

За пределами PageObject

Дмитрий Жарий





**ВАМ ДЕЙСТВИТЕЛЬНО НУЖЕН
PAGE OBJECT?**

Нет. Если у вас парочка НЕСЛОЖНЫХ ТЕСТОВ

```
public void Wikipedia_Smart_Search_Test()
{
    RemoteWebDriver driver = new InternetExplorerDriver();

    driver.Navigate()
        .GoToUrl("http://en.wikipedia.org/wiki/Main_Page");

    driver.FindElementByCssSelector("@div#simpleSearch")
        .SendKeys("Webdriver Selenium");

    Assert.AreEqual("http://en.wiki {...} Driver"
        , driver.Url);
}
```



Нет. Если у вас несколько СЛОЖНЫХ ТЕСТОВ

```
public void TestMethod2()
{
    RemoteWebDriver driver = new InternetExplorerDriver();
    driver.Navigate().GoToUrl("http://localhost:8085/");
    Thread.Sleep(2000);
    driver.Manage().Cookies.DeleteAllCookies();
    Thread.Sleep(2000);
    driver.FindElement(By.XPath("//a[text()='Log On']")).Click();
    Thread.Sleep(2000);
    Assert.That(driver.Title, Is.EqualTo("Task Tracker - Log On"));
    driver.FindElement(By.Id("UserName")).SendKeys("testuser");
    driver.FindElement(By.Id("Password")).SendKeys("password");
    driver.FindElement(By.Id("LogOn")).Click();
    Thread.Sleep(2000);

    Assert.That(driver.Title, Is.EqualTo("Task Tracker - Home"));
    Assert.Throws<NoSuchElementException>(
        () => driver.FindElement(By.XPath("//a[text()='Log On']")));

    driver.FindElement(By.XPath("//a[text()='Log Off']")).Click();
    Thread.Sleep(2000);
    Assert.That(driver.Title, Is.EqualTo("Task Tracker - Home"));
    Assert.Throws<NoSuchElementException>(
        () => driver.FindElement(By.XPath("//a[text()='Log Off']")));
    Assert.Throws<NoSuchElementException>(
        () => driver.FindElement(By.XPath("//span[@welcome=''][text()='Welcome, test user!']")));
}
```



Ведь код можно улучшить

```
public void HomeIndex_LogIn_LogOut()
{
    RemoteWebDriver driver = new InternetExplorerDriver();
    // Navigate to the base url.
    driver.Navigate().GoToUrl("http://localhost:8085/");
    Wait();

    // Delete all cookies on the profile.
    driver.Manage().Cookies.DeleteAllCookies();
    Wait();

    // Click on the Log On link.
    IWebElement logOnLink = driver.FindElement(By.XPath("//a[text()='Log On']"));
    logOnLink.Click();
    Wait();

    // Check that the page title is that of the User/Logon page.
    Assert.That(driver.Title, Is.EqualTo("Task Tracker - Log On"));

    // Enter the username and password.
    IWebElement userName = driver.FindElement(By.Id("UserName"));
    userName.SendKeys("testuser");
    // ...
}
```

Как улучшить?

- ✓ Вынести часто используемый функционал в **общедоступные** методы
- ✓ Отформатировать код
- ✓ Добавить комментарии

```
driver.FindElement(By.XPath("//a[text()='Log On']"))  
.Click();
```



```
// Click on the Log On link.  
var logOnLink = driver.FindElement(  
    By.XPath("//a[text()='Log On']"));  
logOnLink.Click();
```

И, если у вас все под
контролем...





BAM HE HYЖEH PAGE OBJECT!



**ЧТО ОЖИДАЛОСЬ ПОСЛЕ
ПРЕДЫДУЩЕГО СЛАЙДА**



НО, ПОЛУЧИЛОСЬ В ИТОГЕ...

Иногда, мы не можем
контролировать ВСЁ



Иногда, мы не можем контролировать ВСЁ

- ✓ **"Тест-простыней"** становится **слишком много**
- ✓ **В коде не разобраться** без пол-литру
- ✓ На **поддержку** уходит уж **слишком много времени**
- ✓ Легче **всё переписать** заново
- ✓ Начинаете **ненавидеть разработчиков**,
которые вынесли `div` из `span`

