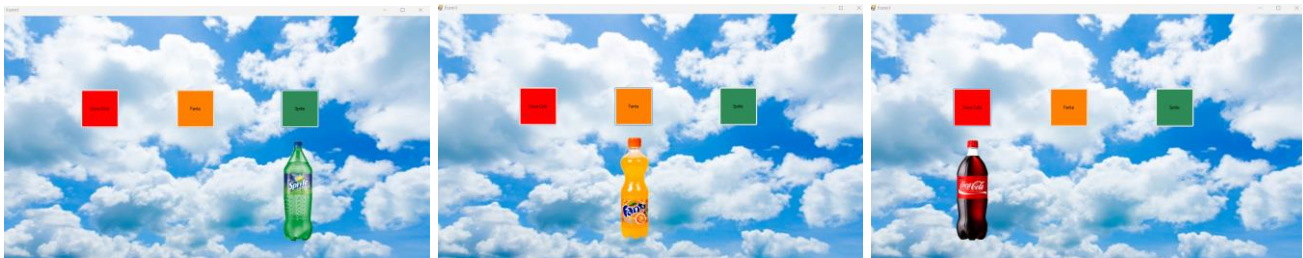


Zadanie 3. (zadanie powtórzeniowe zaliczeniowe L2)



Na podstawie zrzutu zaprojektuj aplikacje w XAML (odzwierciedlić ustawienia + dodatkowo można dołożyć coś od siebie)

C# na poszczególnych buttonach pokazuje krokodyla, słonia i zamyka aplikacje.



Na podstawie zrzutu zaprojektuj aplikacje w XAML (odzwierciedlić ustawienia + dodatkowo można dołożyć coś od siebie)

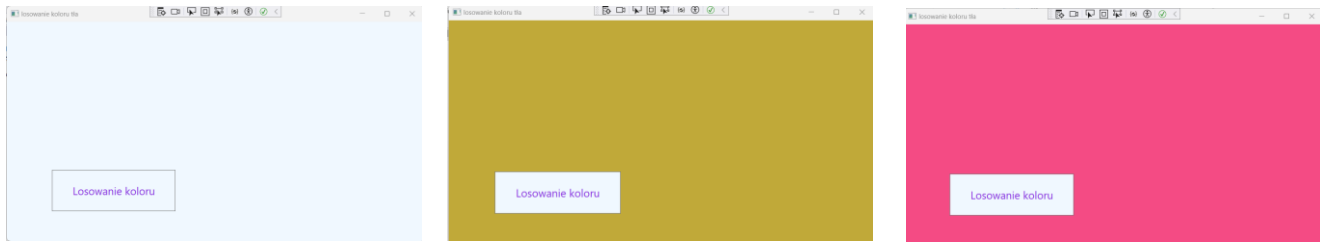
Zadanie 2

Stwórz aplikację okienkową w języku XAML i C#, która będzie losować kolor tła w obszarze okna głównego po kliknięciu przycisku "Losowanie koloru".

Aplikacja powinna zawierać przycisk o nazwie "Losowanie koloru" umieszczony na błękitnym tle.

Po kliknięciu tego przycisku, kolor tła całego okna głównego powinien zmieniać się losowo.

Dodatkowo na starcie , kolor przycisku powinien być w takim samym odcieniu niebieskiego jak tło np. „AliceBlue”, kolor czcionki „BlueViolet”



Wskazówki :

```
Random random = new Random();
byte r= (byte)random.Next(256);
byte g= (byte)random.Next(256);
byte b= (byte)random.Next(256);

Color randomColor = Color.FromRgb(r, g, b);
SolidColorBrush color = new SolidColorBrush(randomColor);

grid.Background=color;
```

Color to struktura reprezentująca kolor w przestrzeni kolorów RGB.

FromRgb(r, g, b) to metoda, która tworzy obiekt **Color** na podstawie trzech składowych koloru

SolidColorBrush to rodzaj pędzla, który reprezentuje jednolity kolor lub wzór w WPF.

new SolidColorBrush(randomColor) tworzy nowy obiekt **SolidColorBrush**, który ma kolor określony przez randomColor

Zadanie 3

Przycisk "zmień" ma spowodować zmianę tła formularza na losowy kolor z palety RGB.

Na TextBlock wyświetla się informacja z wylosowanym kolorem tła w postaci szesnastkowej.

Część dodatkowa:

Dodatkowo można sterować kolorem napisu dla ciemnego tła kolor napisu biały dla jasnego czarny.

