Przedmiot: Pracownia Programowania Aplikacji (WPF / C#) Imię i nazwisko: .....
Kwalifikacja: INF.04 / Technik Programista Klasa: ..... Data: ..... Stanowisko: .....

KARTA PRACY: Interaktywny Mikser Kolorów RGB

Ćwiczenie laboratoryjne (2 godziny lekcyjne – 90 minut) • Kontrolki Slider i ProgressBar w WPF

Cel zajęć: Projektowanie responsywnego interfejsu graficznego w WPF. Praktyczne zastosowanie kontrolek zakresowych Slider oraz ProgressBar, obsługa zdarzenia ValueChanged, eliminacja błędu NullReferenceException podczas inicjalizacji komponentów, a także dynamiczne renderowanie koloru za pomocą SolidColorBrush i formatowanie wartości HEX.

ETAP 1: Projekt interfejsu użytkownika w XAML (ok. 30 min)

Utwórz w Visual Studio nowy projekt WPF Application (.NET 8.0 lub .NET Framework) o nazwie MikserRGB_Nazwisko. W pliku MainWindow.xaml skonfiguruj okno: Title="Mikser Kolorów RGB", Width="460", Height="480", ResizeMode="NoResize", WindowStartupLocation="CenterScreen". Zbuduj interfejs zawierający następujące komponenty ułożone w kontenerze StackPanel lub Grid:

Element	Nazwa (x:Name)	Właściwości i parametry w XAML	Rola w aplikacji
Border	`borderPodglad`	`Height="90", Background="Black", CornerRadius="6", BorderBrush="#94A3B8", BorderThickness="2", Margin="0,0,0,15"`	Duży prostokąt podglądu wymieszanego koloru RGB.
Slider R, G, B (3 sztuki)	`slR, slG, slB`	`Minimum="0" Maximum="255" Value="0" TickFrequency="1" IsSnapToTickEnabled="True" ValueChanged="SuwakRGB_ValueChanged"`	Suwaki sterujące nasyceniem barw: Czerwonej (R), Zielonej (G) i Niebieskiej (B). Tylko liczby całkowite!
ProgressBar (3 sztuki)	`pbR, pbG, pbB`	`Minimum="0" Maximum="255" Value="0" Height="8" Margin="0,2,0,8"`	Wizualny pasek postępu umieszczony pod suwakiem, pokazujący stopień wypełnienia kanału.
TextBlock (wartości)	`lblR, lblG, lblB`	`Text="0", Width="35", TextAlignment="Right", FontWeight="Bold"`	Etykiety liczbowe wyświetlające aktualną wartość (0-255).
TextBlock (HEX)	`txtHexKod`	`Text="#000000", FontSize="16", FontWeight="Bold", HorizontalAlignment="Center"`	Pole wyświetlające kod koloru w formacie szesnastkowym.
Button	`btnKopiuj`	`Content="Kopiuj kod HEX", Width="140", Height="28", Click="btnKopiuj_Click"`	Przycisk kopiujący wartość koloru do schowka Windows.

□□ KLUCZOWA WSKAZÓWKA CKE (Wyeliminowanie ułamków na suwaku):
Domyślny Slider zwraca wartości typu double z wieloma cyframi po przecinku (np. 142.8471). Aby wymusić ruch suwaka co dokładnie 1 jednostkę całkowitą, koniecznie ustaw parę atrybutów: TickFrequency="1" oraz IsSnapToTickEnabled="True"!

```
<!-- Wzorcowy blok dla jednego kanału (np. R) -->
<DockPanel Margin="0,0,0,2">
    <TextBlock Text="R (Czerwony):" Width="110" FontWeight="Bold" />
    <TextBlock x:Name="lblR" Text="0" Width="35" DockPanel.Dock="Right" TextAlignment="Right" />
    <Slider x:Name="slR" Minimum="0" Maximum="255" TickFrequency="1" IsSnapToTickEnabled="True"
            ValueChanged="SuwakRGB_ValueChanged" />
</DockPanel>
<ProgressBar x:Name="pbR" Minimum="0" Maximum="255" Value="0" Height="6" Margin="0,0,0,10" />
```

ETAP 2: Implementacja logiki biznesowej w C# (ok. 35 min)

