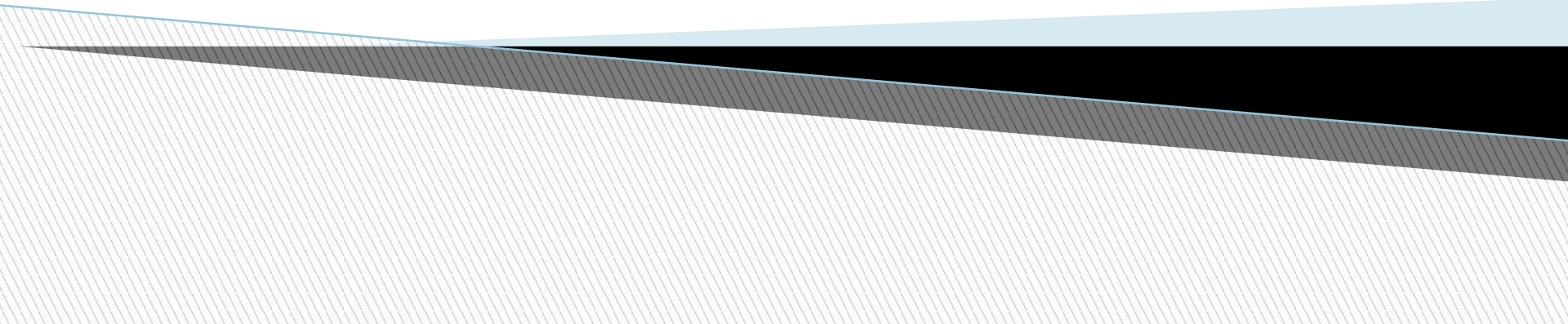


Создание визуальных компонентов

Махнев А.А.