PageObject – он как книжная полка

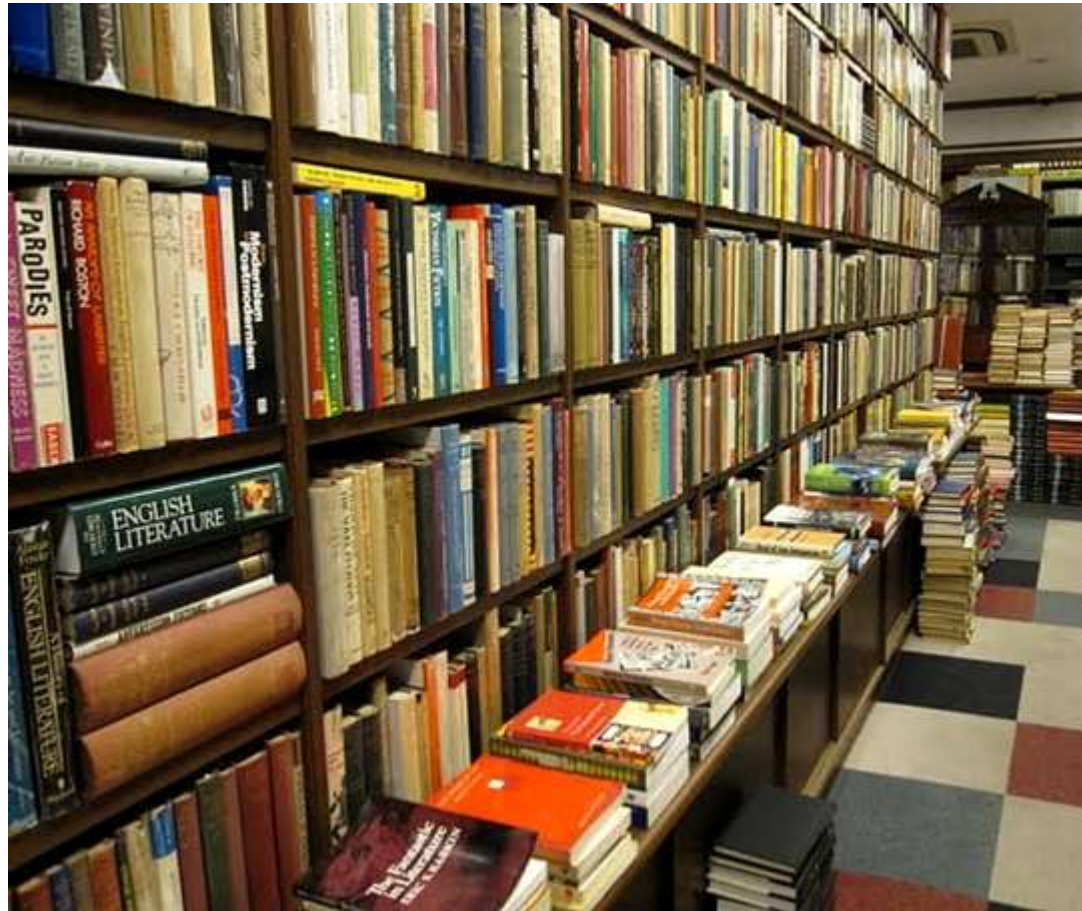
Pages

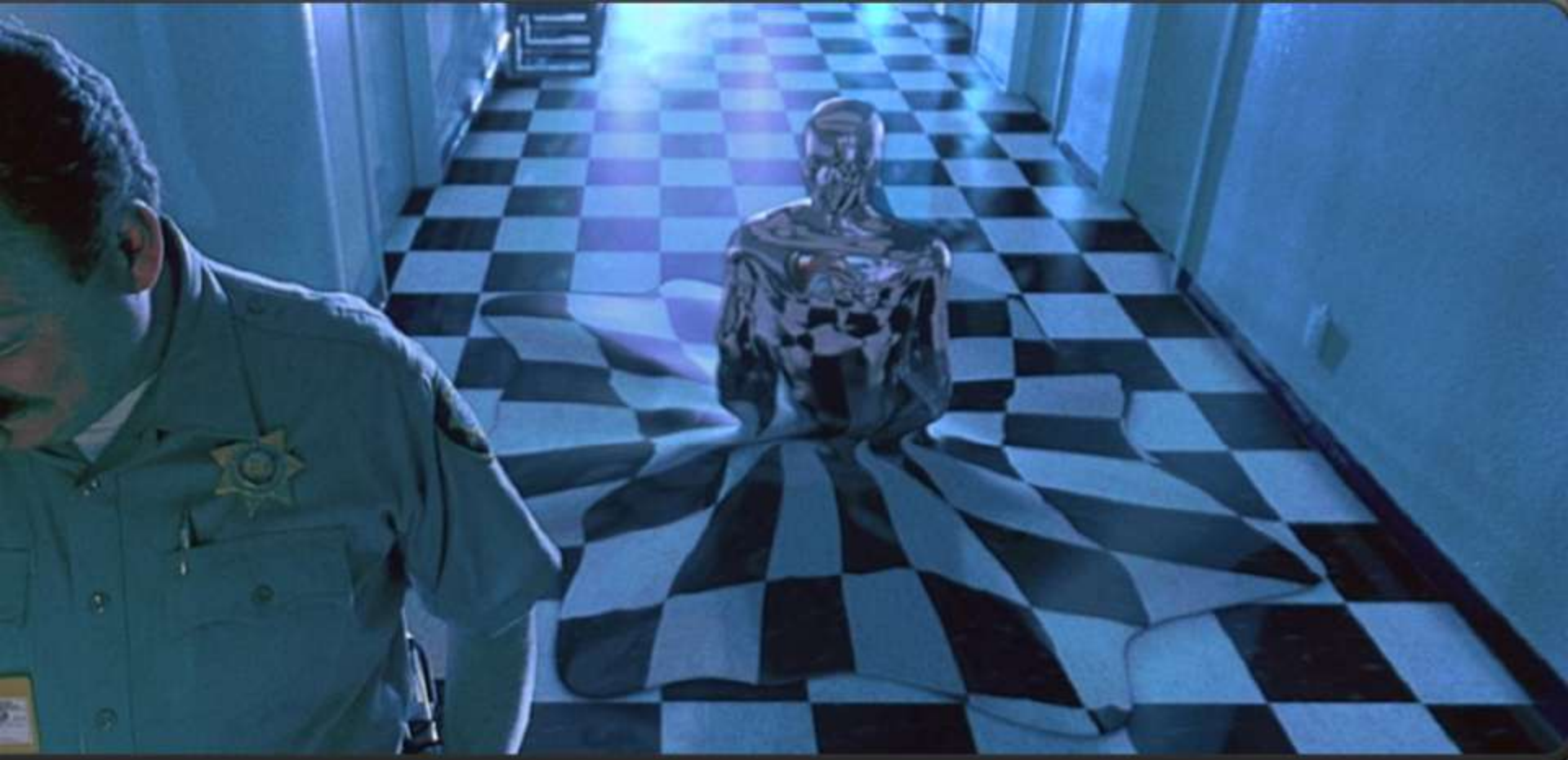
LoginPage

→ Login(name, passwd)

MainPage

→ Logout()
→ GotoProjects()
→ GotoUserProfile()
→ Search(text)





PAGE OBJECT – ОН ГИБКИЙ

PageObject – главная цель

- ✓ Обеспечить **хранение локаторов** в отдельном классе
- ✓ Обеспечить **повторное использование** локаторов и/или действий над страницей без дублирования кода
- ✓ Обеспечить **слой абстракции** от «драйвера» так, чтобы в тестах не использовались физические элементы идентификации элементов управления приложением

Свойства PageObject

- ✓ **Декларирует одну** и лишь одну **страницу** приложения
- ✓ **Страница** может быть **логической** либо **физической**
- ✓ Может содержать другие объекты PageObject (**композиция**)
- ✓ Может наследоваться от базовых классов PageObject (**наследование**)

