

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.4

Хильчук
Алексей Сергеевич

Программное средство автоматической генерации интерфейсных объектов
для тестирования веб и мобильных приложений

АВТОРЕФЕРАТ

на соискание академической степени
магистра технических наук

по специальности 1-40 80 05 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель
Куликов С.С.
к.т.н., доцент

Минск 2019

КРАТКОЕ ВВЕДЕНИЕ

Все последние годы сеть Интернет стремительно развивается, проникая как в подавляющее большинство сфер бизнеса, так и в общественную и культурную жизнь. Создание собственных сайтов играет существенную роль в большинстве сфер бизнеса, будь то осуществление покупок в интернет-магазине или управление производством на предприятии.

В области разработки веб и мобильных приложений образовалась высокая конкуренция вследствие того, что приложение может приносить прибыль для заказчика и даже небольшая ошибка может повлиять на выбор в сторону программного решения конкурента. Разработка программного продукта предполагает решение задач в областях вычислительных систем, дизайна, баз данных, сетевой безопасности и др. Тестирование проводится практически для каждой из стадии разработки.

Тестирование ПО – это процесс исследования, испытания программного продукта, по результатам которого можно выявить ситуации, где поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Разделяют два основных вида тестирования: ручное и автоматизированное. При ручном тестировании инженер по тестированию самостоятельно производит все действия по симулированию действий реального пользователя сайта, в процессе которых следит за корректностью работы функционала сайта, его оформлением и удобством использования и др.

Автоматизированное тестирование подразумевает написание программ, автоматически выполняющие тестовые сценарии [1]. С помощью таких программ инженер по тестированию может получить информацию о возможно обнаруженных дефектах за достаточно короткий промежуток времени. Этот подход требует большое количество ресурсов на начальном этапе внедрения т.к. на разработку каждого автотеста требуется гораздо больше времени чем однократное выполнение сценария вручную, но со временем автоматизированное тестирование начинает приносить выгоду вследствие того, что нет необходимости перепроверять функциональность каждой сборки приложения, покрытого автотестами, а, следовательно, это экономия как времени инженеров по тестированию, так и бюджета компании-разработчика.

Диссертационная работа посвящена разработке программного средства, позволяющего увеличить скорость разработки автоматических тестов путем автоматического генерирования данных, необходимых для написания автоматических тестовых сценариев, тем самым повышая эффективность внедрения и проведения автоматизированного тестирования на проекте по разработке программного продукта, в частности веб и мобильных приложений на базе ОС Android.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является разработка алгоритмов и программного обеспечения, позволяющего увеличить скорость разработки автоматизированных тестов для веб и мобильных приложений путем автоматической генерации кода объявления интерфейсных объектов на языках программирования Java и C#.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить технологии и паттерны проектирования, которые используются в разработке автоматизированных тестов.
2. Разработать требования и будущую архитектуру проектируемого программного средства.
3. Разработать методы и алгоритмы, необходимые для генерации локаторов элементов веб-страницы и страницы мобильного приложения.
4. Реализовать по для автоматической генерации кода объявления интерфейсных объектов.

Объектом исследования является программное обеспечение, предоставляющее возможности автоматической генерации интерфейсных объектов для тестирования веб и мобильных приложений.

Предметом исследования является программное обеспечение компьютерных систем для решения задачи автоматической генерации интерфейсных объектов для тестирования веб и мобильных приложений.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность использования отдельного ПО, работающего на ОС Windows, Linux и MacOS, для решения задачи написания кода объявления интерфейсных объектов, а так же взаимодействия с ними для тестирования веб и мобильных приложений.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии с научно-техническим заданием и планом работ кафедры «Программное обеспечение информационных технологий» по теме «Программное средство автоматической генерации интерфейсных объектов для тестирования веб и мобильных приложений» (ГБ № 16-2004 «Разработка моделей, методов, алгоритмов, повышающих показатели проектирования, внедрения и эксплуатации программных средств для перспективных платформ обработки информации, решения интеллектуальных задач, работы с большими массивами данных и внедрение в современные обу-

чающие комплексы», раздел «Анализ способов повышения показателей эффективности программных средств для решения интеллектуальных задач», научный руководитель НИР – С. С. Куликов).

