

УДК 621.311

SIKULI SCRIPT –ИНСТРУМЕНТ ДЛЯ АВТОМАТИЗАЦИИ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ

Подсеваткина А.А.

Научный руководитель – Попова Ю.Б.

Выбор инструмента для автоматизации тестирования зачастую зависит от объекта тестирования и требований к тестовым сценариям, поскольку один инструмент не может поддерживать абсолютно все технологии, используемые при разработке приложений. На практике приходится сталкиваться с тестированием Web, мобильных, десктопных приложений, и не все инструменты могут реализовать необходимые тесты.

Один из инструментов автоматизации – это SikuliScript. Данный инструмент позволяет на основании распознавания изображений найти некоторый объект на экране и выполнить с ним определенные действия. Иначе говоря, Sikuli — это API, позволяющая писать на Python сценарии автоматизации, опираясь на визуальную составляющую любой программы/сайта.

Прежде всего Sikuli— это бесплатный OpenSource инструмент, который позволяет без труда проводить автоматизацию тестирования. Рассматриваемый инструмент достаточно *прост*: существует собственная SikuliIDE, в правой области располагается тело скрипта, а слева— основные операторы SikuliScript, которые по клику могут переноситься в тело скрипта (рис. 1). Также существует собственный инструмент для захвата скриншотов.

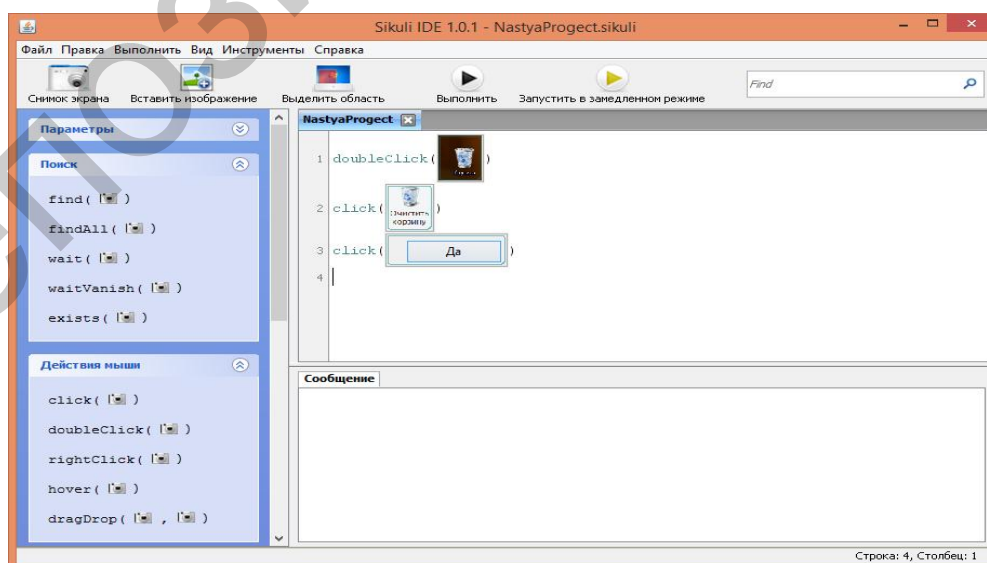


Рисунок 1. Собственная SikuliIDE

Можно распознавать изображения с достаточно тонкой настройкой параметров: процент схожести, размер, количество пикселей. Хранить изображения удобнее всего в ресурсах, разложив их по папкам, соответствующим названиям PageObject-ов.

В SikuliIDE можно писать небольшие сценарии: существует набор встроенных команд. Изображения выступают в качестве локаторов.

Область применения рассматриваемого инструмента является возможность решения прикладных задач. Всё что можно сделать при помощи клавиатуры и мыши, можно сделать при помощи SikuliScript, однако чаще всего применяется в автоматизации Flash, HTML-5 Canvas, Active-X приложений. Работа с ожиданиями и поиском элементов, имитация нажатия различных клавиш, проверка наличия и корректности изображений в приложении—всё это о SikuliScript.

