

ЭРГОНОМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ФРЕЙМВОРКА АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савосик А. А.

Меженная М. М. – канд. техн. наук,
доцент каф ИПиЭ

Цель разработки – создание фреймворка, автоматизирующего рабочий процесс тестирования веб-приложений. Фреймворк представляет собой два модуля (рисунок 1), разработанных с применением Selenium – наиболее популярного фреймворка для автоматизации тестирования [1] и TestNG – фреймворка управления структурой тестов. Следует отметить, что разрабатываемый продукт для автоматического тестирования программного обеспечения не выступает конкурентом вышеперечисленных Selenium и TestNG, а объединяет и расширяет их возможности, адаптирует API (Application Programming Interface - программный интерфейс приложения) под задачи программиста. В результате фреймворк позволяет своим пользователям производить тестирование различных веб-приложений в более простом и понятном виде, не затрачивая много времени на работу с низкоуровневыми фреймворками, такими как Selenium и TestNG, а используя разработанный упрощенный и эргономичный с точки зрения программирования API.

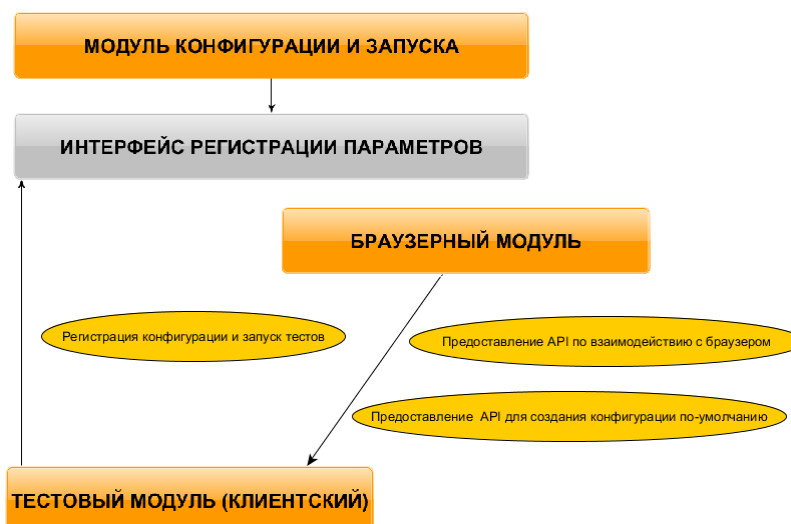


Рисунок 1 – Архитектура фреймворка автоматического тестирования программных приложений

В рамках разрабатываемого фреймворка для автоматического тестирования программного обеспечения реализован модуль конфигурации и запуска тестов, который предоставляет интерфейс для регистрации конфигурационных параметров. После определения финальных настроек, данный модуль осуществляет запуск указанных клиентом тестов. Клиентом и пользователем фреймворка в м случае выступает специалист по автоматизации тестирования конкретного веб-приложения.

Также реализован браузерный модуль фреймворка, который предоставляет API для работы с браузерами различных типов: Google Chrome, Mozilla Firefox и Microsoft Internet Explorer. Данный модуль позволяет осуществлять конфигурацию браузеров под любые нужды и повышает абстрактность кода, оставляя низкоуровневую логику взаимодействия с сущностями браузера внутри модуля. Осуществлена поддержка популярных в автоматизации тестирования шаблонов проектирования Page Object и Page Factory [2-3]. Пользователю – специалисту по автоматизации – остается разработать последний модуль – тестовый, который является уникальным для каждого отдельного веб-приложения.

Список использованных источников:

1. Приемы проектирования тестов // Selenium2 [Электронный ресурс]. – 2017. – Режим доступа: <http://selenium2.ru/docs/test-design-considerations.html>. – Дата доступа: 18.11.2017.
2. Паттерн Factory Method (фабричный метод) // CPP-Reference [Электронный ресурс]. – 2017. – Режим доступа: <http://cpp-reference.ru/patterns/creational-patterns/factory-method/>. – Дата доступа: 25.11.2017.
3. Куликов, С. Тестирование программного обеспечения – Базовый курс / С. Куликов. – М.: EPAM Systems, 2015г. - 289 с.