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя С. С. Куликова, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на научно-технической и научно методической конференции «Современные технологии в науке и образовании – СТНО-2017» (Рязань, Россия, 2017); 53-ей научной конференции аспирантов, магистрантов и студентов БГУИР (Минск, Беларусь, 2017); X Международная научно-методическая конференция «Дистанционное обучение – образовательная среда XXI века» (Минск, Беларусь, 2017); 54-ой научной конференции аспирантов, магистрантов и студентов БГУИР (Минск, Беларусь, 2018).

Опубликованность результатов диссертации

По теме диссертации опубликовано 4 электронных научных работ в сборниках трудов и материалах международных и научных конференций.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, шести глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе предоставлен анализ предметной области, технологий и паттернов, используемых в разработке и автоматизации тестирования веб и мобильных приложений, а так же описаны требования к проектируемому программному средству. Вторая глава посвящена моделированию предметной области, где описываются функции разрабатываемого ПО, а так же разработке функциональных требований к программному продукту. Третья глава посвящена разработке методов и алгоритмов генерации локаторов различных типов к элементам веб страницы или страницы мобильного приложения, генерации класса, в котором должны быть описаны вышеупомянутые объекты. В четвертой главе предложены описания классов и методов разрабатываемого программного средства. Тестирование разработанного ПО и методика использования разработанного программного средства предложены в пятой и шестой главе соответственно.

Общий объем работы составляет 149 страниц, из которых основного текста – 54 страницы, 37 рисунков на 20 страницах, 1 таблица на 21 странице, библиографический список из 34 наименований на 3 страницах и 2 приложения на 51 странице.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначена практическая ценность работы.

В **первой главе** проведен анализ процесса улучшения регрессионного тестирования программных средств путем внедрения автоматизации тестирования [2]. Проведен анализ применяемых технологий и фреймворков в процессе разработки автоматизированных тестов [3], проведен обзор аналогов программных средств, сформированы требования к проектируемому программному средству.

В настоящий момент регрессионное тестирование может занимать до 90% от общего объема времени, затрачиваемого на проверку очередной сборки ПО. Для снижения данных затрат внедряется автоматизация проведения тестирования.

К плюсам автоматизированного тестирования можно отнести быстрое выполнение тестовых сценариев, исключение «человеческого фактора», возможность высвобождения времени тестировщика, а также автоматическая генерация отчетов. Но существует и ряд минусов, основным из которых можно назвать трудоёмкость проектирования автоматизированных тестов, а также их последующая однотипность, которая отсутствует при ручном тестировании. В разработке автоматизированных тестов используются такие технологии и фреймворки, как HTML/XHTML, XML Path Language, CSS, Selenium WebDriver, Appium [4] и прочие.

Вторая глава посвящена разработке функциональных требований к программному средству генерации интерфейсных объектов для тестирования веб и мобильных приложений.

В главе предложены следующие функции приложения (рисунок 1):

1) Чтение HTML кода веб-страницы из локального файла. После выбора пользователем кнопки вызова функции загрузки из файла ему будет предложено диалоговое окно навигации по файловой системе компьютера, на котором запущено приложение. После выбора файла производится его чтение и запись в память приложения.

2) Чтение HTML кода веб-страницы из сети интернет. Пользователь должен ввести URL веб страницы, анализ которой необходимо провести, и подтверждает ввод кнопкой вызова функции загрузки из сети интернет. После загрузки страницы, ее код должен быть записан в память приложения.

