

C# Design Patterns: Null Object



David Starr

@elegantcoder

elegantcode.com



Overview



Origins of null

The problem to solve

Introducing the Null Object pattern

Using a Null Object in practice





Sir Tony Hoare

Inventor of quicksort

Created null reference in 1965

“Billion-dollar mistake”



Null Checks

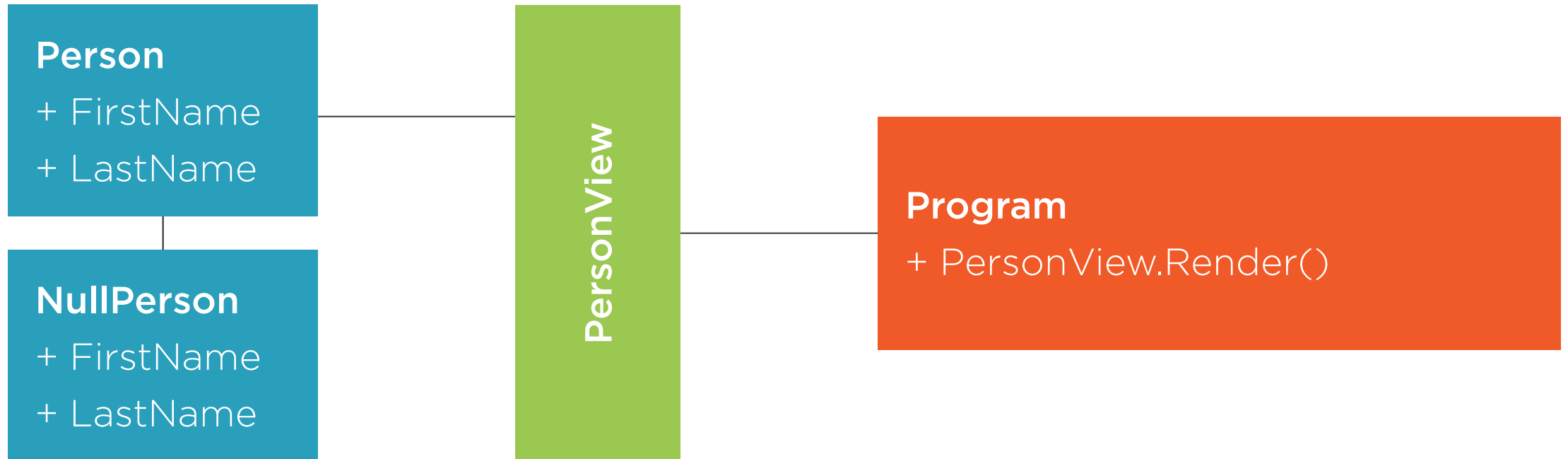
The problem to solve

PersonView.cs

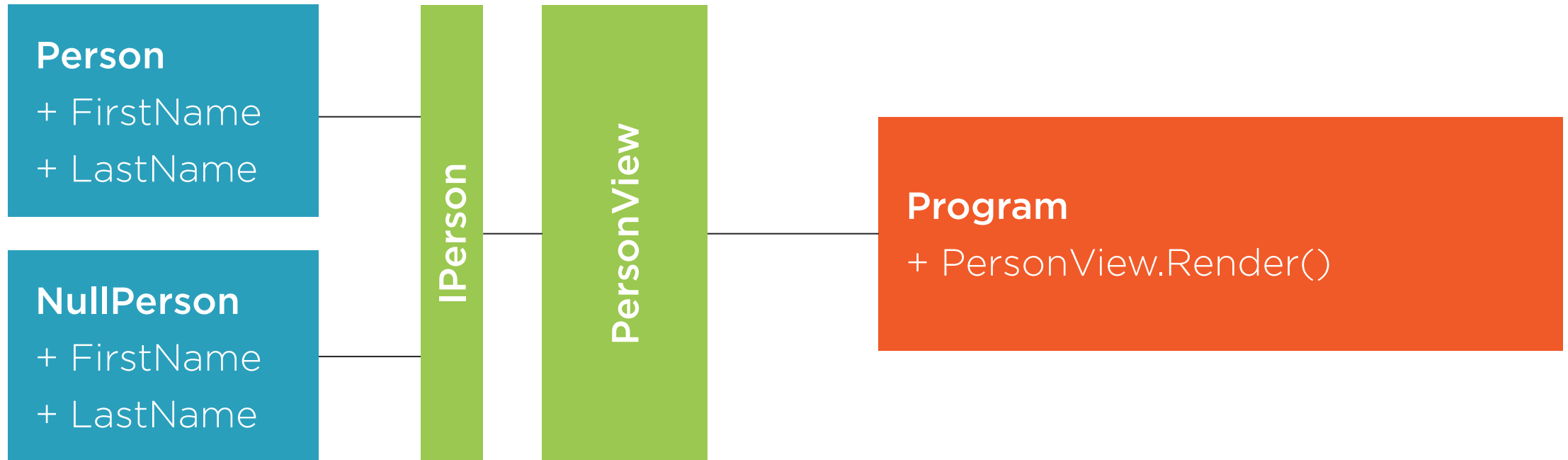
```
public class PersonView
{
    public PersonView(Person person)
    {
        if (person == null) throw new ArgumentNullException();
        if (person.FirstName == null) throw new ArgumentNullException();
        if (person.LastName == null) throw new ArgumentNullException();

        // save person to local field
    }
}
```

The Null Object Pattern



The Null Object Pattern



A Null Object

Returned instead of null

NullPerson.cs

```
public class NullPerson : IPerson
{
    public string FirstName
    {
        get { return "David"; }
    }

    public string LastName
    {
        get { return "Starr"; }
    }
}
```

With Null Objects

No need for null checks

PersonView.cs

```
public class PersonView
{
    private readonly Person _person;

    public PersonView(Person person)
    {
        _person = person;
    }
}
```


Demo



Making it tangible

See it in real code



Summary



Use Null Object to avoid null checks

Returned object is not null

Use interface or inheritance
as appropriate

Cleaner code

