

Attribútumok, Reflection

Csorba Kristóf

Unit tesztekben attribútumok

```
public class TaskTests
{
    private readonly Solutions s1;
    private readonly Solutions s2;

    public TaskTests()
    {
        s1 = new Solutions("rectangles1.svg");
        s2 = new Solutions("rectangles2.svg");
    }

    [Fact]
    public void GetAllRectangles()
    {
        Assert.Equal(7, s1.GetAllRectangles().Count());
        Assert.Equal(12, s2.GetAllRectangles().Count());
    }
}
```

Linq2XmlSvgLab\TaskTests.cs

Serializer példa (félév elejéről)

- A háttérben mi történik?
- Sima memóriakép elmentés nem jó?

```
...  
[XmlRoot("PurchaseOrder", IsNullable = false)]  
public class PurchaseOrder  
{  
    public Address ShipTo;  
    public string OrderDate;  
    // The XmlArray attribute changes the XML element name  
    // from the default of "OrderedItems" to "Items".  
    [XmlArray("Items")]  
    public OrderedItem[] OrderedItems;  
    public decimal SubTotal;  
    public decimal ShipCost;  
    public decimal TotalCost;  
}
```

```
<?xml version="1.0" encoding="utf-8"?>  
<PurchaseOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema  
    <ShipTo Name="Minta Márton">  
        <Line>Fontos utca</Line>  
        <City>Teszvesz város</City>  
        <State>Mozgalmas ország</State>  
        <Zip>1234</Zip>  
    </ShipTo>  
    <OrderDate>Thursday, June 18, 2020</OrderDate>  
    <Items>  
        <OrderedItem>  
            <ItemName>Kis cucc</ItemName>  
            <Description>Egy egészen kicsi cucc</Description>  
            <UnitPrice>5.23</UnitPrice>  
            <Quantity>3</Quantity>  
            <LineTotal>15.69</LineTotal>  
        </OrderedItem>  
        <OrderedItem>  
            <ItemName>Nagyobb cucc</ItemName>
```

Típusok lekérdezése, vizsgálata

```
// EVIP: assert on type of object
Assert.IsInstanceOfType(ParseSingle(@"v:"), typeof(VisibleCommand));
Assert.IsInstanceOfType(ParseSingle(@"e:"), typeof(EmphasizedCommand));
```

```
private AnimationCommand ParseSingle(string s)
```

svg2pptx\Tests\AnimationCommandTests.cs

Típusok kigyűjtése (AttaxxPlus boosterek)

```
// EVIP: using reflection to find all non-abstract
// implementations of an interface
var currentAssembly = this.GetType().GetTypeInfo().Assembly;
// EVIP: using Linq for searching
var allIBoosterTypes = currentAssembly.DefinedTypes
    .Where(t => t.ImplementedInterfaces.Any(i => i == typeof(IBooster)))
    .Where(t => !t.IsAbstract).ToList();
```

AttaxxPlus\AttaxxPlus\ViewModel\BoosterListViewModel.cs

Activator

```
foreach (var boosterType in allIBoosterTypes)
{
    // EVIP: using reflection to instantiate objects
    IBooster booster = Activator
        .CreateInstance(boosterType.AsType()) as IBooster;
    booster.GameViewModel = gameViewModel;
    Boosters.Add(booster);
}
```

AttaxPlus\AttaxPlus\ViewModel\BoosterListViewModel.cs

Activator: default ctor kell...

```
public abstract class BoosterBase : ObservableObject, IBooster
{
    // EVIP: specialties when instantiated using Activator
    // Warning: Instantiated by Activator using default ctor.
    // GameViewModel is set after that, so in derived classes,
    // methods invoked from ctor should not rely on GameViewModel.
    public GameViewModel GameViewModel { get; set; }
```

AttaxxPlus\AttaxxPlus\Boosters\BoosterBase.cs

UI-on Enum értékkészlete

```
public enum InitialShapeEnum
{
    Rectangle, Triangle
}

// This is used as ItemSource
// EVIP: reflection to get a list of enum values
public InitialShapeEnum[] InitialShapeEnumValues =>
    Enum.GetValues(typeof(InitialShapeEnum))
        .Cast<InitialShapeEnum>().ToArray();
```



```
<DataTemplate x:Key="StrokeVMDataTemplate" x:DataType="local:StrokeVM">
    <StackPanel>
        <TextBlock>Stroke</TextBlock>
        <TextBlock Text="{x:Bind NumberOfIterations}"/>
        <ComboBox x:Name="shapeSelector"
            ItemsSource="{x:Bind InitialShapeEnumValues}"
            SelectedItem="{x:Bind Mode=TwoWay, Path=InitialShape}"/>
    </StackPanel>
</DataTemplate>
```

SpirographLab\SpirographViewer\ViewModel\StrokeVM.cs

Metódusok lekérdezése, meghívása

- `(string)user.GetType().GetField("Id").GetValue(user)`
- `GetMethod`
- `FindInterfaces`
- `GetConstructor`
- `GetEvents`
- `GetMember`
- `GetProperties`
- `GetCustomAttributes`