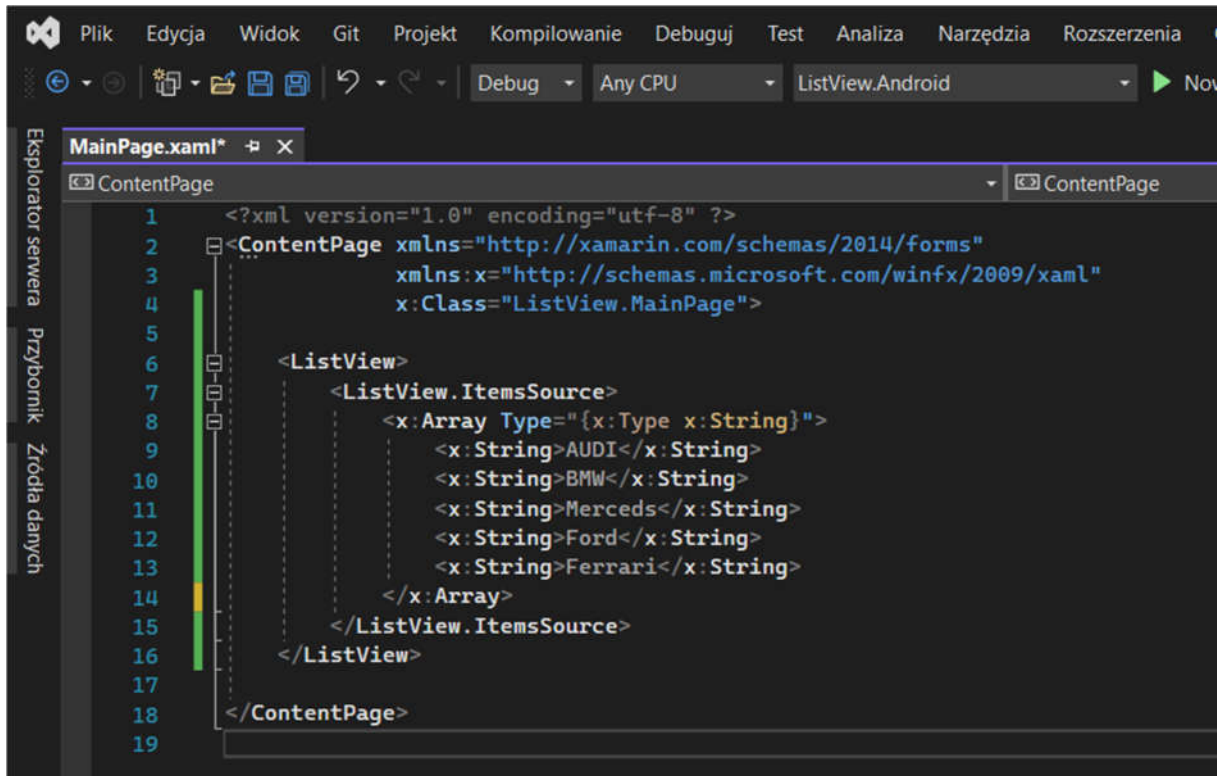


1. Do czego wykorzystuje się ListView?

2. Za co odpowiada znacznik ListView? Jakie właściwości może obsługiwać?

3. Przepisz i przeanalizuj poniższy kod. Za co odpowiada kod w <ListView> dokonaj odpowiedniego opisu. Np. za co odpowiada ItemSource



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             x:Class="ListView.MainPage">
5
6     <ListView>
7         <ListView.ItemsSource>
8             <x:Array Type="{x:Type x:String}">
9                 <x:String>AUDI</x:String>
10                <x:String>BMW</x:String>
11                <x:String>Merceds</x:String>
12                <x:String>Ford</x:String>
13                <x:String>Ferrari</x:String>
14            </x:Array>
15        </ListView.ItemsSource>
16    </ListView>
17
18 </ContentPage>
19
```

Za co odpowiada: `<x:Array Type="{x:Type x:String}">`

4. Przedstaw efekt swojej pracy na emulatorze UWP oraz Android.

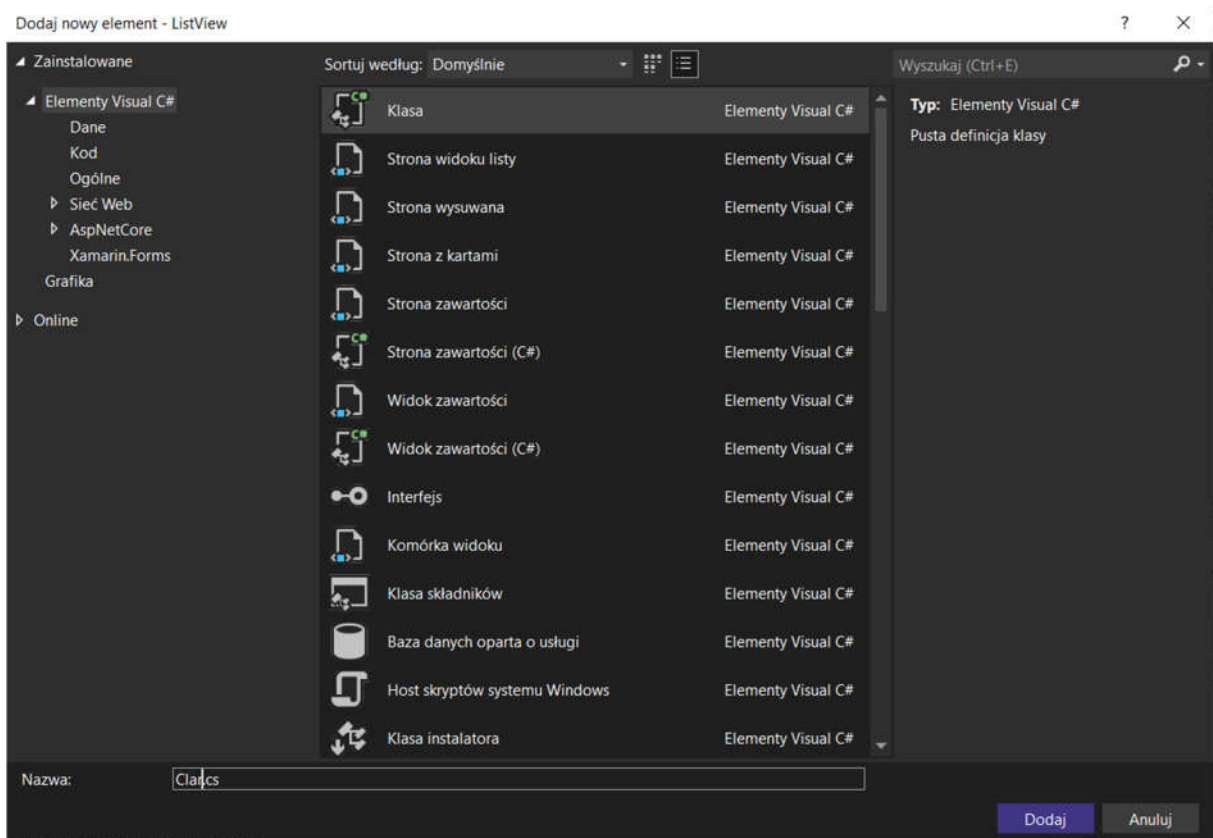
5. W znaczniku ListView możemy modyfikować ustawienia separatora, np. wyłączyć go, zmienić jego kolor.

Zanotuj jak zmienisz kolor separatora:

Za pomocą jakiego polecenia ukryjesz separator:

Dodaj kilka swoich modyfikacji do kodu. Za co one odpowiadają:

6. W kolejnych krokach postaramy się rozszerzyć naszą listę o grupy obiektów, nie tylko tekst. Na początek w Solution Explorer dodajmy sobie nowy folder o nazwie Models, będziemy w nim przechowywać wszystkie modele danych naszej aplikacji. Tworzymy w nim nową klasę, która będzie reprezentowała nasz samochód. Nazwiemy ją Car.cs



Klasę zmieniamy na publiczną i dodajemy do niej pola Name oraz Year.

```
using System;
using System.Collections.Generic;
using System.Text;