А ВЫГЛЯДИТ ВСЁ ВОТ ТАК



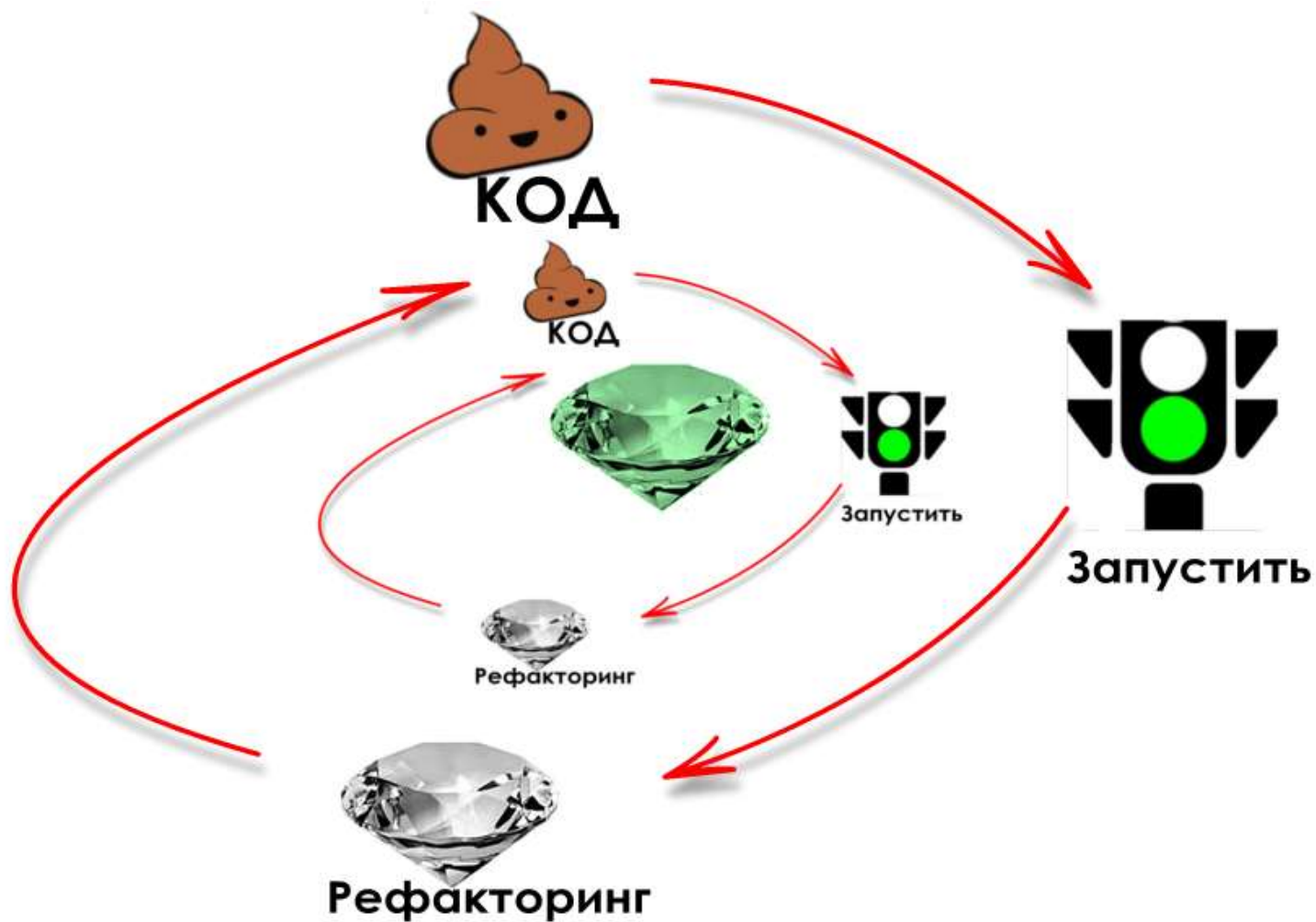
MainPage



**Create
Account
Page**

**T
E
S
T**

Демо!





**ЧТО ЭТО БЫЛО? –
СТАТИЧЕСКИЙ PAGE OBJECT!**

Позитив 01: Тест став читабелен

```
public static RemoteWebDriver Driver { get { return WebBrowser.Driver; } }
```

```
[TestMethod]
```

```
public void Donate_test_static()
```

```
{
```

```
    MyPages.MainPage.Open();
```

```
    MyPages.MainPage.GoToDonatePage();
```

```
