**Page Object Model(POM)**

Если вы знакомы с любым видом фронтенд-веб-разработки, то, возможно, сталкивались с Document Object Model (DOM). DOM широко признан как тип API для документов HTML и XML, позволяющий получать доступ к логической структуре документа и манипулировать ей. Page Object Model работает по схожему принципу. Можно сказать, что POM относится к тест-автоматизации веб-приложений так же, как DOM – к фронтенд-веб-разработке. Page Object Model – это паттерн проектирования в Selenium, создающий репозиторий объектов для хранения всех веб-элементов. Это полезно, так как помогает инженерам-автоматизаторам снизить дупликацию кода и улучшить поддержку тест-кейсов. Каждый файл класса хранит все элементы определенной веб-страницы. Эти элементы затем используются тестировщиками для выполнения операций над тестируемым сайтом.

**Преимущества**

1. Очевидное преимущество POM – это простота, позволяющая поддерживать тест-кейсы в случае изменений в UI/веб-элементе. Хороший пример: выпадающее меню изменилось на чек-бокс. POM поможет вам выявить изменившуюся страницу, так как каждая страница имеет отдельный файл класса.
2. Репозиторий объектов не зависит от тест-кейсов – это отдельные сущности, и это значит, что один и тот же репозиторий можно использовать для разных задач и при помощи различных инструментов. К примеру, репозиторий можно использовать для запуска функциональных тестов через фреймворк вроде TestNG, и в то же самое время – для приемочного тестирования с Cucumber.
3. При разделении объектов репозитория и тест-кейсов требуется меньше кода, и операции оптимизируются, так как страницы можно использовать повторно.
4. Скрипты становятся читабельнее, и за ними легко следить – методы получают более реалистичные имена, которые легко ассоциировать с выполняемыми ими операциями.