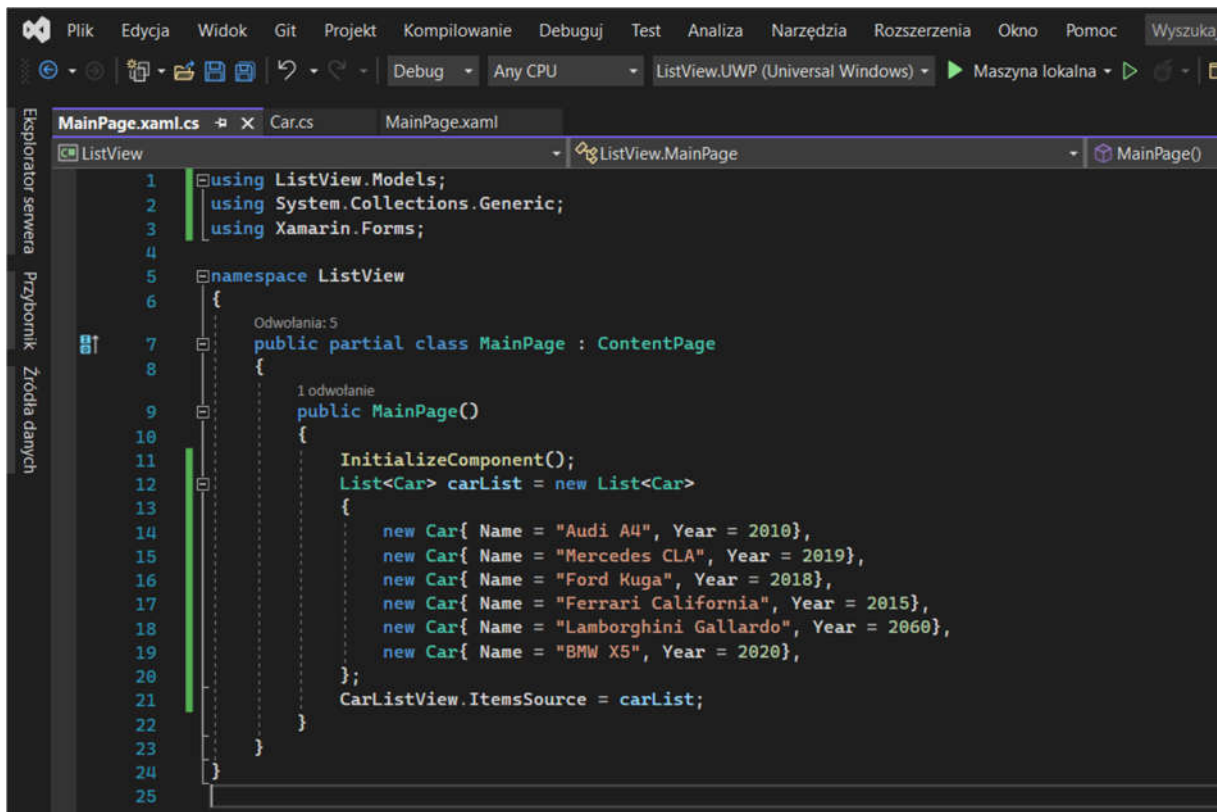
namespace ListView.Models
{
    public class Car
    {
        public int Name { get; set; }
        public int Year { get; set; }
    }
}
```

7. Za co odpowiadają accessory? W jakim celu je stosujemy? Zanotuj długi i krótki sposób ich zapisu. Stwórz własny oryginalny przykład, aby zaprezentować ich działanie.

W znaczniku `ListView` kasujemy dotychczasową zawartość. Dodamy mu nazwę abyśmy mogli się do niego odwołać z `CodeBehind`

```
<ListView x:Name="CarListView">
</ListView>
```

Przejdźmy do `CodeBehind` `MainPage.xaml.cs` gdzie zadeklarujemy sobie nową listę samochodów.



```
1 using ListView.Models;
2 using System.Collections.Generic;
3 using Xamarin.Forms;
4
5 namespace ListView
6 {
7     public partial class MainPage : ContentPage
8     {
9         public MainPage()
10        {
11            InitializeComponent();
12            List<Car> carList = new List<Car>
13            {
14                new Car{ Name = "Audi A4", Year = 2010},
15                new Car{ Name = "Mercedes CLA", Year = 2019},
16                new Car{ Name = "Ford Kuga", Year = 2018},
17                new Car{ Name = "Ferrari California", Year = 2015},
18                new Car{ Name = "Lamborghini Gallardo", Year = 2060},
19                new Car{ Name = "BMW X5", Year = 2020},
20            };
21            CarListView.ItemsSource = carList;
22        }
23    }
24 }
25
```

Po raz kolejny przeanalizuj kod, co nowego zostało zastosowane. Dokonaj odpowiedniego opisu. Za co dokładnie odpowiada linijka 21:  
`CarListView.ItemsSource = carList;`

Jeśli korzystasz z VS 2022, popraw następujący błąd:

```

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using Xamarin.Forms;
8
9 namespace ListView
10 {
11     public partial class MainPage : ContentPage
12     {
13         public MainPage()
14         {
15             InitializeComponent();
16             List<Car> carList = new List<Car>
17             {
18                 new Car { Year = 2010 },
19                 new Car { Year = 2011 }
20             };
21             CarListView.ItemsSource = carList;
22         }
23     }
24 }

```

The context menu is open over the `Car` type in line 16, showing options like `using ListView.Models;` and `using System;`. An error message is visible: `CS0246 Nie można znaleźć nazwy typu lub przestrzeni nazw „Car” (brak dyrektywy using lub odwołania do zestawu?)`.

8. Aby skompilować poprawnie kod należało poprawić inny błąd na czym on polegał ?

9. Uruchom emulator jaki efekt otrzymałeś/aś ? Czym jest to spowodowane?

10. Dokonujemy kolejnych zmian w kodzie:

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             x:Class="ListView.MainPage">
5
6     <ListView x:Name="CarListView">
7         <ListView.ItemTemplate>
8             <DataTemplate>
9                 <TextCell Text="{Binding Path=Name}" Detail="{Binding Year}"/>
10            </DataTemplate>
11        </ListView.ItemTemplate>
12    </ListView>
13 </ContentPage>

```

Po dopisaniu kolejnych linijek kodu uruchom emulator. Jaki efekt otrzymałeś/aś?

Za co odpowiada komórka `TextCell`