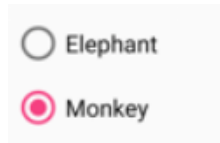


Работа с элементом управления RadioButton

Изучим работу с элементом управления RadioButton, который позволяет пользователям выбрать один вариант из нескольких возможных вариантов решения задачи, каждый из которых представлен одним элементом RadioButton.



Для использования элемента управления RadioButton необходимо в программном приложении Xamarin.Forms установить флаг `RadioButton_Experimental`.

Элемент управления RadioButton имеет следующие свойства:

свойство `IsChecked` определяет, выбран ли элемент управления RadioButton или не выбран (по умолчанию имеет значение `false` – не выбран);

свойство `GroupName` определяет имя группы элементов управления RadioButton, внутри которой может быть выбран только один элемент управления RadioButton (по умолчанию имеет значение `null`).

Элемент управления RadioButton также имеет некоторые свойства, аналогичные свойствам элемента управления Button:

`Command` - команда, которая выполняется при выборе RadioButton;

`CommandParameter` - параметр, который передается в команду `Command`;

`FontAttributes` - определяет стиль поясняющего текста для Radiobutton;

`FontFamily` - определяет семейство шрифтов для поясняющего текста для Radiobutton;

`FontSize` (тип `double`) - определяет размер шрифта в поясняющем тексте для Radiobutton;

`Content` – задает поясняющий текст для Radiobutton;

`TextColor` – задает цвет поясняющего текста для Radiobutton.

Элементу управления RadioButton соответствует событие `CheckedChanged`, которое возникает при изменении значения свойства `IsChecked` при изменении состояния элемента управления.

Событию `CheckedChanged` соответствует объект `CheckedChangedEventArgs`, который имеет свойство `Value` (тип `bool`).

При возникновении события `CheckedChanged` в качестве значения свойства `Value` принимается значение свойства `IsChecked`.

Элементы управления RadioButton обычно используются в составе групп, внутри которых может быть выбран лишь один элемент управления RadioButton.

Существует два подхода к группированию элементов управления RadioButton:

1. Неявная группировка. Помещение их в один и тот же макет.

Показаны фрагменты программного кода, соответствующие неявной группировке элементов управления RadioButton в XAML и Си Шарп

```
<StackLayout>
  <RadioButton Content ="Cat" />
  <RadioButton Content ="Dog" IsChecked="true" />
</StackLayout>
```

```
StackLayout stackLayout = new StackLayout
{
  Children =
  {
    new RadioButton { Content = "Cat" },
    new RadioButton { Content = "Dog", IsChecked = true }
  }
};
```

2. Явная группировка. Задание одного и того же значения свойства GroupName для элементов управления RadioButton, которых необходимо объединить в одну группу.

Показан фрагмент программного кода, соответствующий явной группировке элементов управления RadioButton путем задания значения свойства GroupName в XAML и Си Шарп:

```
<RadioButton Content="Red"
  TextColor="Red"
  GroupName="colors" />
<RadioButton Content ="Green"
  TextColor="Green"
  GroupName="colors" />
```

```
new RadioButton { Content = "Red", GroupName="colors" },
new RadioButton { Content = "Green", GroupName="colors", IsChecked = true }
```

Выбор элемента управления RadioButton можно отменить, выбрав другой элемент управления RadioButton в той же группе. Также выбор элемента управления RadioButton можно отменить в программном коде, установив для свойства IsChecked значение false.

При этом, нельзя отменить выбор элемента управления RadioButton с помощью повторного клика мыши или повторного касания пальцем.

Когда значение свойства IsChecked изменяется, вызывается событие CheckedChanged, которому соответствует обработчик события. В презентации обработчик события имеет название OnRadioButtonChanged.

```
<RadioButton Content ="Red"
  TextColor="Red"
  GroupName="colors"
  CheckedChanged="OnRadioButtonChanged" />
```

Для реагирования на изменение состояния `RadioButton` необходимо создать обработчик события, общий вид которого следующий:

```
void OnRadioButtonChanged (object sender,  
                             CheckedChangedEventArgs e)  
{  
    // Операции, выполняемые при вызове события CheckedChanged  
}
```

Аргумент `sender` соответствует элементу управления `RadioButton`, с которым происходит событие. Его можно использовать для доступа к обработчику события нескольким элементам управления `RadioButton`, совместно использующих один и тот же обработчик события `CheckedChanged`.

Кроме того, обработчик события `CheckedChanged` может быть зарегистрирован непосредственно в программном коде:

```
radioButton.CheckedChanged += (sender, e) =>  
{  
    // Операции, выполняемые при вызове события CheckedChanged  
};
```

Для работы с элементом управления `RadioButton` создайте проект `RadioButtonDemo` для мультиплатформенного приложения `Xamarin.Forms` с использованием шаблона «Пустой».

Откройте файл `App.xaml.cs` и наберите программный код, приведенный в презентации.

В программном коде производится установка флага `RadioButton_Experimental`, что делает возможным использование элементов управления `RadioButton`.

```
using Xamarin.Forms;  
  
namespace RadioButtonDemo  
{  
    public partial class App : Application  
    {  
        public App()  
        {  
            InitializeComponent();  
  
            Device.SetFlags(new string[] { "RadioButton_Experimental" });  
            MainPage = new MainPage();  
        }  
    }  
}
```

Откройте файл `MainPage.xaml` и наберите программный код, приведенный в презентации.