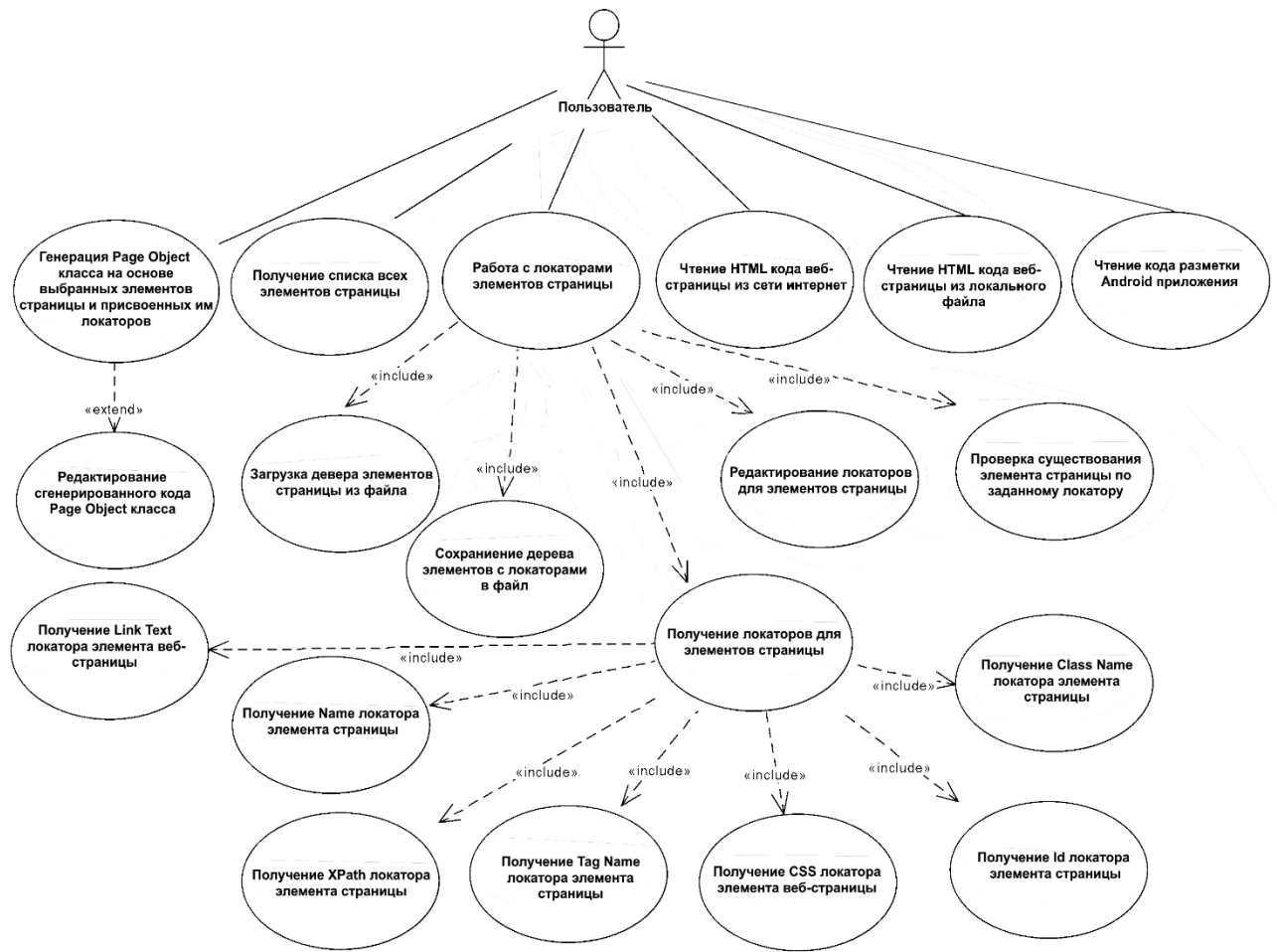


Рисунок 1 – Диаграмма вариантов использования разрабатываемого программного средства

3) Чтение кода разметки Android приложения. Пользователь должен ввести название пакета тестируемого Android приложения, путь к APK файлу Android приложения, а также название Activity компонента для запуска приложения.

4) Получение списка всех элементов страницы приложения. Пользователю выйдется список элементов выбранной страницы после выбора кнопки вызова функции анализа страницы.

5) Получение локов для элементов на странице. Пользователю, после выбора элемента страницы из списка, будет предложено выбрать тип локатора данного элемента. После выбора пользователем типа локатора, элементу будет присвоен полученный локатор.

6) Редактирование локов элементов страницы. Пользователю, после выбора элемента страницы из списка, будет предложено присвоить тип и значение локатора вручную.

7) Проверка существования элемента страницы по заданному локатору. Пользователь сообщает локатор элементу приложения, приложение должно ответить пользователю, существует ли элемент с таким локатором на анализируемой странице.

8) Сохранение списка найденных и созданных локаторов элементов страницы в файл. Пользователь сохраняет список элементов страницы в файл.

9) Загрузка списка элементов страницы из файла. Пользователь указывает файл, содержащий в себе список элементов страницы, для продолжения работы с ними.

10) Генерация Page Object класса на основе выбранных элементов страницы и присвоенных им локаторов. Пользователь указывает, какие элементы должны быть в сгенерированном Page Object классе. После выбора кнопки вызова функции генерации класса получает файл-класс, готовый к внедрению в фреймворк для проведения тестов.

11) Редактирование сгенерированного кода Page Object класса. Пользователь может внести изменения в сгенерированный Page Object класс.

