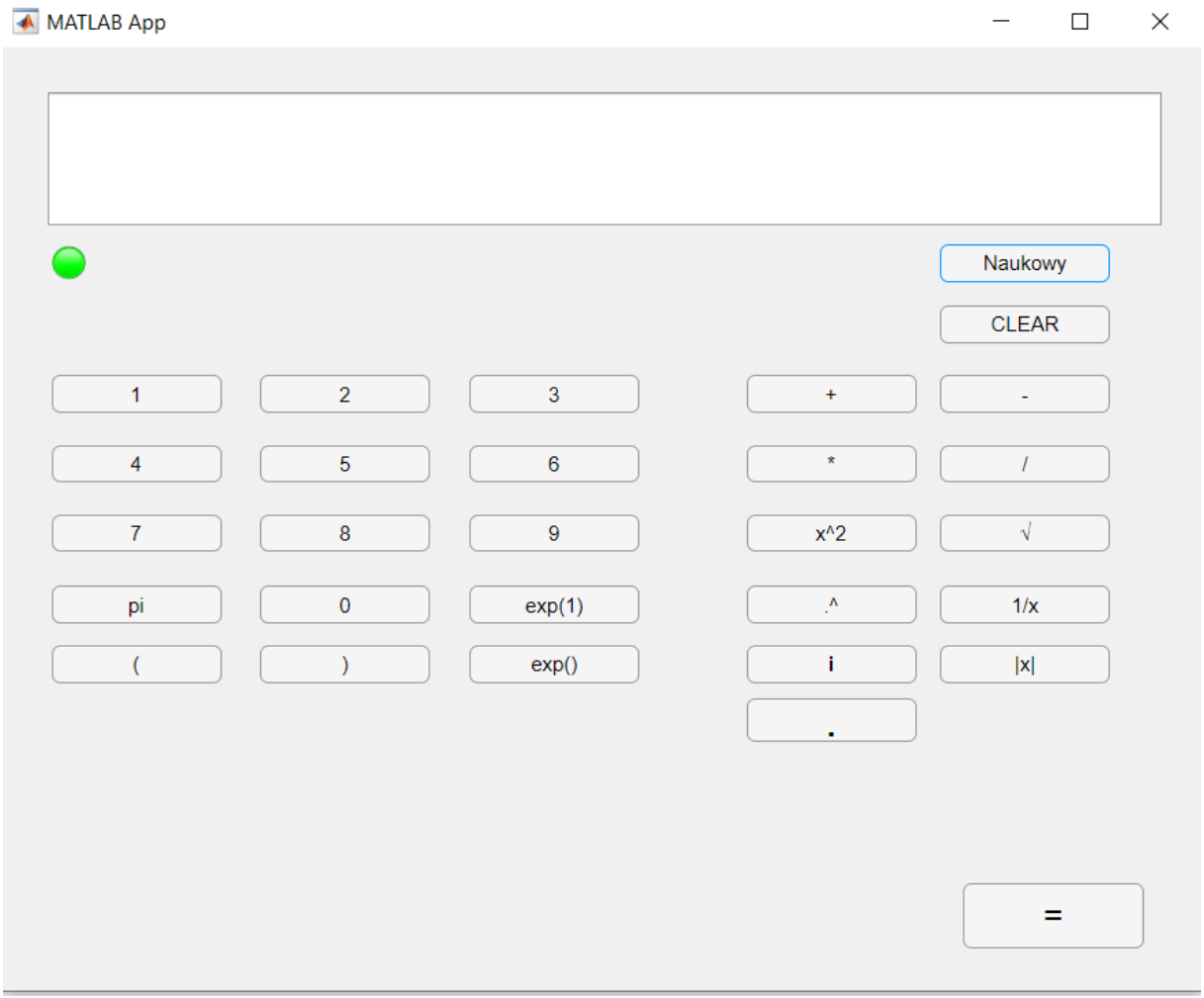
Dokumentacja projektu

Matlab Calculator

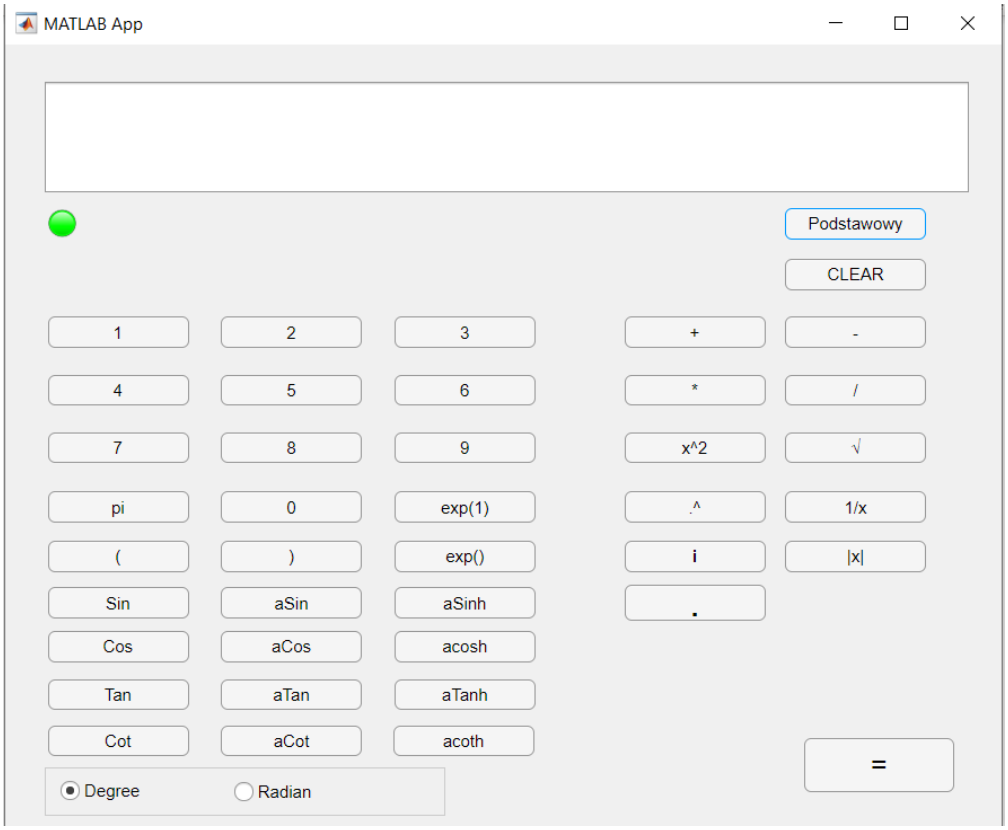
Mikołaj Bańkowski

Interfejs użytkownika

Interfejs kalkulatora podstawowego



Interfejs kalkulatora naukowego



Funkcje kalkulatora

- dodawanie
- odejmowanie
- dzielenie
- mnożenie
- podnoszenie liczby do kwadratu
- pierwiastkowanie
- funkcja e^(x)
- odwrotność liczby
- wartość bezwzględna z liczby
- działania na liczbach zespolonych(dodawanie, odejmowanie, mnożenie, dzielenie)
- funkcje trygonometryczne
