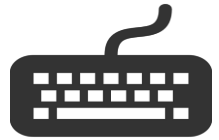
MVVM architektúra, ICommand

Csorba Kristóf

MVVM alapok

Ki válogatja össze a konkrét megjelenítésnek az adatokat?

ItemsControlMvvmDemo



- MVVM részei a forráskódban
- Függőségek továbbítása
- Adatkötés: xaml UI és view model között
- INotifyPropertyChanged láncok

MVVM – példányosítási példa

```
// EVIP: almost empty view. Instantiates view model and model.
public sealed partial class GameView : UserControl
{
    public GameViewModel ViewModel;

    public GameView()
    {
        this.InitializeComponent();
        ViewModel = new GameViewModel(new SimpleGame(5));
    }
}
```

AttaxxPlus\AttaxxPlus\View\GameView.xaml.cs

Függő property és INPC

AttaxxPlus\AttaxxPlus\Boosters\DummyBooster.cs

```
private int usableCounter;
public int UsableCounter
{
    get => usableCounter;
    private set
    {
        if (usableCounter != value)
        {
            usableCounter = value;
            Notify();
            // EVIP: PropertyChange for another property
            Notify(nameof(Title));
        }
    }
}

// EVIP: overriding abstract property in base class.
public override string Title { get => $"Dummy ({UsableCounter})"; }
```

INPC továbbítás a modell felől

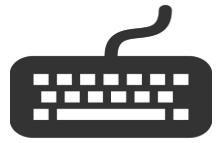
```
// EVIP: derived from ObservableObject to have Notify()
public class FieldViewModel : ObservableObject
{
    public Field Model { get; internal set; }

    public FieldViewModel(Field model)
    {
        Model = model;
        // EVIP: VM observes Model and translates PropertyChanged events
        model.PropertyChanged += Model_PropertyChanged;
    }

    private void Model_PropertyChanged(object sender,
        System.ComponentModel.PropertyChangedEventArgs e)
    {
        // EVIP: forwarding INPC
        // EVIP: nameof operator
        if (e.PropertyName == nameof(Field.Owner))
            Notify(nameof(Owner));
    }

    // EVIP: read-only property. Needs to emit (forward)
    //  PropertyChanged if underlying model changes.
    public int Owner { get => Model.Owner; }
```

AttaxxPlus\AttaxxPlus\ViewModel\FieldViewModel.cs

ICommand

```
// EVIP: Commands to bind to.
public AddToFavoritesCommand AddToFavoritesCommand { get; set; }
```

UwpBindingAndCommandDemo (ICommand)

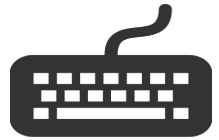


```
class IncreaseCommand : ICommand
{
    Counters model;

    public IncreaseCommand(Counters dataModel)...

    private void Model_PropertyChanged(object sender,
        PropertyChangedEventArgs e)...

    public event EventHandler CanExecuteChanged;

    public bool CanExecute(object parameter)...

    public void Execute(object parameter)
    {
        model.CounterA++;
    }
}
```

```
public sealed partial class MainPage : Page
{
    IncreaseCommand IncCommand;
```

```
<Button Command="{x:Bind IncCommand}" Content="Increase!"
    Background="{x:Bind CountersModel.CounterA, Mode=OneWay,
    Converter={StaticResource converter}}"/>
```

AttaxxPlus



(Áttekintő MVVM ábra)