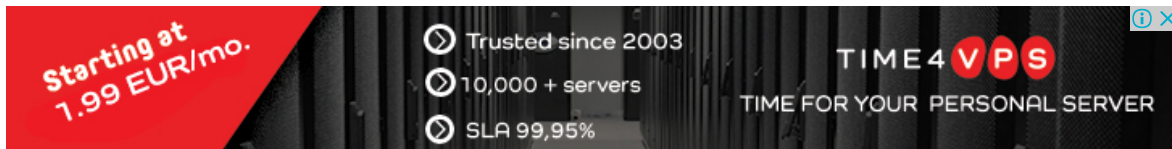




Передача данных при навигации

Последнее обновление: 25.06.2016



Рассмотрим, как мы можем передавать данные из одной страницы в другую. Здесь могут быть различные варианты. Наиболее распространенный способ представляет передача данных через конструктор.

Например, пусть у нас есть две страницы:

```
class PageA : ContentPage
{
    public PageA()
    {
        //.....
    }
}

class PageB : ContentPage
{
    public PageB(int x)
    {
        //.....
    }
}
```

Страница PageB в конструкторе в качестве параметра принимает некоторое число, и чтобы перейти на нее из страницы PageA, мы могли бы использовать следующее выражение:

```
await Navigation.PushAsync(new PageB(56));
```

Другой вариант состоит в использовании методов или свойств второй страницы для передачи в нее данных. Например, пусть в PageB определен некий метод, принимающий в качестве параметра число:

```
class PageB : ContentPage
{
    public void Calculate(int x)
    {
    }
}
```

Тогда передать данные в этот метод из PageA мы могли бы следующим способом:

```
PageB pageB = new PageB();
await Navigation.PushAsync(pageB);
pageB.Calculate(56);
```

Теперь рассмотрим на реальном примере. Вначале добавим в проект класс Phone, с объектами которого мы будем работать в программе:

```
using System.ComponentModel;

namespace NavigationApp
{
    public class Phone : INotifyPropertyChanged
    {
        private string name;
        private string company;
        private int price;

        public string Name
```

```

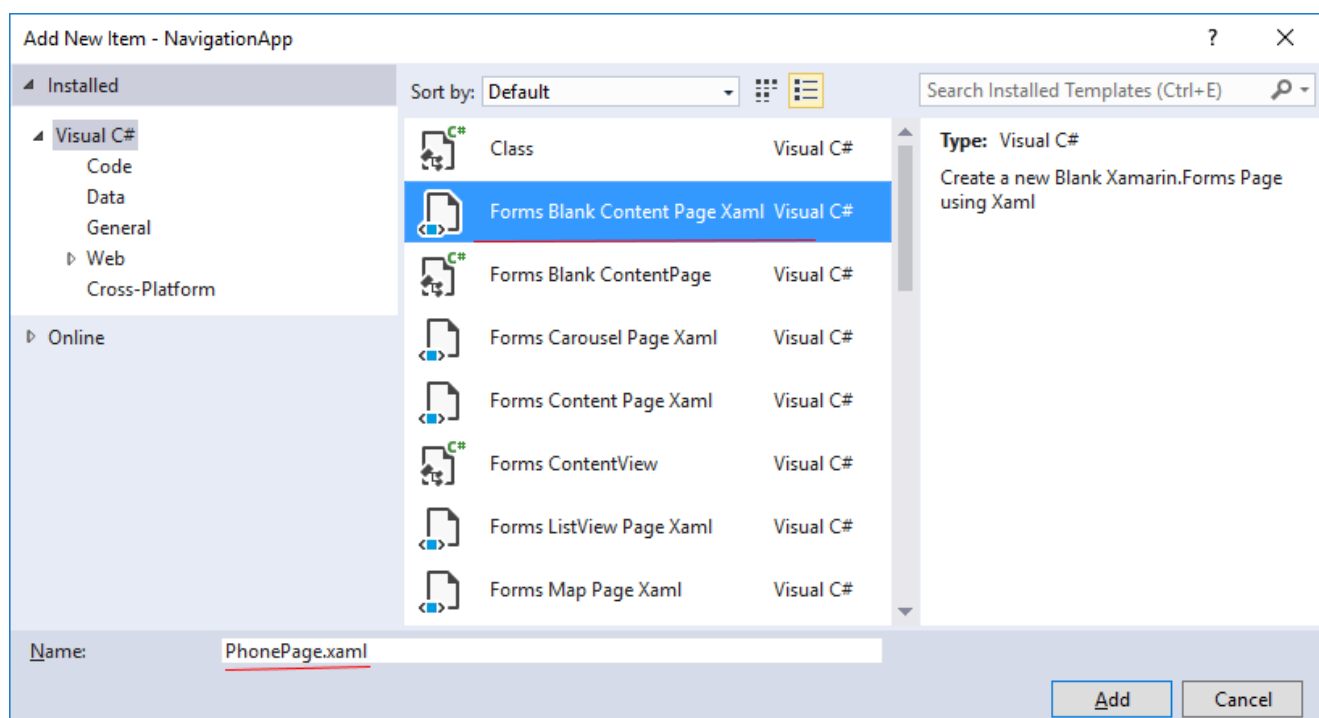
{
    get { return name; }
    set
    {
        if (name != value)
        {
            name = value;
            OnPropertyChanged("Name");
        }
    }
}
public string Company
{
    get { return company; }
    set
    {
        if (company != value)
        {
            company = value;
            OnPropertyChanged("Company");
        }
    }
}
public int Price
{
    get { return price; }
    set
    {
        if (price != value)
        {
            price = value;
            OnPropertyChanged("Price");
        }
    }
}

public event PropertyChangedEventHandler PropertyChanged;
public void OnPropertyChanged(string prop = "")
{
    if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs(prop));
}
}
}

```

Важно, что этот класс реализует интерфейс `INotifyPropertyChanged`, что даст возможность элементам пользовательского интерфейса автоматически синхронизировать значения со значениями свойств объекта `Phone`.

Далее добавим в проект новую страницу по типу **Forms Blank Content Page Xaml**, которую назовем **PhonePage**:



В XAML-коде главной страницы MainPage определим следующий интерфейс:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="NavigationApp.MainPage"
    Title="Список смартфонов">
    <StackLayout>
        <Button Text="Добавить" Clicked="AddButton_Click" />
        <ListView x:Name="phonesList" ItemsSource="{Binding}" ItemSelected="OnListViewItemSelected">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <ViewCell.View>
                            <Grid>
                                <Grid.ColumnDefinitions>
                                    <ColumnDefinition Width="*" />
                                    <ColumnDefinition Width="*" />
                                    <ColumnDefinition Width="*" />
                                </Grid.ColumnDefinitions>
                                <Label Text="{Binding Name}" FontSize="Medium" Grid.Column="0" />
                                <Label Text="{Binding Company}" FontSize="Medium" Grid.Column="1" />
                                <Label Text="{Binding Price}" FontSize="Medium" Grid.Column="2" />
                            </Grid>
                        </ViewCell.View>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>
</ContentPage>
```

Здесь в ListView будет выводиться список объектов Phone. По нажатию на кнопку будет открываться форма для добавления нового элемента, а в результате выбора элемента в списке ListView выбранный элемент будет передаваться на ту же форму для редактирования.

И в коде с# у MainPage определим обработчики кнопок и выделения списка:

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

namespace NavigationApp
{
    public partial class MainPage : ContentPage
    {
        protected internal ObservableCollection<Phone> Phones { get; set; }

        public MainPage()
        {
            InitializeComponent();

            Phones = new ObservableCollection<Phone>
            {
                new Phone {Name="iPhone 7", Company="Apple", Price=52000},
                new Phone {Name="Galaxy S8", Company="Samsung", Price=50000},
                new Phone {Name="LG G6", Company="LG", Price=45000},
                new Phone {Name="Huawei P10", Company="Huawei", Price=35000}
            };
            phonesList.BindingContext = Phones;
        }
        // обработчик выбора элемента в списке
        private async void OnListViewItemSelected(object sender, SelectedItemChangedEventArgs args)
        {
            // Получаем выбранный элемент
            Phone selectedPhone = args.SelectedItem as Phone;
            if (selectedPhone != null)
            {
                // Снимаем выделение
                phonesList.SelectedItem = null;
                // Переходим на страницу редактирования элемента
                await Navigation.PushAsync(new PhonePage(selectedPhone));
            }
        }
        // переходим на страницу PhonePage для добавления нового элемента
        private async void AddButton_Click(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new PhonePage(null));
        }
        // вспомогательный метод для добавления элемента в список
    }
}
```

```

        protected internal void AddPhone(Phone phone)
        {
            Phones.Add(phone);
        }
    }
}

```

На странице PhonePage в xaml определим следующий код:

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="NavigationApp.PhonePage"
              Title="Информация о смартфоне">
    <StackLayout>
        <Entry x:Name="nameEntry" Text="{Binding Path=Name}" />
        <Entry x:Name="companyEntry" Text="{Binding Path=Company}" />
        <StackLayout Orientation="Horizontal">
            <Stepper x:Name="priceStepper" Minimum="0" Maximum="100000" Increment="100"
                    Value="{Binding Path=Price}"/>
            <Label x:Name="priceLbl" FontSize="Large"
                  Text="{Binding Source={x:Reference Name=priceStepper}, Path=Value }" />
        </StackLayout>
        <Button Text="Сохранить" Clicked="SavePhone" />
    </StackLayout>
</ContentPage>

```

А в файле кода для PhonePage следующие строки:

```

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

namespace NavigationApp
{
    public partial class PhonePage : ContentPage
    {
        bool edited = true; // флаг редактирования
        public Phone Phone { get; set; }
        public PhonePage(Phone phone)
        {
            InitializeComponent();

            Phone = phone;
            if(phone==null)
            {
                Phone = new Phone();
                edited = false;
            }
            this.BindingContext = Phone;
        }

        async void SavePhone(object sender, EventArgs e)
        {
            await Navigation.PopAsync();

            // если добавление
            if (edited == false)
            {
                // находим в стеке предпоследнюю страницу - то есть MainPage
                NavigationPage navPage = (NavigationPage)Application.Current.MainPage;
                IReadOnlyList<Page> navStack = navPage.Navigation.NavigationStack;
                MainPage homePage = navStack[navPage.Navigation.NavigationStack.Count - 1] as MainPage;

