

< Teach  
Me  
Skills />

## Архитектура. Page Object



# С чего начинается автоматизация



1

Определение логически-выделенных областей приложения

2

Создание чеклиста и выбор проверок для автоматизации

3

Разделение на страницы/элементы и создание архитектуры  
фреймворка

4

Написание тестов

# Page Object Pattern

**Page Object** – это шаблон проектирования, который широко используется в автоматизированном тестировании и позволяет разделять логику выполнения тестов от их реализации. **Page Object** как бы моделирует страницы тестируемого приложения в качестве объектов в коде. В результате его использования у вас получатся отдельные классы, отвечающие за работу с HTML каждой конкретной веб-страницы. Такой подход значительно уменьшает объем повторяющегося кода, потому что одни и те же объекты страниц можно использовать в различных тестах. Основное преимущество **Page Object** заключается в том, что в случае изменения пользовательского интерфейса, можно выполнить исправление только в одном месте, а не исправлять каждый тест, в котором этот интерфейс используется.

Класс **PageObject** не обязательно должен представлять собой всю страницу. Он может быть частью страницы, которая часто используется на сайте (или даже на одной странице). Это может быть, например, пагинатор или меню. Основной принцип состоит в том, что есть только один класс в Вашем проекте, который знает о структуре HTML конкретной страницы или ее части.

# SauceDemo

<https://www.saucedemo.com/index.html>



 SWAGLABS

Username

Password

**LOGIN**



Accepted usernames are:

- standard\_user
- locked\_out\_user
- problem\_user
- performance\_glitch\_user

Password for all users:

secret\_sauce

# Страница продуктов

The screenshot shows a product listing page for SWAGLABS. The header features the SWAGLABS logo and a search bar labeled "Name (A to Z)". The products are displayed in a grid format:

- Sauce Labs Backpack**: \$29.99. Description: carry all the things! with the sleek, streamlined Sly Pack that melds uncompromising style with unequalled laptop and tablet protection.
- Sauce Labs Bike Light**: \$9.99. Description: A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.
- Sauce Labs Bolt T-Shirt**: \$15.99. Description: Get your testing superhero on with the Sauce Labs bolt t-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt.
- Sauce Labs Fleece Jacket**: \$49.99. Description: It's not every day that you come across a midweight quarter-zip fleece jacket capable of handling everything from a relaxing day outdoors to a busy day at the office.
- Sauce Labs Onesie**: \$7.99. Description: Rib snap infant onesie for the junior automation engineer in development. Reinforced 3-snap bottom closure, two-needle hemmed sleeves and bottom won't unravel.
- Test.allTheThings() T-Shirt (Red)**: \$15.99. Description: This classic Sauce Labs t-shirt is perfect to wear when cozying up to your keyboard to automate a few tests. Super-soft and comfy ringspun combed cotton.

Each product listing includes a small image, the product name, its price, and a red "ADD TO CART" button.

# Страница корзины



## Your Cart

QTY DESCRIPTION

### Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequalled laptop and tablet protection.

1

29.99

REMOVE

CONTINUE SHOPPING

CHECKOUT

# Страница checkout



## Checkout: Your Information

First Name

Last Name

Zip/Postal Code

CANCEL

CONTINUE

# Page Object Pattern - пример

Идея паттерна:

- Описывать каждую веб-страницу в отдельном классе

Плюсы:

- Работать с файлами на git проще, реже конфликты
- Тесты становятся читабельнее
- Переиспользование функционала

```
public class NewProjectPage extends BasePage {  
  
    private By CREATE_BUTTON = By.xpath("//button[text()='Create']");  
    private By NAME = By.name("name");  
    private By DESCRIPTION = By.xpath("//textarea[@name='description']");  
    private By DOMAINS_INPUT = By.name("domains[]");  
  
    public NewProjectPage(WebDriver driver) {  
        super(driver);  
    }  
  
    public void openPage() {  
        driver.get("https://dev.integrivideo.com/app/projects/new");  
    }  
  
    public NewProjectPage fillInFields(Project project) {  
        driver.findElement(NAME).sendKeys(project.getName());  
        driver.findElement(DESCRIPTION).sendKeys(project.getDescription());  
        for(int i = 0; i < project.getDomains().size(); i++) {  
  
            driver.findElements(DOMAINS_INPUT).get(i).sendKeys(project.getDomains().g  
et(i));  
        }  
        return this;  
    }  
  
    public NewProjectPage clickCreate() {  
        driver.findElement(CREATE_BUTTON).click();  
        return this;  
    }  
}
```

# BasePage

Класс, хранящий в себе драйвер и весь код, который многократно может быть использован в конкретных классах страниц

```
abstract class BasePage {  
    WebDriver driver;  
  
    BasePage(WebDriver driver) {  
        this.driver = driver;  
    }  
}
```

# BaseTest

Класс, использующий для настройки, открытия/закрытия драйвера и инициализации объектов страниц

```
public class BaseTest {  
  
    private WebDriver driver;  
    LoginPage loginPage;  
    NewProjectPage newProjectPage;  
  
    @BeforeMethod  
    public void setUp() {  
        System.setProperty("webdriver.chrome.driver",  
"src/test/resources/webdrivers/chromedriver");  
        driver = new ChromeDriver();  
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);  
  
        loginPage = new LoginPage(driver);  
        newProjectPage = new NewProjectPage(driver);  
    }  
  
    @AfterMethod(alwaysRun=true)  
    public void closeBrowser() {  
        driver.quit();  
    }  
}
```