    MyPages.DonatePage.Donate_50_UAH_Using_Debit_Card();
```

```
    MyPages.DonationPaymentsForm.FillDonationForm
```

```
    (
```

```
        firstName : "Vasya",
```

```
        lastName : "Pupkin",
```

```
        securityCode : "555"
```

```
    );
```

```
    // CUT
```

```
}
```


Позитив 2: Удобная «точка входа» в страницы

```
19 [TestMethod]
20 public void Donate_test_static()
21 {
22
23
24
25
26
27
28
29
30
31
32
```

MyPages.

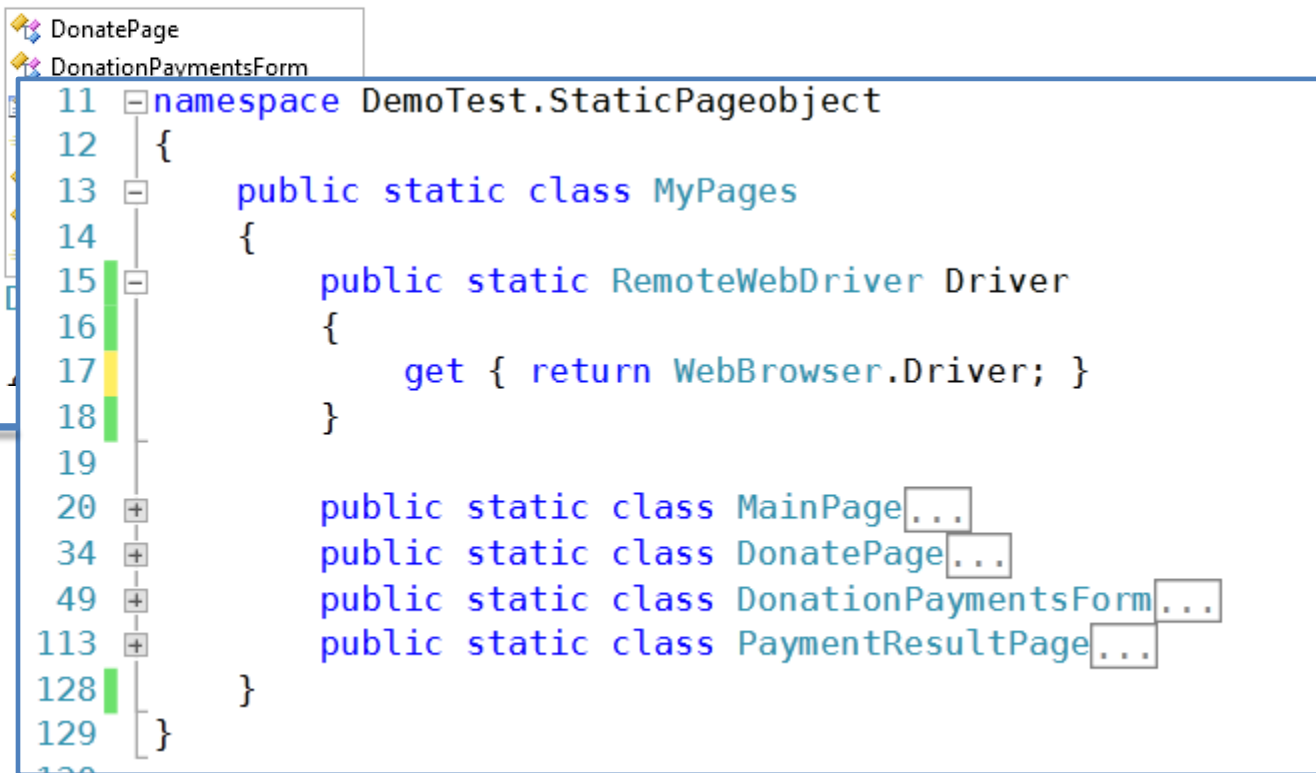
MyPages.

MyPages.

MyPages.

MyPages.

(



Позитив 3: Driver стал доступен отовсюду!

```
public static class WebBrowser
{
    public static RemoteWebDriver _driver = null;

    public static RemoteWebDriver Driver
    {
        get
        {
            _driver = _driver ?? new InternetExplorerDriver();
            return _driver;
        }
    }
}
```

```
public static RemoteWebDriver Driver {
    get { return WebBrowser.Driver; }
}
```

И еще раз о позитиве

- ✓ Теперь **жизнь** браузера (**драйвера**) контролирует класс **WebBrowser**.
- ✓ **Текущий драйвер** доступен из любого участка кода
- ✓ Любая **страница доступна** из **MyPages.***
- ✓ Наш тест теперь помещается на один экран монитора

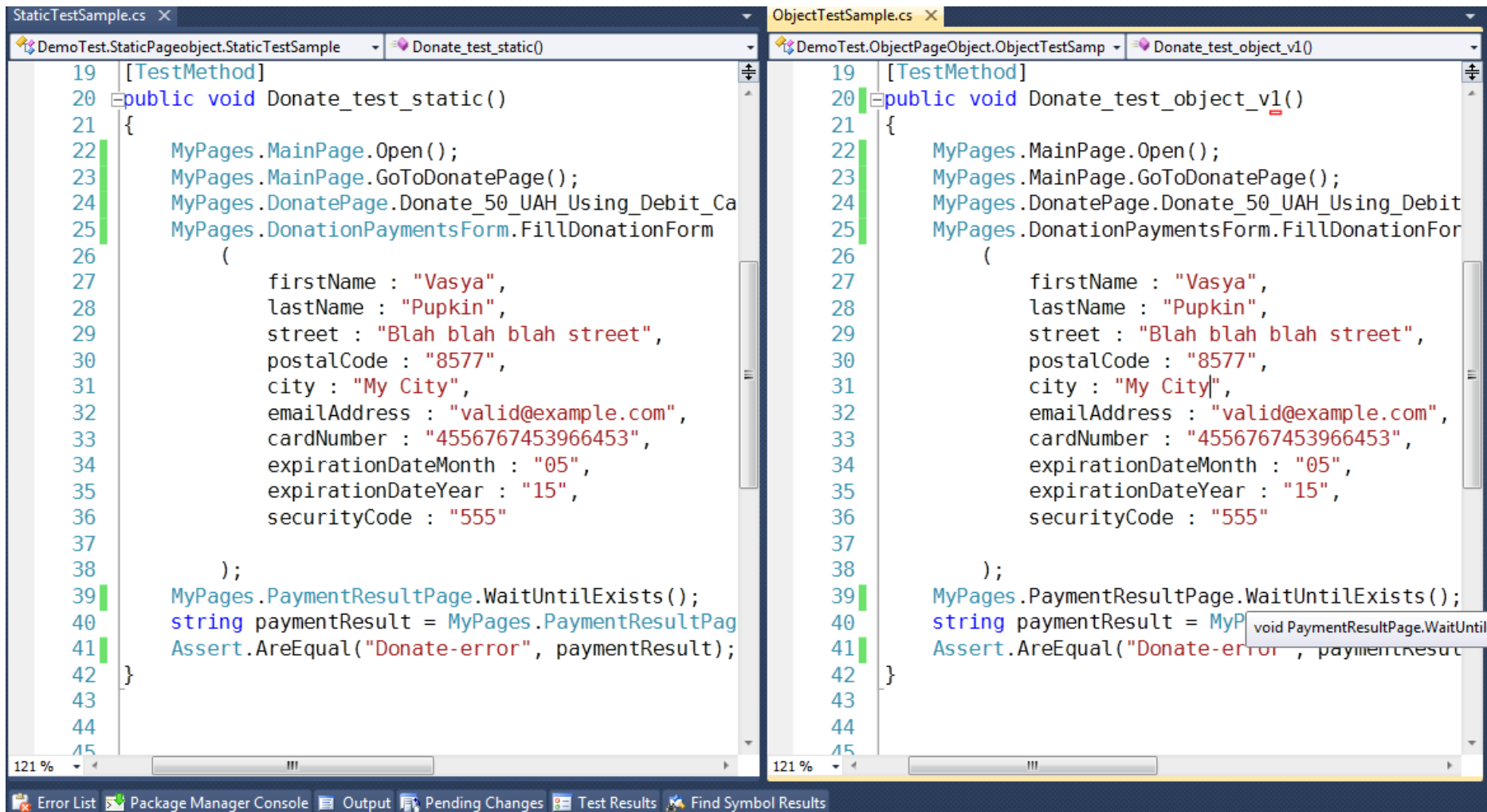
Ограничения статического класса в C#

	class	static class
Наследоваться	можно	нельзя
Породить много разных	можно	нельзя
Инициализировать параметрами через конструктор	можно	нельзя
Завалить	можно	полностью нельзя. Очень живучие
Присвоить переменной	можно	нельзя
Передать как параметр другому методу	можно	нельзя
Реализовать интерфейс	можно	нельзя
Использовать статические методы	можно	можно
Использовать статические поля	можно	можно
Инициализировать при помощи Webdriver PageFactory	можно	нельзя "из коробки", нужно написать свою фабрику



