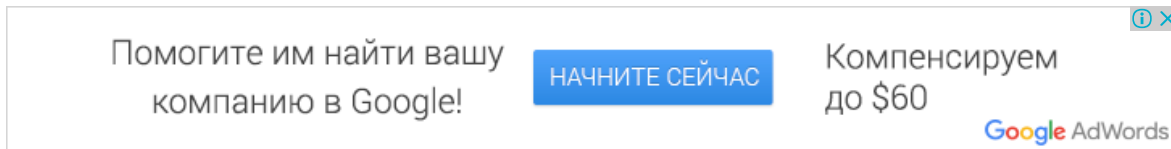




Изображения в ListView. ImageCell и ViewCell

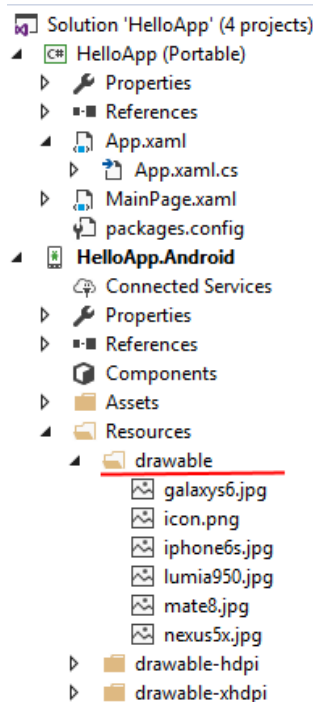
Последнее обновление: 29.03.2017



Для отображения изображений в ListView в простых случаях мы можем использовать класс **ImageCell**, а в более сложных можно воспользоваться ViewCell.

Класс ImageView расширяет класс TextCell, добавляя свойство ImageSource, которое указывает на источник изображения.

Ранее в теме [Работа с изображениями. Элемент Image](#) уже рассматривалось, как добавлять изображения проект и выводить их на страницу. И аналогичным образом добавим в проекты несколько изображений:



В проект для Android изображения добавляются в папку Drawables, в проект для iOS - в папку Resources, в проект для UWP - в корневую папку проекта.

Теперь определим класс страницы, которая будет выводить эти изображения через ImageCell:

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

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

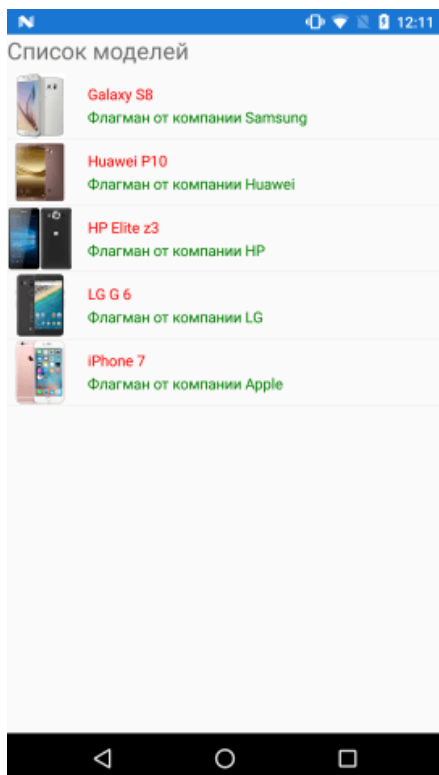
        public MainPage()
        {
            Phones = new List<Phone>
            {
```

```

new Phone {Title="Galaxy S8", Company="Samsung", Price=48000, ImagePath="galaxys6.jpg" },
new Phone {Title="Huawei P10", Company="Huawei", Price=35000, ImagePath="mate8.jpg" },
new Phone {Title="HP Elite z3", Company="HP", Price=42000, ImagePath="lumia950.jpg" },
new Phone {Title="LG G 6", Company="LG", Price=42000, ImagePath="nexus5x.jpg" },
new Phone {Title="iPhone 7", Company="Apple", Price=52000, ImagePath="iphone6s.jpg" }
};
Label header = new Label
{
    Text = "Список моделей",
    FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label))
};
ListView listView = new ListView
{
    HasUnevenRows = true,
    ItemsSource = Phones,
    // Определяем формат отображения данных
    ItemTemplate = new DataTemplate(() =>
    {
        ImageCell imageCell = new ImageCell { TextColor = Color.Red, DetailColor = Color.Green };
        imageCell.SetBinding(ImageCell.TextProperty, "Title");
        Binding companyBinding = new Binding { Path = "Company", StringFormat="Флагман от компании {0}"};
        imageCell.SetBinding(ImageCell.DetailProperty, companyBinding);
        imageCell.SetBinding(ImageCell.ImageSourceProperty, "ImagePath");
        return imageCell;
    })
};
listView.ItemTapped += OnItemTapped;
this.Content = new StackLayout { Children = { header, listView } };
}
public async void OnItemTapped(object sender, ItemTappedEventArgs e)
{
    Phone selectedPhone = e.Item as Phone;
    if (selectedPhone != null)
        await DisplayAlert("Выбранная модель", $"{selectedPhone.Company} - {selectedPhone.Title}", "OK");
}
}
public class Phone
{
    public string Title { get; set; }
    public string ImagePath { get; set; }
    public string Company { get; set; }
    public int Price { get; set; }
}
}

```

По сравнению с прошлой темой в класс Phone было добавлено свойство ImagePath, которое хранит путь к изображению в проекте. К этому свойству осуществляется привязка свойства ImageSource объекта ImageCell.



Аналогичный пример в Xaml:

```
<?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>
                    <ImageCell
                        ImageSource="{Binding ImagePath}"
                        Text="{Binding Title}"
                        Detail="{Binding Company, StringFormat='Флагман от компании {0}'}"
                        TextColor="Red"
                        DetailColor="Green"
                    />
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>
</ContentPage>
```

И в файле кода происходит создание источника данных:

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

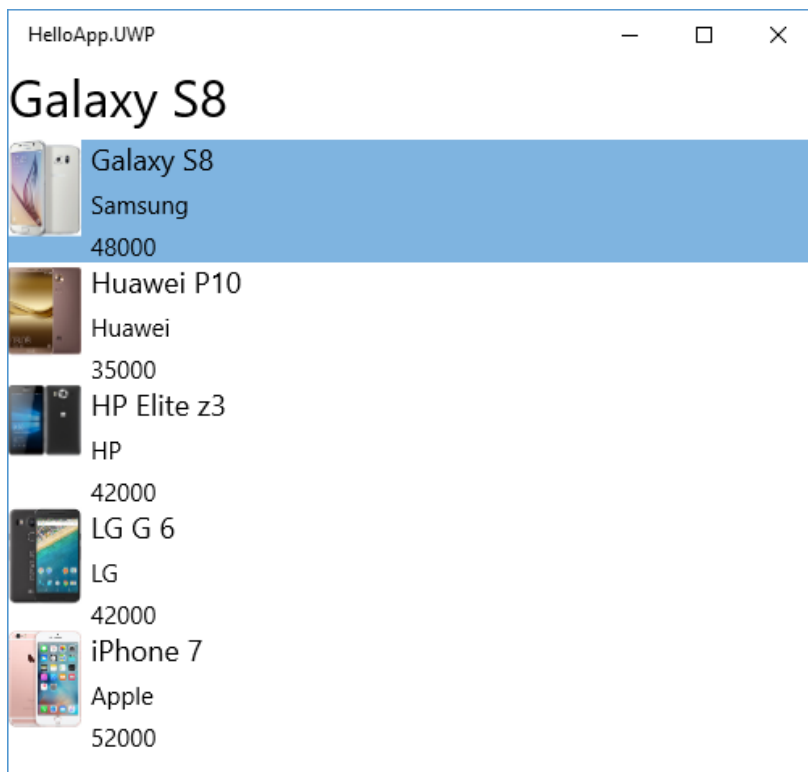
    public MainPage()
    {
        InitializeComponent();
        Phones = new List<Phone>
        {
            new Phone {Title="Galaxy S8", Company="Samsung", Price=48000, ImagePath="galaxys6.jpg" },
            new Phone {Title="Huawei P10", Company="Huawei", Price=35000, ImagePath="mate8.jpg" },
            new Phone {Title="HP Elite z3", Company="HP", Price=42000, ImagePath="lumia950.jpg" },
            new Phone {Title="LG G 6", Company="LG", Price=42000, ImagePath="nexus5x.jpg" },
            new Phone {Title="iPhone 7", Company="Apple", Price=52000, ImagePath="iphone6s.jpg" }
        };
        this.BindingContext = this;
    }
}
```

Вывод изображения в ViewCell

В то же время следует отметить, что на UWP, в отличие от Android и iOS, отсутствует масштабирование изображения. И оно отображается в тех размерах, которые имеет. В этом случае мы можем выводить элементы списка через ViewCell, задавая у элемента Image явным образом высоту и ширину:

```
<?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}"
            ItemTapped="OnItemTapped">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout Orientation="Horizontal">
                            <Image Source="{Binding ImagePath}" WidthRequest="45" HeightRequest="60" />
                            <StackLayout>
                                <Label Text="{Binding Title}" FontSize="18" />
                                <Label Text="{Binding Company}" />
                                <Label Text="{Binding Price}" />
                            </StackLayout>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>
</ContentPage>
```

```
</ListView.ItemTemplate>
</ListView>
</StackLayout>
</ContentPage>
```



Яндекс.Директ

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

G+



1 Комментарий metanit.com

1 Войти ▾

 Рекомендовать  Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS **Александр Городилов** • 4 месяца назад

Здравствуйте. Возникла проблема с изображениями на iOS: после того как я добавляю мои изображения в папку Resource у меня сразу же будто бы пропадает ссылка на мой Portable проект из references (соответственно все классы оттуда не видны). Никак не могу понять, с чем это связано, возможно, кто-то знает решение?

  • [Ответить](#) • [Поделиться](#) >

ТАКЖЕ НА METANIT.COM

Kotlin | Введение в язык. Первая программа

1 комментарий • 2 месяца назад



Jonny Manowar — Есть мнение что на самом деле язык был создан и продвигается компанией JetBrains для продажи IDE, по сути не IDE пислась под язык а язык

Vue.js | Навигация и ссылки

1 комментарий • 3 месяца назад



Delirium4Dude — Огромное спасибо тебе за твои труды! Одни из самых толковых мануалов в Рунете

C++ | Указатель на функцию как возвращаемое значение

1 комментарий • 4 месяца назад




Даниил Данилов — Не знал что можно декораторы делать как JS

React | React-router и webpack

8 комментариев • 3 месяца назад



Юрий Демин — да, установлен. решил проблему обновив OS, но конкретной причины и связи с обновлением системы так и не нашел!) Спасибо за помощь и за ...

 Подписаться  Добавь Disqus на свой сайт [Добавить Disqus](#) [Добавить](#)  Конфиденциальность