Otwórz plik MainWindow.xaml.cs. Twoim zadaniem jest zaprogramowanie jednej wspólnej metody SuwakRGB_ValueChanged, która obsługuje zdarzenie przesuwania dla wszystkich trzech suwaków:

```
using System.Windows;
using System.Windows.Media; // Niezbędne dla: SolidColorBrush oraz Color!

private void SuwakRGB_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
{
    // 1. STRAŻNIK (Null-Check): Chroni przed błędem podczas startu okna!
    if (slR == null || slG == null || slB == null || borderPodglad == null) return;

    // 2. Pobranie wartości i rzutowanie z double na byte (zakres 0 - 255)
    byte r = (byte)slR.Value;
    byte g = (byte)slG.Value;
    byte b = (byte)slB.Value;

    // 3. Aktualizacja pasków postępu (ProgressBar) oraz etykiet tekstowych
    pbR.Value = r;  lblR.Text = r.ToString();
    pbG.Value = g;  lblG.Text = g.ToString();
    pbB.Value = b;  lblB.Text = b.ToString();

    // 4. Utworzenie koloru i pomalowanie tła podglądu za pomocą SolidColorBrush
    Color wybranyKolor = Color.FromRgb(r, g, b);
    borderPodglad.Background = new SolidColorBrush(wybranyKolor);

    // 5. Wygenerowanie i wyświetlenie kodu w formacie szesnastkowym (HEX)
    txtHexKod.Text = $"#{r:X2}{g:X2}{b:X2}";
}
```

 DLACZEGO STRAŻNIK JEST KONIECZNY? (Pytanie z obrony projektu):
Gdy okno się ładuje, metoda InitializeComponent() czyta plik XAML od góry do dołu. Gdy napotka pierwszy suwak slR z atrybutem Value="0", natychmiast odpala zdarzenie ValueChanged. W tym momencie suwaki slG i slB jeszcze nie istnieją w pamięci RAM (są równe null)! Brak sprawdzenia if (slR == null || ...) spowoduje natychmiastowy crash aplikacji (NullReferenceException).

ETAP 3: Zadania rozszerzone dla dociekliwych (na ocenę 5 / 6) (ok. 15 min)

1. Kopiowanie do schowka: Oprogramuj przycisk btnKopiuj_Click tak, aby za pomocą klasy Clipboard.SetText(txtHexKod.Text); skopiować kod do schowka i wyświetlić komunikat w MessageBox.Show.
2. Przyciski presetów kolorów: Dodaj poziomy panel (WrapPanel lub StackPanel) z 4 przyciskami szybkiego wyboru: Czerwony (255,0,0), Zielony (0,255,0), Niebieski (0,0,255), Żółty (255,255,0). Kliknięcie przycisku ma przestawić wartości właściwości Value suwaków (np. slR.Value = 255;).

KRYTERIA OCENY I PUNKTACJA (Maksymalnie 15 punktów)

Lp.	Kryterium sprawdzane przez nauczyciela	Maks. pkt	Zdobyte
1.	Poprawny układ XAML: zdefiniowano Border, 3 suwaki, 3 paski postępu, etykiety wartości oraz pole HEX.	**3 pkt**	..... / 3
2.	Właściwa konfiguracja Sliderów: TickFrequency="1" oraz IsSnapToTickEnabled="True" (tylko int).	**2 pkt**	..... / 2
3.	Implementacja ValueChanged ze „strażnikiem” null – aplikacja nie crashuje przy starcie.	**3 pkt**	..... / 3
4.	Dynamiczna zmiana koloru tła za pomocą SolidColorBrush + synchronizacja ProgressBar.	**3 pkt**	..... / 3
5.	Poprawne generowanie i formatowanie wartości szesnastkowej HEX (np. #FF0000).	**2 pkt**	..... / 2
6.	Zadanie rozszerzone: Obsługa schowka (Clipboard) lub przyciski presetów barw.	**2 pkt**	..... / 2
	SUMA PUNKTÓW I OCENA: 0-6 pkt: ndst (1) | 7-8 pkt: dop (2) | 9-10 pkt: dst (3) | 11-12 pkt: db (4) | 13-14 pkt: bdb (5) | 15 pkt: cel (6)	**15 pkt**	**..... / 15** **Ocena: .....**