**ГИБКОСТЬ – ЭТО ВАЖНО.
PAGE OBJECT КАК OBJECT**

Тест не изменился! Спасибо, MyPages



```
StaticTestSample.cs x
DemoTest.StaticPageObject.StaticTestSample Donate_test_static()
19 [TestMethod]
20 public void Donate_test_static()
21 {
22     MyPages.MainPage.Open();
23     MyPages.MainPage.GoToDonatePage();
24     MyPages.DonatePage.Donate_50_UAH_Using_Debit_Ca
25     MyPages.DonationPaymentsForm.FillDonationForm
26     (
27         firstName : "Vasya",
28         lastName : "Pupkin",
29         street : "Blah blah blah street",
30         postalCode : "8577",
31         city : "My City",
32         emailAddress : "valid@example.com",
33         cardNumber : "4556767453966453",
34         expirationDateMonth : "05",
35         expirationDateYear : "15",
36         securityCode : "555"
37     );
38     MyPages.PaymentResultPage.WaitUntilExists();
39     string paymentResult = MyPages.PaymentResultPag
40     Assert.AreEqual("Donate-error", paymentResult);
41 }
42
43
44
45
121 %

ObjectTestSample.cs x
DemoTest.ObjectPageObject.ObjectTestSamp Donate_test_object_v1()
19 [TestMethod]
20 public void Donate_test_object_v1()
21 {
22     MyPages.MainPage.Open();
23     MyPages.MainPage.GoToDonatePage();
24     MyPages.DonatePage.Donate_50_UAH_Using_Debit
25     MyPages.DonationPaymentsForm.FillDonationFor
26     (
27         firstName : "Vasya",
28         lastName : "Pupkin",
29         street : "Blah blah blah street",
30         postalCode : "8577",
31         city : "My City",
32         emailAddress : "valid@example.com",
33         cardNumber : "4556767453966453",
34         expirationDateMonth : "05",
35         expirationDateYear : "15",
36         securityCode : "555"
37     );
38     MyPages.PaymentResultPage.WaitUntilExists();
39     string paymentResult = MyP
40     Assert.AreEqual("Donate-error", paymentResult);
41 }
42
43
44
45
121 %

Error List Package Manager Console Output Pending Changes Test Results Find Symbol Results
```


Изменения в MyPages

```
public static class MyPages  
{
```

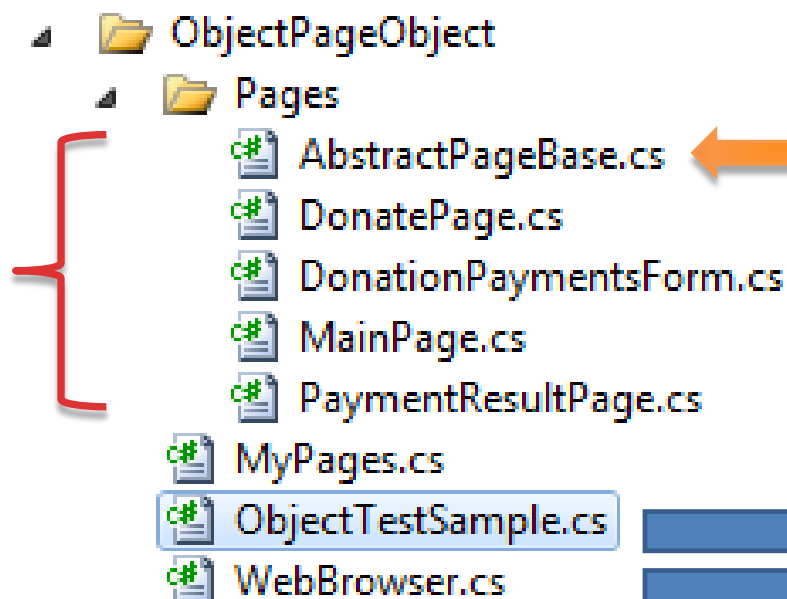
```
    public static MainPage MainPage  
    {  
        get { return new MainPage(); }  
    }
```

```
    public static DonatePage DonatePage  
    {  
        get { return new DonatePage(); }  
    }
```

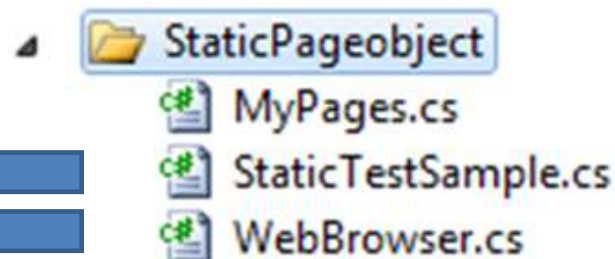
```
}
```

Новая структура проекта

✓ Object



✓ Static



AbstractPageBase

```
public abstract class AbstractPageBase
{
    public RemoteWebDriver Driver
    {
        get { return WebBrowser.Driver; }
    }
}
```

