Zadanie 1. Owocowe niespodzianki.

Opis:

- Program losuje
 - nazwę owocu
 - jego obrazek
 - kolor tła aplikacji.

Po wywołaniu metody MouseDown()

<u>Wskazówki :</u>

Deklaracje tablic :

```
// Tablica owoców i ich obrazków
string[] fruits = { "Apple", "Banana", "Orange", "Strawberry", "Grapes" };
string[] img = { "apple.png", "banana.png", "orange.png", "strawberry.png", "grapes.png" };
// Tablica kolorów tła
Brush[] backgroundColors = { Brushes.LightGreen, Brushes.LightYellow, Brushes.LightCoral,
Brushes.LightPink, Brushes.LightSalmon };
```

// Zmiana koloru tła
this.Background = backgroundColors[index];

// Zmiana obrazka w Image kontrolce
fruitImage.Source = new BitmapImage(new Uri(img[index], UriKind.Relative));

Widok startowy :





Zadanie 2. Formatowanie tekstu.

Zadanie polega na stworzeniu okna aplikacji WPF (Windows Presentation Foundation) w języku XAML, które umożliwia użytkownikowi formatowanie tekstu w zależności od wybranych opcji.

Aplikacja pozwala na:

- 1. Wprowadzenie wartości liczbowej do pola tekstowego (TextBox).
- 2. Wybór stylu formatowania tekstu poprzez rozwijane menu (ComboBox), gdzie dostępne są dwie opcje:
 - Prosty tekst,
 - Pochyły tekst.
- 3. Zmianę koloru tekstu poprzez kliknięcie jednego z trzech przycisków (Button), które reprezentują kolory:
 - Czerwony,
 - o Zielony,
 - Niebieski.

4. Wyświetlenie sformatowanego tekstu w polu tekstowym (TextBlock), który domyślnie zawiera przykład tekstu do formatowania. Na podstawie wyborów użytkownika (stylu i koloru) tekst ten zmienia swój wygląd.

Każdy przycisk zmienia kolor tekstu, a wybór stylu z ComboBox zmienia wygląd czcionki (np. pochyły dla opcji "Pochyły").

Formatowanie tekstu		(\times
	%		
	Prosty ~		
	Jakim kolorem sformatować tekst> Wybierz przycisk.		
	To jest przykład paragrafu, który może być dowolnie formatowany przez WPF		
Formatowanie tekstu		_	×
	200 %		
	Prosty ~		
	Jakim kolorem sformatować tekst> Wybierz przycisk.		

To jest przykład paragrafu, który może być dowolnie formatowany przez WPF

Formatowanie tekstu

200		%	
Pochyły	~		

Jakim kolorem sformatować tekst> Wybierz przycisk.



To jest przykład paragrafu, który może być dowolnie formatowany przez WPF

Zadanie 3. Kelwiny - Fahrenheity

Zadanie polega na stworzeniu aplikacji WPF, która przelicza temperaturę podaną w stopniach Celsjusza na inne skale: Kelvin lub Fahrenheit. Aplikacja zawiera prosty interfejs użytkownika z polami wejściowymi i przyciskami do obsługi przeliczania temperatury.

Funkcjonalność aplikacji obejmuje:

- 1. Pole tekstowe do wprowadzenia temperatury w stopniach Celsjusza.
- Przyciski wyboru skali dwa przyciski, które pozwalają użytkownikowi przeliczyć temperaturę na:
 - Kelviny,
 - Fahrenheity.
- 3. Przetwarzanie danych wejściowych:
 - Użytkownik wprowadza wartość liczbową w stopniach Celsjusza.
 - Po kliknięciu jednego z przycisków, aplikacja przelicza temperaturę odpowiednio na stopnie Kelvina lub Fahrenheita.

- Wynik jest wyświetlany w odpowiedniej jednostce w postaci sformatowanej (do dwóch miejsc po przecinku).
- 4. Obsługa błędów: w przypadku wprowadzenia nieprawidłowych danych (np. litery zamiast liczby), wyświetlany jest komunikat o błędzie.

Aby przeliczyć temperaturę w stopniach Celsjusza (t) na temperaturę w stopniach Fahrenheita (T) należy skorzystać ze wzoru: $T = 9/5 \cdot t + 32$.