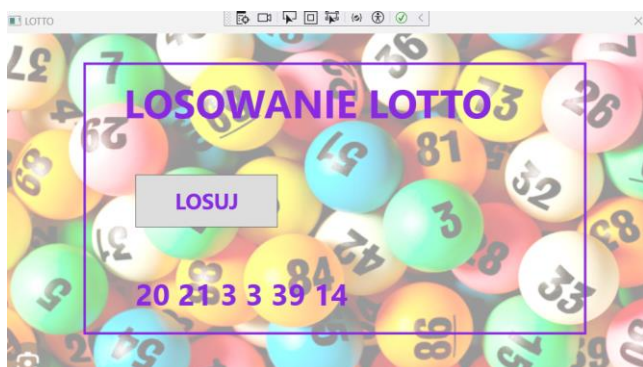
Zadanie 4.

Aplikacja ma realizować prostą funkcjonalność losowania z duplikatami liczb w aplikacji WPF i aktualizować wyświetlane liczby po kliknięciu przycisku "LOSUJ".

Interfejs graficzny aplikacji "LOTTO" :

1. Okno główne o tytule "LOTTO" o wymiarach 800x450 pikseli, zabezpieczone przed zmianą szerokości i wysokości.
2. Tło okna powinno być zdefiniowane jako obraz o nazwie "lotto.png" umieszczony w tle, z przezroczystością ustawioną na 40%.
3. Umieść na oknie głównym siatkę (**Grid**), która będzie zawierała następujące elementy:
 - a) Ramkę (**Frame**) umieszczoną z lewej strony okna, o wysokości 329 pikseli i szerokości 608 pikseli, z obramowaniem w kolorze BlueViolet o grubości 3 pikseli.
 - b) Etykietę (**Label**) o treści "LOSOWANIE LOTTO", wyśrodkowaną poziomo, o wysokości 75 pikseli i szerokości 502 pikseli, z rozmiarem czcionki 48 punktów, pogrubioną, w kolorze tekstu BlueViolet.
 - c) Przycisk (**Button**) o treści "LOSUJ", umieszczony z lewej strony, o wysokości 64 pikseli i szerokości 172 pikseli, z rozmiarem czcionki 26 punktów, pogrubiony, w kolorze tekstu BlueViolet. Przycisk będzie reagował na kliknięcie poprzez wywołanie zdarzenia **Button_Click**.
 - d) Blok tekstu (**TextBlock**) o nazwie **liczby**, umieszczony z lewej strony, o wysokości 64 pikseli i szerokości 499 pikseli, z rozmiarem czcionki 36 punktów, pogrubiony, w kolorze tekstu BlueViolet, który będzie wyświetlał wynik losowania liczb.

Zdjęcie poglądowe:



```
Random random = new Random();
String wylosowane="";

for(int i=0;i<6;i++)
{
    int liczba = random.Next(1, 50);
    wylosowane+= liczba.ToString() + " ";
}

liczby.Text = wylosowane;
```

Zadanie 4.2.

Do powyższej aplikacji dołącz funkcję, która realizuje losowanie bez duplikatów.

```
Random random = new Random();  
1 odwołanie  
private void los_bez_duplikatow()  
{  
    HashSet<int> set = new HashSet<int>();  
  
    while (set.Count() != 6)  
    {  
        int liczba = random.Next(1, 50);  
        set.Add(liczba);  
    }  
  
    String wylos = string.Join(" ", set);  
  
    liczby.Text = wylos;  
}
```

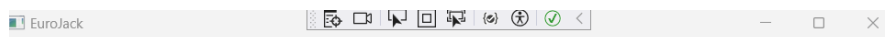
- HashSet<int> to zbiór, który przechowuje tylko unikalne wartości. Oznacza to, że każda liczba może zostać dodana tylko raz. To automatycznie zapewnia brak duplikatów w zestawie wylosowanych liczb.
- string.Join(" ", set) konwertuje liczby w zbiorze set na pojedynczy ciąg znaków, w którym liczby są oddzielone spacją.

Zadanie 5 losowanie liczb w stylu EuroJackpot.

Zbudujmy prostą aplikację symulującą losowanie liczb w stylu EuroJackpot w technologii WPF.

W EuroJackpot gracz wybiera 5 liczb z puli 50 oraz 2 dodatkowe liczby (tzw. „Euro Numery”) z puli 10.

Symulujemy losowanie tych liczb, a następnie wyświetlamy wynik użytkownikowi.



Losuj liczby

Wylosowane liczby: 14, 18, 6, 23, 16
Euro Numery: 3, 4



Losuj liczby

Wylosowane liczby: 16, 13, 45, 11, 40
Euro Numery: 4, 3

Rozbudowa (opcjonalne):

- **Liczba prób:** Można dodać licznik pokazujący, ile razy użytkownik losował liczby.
- **Historia losowań:** Można zaimplementować funkcjonalność przechowywania historii ostatnich losowań.
- **Symulacja wielu losowań:** Aplikacja mogłaby umożliwiać symulację wielu losowań na raz, co pozwoliłoby zbadać prawdopodobieństwo wygranej