Третья глава посвящена разработке архитектуры ПО, алгоритмов для генерации локаторов элементов страницы веб или мобильного приложения, модулей приложения.

На рисунке 2 изображена схема высокоуровневой архитектуры программного средства:

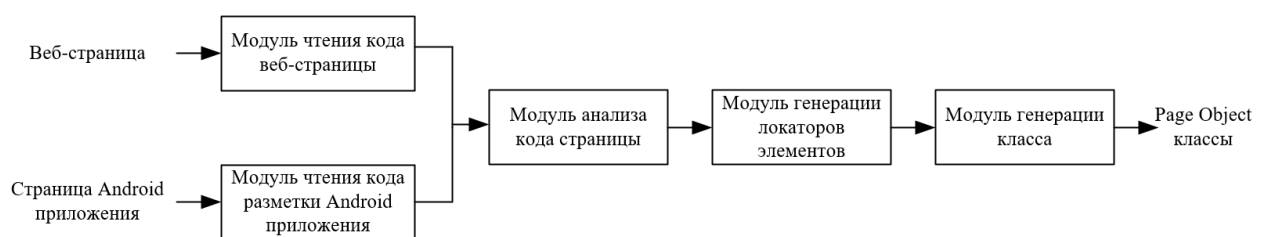


Рисунок 2 – Схема высокоуровневой архитектуры программного средства

В модуль чтения кода веб-страницы подается ссылка на локальный файл веб-страницы в файловой системе компьютера, на котором запущено приложение либо интернет ссылка на интересующую веб-страницу. Чтение страницы осуществляется браузером.

В модуль чтения кода разметки Android приложения подается ссылка на APK файл приложения в файловой системе компьютера, название пакета приложения и название запускаемой Activity приложения. Чтение страницы осуществляется посредством Android Debugging Bridge, поставляемое с Android SDK.

Результатом работы модуля анализа кода страницы будет иерархическая структура элементов страницы, полученная с помощью библиотеки JSoup. Данная библиотека имеет удобный API для извлечения и манипулирования данными, используя JQuery-методы взаимодействия с объектной модели и CSS страницы. С помощью интерфейса пользователь может выбрать элементы из иерархии, для которых необходимо генерировать локаторы.

Модуль генерации локаторов должен осуществить генерацию шести видов локаторов для элемента (Name, Id, Class Name, Tag Name, CSS, XPath). Генерация локаторов осуществляется на основе атрибутов, присущих данному элементу, а также его положению в иерархии элементов страницы.

Генерация кода класса определения интерфейсных объектов, а также методов взаимодействия с элементами страницы (Page Object класс) производится на основе выбранного локатора элемента и тэга, где он содержится.

В **четвертой главе** рассмотрена практическая реализация программного средства генерации интерфейсных объектов для тестирования веб и мобильных приложений.

В качестве языка программирования был использован Java, так как данный язык позволяет создавать достаточно расширяемые программные решения, а также имеет наибольшую программную совместимость с фреймворком Selenium WebDriver и Appium, так как фреймворк Selenium также написан с использованием языка Java, а для работы с Appium предоставлена Java библиотека.

Одним из показателей для выбора библиотеки парсинга веб-страниц является совместимость с версией HTML 5, которая на данный момент набирает все большую популярность в разработке сайтов. Также необходимо, чтобы библиотека смогла определять элементы, которые имеют старые принципы объявления атрибутов, которые присущи HTML 4.X, а также игнорирует лишние атрибуты и скрипты, никак не совершенствующие имеющийся код страницы. Среди многих библиотек, под данные критерии подходит JSoup, которая имеет один из самых удобных интерфейсов для работы с HTML и XML, а также активно поддерживается разработчиками.

Вследствие того, что код Page Object классов обычно делается по одному шаблону, для более быстрой и удобной постановки объявлений интерфейсных объектов, а также коды методов для взаимодействия с элементами будут использоваться шаблонизаторы. В основном они используются для генерации HTML страниц, поэтому одним из критериев для отбора шаблонизатора была возможность его работы не только с HTML шаблонами, но и с другими форматами. Под такие критерии подошел шаблонизатор Water Template Engine.

Система построена по модульно-функциональному принципу, когда основные функциональные действия реализуются отдельными классами. Диаграмма классов программного средства представлена на рисунке 3.

Пятая глава посвящена тестированию разработанного программного средства. Для автоматизации реализованных модульных и интеграционных тестов используется сервис непрерывной интеграции Jenkins CI, предназначенный для сборки приложения, а также проведения автоматизированного тестирования с функцией агрегации результатов выполнения тестов с помощью библиотеки Allure для Java.

В **шестой главе** предложена методика использования разработанного программного средства.

Для корректной работы данного программного средства необходим компьютер следующей минимальной конфигурации:

– ОС: Linux (Ubuntu 14.04 и выше, Mint 17 и выше, Debian 8 и выше), Windows (версии 7, 8.1, 10) или Mac OS X (Mavericks 10.9 и выше);

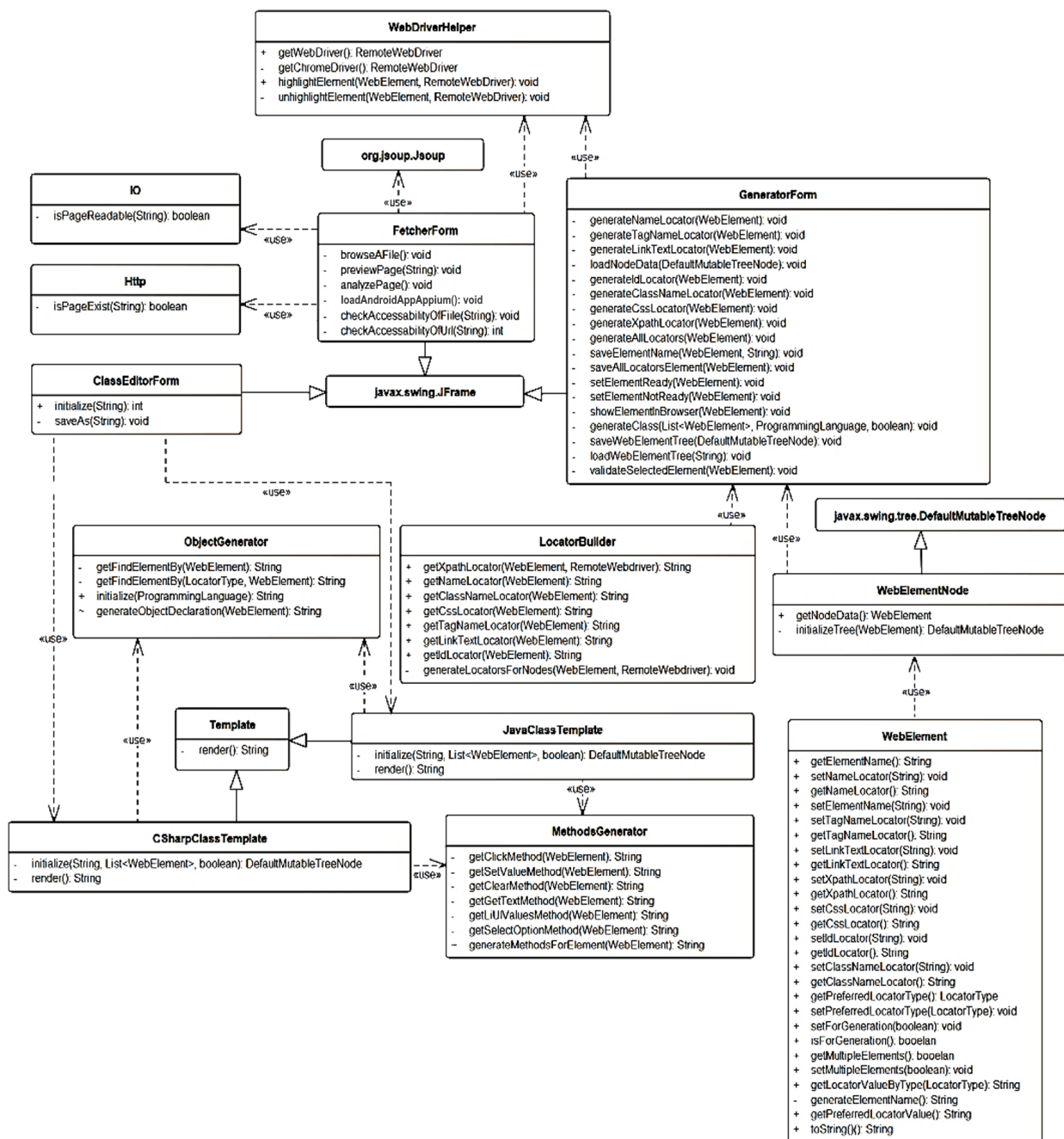


Рисунок 3 – Диаграмма классов программного средства

- Процессор: Intel Core i3 с поддержкой SSE2 и Intel VT;
- RAM: 2Гб;
- HDD: 4 Гб свободного места.