Для того чтобы установить данный интерфейс необходимо следующее:

- инструмент для снятия скриншотов (SnagIt);
- JDK (предполагается работа с Java);
- IDE для работы с Java;
- библиотекasikuli-script.jar (необходимо скачать совместно с инструментомSikuliX).

Базовыми сущностями в SikuliScript являются (рис. 2):

- App— используется для запуска приложения (desktop);
- Screen— захватывает весь экран, производит поиск эталонов по всему экрану;
- Region— заданный регион на экране(для одинаковых объектов);
- Pattern— шаблон для распознавания, нахождения изображения на экране.

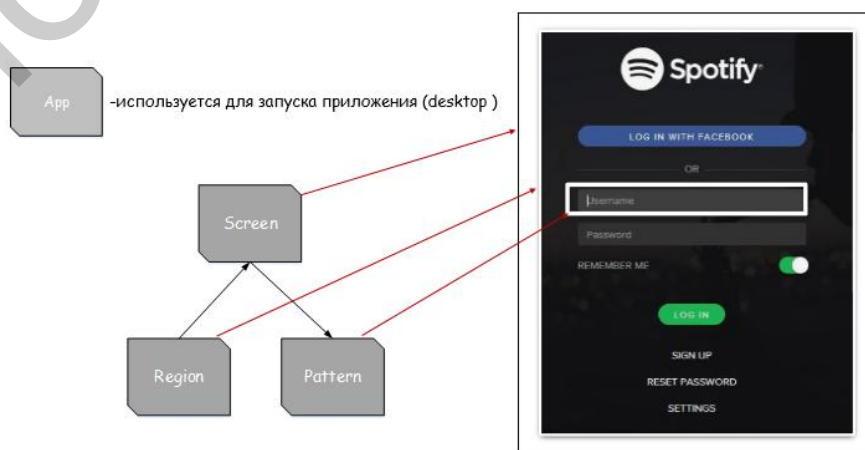


Рисунок 2. Базовые сущности в SikuliScript

На базе SikuliScript можно строить привычный PageObject, но с ссылками на картинки вместо привычных для WebDriver локаторов. Также существует SikuliFirefoxDriver – расширение WebDriver, использующее возможности работы с изображениями, доступными в Sikuli (например, может проверить наличие изображения на экране).

В качестве других примеров можно привести реализации следующих задач:

1. Проверка новых сообщений на Facebook от определенного пользователя и автоматическая отметка их кнопкой Like.
2. Система фильтрации входящих звонков в Skype.
3. Видеонаблюдение с помощью веб-камеры, которая отслеживает появление и исчезновение разных объектов.

На примере приложения Spotify можно написать тесты, реализующие успешную/не успешную авторизацию пользователя (рис. 3).

```

15 public class LoginTest extends BaseTestCase {
16
17     @Test
18     public void testInvalidLogin() throws Exception {
19         LoginScreen login = new LoginScreen();
20         login.enterLoginData(
21             DataProperties.get("invalid.login"),
22             DataProperties.get("invalid.password"))
23         .clickLogIn();
24         assertTrue(login.isErrorExist());
25     }
26
27     @Test
28     public void testValidLogin() throws Exception {
29         LoginScreen login = new LoginScreen();
30         MainScreen main = login
31             .enterLoginData(
32                 DataProperties.get("valid.login"),
33                 DataProperties.get("valid.password"))
34             .clickLogIn("and wait for vanish");
35         assertFalse(login.isLoginWindowExist());
36     }
37 }

```

Рисунок 3. Пример реализации тестов на языке Java

Таким образом, используя инструмент SikuliScript можно автоматизировать все видимое на экране без использования каких-либо специализированных API.

Литература

1. Bosch J. Continuous Software Engineering. - New York Dordrecht London, 2014. -133-135 p.
2. Сомаца: SikuliScript как инструмент автоматизированного тестирования [Электронный ресурс] Режим доступа: <https://comaqa.by/tag/sikuli-script/> - дата доступа: 11.05.2016