- funkcje cyklometryczne
- funkcje hiperboliczne
- funkcjonalność liczenia w stopniach i radianach
- CLEAR(czyszczenie ekranu)

Potęgowanie/pierwiastkowanie dowolnego stopnia

Potęgowanie a.^b

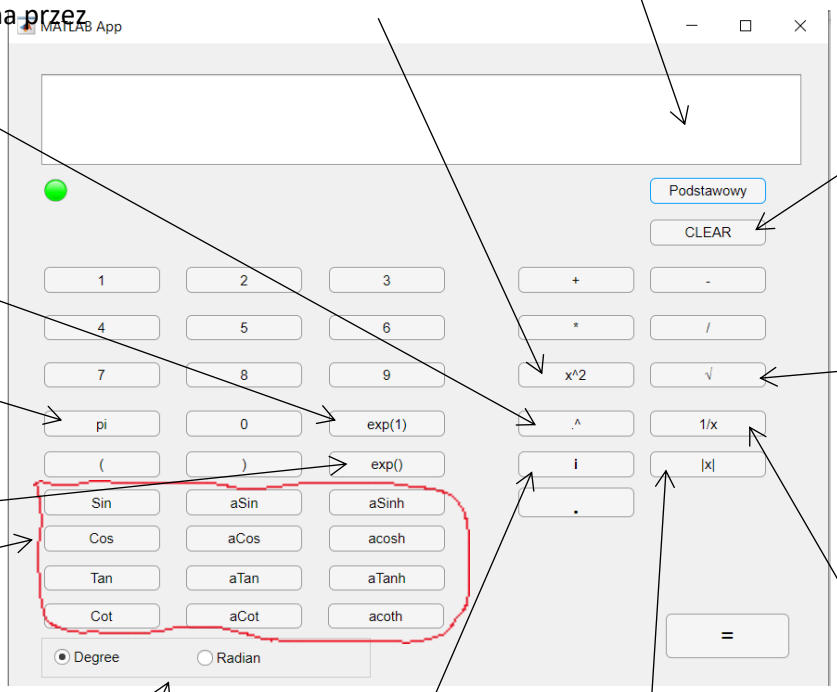
Pierwiastkowanie a.^(1/b)

a - liczba wprowadzona przez użytkownika

b – druga liczba wprowadzona przez użytkownika

Przycisk do zmieniania trybu z naukowego na podstawowy

Potęgowanie do kwadratu



Przycisk do czyszczenia ekranu kalkulatora

Podstawa logarytmu naturalnego(zapisana jako exp(1)