PaymentResultPage

```
public class PaymentResultPage : AbstractPageBase
{
    public void WaitUntilExists()
    {
        WebDriverWait wait = new WebDriverWait(Driver,
                                                TimeSpan.FromSeconds(30));
        wait.Until(ExpectedConditions.TitleContains(
            @"Donate-error - Payments"));
    }

    public string GetResultHeaderText()
    {
        var lblFirstHeader = Driver.
            FindElementById(@"firstHeading");
        return lblFirstHeader.Text;
    }
}
```

Changelog

- ✓ Появился **базовый класс** `AbstractPageBase`
- ✓ Из **MyPages.*** можно **по-прежнему получить любую страницу**
- ✓ Декларация страниц была вынесена из **MyPages** в отдельные файлы
- ✓ Мы лишились ограничений статических классов
- ✓ **ТЕПЕРЬ НАС НЕ ОСТАНОВИТЬ!**



ИНТЕРФЕЙСЫ И РОЛИ

linvokable

```
public interface IInvokable  
{  
    void Invoke();  
    bool Exists();  
}
```

Invoke – ВЫЗВАТЬ

Invokable – ТО, ЧТО МОЖНО ВЫЗВАТЬ

I invokable – Я ВЫЗЫВАЕМЫЙ(-оя, -ое)!

НужнаяPage.Invoke():

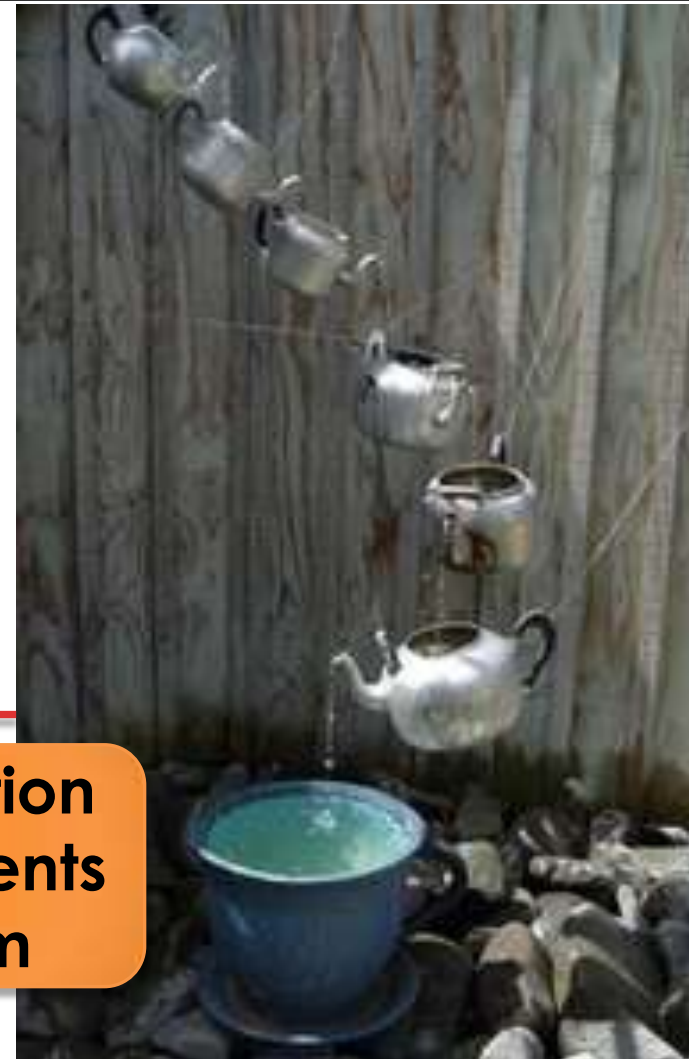
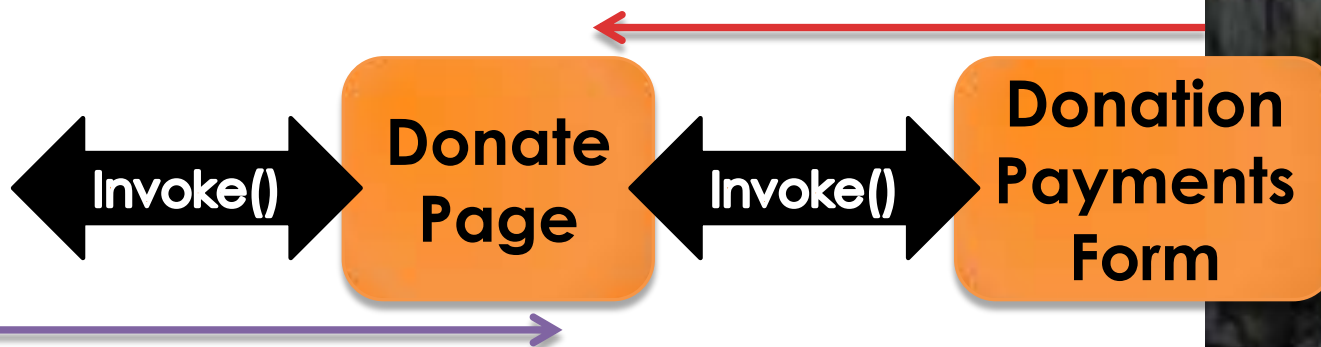
НужнаяPage.Invoke():

Если я уже **Exists()** – то вот она я!

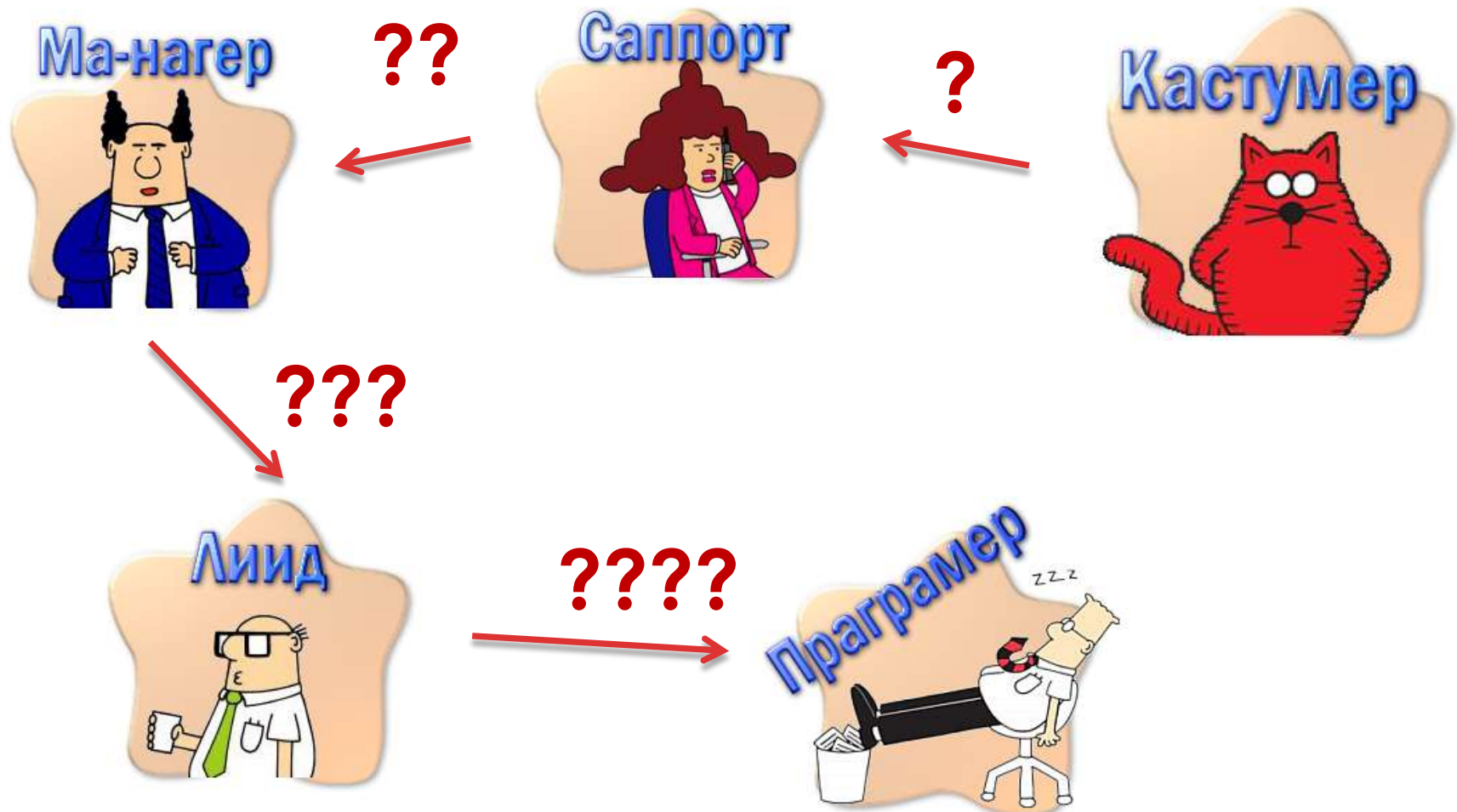
Если я **не Exists()**, то я сделаю:

ПредыдущаяPage.Invoke()

Потом **что-то нажму** – вот она я!



Chain of Responsibility



Что дает нам invocable?

- ✓ Любую страницу можно вызвать с параметрами **по умолчанию**

```
var page = MyPages.  
    PaymentResultErrorInvalidCreditCard;  
  
page.Invoke();
```

Что дает нам `Invokable`?

- ✓ Передать страницу как параметр метода

```
public void TestThatPageExists(Invokable page)
{
    page.Invoke();
    Assert.IsTrue(page.Exists());
}
```

```
[TestMethod]
public void Test_PaymentResultErrorInvalidCreditCard()
{
    TestThatPageExists(
        MyPages.PaymentResultErrnvalidCreditCard
    );
}
```