Aby przeliczyć temperaturę w stopniach Celsjusza (t) na temperaturę w stopniach Kelwiny należy skorzystać ze wzoru:

t [°C] = t [K] + 273.15

🔳 Celcjusze		\times
Wp	rowadź temperaturę w stopniach Celsjusza:	
	Zmień na:	
	Kelviny Fahrenheity	



Wprowadź	temperaturę	w stopniach	Celsjusza:
----------	-------------	-------------	------------

30	
	Zmień na:
Kelvin	Fahrenheity
	303,15 K
🔳 Celcjusze	

Wprowadź temperaturę w stopniach Celsjusza:

30					
Zn	nień na:				
Kelviny Fahrenheity					
86,00 F					

Celcjusze	ि		R (0	} 🛞	\checkmark	< _	\times

Wprowadź temperaturę w stopniach Celsjusza:

	Zmień na:		×
	Kelviny Fahrenheity	Błąd danych	
		ОК	
🔳 Celcjusze) 🔽 🗖 🛱 🐼 🕢	<	×

Wprowadź temperaturę w stopniach Celsjusza:

WW	
Zmień na:	×
Kelviny Fahrenheity	Błąd danych
	ОК

Zadanie 3.2.

a) Utworzyć aplikację Obliczenia z GUI jak pokazano na rysunku 3.15.

•	Obliczenia	_ 🗆 🗙
Wpisz liczby:		Wskaż działanie:
X:		⊖ X+Y
×		⊖ X-Y
т		○ X*Y
		O X∕Y
Oblicz Wynik:		

Rys.3.15. Widok formularza obliczeniowego

- b) zdefiniować delegata Dzialanie oraz odpowiadające mu metody Dodawanie, Odejmowanie, Mnozenie i Dzielenie.
- c) dodać obsługę przycisku Oblicz, który w zależności od wybranego działania utworzy odpowiedni obiekt delegata i wyświetli wynik działania wskazanego .przez użytkownika,

	Obliczenia	_ 🗖	x
Wpisz liczby: X: 1234 Y: 0		Wskaż działa X+Y X-Y X'Y X'Y	anie:
Oblicz Wynik	c: Dzielenie przez 0!		

Rys.3.16. Widok formularza obliczeniowego

•	Obliczenia	_ 🗆 🗙
Wpisz liczby: X: 10 Y: 10		Wskaż działanie: X+Y X-Y X*Y X*Y XY
Oblicz Wynik	: Iloczyn: 100	

Zadanie 4. Zgadnij Liczbę

Aplikacja **"zgadnij liczbę"**, w której użytkownik będzie próbował odgadnąć losowo wygenerowaną liczbę. Program będzie informować, czy podana przez użytkownika liczba jest za duża, za mała, czy poprawna.

Użytkownik ma możliwość wpisania liczby, a aplikacja generuje losową liczbę w zakresie od 1 do 100. Na podstawie podanej liczby użytkownik otrzymuje wskazówki, aż w końcu odgadnie liczbę.

Krok 1: Projektowanie interfejsu w XAML

W pliku XAML możemy stworzyć prosty interfejs składający się z:

- pola do wpisywania liczby,
- przycisku do sprawdzania liczby,
- MessageBox() do wyświetlania wyniku (czy liczba jest za duża, za mała lub poprawna).

Wskazówki:

- 1. Losowanie liczby:
 - W konstruktorze klasy MainWindow generujemy losową liczbę z zakresu od 1 do 100 za pomocą random.Next(1, 101).
- 2. Sprawdzanie liczby:
 - Gdy użytkownik kliknie przycisk "Check Number", aplikacja pobiera liczbę z pola TextBox. Jeśli wprowadzona liczba jest poprawna (parsujemy za pomocą int.TryParse()), sprawdzamy, czy jest większa, mniejsza, czy równa wylosowanej liczbie.
- 3. Informowanie użytkownika:
 - Wynik porównania jest wyświetlany w MessageBox(). Użytkownik otrzymuje wskazówki - za duża lub za mała, aż do momentu, gdy poda właściwą liczbę, co wywołuje komunikat "Congratulations!".

Rozszerzenie gry (opcjonalne)

Aby rozbudować grę, można dodać:

• Licznik prób, aby informować użytkownika, ile razy próbował odgadnąć.

- Przycisk do ponownej gry, który generuje nową losową liczbę bez potrzeby ponownego uruchamiania aplikacji.
- Zakres trudności, aby umożliwić wybór trudniejszego zakresu (np. 1-500).

Zadanie 6.

Jakiś czas temu popularny był żart, rozsyłany między innymi przez szefów do pracowników. Zawierał pytanie "Czy chcesz podwyżkę" poniżej były dwa przyciski: TAK i NIE, żart polegał na tym, że gdy próbowaliśmy najechać myszą na przycisk TAK, on oddalał się (uciekał od wskaźnika myszy). Wykonaj taki program w wersji podmiany przycisków w momencie najechania.

wskazówka:

```
var tmpMargin = btnYes.Margin; // Zapamiętanie początkowego położenia przycisku
Tak
btnYes.Margin = btnNo.Margin; // Przycisk Tak przyjmuje położenie przycisku Nie
btnNo.Margin = tmpMargin; // Przycisk Nie przyjmuje początkowe położenie
przycisku Tak
```