**Паттер Page Object**

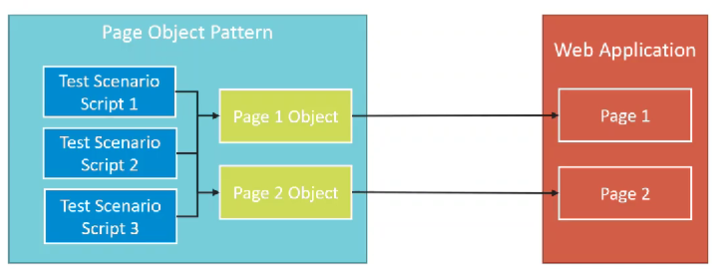
Широко используется при автоматизации UI, user interface.

**Core Ideas**

**Основные идеи, заложенные в паттерне:**

1. Разделение тестовых сценариев и контента страниц;
2. Все взаимодействие со страницей должно находится в одном месте.

В результате преобразования кода мы получаем 2 слоя. Тестовые сценарии, которые непосредственно не обращаются к страницам, а обращаются через слой page object (группу классов)



**Ideas vs Development**

**Ключевые принципы, для сохранения чистого кода:**

1. *DSL (Domain Specific Language)* – специфичный язык для доменов. Именование должно говорить о специфике объекта;
2. *DRY (Don’t Repeat Yourself)* – не повторяться;
3. *KISS (Keep It Simple Stupid)* – делать проще;
4. *YAGNI (You Aren’t Gonna Need It)* – не реализовывать с ходу весь функционал.

**Structure - must have rules**

**Список критериев для оценки качества page object-ов:**

1. Содержимое страницы должно быть спрятано (с приватным модификатором доступа);
2. Все ассерты должны осуществляться за рамками класса page object;
3. Реализовывать только те взаимодействия с элементами страницы, относящиеся к тесту;
4. Несколько небольших объектов будет лучше, чем один огромный;
5. *Page Object* – динамическая сущность. Избегание static полей/объектов и т.д.

**Методы:**

1. Не должны содержать в себе функциональную логику;
2. Возвращать объекты страниц, не должны быть *void*;
3. Не злоупотреблять переопределением методов.

**Page Factory internals**

Эффективный механизм создания page object-ов.

**Основные аспекты работы:**

1. Ключевую роль играет метод *initElements*, удалять его нельзя;
2. Статичность, неизменность локаторов;
3. Компактный синтаксис.

**Page Factory - FindBys vs FindBy**

Способы записи аннотаций:

