



ObservableCollection

Последнее обновление: 29.03.2017



| | | | | | | | |
|--|------------|------------|---------|------------|-------------|------------|--------------------|
| | | | | | | | |
| | 40,85 руб. | 60,14 руб. | 84 руб. | 63,65 руб. | 107,60 руб. | 50,80 руб. | 70 руб. "Триовист" |

При работе со списком объектов мы можем столкнуться с проблемой добавления или удаления объектов в этом списке. Так, возьмем пример из прошлой темы и добавим в код визуального интерфейса пару кнопок, которые будут выполнять добавление и удаление:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:local="clr-namespace:HelloApp;assembly=HelloApp"
  x:Class="HelloApp.MainPage">
  <StackLayout>
    <Label Text="{Binding Source={x:Reference Name=phonesList}, Path=SelectedItem.Title}"
      FontSize="Large" />
    <ListView x:Name="phonesList"
      HasUnevenRows="True"
      ItemsSource="{Binding Phones}">
      <ListView.ItemTemplate>
        <DataTemplate>
          <ViewCell>
            <ViewCell.View>
              <StackLayout>
                <Label Text="{Binding Title}" FontSize="Large" />
                <Label Text="{Binding Company}" />
                <Label Text="{Binding Price}" />
              </StackLayout>
            </ViewCell.View>
          </ViewCell>
        </DataTemplate>
      </ListView.ItemTemplate>
    </ListView>
    <StackLayout Orientation="Horizontal" HorizontalOptions="Center" >
      <Button Clicked="AddItem" Text="Добавить" />
      <Button Clicked="RemoveItem" Text="Удалить" />
    </StackLayout>
  </StackLayout>
</ContentPage>
```

В файл кода C# добавим к странице обработчики кнопок:

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using Xamarin.Forms;

namespace HelloApp
{
  public partial class MainPage : ContentPage
  {
    public List<Phone> Phones { get; set; }

    public MainPage()
    {
      InitializeComponent();
      Phones = new List<Phone>
      {
        new Phone { Title = "HTC U Ultra", Company = "HTC", Price = 36000 },
        new Phone { Title = "Huawei P10", Company = "Huawei", Price = 35000 },
      };
    }
  }
}
```

```

        new Phone {Title="LG G 6", Company="LG", Price=42000 },
        new Phone {Title="iPhone 7", Company="Apple", Price=52000 }
    };

    this.BindingContext = this;
}
// добавление объекта
private void AddItem(object sender, EventArgs e)
{
    Phones.Add(new Phone { Title = "Galaxy S8", Company = "Samsung", Price = 48000 });
}
// удаление выделенного объекта
private void RemoveItem(object sender, EventArgs e)
{
    Phone phone = phonesList.SelectedItem as Phone;
    if (phone != null)
    {
        Phones.Remove(phone);
        phonesList.SelectedItem = null;
    }
}

public class Phone
{
    public string Title { get; set; }
    public string Company { get; set; }
    public int Price { get; set; }
}
}

```

В данном случае типом коллекции является стандартный класс List, который поддерживает добавление и удаление с помощью методов Add() и Remove(). Однако при запуске приложения, если мы будем нажимать на кнопки, то никаких изменений в ListView, который отображает данный список, мы не увидим. Хотя в реальности коллекция Phones будет изменяться. Более того, мы можем столкнуться с исключением.

Чтобы решить эту проблемы в качестве типа коллекции, как правило, используется не класс List, а класс **ObservableCollection** из пространства имен System.Collections.ObjectModel. За счет реализации интерфейса INotifyCollectionChanged при добавлении или удалении объектов в ObservableCollection автоматически будут изменяться все привязанные к этой коллекции объекты, в том числе и ListView.

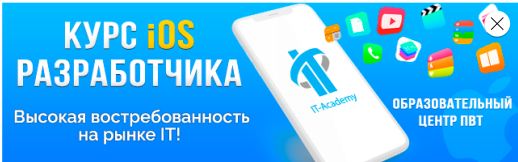
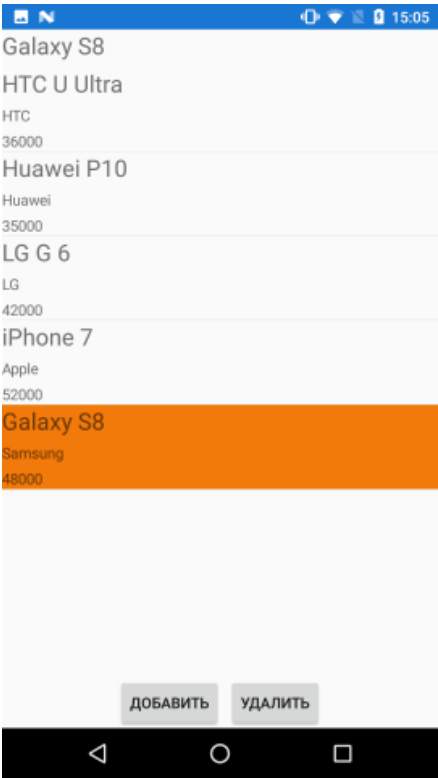
Итак, изменим определение коллекции Phones:

```

public ObservableCollection<Phone> Phones { get; set; }
public MainPage()
{
    InitializeComponent();
    Phones = new ObservableCollection<Phone>
    {
        new Phone { Title = "HTC U Ultra", Company = "HTC", Price = 36000 },
        new Phone {Title="Huawei P10", Company="Huawei", Price=35000 },
        new Phone {Title="LG G 6", Company="LG", Price=42000 },
        new Phone {Title="iPhone 7", Company="Apple", Price=52000 }
    };
    this.BindingContext = this;
}

```

И теперь у нас не возникнет проблем с добавлением или удалением.



Яндекс.Директ

[Назад](#) [Содержание](#) [Вперед](#)



| | | | | | | | |
|--|------------|------------|---------|---------|------------|------------|------------|
| | | | | | | | |
| | 40,85 руб. | 51,30 руб. | 84 руб. | 70 руб. | 60,14 руб. | 77,86 руб. | 63,65 руб. |

ООО "Триовист"

3 Комментариев metanit.com

1 Войти ▾

♥ Рекомендовать  Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS **Mr. Wolf** • месяц назад

Работает ли ObservableCollection для списков с ImageCell?

^ | ▾ • Ответить • Поделиться ›

**Влад** • год назад

Для чего используется this.BindingContext = this; ?

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор ➔ Влад • год назад

В качестве контекста привязки выступает сама страница

^ | ▾ • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Руководство по языку Go

2 комментария • месяц назад

**Alexey** — О, спасибо большое, что уделили внимание этому языку

Kotlin | Переменные

1 комментарий • 2 месяца назад

**Иван Китаев** — Подскажите, как мы можем ввести переменные с клавиатуры? String Вводится через функцию readLine(), а другие переменные типа Int,

Angular в ASP.NET Core | Первый проект

42 комментария • 2 месяца назад

**Metanit** — для управления пакетами

Kotlin | Руководство

3 комментария • 2 месяца назад

**Артур Голояд** — Просто шикарно, может подкинете материалов по работе с сетями на Kotlin? Подписаться  Добавить Disqus на свой сайт [Добавить Disqus](#) [Добавить](#)  Конфиденциальность