Pierwiastek kwadratowy

Liczba pi

e^(x)

Funkcje:

Trygonometryczne

Cyklometryczne

Hiperboliczne

Odwrotność liczby

Jednostka urojona

Wartość bezwzględna z liczby

Funkcja liczenia w stopniach

Funkcja liczenia w radianach

```
function PRZYCISK(app, event)
    app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text];
```

Funkcja „PRZYCISK” odpowiada za wyświetlanie na obszarze tekstowym znaków, odpowiadających wartościom umieszczonym na polach tekstowych przycików, funkcjonalność przypisana jest do większosci przycisków poza funkcjami matematycznymi

```
function WYNIK(app, event)
    try
        app.TextArea.Value = [app.TextArea.Value{1} event.Source.Text ...
            num2str(eval(app.TextArea.Value{1}))];
    catch
        app.TextArea.Value='Error!';
    end
```

Funkcja „WYNIK” odpowiada za obliczenia działania znajdującgo się w obszarze tekstowym.

Funkcja odpowiada także za wyświetlenie komunikatu wystąpienia błedu w przypadku podwójnego klikniecia przycisku „=” lub gdy przycisk zostanie naciśnęty w przypadku pustego pola tekstowego.

Podczas wywoływania eval z niezgonymi danymi wejściowymi użytkownika sprawdza poprawność danych wejściowych używając polecenia (try , catch ,end), aby uniknąć nieoczekiwanego wykonania kodu.

Polecenie (try , catch ,end) wykonuje instrukcje z bloku try i przechwytuje powstałe błędy w bloku catch. Takie podejście umożliwia zastąpienie domyślnego zachowania błędu dla zestawu instrukcji programu. Jeśli jakakolwiek instrukcja w bloku try wygeneruje błąd, sterowanie programem przechodzi natychmiast do bloku catch, który zawiera instrukcje obsługi błędów.

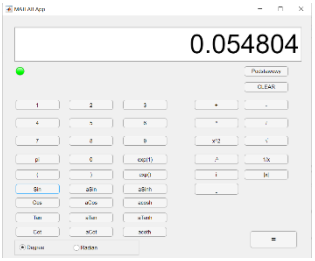
```
function SIN(app, event)
     inputValue = str2num(app.TextArea.Value{1});

    if(app.DegreeButton.Value)
        app.TextArea.Value = num2str(sind(inputValue));
    else(app.RadianButton.Value)
        app.TextArea.Value = num2str(sin(inputValue));
    end
```

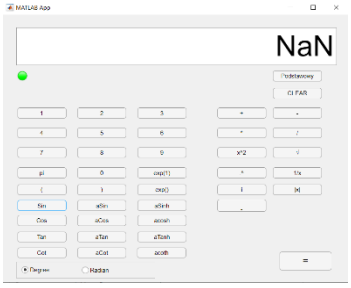
W każdej funkcji trygonometrycznej, hiperbolicznej, cyklometrycnzej użyłem „str2num(x,p)

S = str2num(x,p) – zmienia wartość wyrażenia na łańcuch znaków S, opcjonalnie uwzgledniając p miejsc dziesiętnych po przecinku

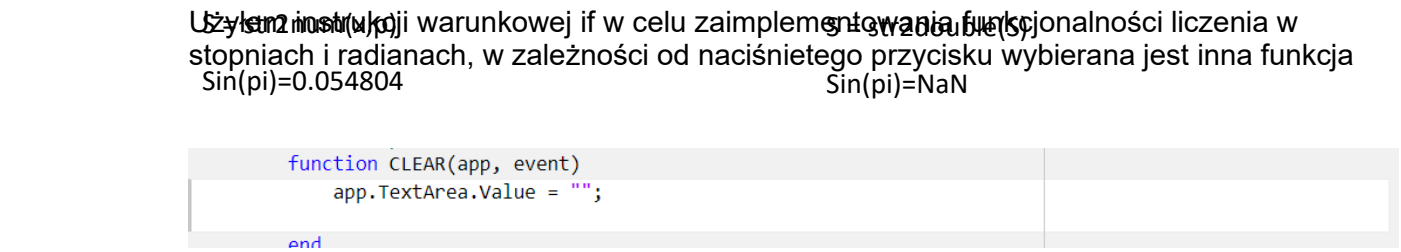
X = str2double(S) – zmienia łancuch znaków S reprezentujących liczbę na liczbę typu double

