

# Dokumentacja projektu

Przedmiot: Programowanie

Tytuł projektu:

# KALKULATOR MATERIAŁOWY

Prowadzący: Wykonawca:

Dr Marek Jaszuk Mateusz Bochnia (W60116)

Semestr i symbol kierunku: 4IIZ

Grupa:

Grupa na platformie BB:

## 1. Opis założeń projektu (wymagania biznesowe)

Stworzenie aplikacji, która będzie umożliwiała obliczenie wagi materiału po podaniu potrzebnych parametrów. Aplikacja powinna umożliwić sprawdzenie wagi dla różnego rodzaju produktów oraz w różnej konfiguracji ilościowej. Program powinien umożliwić obliczenie wartości dla różnego rodzaju materiału oraz gatunku.

# 2. Specyfikacja wymagań

#### Wymagania funkcjonalne:

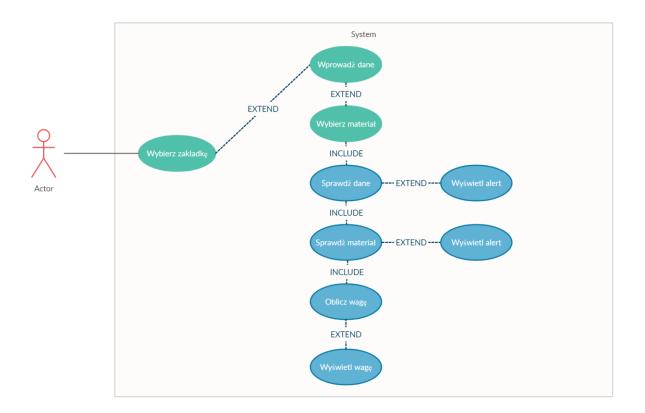
- możliwość wpisania parametrów potrzebnych do obliczenia wagi
- możliwość policzenia wagi dla różnych produktów (blachy/płaskowniki, profile, pręty, rury)
- brak możliwości policzenia wagi w przypadku niewystarczającej ilości danych
- brak możliwości policzenia wagi w przypadku niewybrania materiału

#### Wymagania niefunkcjonalne:

- program powinien być intuicyjny przejrzysty i powinien zawierać łatwo dostępne przyciski, oraz zrozumiałe komunikaty
- program powinien być wydajny zapewnienie szybkiego działania aplikacji tak aby użytkownik był zadowolony z działania systemu
- jakość aplikacja musi być wykonana w przyjemny dla oka sposób
- dopasowanie aplikacja musi zawierać rozwiązania dla wszystkich wymagań.

# 3. Diagram przypadków użycia z wyróżnieniem aktorów projektu oraz przypadków użycia odzwierciedlających wymagania funkcjonalne.

Do rysowania diagramu użyłem oprogramowania ze strony: https://app.creately.com/



# 4. Harmonogram realizacji projektu

Harmonogram realizacji projektu przedstawiony przy pomocy diagramu Gantta.



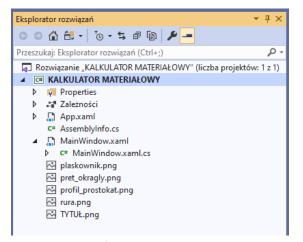
## 5. Opis techniczny projektu

Aplikacja została stworzona na podstawie języka C# i frameworku .Net Core z technologią Windows Presentation Foundation (WPF). Projekt tworzyłem przy użyciu Microsoft Visual Studio Community 2019 w jednoosobowym zespole, więc samodzielnie wykonałem projekt zarówno, jeśli chodzi o programowanie logiki programu, jak i wygląd aplikacji. Grafika wykorzystana w projekcie została wykonana przeze mnie w programie Adobe Photoshop 2020.

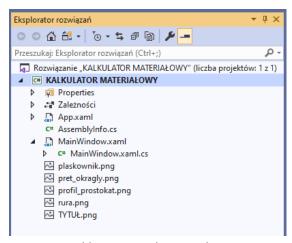
Cała logika programu znajduje się w pliku *Mainwindow.xaml.cs*, natomiast cały wygląd aplikacji znajduje się w *Mainwindow.xaml*.

#### Struktura kodu programu:

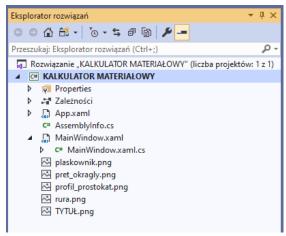
Struktura katalogów:



Struktura katalogów w programie MS Visual Studio.