                if (homePage != null)
                {
                    homePage.AddPhone(Phone);
                }
            }
        }
    }
}

```

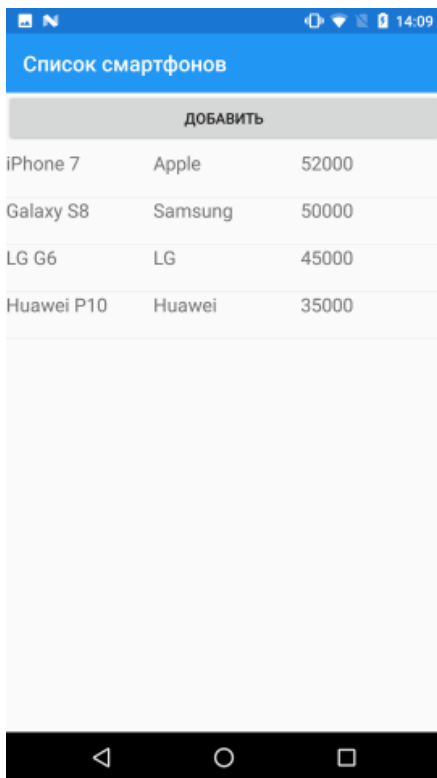
Для разграничения ситуаций добавления и редактирования здесь определена дополнительная переменная `edited`. Если в конструктор страницы передается объект `Phone`, то эта переменная равна `true`. Если же в конструктор передается значение `null`, то переменная равна `false`, и создается новый объект `Phone`.

В обработчике кнопки `SavePhone()`, если у нас редактирование, то мы просто возвращаемся назад на предыдущую страницу. Поскольку редактируемый объект `Phone` реализует интерфейс `INotifyPropertyChanged`, то нам не надо выполнять никаких

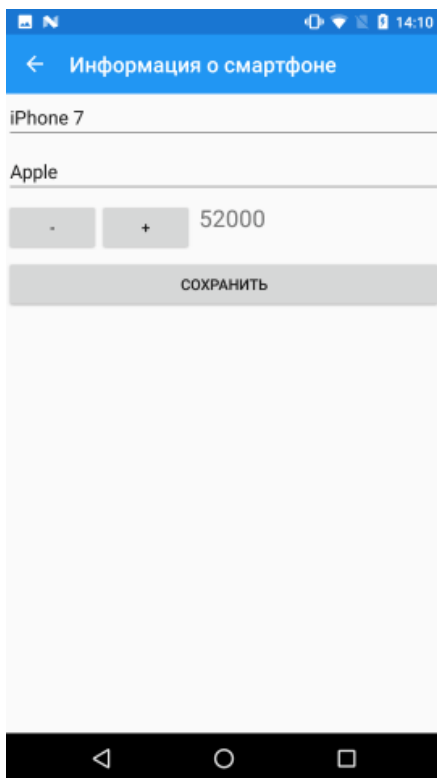
дополнительных действий, и соответствующий объект на главной станции MainPage будет сам автоматически обновлен.

Если же у нас добавление, то получаем из стека навигации страницу MainPage и вызываем у нее метод AddPhone() для добавления объекта. Стоит также заметить, что на странице MainPage привязка списка идет не к простому списку List, а к объекту ObservableCollection, что позволяет автоматически обновить привязанный объект ListView в случае добавления нового объекта.

Запустим проект, и на странице отобразится список объектов:



Выберем какой-либо объект, и он перейдет на страницу PhonePage:

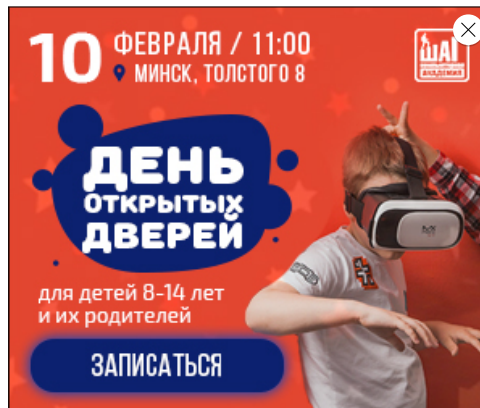




ОБРАЗОВАТЕЛЬНЫЙ
ЦЕНТР ПВТ

**КУРС iOS
РАЗРАБОТЧИКА**

Высокая востребованность
на рынке IT!



10 ФЕВРАЛЯ / 11:00
МИНСК, ТОЛСТОГО 8

**ДЕНЬ
ОТКРЫТЫХ
ДВЕРЕЙ**

для детей 8-14 лет
и их родителей

ЗАПИСАТЬСЯ

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



G+



1 Комментарий metanit.com

1 Войти ▾

♥ Рекомендовать  Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS ?

Имя



Олег Ладыко • 3 месяца назад

// вспомогательный метод для добавления элемента в список
protected internal void AddPhone(PHONE phone)
{
Phones.Add(phone);
}

У меня в добавлении вылетает exception. тестирую на своем телефоне. Не могу понять почему.

^ | ▾ • Ответить • Поделиться ▸

ТАКЖЕ НА METANIT.COM

Angular в ASP.NET Core | Введение

16 комментариев • 2 месяца назад



Igor Teterkin — Если у меня есть в контроллере (UserController) метод Send (отправка почты сотруднику) [HttpPost("{id}")] public IActionResult Send(int id), то как ...

Kotlin | Руководство

3 комментария • 2 месяца назад



Артур Голояд — Просто шикарно, может подкинете материалов по работе с сетями на Kotlin?

Angular в ASP.NET Core | Первый проект

42 комментария • 2 месяца назад



Роман Тимохов — На локальной машине все норм работает, но когда выливаю код на AZURE сервер не запускается((<https://uploads.disquscdn.c...>

Паттерны в C# и .NET | Fluent Builder

10 комментариев • 3 месяца назад



LamaK — При том, что, если забыть какой-то параметр, то что-то может обвалиться. По-хорошему, в метод Build() можно добавить проверку всех необходимых ...

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