Программный код показывает, что в странице типа `ContentPage` содержится макет `StackLayout`, внутри которого находятся:

элемент управления Label с названием страницы «Работа с RadioButton»;
элемент управления Label для обозначения блока элементов управления
RadioButton для выбора цвета;

группа из четырех элементов управления RadioButton для выбора цвета;
элемент управления Label для отображения выбранного цвета, который
с помощью атрибута x:Name и имени colorLabel связан с обработчиком
события OnColorChanged, программный код которого приведен в файле
MainPage.xaml.cs.;

элемент управления Label для обозначения блока элементов управления
RadioButton для выбора фруктов;

группа из четырех элементов управления RadioButton для выбора
фруктов;

элемент управления Label для отображения выбранного фрукта,
который с помощью атрибута x:Name и имени fruitLabel связан с
обработчиком события OnFruitChanged, программный код которого приведен
в файле MainPage.xaml.cs.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="RadioButtonDemo.MainPage"
  BackgroundColor="LightBlue">
  <StackLayout Margin="10">
    <Label Text="Работа с RadioButton"
      TextColor="Black"
      FontSize="50"
      FontAttributes="Bold"
      HorizontalOptions="Center" />
    <Label Text="Какой Ваш любимый цвет?"
      FontAttributes="Bold"
      FontSize="20" TextColor="Brown"/>
    <RadioButton Content = "Red"
      FontSize="16" FontAttributes="Bold"
      TextColor="Red"
      GroupName="colors"
      CheckedChanged="OnColorChanged" />
    <RadioButton Content = "Green"
      FontSize="16" FontAttributes="Bold"
      TextColor="Green"
      GroupName="colors"
      CheckedChanged="OnColorChanged" />
    <RadioButton Content = "Blue"
      FontSize="16" FontAttributes="Bold"
      TextColor="Blue"
      GroupName="colors"
      CheckedChanged="OnColorChanged" />
    <RadioButton Content = "Other"
      FontSize="16" FontAttributes="Bold"
      GroupName="colors"
```

```

        CheckedChanged="OnColorChanged" />
<Label x:Name="colorLabel"
    FontSize="20" FontAttributes="Bold"
    TextColor="Brown"
    Text="Ваш любимый цвет:" />
<Label Text="Какой Ваш любимый фрукт?"
    FontAttributes="Bold"
    FontSize="20" TextColor="DarkMagenta"/>
<RadioButton Content ="Apple"
    FontSize="16" FontAttributes="Bold"
    GroupName="fruits"
    CheckedChanged="OnFruitChanged" />
<RadioButton Content ="Banana"
    FontSize="16" FontAttributes="Bold"
    GroupName="fruits"
    CheckedChanged="OnFruitChanged" />
<RadioButton Content ="Pineapple"
    FontSize="16" FontAttributes="Bold"
    GroupName="fruits"
    CheckedChanged="OnFruitChanged" />
<RadioButton Content ="Other"
    FontSize="16" FontAttributes="Bold"
    GroupName="fruits"
    CheckedChanged="OnFruitChanged" />
<Label x:Name="fruitLabel"
    FontAttributes="Bold"
    FontSize="20" TextColor="DarkMagenta"
    Text="Ваш любимый фрукт:" />
</StackLayout>
</ContentPage>

```

Откройте файл MainPage.xaml.cs и наберите программный код для обработчиков события CheckedChanged, на которые происходит ссылка из файла MainPage.xaml.

Обработчик OnColorChanged принимает в качестве параметра sender элемент управления RadioButton из группы colors, который был включен для выбора цвета. Обработчик производит изменение текста, отображаемого в элементе управления Label с помощью имени colorLabel, переданного из файла MainPage.xaml.

Обработчик OnFruitChanged принимает в качестве параметра sender элемент управления RadioButton из группы fruits, который был включен для выбора фрукта. Обработчик производит изменение текста, отображаемого в элементе управления Label с помощью имени fruitLabel, переданного из файла MainPage.xaml.

```
using Xamarin.Forms;
```

```
namespace RadioButtonDemo
```

```

{
    public partial class MainPage: ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
        void OnColorChanged(object sender, CheckedChangedEventArgs e)
        {
            RadioButton button = sender as RadioButton;
            colorLabel.Text = $"Ваш любимый цвет: {button.Content}";
        }
        void OnFruitChanged(object sender, CheckedChangedEventArgs e)
        {
            RadioButton button = sender as RadioButton;
            fruitLabel.Text = $"Ваш любимый фрукт: {button.Content}";
        }
    }
}

```

Запустите программное приложение для работы с элементом управления `RadioButton`. В пользовательском интерфейсе отображаются две группы элементов управления `RadioButton`. При выборе цвета изменяется текст, располагающийся под первой группой элементов управления `RadioButton`. При выборе фрукта изменяется текст, расположенный под второй группой элементов управления `RadioButton`.

Элементы управления `RadioButton` в обеих группах являются взаимоисключающими. То есть, внутри каждой группы во включенном состоянии может быть только один элемент управления `RadioButton`.