Plik Mainwindow.xaml.cs



Plik Mainwindow.xaml

#### Metody obliczeniowe:

```
private void ButtButton_Click_Licz_Plaskownik(object sender, RoutedEventArgs e)

{
    string messageBoxText2 = "Podaj potrzebne dane";
    string caption2 = "UNAGAI";
    MessageBoxSutton button2 = MessageBoxButton.OK;
    MessageBoxSutton button2 = MessageBoxButton.OK;
    MessageBoxSutson button2 = MessageBoxButton.OK;
    MessageBoxImage icon2 = MessageBoxImage.Warning;
    if (plaskownik_szerokosc.Text == "" || plaskownik_dlugosc.Text == "" || plaskownik_grubosc.Text == "") MessageBox.Show(messageBoxText2, caption2, button2, icon2);
    else

{
        string messageBoxText = "Wybierz material z listy";
        string caption = "UNAGAI";
        MessageBoxImage icon = MessageBoxImage.Warning;
        double wynik = 0;
        short pomoc = 1;
        double da = convert.ToDouble(plaskownik_dlugosc.Text);
        double sizer = Convert.ToDouble(plaskownik_szerokosc.Text);
        double sizer = Convert.ToDouble(plaskownik_szerokosc.Text);
        double gr = Convert.ToDouble(plaskownik_s
```

#### Metoda obliczania wagi dla płaskowników i blach

Metoda obliczania wagi dla profili

```
| String messageBoxText2 = "Podaj potrzebne dane";
| string messageBoxText2 = "Podaj potrzebne dane";
| string caption2 = "UNAGA!";
| MessageBoxMider ton button2 = MessageBoxMatton.OK;
| MessageBoxMider ton con2 = MessageBoxMamage.Warning;
| string messageBoxText = "Wybierz material z listy";
| string caption = "UNAGA!";
| MessageBoxMutton button = MessageBoxButton.OK;
| MessageBoxMamage icon = MessageBoxButton.OK;
| MessageBoxImage icon = MessageBoxImage.Warning;
| if (prety_srednica.Text == "" || prety_dlugosc.Text == "") | MessageBox.Show(messageBoxText2, caption2, button2, icon2);
| else |
| double | mynik = 0;
| short pomoc = 1;
| double of = Convert.ToDouble(prety_srednica.Text);
| double of = Convert.ToDouble(prety_dlugosc.Text);
| wynik = (3.14 * Math.Pow((sr / 2), 2) * dl) / 10000000;
| if (prety_material.SelectedItem == Pret_Alu_7075) | wynik = wynik * 2.9;
| else if (prety_material.SelectedItem == Pret_Alu_7075) | wynik = wynik * 2.8;
| else if (prety_material.SelectedItem == Pret_Alu_7075) | wynik = wynik * 2.7;
| else if (prety_material.SelectedItem == Pret_Alu_7075) | wynik = wynik * 7.85;
| else if (prety_material.SelectedItem == Pret_Stalczarna) | wynik = wynik * 7.85;
| else if (prety_material.SelectedItem == Pret_Stalczarna) | wynik = wynik * 7.85;
| else if (prety_material.SelectedItem == Pret_Stalczarna) | wynik = wynik * 7.85;
| else if (prety_material.SelectedItem == Pret_Brety_Material.SelectedItem == Prety_Material.SelectedItem == Prety_Material.SelectedItem == Prety_Material.SelectedItem =
```

