

Ex 13

Рассуждения о текущей реализации кода и предложения по изменению архитектуры тестового фреймворка под iOS и Android

В текущей реализации кода (Ex 11, Ex 12), на мой взгляд? присутствует ряд недостатков, которые могут негативно влиять на читаемость кода, а также усложнять его поддержку, модификацию и расширение.

Из-за того, что тестируемые приложения под iOS и Android не идентичны, в коде тестов присутствуют проверки, на какой платформе в данный момент выполняется код. Это делает код менее читабельным, чтобы понять логику его работы требуется больше времени. Также это – дополнительные сложности как в поддержке кода (сложнее локализовать ошибки в коде), так и в его модификации (если функционал тестов будет расти, будет расти и количество проверок платформы «if», замусоривающих код). Еще одна проблема состоит в сложности масштабирования. К примеру, нам потребуется тестировать отдельно приложение под мобильный WEB, Tizen, Harmony (или какую-нибудь вновь появившуюся мобильную ОС). В этом случае модифицировать код тестов будет крайне проблематично.

Другой момент, менее критичный, это – использование паттерна Factory Method для объектов PageObject. Родительские классы PageObject содержат методы для работы с UI всех тестируемых платформ. При этом, читая код, трудно сразу понять к какой платформе относится тот или иной метод. Также мы унифицируем наименование переменных, содержащих локаторы элементов интерфейса тестируемых приложений. Опять же, из-за различий приложений под iOS и Android, элементы в этих приложениях называются по-разному. Для облегчения поддержки тестов, как мне кажется, целесообразно именовать переменные близко к именованию соответствующих элементов в приложении. Например, кнопка отмены поиска в приложении под iOS именуется «Cancel», а под Android – «Close Search». Если в тестах под iOS у нас будет переменная cancelButton, а в тестах под Android – closeSearchButton, это сделает поддержку тестов более простой, чем если бы мы использовали одинаковый для обеих платформ идентификатор переменной cancelSearchButton, например.

В связи с этим, предлагается внести следующие изменения в архитектуру тестового фреймворка:

1. При реализации тестов использовать паттерн Template Method;
2. Разделить тесты на отдельные классы для каждой тестируемой платформы;
3. Использовать у классов тестов аннотацию @Category, что позволит запускать из CI / CD тесты под нужную платформу;
4. Так как определение платформы из переменной окружения больше не требуется, специализированный класс Platform больше не нужен. Что снижает риски утечки памяти из-за отказа от Singleton-объекта (чисто мое предубеждение 😊);
5. Так же из-за отмены необходимости проверять платформу, отказываемся от паттерна Factory Method для объектов PageObject.