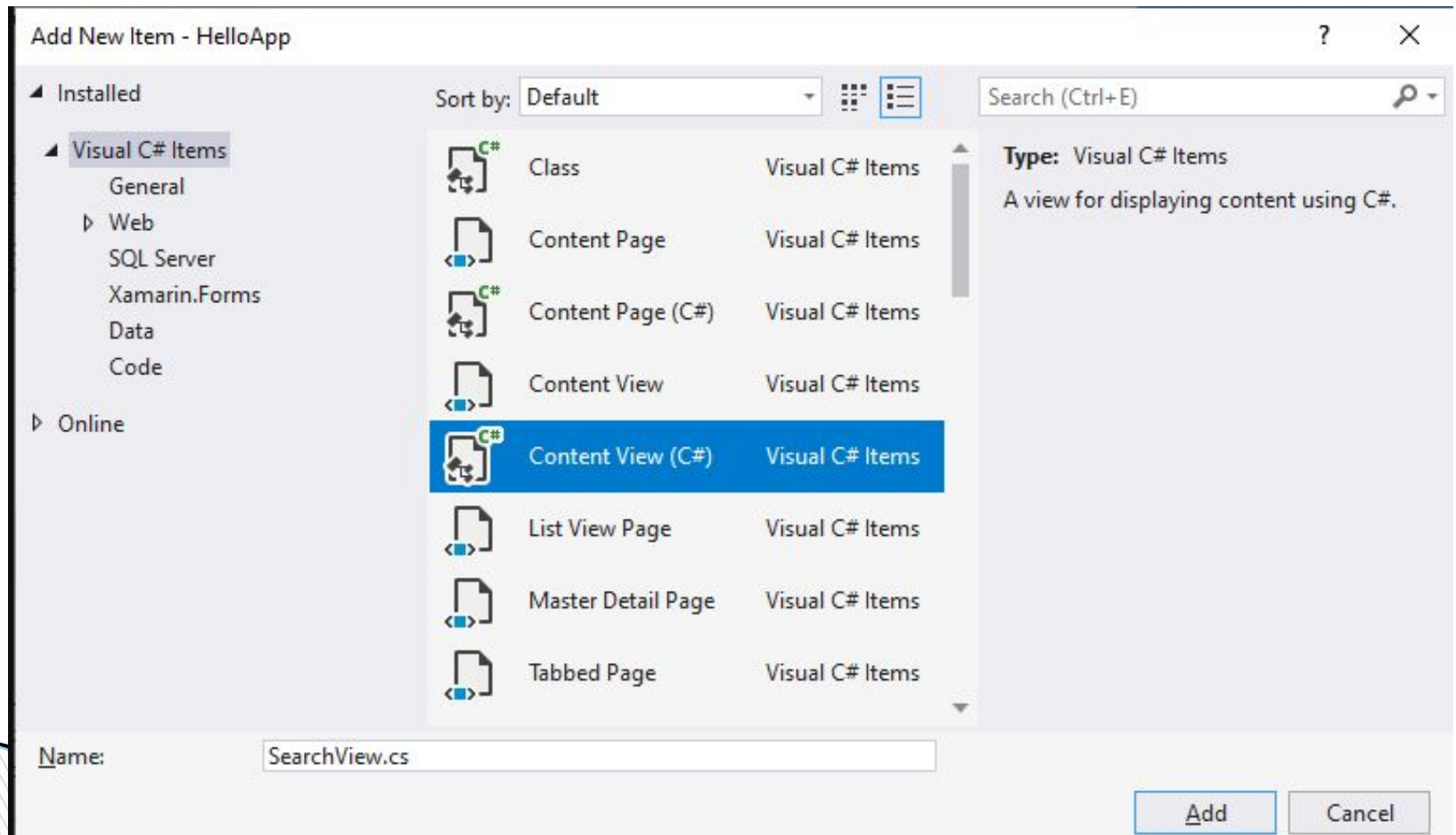


Применение

- Использовать в приложениях повторяющиеся элементы разметки `html`
- форма входа - ввод логина и пароля
- форма поиска – поле ввода текста и кнопка

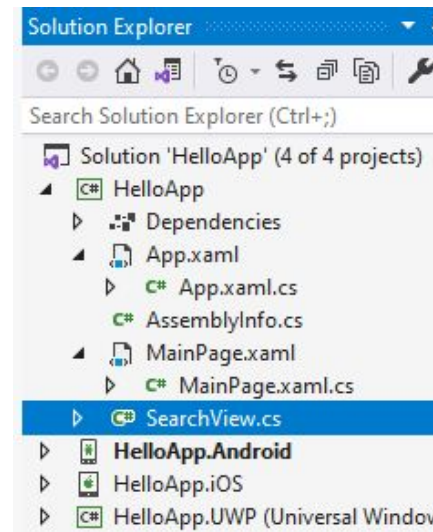
Создание визуальных компонентов

□ Класс **ContentView**



Добавится обычный класс на языке C#

```
public class SearchView : ContentView
{
    public SearchView()
    {
        Content = new StackLayout
        {
            Children = {
                new Label { Text = "Welcome to
Xamarin.Forms!" }
            }
        };
    }
}
```



Внести в него изменения (элементы формы поиска)

```
public class SearchView : ContentView
{
    public event SearchEventHandler Search;
    public SearchView()
    {
        Button searchBtn = new Button { Text = "Поиск" };
        Entry searchEntry = new Entry { HorizontalOptions =
LayoutOptions.FillAndExpand };

        searchBtn.Clicked += (sender, e) => Search?.Invoke(searchEntry.Text);
        Content = new StackLayout
        {
            Orientation = StackOrientation.Horizontal,
            Spacing = 5,
            Children =
            {
                searchEntry,
                searchBtn
            }
        };
    }
}
```


Что определено в классе?

- Кнопка, текстовое поле
- событие Search, представляющее делегат SearchEventHandler

Делегат C#

- Это тип, который представляет ссылки на методы с определенным списком параметров и типом возвращаемого значения.
-
- При создании экземпляра делегата этот экземпляр можно связать с любым методом с совместимой сигнатурой и типом возвращаемого значения.
- Метод можно вызвать (активировать) с помощью экземпляра делегата.