Wybrałem S = str2num(x,p) zamiast S = str2double(S), ponieważ lepiej się sprawdziła w moim kalkulatorze, pierwotnie używałem S = str2double(S), jednak pjawiali się problemy z wyrażeniem „pi”, kalkulator niepoprawnie liczył wyrażenia trygonometryczne oraz funkcje w których pojawiało się „pi”





Użyłem instrukcji warunkowej if w celu zaimplementowania funkcjonalności liczenia w stopniach i radianach, w zależności od naciśniętego przycisku wybierana jest inna funkcja




```
function CLEAR(app, event)
    app.TextArea.Value = "";

end
```

Funkcja „CLEAR” czyści ekran z ekranu kalkulatora, w taki sposób, że przypisuje mu pustą wartość

```
function TRYB_NAUKOWY(app, event)
    switch(app.PodstawowyButton.Text)
        case 'Podstawowy'
            app.PodstawowyButton.Text = 'Naukowy';
            app.SinButton.Visible = 'off';
            app.CosButton.Visible = 'off';
            app.TanButton.Visible = 'off';
            app.CotButton.Visible = 'off';
            app.aSinButton.Visible = 'off';
            app.aCosButton.Visible = 'off';
            app.aTanButton.Visible = 'off';
            app.aCotButton.Visible = 'off';
            app.aSinhButton.Visible = 'off';
            app.aCoshButton.Visible = 'off';
            app.aTanhButton.Visible = 'off';
            app.aCothButton.Visible = 'off';
            app.TanButton.Visible = 'off';
            app.ButtonGroup.Visible = 'off';

        case 'Naukowy'
            app.PodstawowyButton.Text = 'Podstawowy';
            app.SinButton.Visible = 'on';
            app.CosButton.Visible = 'on';
            app.TanButton.Visible = 'on';
            app.CotButton.Visible = 'on';
            app.aSinButton.Visible = 'on';
            app.aCosButton.Visible = 'on';
            app.aTanButton.Visible = 'on';
            app.aCotButton.Visible = 'on';
            app.aSinhButton.Visible = 'on';
            app.aCoshButton.Visible = 'on';
            app.aTanhButton.Visible = 'on';
            app.aCothButton.Visible = 'on';
            app.TanButton.Visible = 'on';
            app.ButtonGroup.Visible = 'on';
    end
```

Funkcja „TYRYB_NAUKOWY” działa w bardzo prosty sposób, po naciśnięciu przycisku „Podstawowy” chowa przed użytkownikiem przyciski z interfacu kalkulatora przed użytkownikiem oraz przypasuje nowa nazwę przyciskowi Podstawowybutton na nazwę

„Naukowy”, chcąc powrócić do trybu interfacu naukowego musimy nacisnąć przycisk „Naukowy”(analogicznie przyciskowi Podstawowybutton przypisywana jest nazwa „Podstawowy”)

```
% Button pushed function: expButton
function EXPONENS(app, event)
    inputValue = str2num(app.TextArea.Value{1});
    app.TextArea.Value = num2str(exp(inputValue));
end

% Button pushed function: Button_21
function PIERWIASTEK(app, event)
    inputValue = str2num(app.TextArea.Value{1});
    app.TextArea.Value = num2str(inputValue.^(1/2));
end

% Button pushed function: x2Button
function DOKWADRATU(app, event)
    inputValue = str2num(app.TextArea.Value{1});
    app.TextArea.Value = num2str(inputValue.^2);
end

% Button pushed function: xButton
function ODWRTONOSC(app, event)
    inputValue = str2num(app.TextArea.Value{1});
    app.TextArea.Value = num2str(inputValue.^(-1));
end

% Button pushed function: xButton_2
function MODUL(app, event)
    inputValue = str2num(app.TextArea.Value{1});
    app.TextArea.Value = num2str(abs((inputValue)));
end
```

Reszta funkcjonalności otrzymywana jest w bardzo prosty i nieskomplikowany sposób

Na wpisanym przez użytkownika ciąg znaków konwertowany jest na wybrane działanie matematyczne