Дополнительно: наличие установленной в операционной системе Java Runtime Environment 8, Node JS и браузера Google Chrome версии 60 и выше.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

В результате выполнения работы было разработано программное средство автоматической генерации интерфейсных объектов для тестирования веб и мобильных приложений.

До реализации программного средства были исследованы технологии и фреймворки, которые используются в автоматизации тестирования веб и мобильных приложений. Также были проанализированы аналоги разрабатываемого программного средства, выявлены их достоинства и недостатки.

Кроме того, были исследованы типы локаторов, определяющие элементы страниц, способы их создания. Был изучен шаблон проектирования автоматизированных тестов Page Object, который сейчас является основополагающим паттерном при проектировании тестов. Также были изучены возможности фреймворка Selenium WebDriver и Appium.

Разработанное программное средство полностью соответствует заявленным функциональным требованиям и предлагает удобный и в то же время простой интерфейс взаимодействия.

В рамках разработки программного средства, описанного в данном проекте, были предложены и реализованы такие функции и алгоритмы, как:

- загрузка исходного кода локальной веб-страницы;
- загрузка исходного кода веб-страницы из сети интернет;
- загрузка исходного кода страницы Android приложения;
- получение дерева элементов на основе исходного кода страницы;
- генерация «Name», «Id», «Class Name», «Tag Name», «Link Text», «CSS» и «XPath» локаторов элементов;
- определение количества элементов страницы, которое определяет заданный локатор;
- генерирование Page Object класса, который содержит в себе код объявления интерфейсных объектов, а также кода методов взаимодействия с элементами страницы;
- сохранение сгенерированного кода Page Object класса на языке программирования Java или C# в файл.

Вследствие реализации программного средства на языке программирования Java, данное программное средство можно использовать на всех популярных операционных системах.

Результаты тестирования позволяют сделать вывод о надлежащем уровне качества данного программного средства. Было разработано 44 тест-кейса, процентный уровень прохождения всех тест-кейсов составил 100 %.

Рекомендации по практическому использованию результатов

Предложенная архитектура программного средства предполагает дальнейшую реализацию модулей чтения страниц приложений для других платформ (например iOS приложения).

Предложенная архитектура генератора кода предоставляет возможность добавления дополнительных языков программирования, на для которого будет сгенерирован код вследствие использования специализированной библиотеки, которая выполняет функцию шаблонизатора кода классов.

Предложенные алгоритмы могут применяться для создания схожих приложений и фреймворков создания автоматизированных тестов, генерации интерфейсных объектов для фреймворков отличных от Selenium и Appium, генерации кода, подходящего для любых паттернов проектирования автоматизированных тестов.

Разработанное программное средство позволяет снизить количество ручного труда при разработке автоматизированных тестов, а также снизить фактор допущения ошибки разработчиком автоматизированных тестов веб и мобильных приложений.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1. Хильчук А.С. Автоматизация тестирования образовательных ИТ-ресурсов / А.С. Хильчук, С.С. Куликов // «Современные технологии в науке и образовании – СТНО-2017»: материалы II междунар. науч.-техн. и науч.-метод. конф. (РГРТУ, Рязань, 2-3 мар. 2017 г.). – Рязань, 2017. – С. 185-187.

2. Хильчук А.С. Автоматизация тестирования веб-приложений / А.С. Хильчук, С.С. Куликов (науч. рук.) // Компьютерные системы и сети: материалы 53-й научной конференции аспирантов, магистрантов и студентов (Минск, 2-6 мая 2017 г.). – Минск: БГУИР, 2017. – С. 125-126.

3. Хильчук А.С. Автоматизация тестирования безопасности образовательных ресурсов / А.С. Хильчук, С.С. Куликов (науч. рук.) // «Дистанционное обучение – образовательная среда XXI века»: материалы X Международной научно-методической конференции (Минск, 7–8 декабря 2017 г.). – Минск: БГУИР, 2017. – С. 72.

4. Хильчук А. С. Автоматизация тестирования мобильных приложений / А. С. Хильчук // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов (Минск, 23 – 27 апреля 2018 г.) / Белорусский государственный университет информатики и радиоэлектроники. – Минск: БГУИР 2018. – С. 115 - 116.