Особенности делегата

- ▣ Делегату можно назначить любой метод из любого доступного класса или структуры, соответствующей типу делегата.
 - ▣ Этот метод должен быть статическим методом или методом экземпляра.
 - ▣ Позволяет программно изменять вызовы метода, а также включать новый код в существующие классы.
- 

Код страницы MainPage

```
public MainPage()
{
    List<string> users = new List<string>()
    {
        "Иван Иванов",
        "Олег Кузнецов",
        "Денис Петров",
        "Иван Сидоров",
        "Петр Денисов"
    };

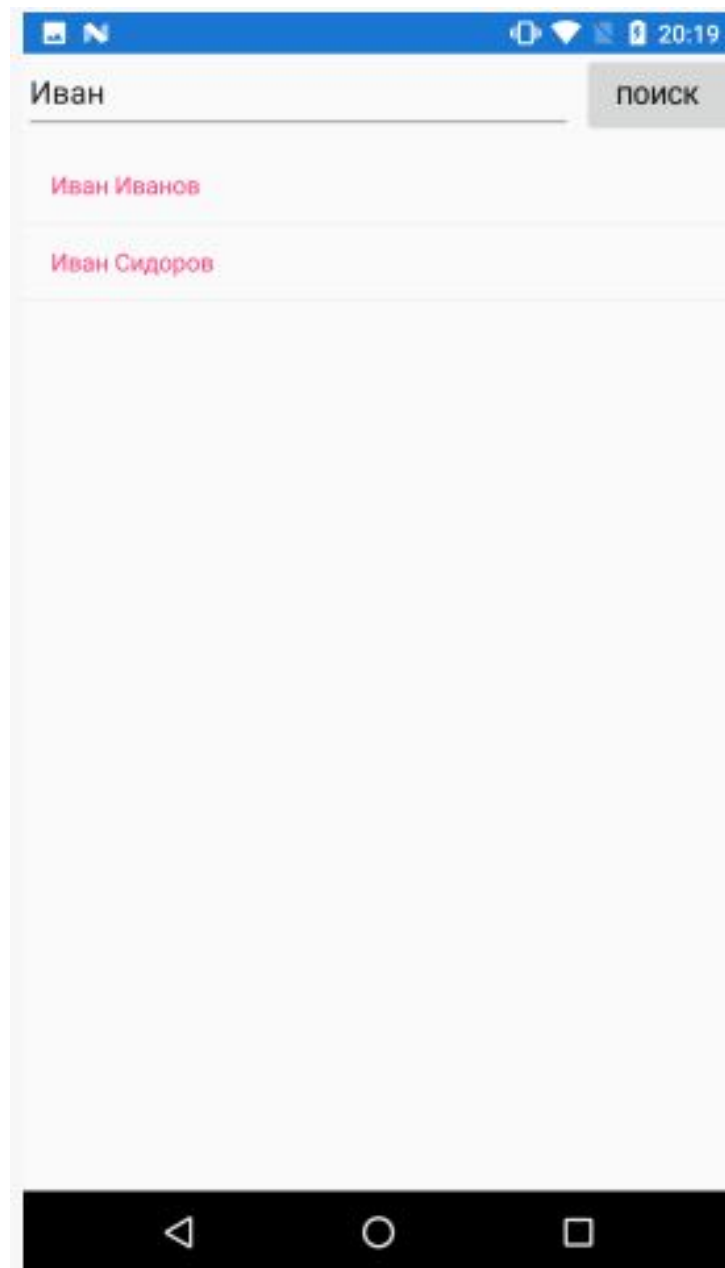
    ListView usersList = new ListView
    {
        ItemsSource = users
    };
}
```

коде страницы MainPage

```

□
□     SearchView searchView = new SearchView();
□     searchView.Search += (text) =>
□     {
□         if(!string.IsNullOrEmpty(text))
□         {
□             usersList.ItemsSource =
users.Where(u=>u.Contains(text));
□         }
□         else
□         {
□             usersList.ItemsSource = users;
□         }
□     };
□     Content = new StackLayout { Children = { searchView,
usersList } };
□ }
```

Итог



Как использовать ContentView в коде хамл? (способ 2)

- Создать страницу MainPage.хамл с элементами
- В коде страницы прописать установку списка и обработчик события Search

MainPage.xaml

- <StackLayout>
- <local:SearchView
 Search="SearchUsers"></local:SearchView>
- <ListView x:Name="usersList" />
- </StackLayout>
- </ContentPage>

Обработчик события Search в коде

```
public partial class MainPage : ContentPage
{
    List<string> users;

    public MainPage()
    {
        InitializeComponent();

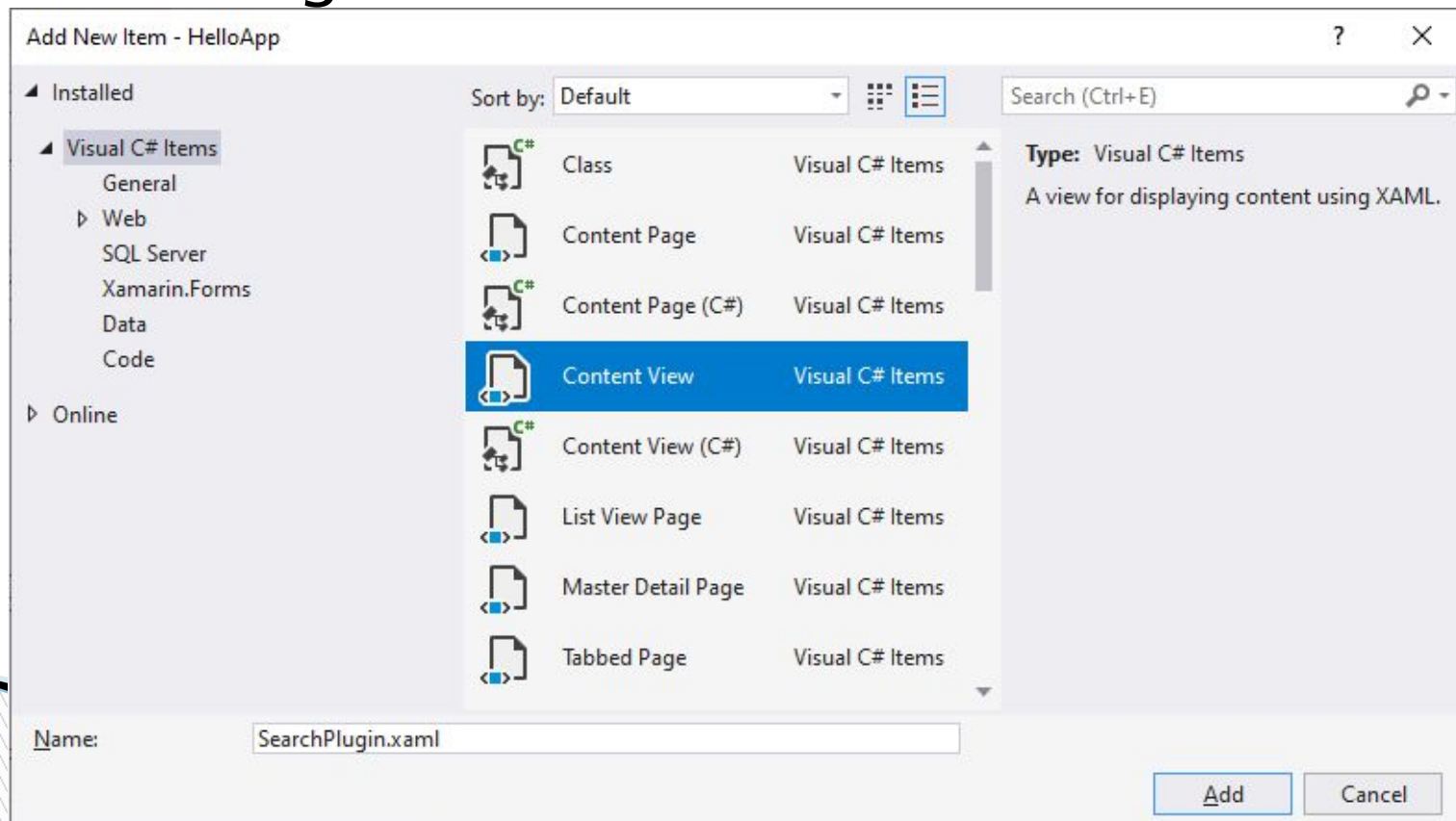
        users = new List<string>()
        {
            "Иван Иванов",
            "Олег Кузнецов",
            "Денис Петров",
            "Иван Сидоров",
            "Петр Денисов"
        };
        userList.ItemsSource = users;
    }
}
```