#### Metoda obliczania wagi dla prętów

```
loaboutnies
private void Button_Click_Licz_Rury(object sender, RoutedEventArgs e)

{
    string messageBoxText2 = "Podaj potrzebne dane";
    string caption2 = "UNAGA!";
    HessageBoxSutton button2 = MessageBoxButton.OK;
    MessageBoxImage Con2 = MessageBoxImage.Warning;
    string messageBoxText = "Wybierz material z listy";
    string messageBoxImage con = MessageBoxButton.OK;
    MessageBoxImage icon = MessageBoxImage.Warning;
    if (rury_srednica_zew.Text == "" || rury_dlugosc.Text == "") MessageBox.Show(messageBoxText2, caption2, button2, icon2);
    else

{
        double wynik = 0;
        short pomoc = 1;
        double gr = Convert.ToDouble(rury_gr.Text);
        double sr_zew = convert.ToDouble(rury_srednica_zew.Text);
        double sr_zew = convert.ToDouble(rury_dlugosc.Text);
        double sr_zew = convert.ToDouble(rury_dlugosc.Text);
        double sr_zew = sr_zew - (gr * 2);
        wynik = (3.14 * 41 * (sr_zew - sr_wew)) / 1000000;
        if (rury_material.SelectedItem == Rur_Alu_6063) wynik = wynik * 2.9;
        else if (rury_material.SelectedItem == Rur_Alu_6082) wynik = wynik * 2.8;
        else if (rury_material.SelectedItem == Rur_Alu_6082) wynik = wynik * 2.8;
        else if (rury_material.SelectedItem == Rur_Cynk) wynik = wynik * 2.7;
        else if (rury_material.SelectedItem == Rur_Cynk) wynik = wynik * 2.7;
        else if (rury_material.SelectedItem == Rur_Cynk) wynik = wynik * 7.8;
        else if (rury_material.SelectedItem == Rur_Deraz || rury_material.SelectedItem == Rur_Miedz) wynik = wynik * 8.9;
        else if (rury_material.SelectedItem == Rur_Miedz) wynik = wynik * 8.9;
        else pomoc = 0;

        if (pomoc == 1) rury_wyswietlanie.Content = wynik;
        else MessageBox_Show(messageBoxText, caption, button, icon);
}
```

Metoda obliczania wagi dla rur

# Metody czyszczące dane:

```
lodwotanie
private void Button_Click_Clear_Wysw_Dane_Plaskownik(object sender, RoutedEventArgs e)
{
    plaskownik_grubosc.Text = "";
    plaskownik_szerokosc.Text = "";
    plaskownik_dlugosc.Text = "";
}

lodwotanie
private void Button_Click_Clear_Wysw_Dane_Profile(object sender, RoutedEventArgs e)
{
    profile_grubosc.Text = "";
    profile_wysokosc.Text = "";
    profile_szerokosc.Text = "";
    profile_dlugosc.Text = "";
}
```

```
lodwolanie
private void Button_Click_Clear_Wysw_Dane_Prety(object sender, RoutedEventArgs e)
{
    prety_srednica.Text = "";
    prety_dlugosc.Text = "";
}

lodwolanie
private void Button_Click_Clear_Wysw_Dane_Rury(object sender, RoutedEventArgs e)
{
    rury_gr.Text = "";
    rury_srednica_zew.Text = "";
    rury_dlugosc.Text = "";
}
```

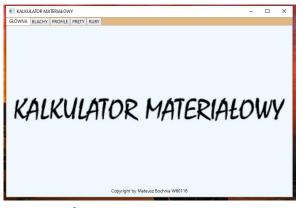
#### Metoda wyczyszczenia wprowadzonych danych

```
1 Odwołanie
private void Button_Click_Clear_Wysw_Wynik_Plaskownik(object sender, RoutedEventArgs e)
{
    plaskownik_wyswietlanie.Content="";
}
```

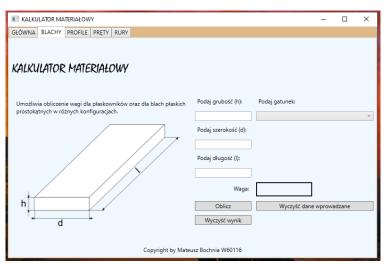
Metoda wyczyszczenia wyświetlanego wyniku, z racji tego, że jest bliźniacza dla każdej z 4 części kalkulatora różni się tylko w nazwie pola do edycji według schematu:

<nazwa\_opcji>\_wyswietlanie.Content="";

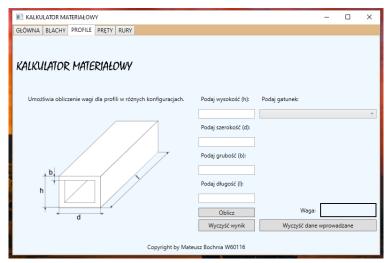
# 6. Prezentacja warstwy użytkowej projektu



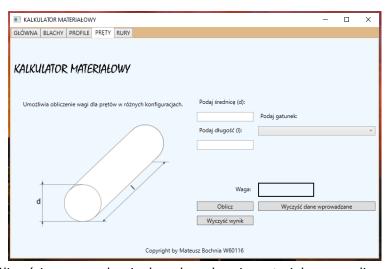
Strona startowa programu



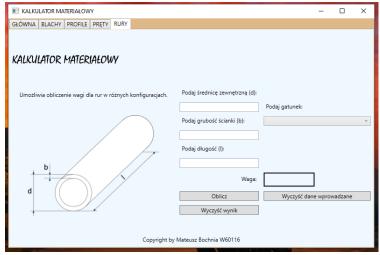
Okno z możliwością wprowadzenia danych, wybrania materiału oraz policzenia wagi dla arkusza blachy lub płaskownika. Widoczny podgląd wymiarów na rysunku poglądowym.