Обработчик события Search в коде

```
private void SearchUsers(string text)
{
    if (!string.IsNullOrEmpty(text))
    {
        userList.ItemsSource = users.Where(u =>
u.Contains(text));
    }
    else
    {
        userList.ItemsSource = users;
    }
}
}
```

Определение **ContentView** в XAML

- Добавить в проект новый элемент по типу **ContentView Xaml**, который назовем SearchPlugin:



Изменить код XAML

- <?xml version="1.0" encoding="utf-8" ?>
- <ContentView
xmlns="<http://xamarin.com/schemas/2014/forms>"
- xmlns:x="<http://schemas.microsoft.com/winfx/2009/xaml>"
- x:Class="HelloApp.SearchPlugin">
- <StackLayout Orientation="Horizontal" Spacing="5">
- <Entry x:Name="searchEntry"
- HorizontalOptions="FillAndExpand" />
- <Button Clicked="OnClicked" Text="Поиск" />
- </StackLayout>
- </ContentView>

Изменить файл связанного кода

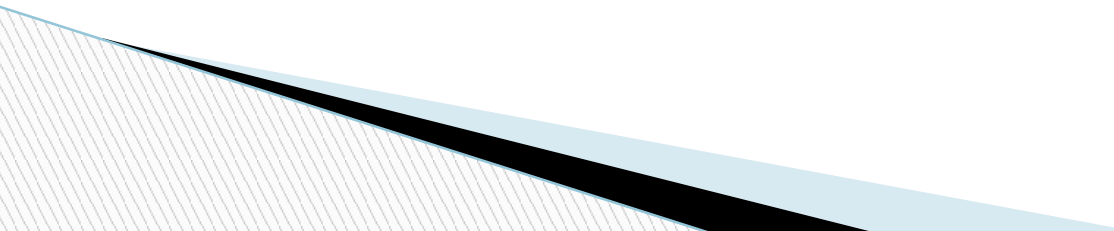
```
namespace HelloApp
{
    public delegate void SearchPluginEventHandler(string text);
    public partial class SearchPlugin : ContentView
    {
        public event SearchPluginEventHandler Search;

        public SearchPlugin()
        {
            InitializeComponent();
        }
        private void OnClicked(object sender, EventArgs e)
        {
            Search?.Invoke(searchEntry.Text);
        }
    }
}
```

Использование класса

```
<?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>
    <local:SearchPlugin
      Search="SearchUsers"></local:SearchPlugin>
    <ListView x:Name="usersList" />
  </StackLayout>
</ContentPage>
```

Задание

- 1. Реализовать пример создания формы поиска по примеру (любым способом)
 - 2. Создать самостоятельно форму **Авторизации пользователя**
 - 3. Создать форму Регистрации пользователя для
 - 4. Объединить все формы в одном мобильном приложении с переходами между ними
- 

Сдача курсового проекта

- Руководителю проекта до 07.12.2021
- Защита курсового проекта 14.12.2021