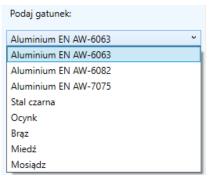
Okno z możliwością wprowadzenia danych, wybrania materiału oraz policzenia wagi dla profilu. Widoczny podgląd wymiarów na rysunku poglądowym.



Okno z możliwością wprowadzenia danych, wybrania materiału oraz policzenia wagi dla pręta. Widoczny podgląd wymiarów na rysunku poglądowym.



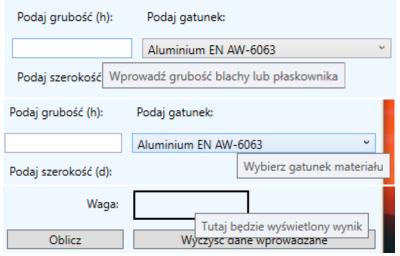
Okno z możliwością wprowadzenia danych, wybrania materiału oraz policzenia wagi dla rury. Widoczny podgląd wymiarów na rysunku poglądowym.



Lista rozwijalna z możliwością wyboru gatunku materiału

	Podaj gatu	nek:	
	Aluminium EN AW-6063		•
Wybrany gatunek materiału			
Podaj grubość (h):	Podaj wysokość (h):	Podaj średnicę (d):	Podaj średnicę zewnętrzną (d):
	Podaj szerokość (d):		
Podaj szerokość (d):	Podaj grubość (b):		Podaj grubość ścianki (b):
Podaj długość (I):		Podaj długość (I):	Podaj długość (I):
	Podaj długość (I):		r outjungest (i/i
Pola do wprowadzania danych			
	Waga:		
Pole do wyświetlania wyniku			
Oblicz Wyczyść dane wprowadzane			
	Wyczyść wynik		

Przyciski do obliczenia wagi oraz wyczyszczenia danych wprowadzonych oraz wyświetlonych



Przykładowo wyglądające podpowiedzi w programie wyświetlane po najechaniu kursora na dane pole



Widoczne przyciski nawigacyjne służace do przełączenia trybu liczenia danego produktu

# 7. System kontroli wersji

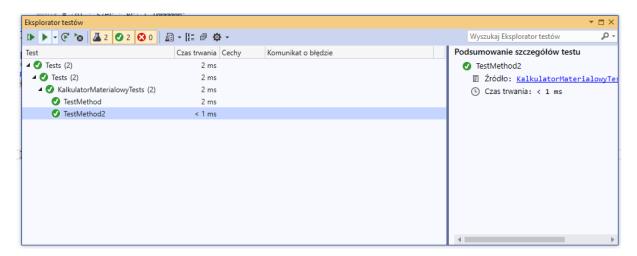
System kontroli wersji jaki użyłem do projektu to git, a platforma na której umieściłem swój kod to github.com Link do repozytorium:

https://github.com/mateuszbochnia/kalkulator\_materialowy

# 8. Raporty z testów

Przeprowadziłem dwa testy jednostkowe.

Moduł testowy zamieściłem w repozytorium. Tutaj umieściłem zrzut ekranu z pozytywnie przeprowadzonymi testami.



# 9. Materiały źródłowe

Przy realizacji projektu korzystałem z wiedzy nabytej podczas wykładów.

Również korzystałem z samouczków dostępnych na https://docs.microsoft.com/pl-pl/dotnet/csharp/tutorials/

Również do wielu elementów "doszedłem" metodą prób i błędów, szczególnie podczas rozwiązywania problemów z kodem.

# 10. Dokumentacja komentarzy:

Komentarze dodane przy pomocy programu Doxygen. Link do komentarzy:

 $\frac{https://drive.google.com/drive/folders/176iq21m6m5BkN05oEbxwy3Mlf8JMCqgp?usp=sharing}{}$