

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ПРИМЕНЕНИЕ ИНСТРУМЕНТОВ С ИСПОЛЬЗОВАНИЕМ
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ТЕСТИРОВАНИИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Стрельниковой Анны Андреевны

Научный руководитель

к. ф.-м. н., доцент

А. С. Иванов

Заведующий кафедрой

к. ф.-м. н., доцент

А. С. Иванов

Саратов 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Применение искусственного интеллекта в тестировании	4
1.1 Тестирование ПО	4
1.1.1 Искусственный интеллект и тестирование	4
1.2 Средства разработки сайта	4
1.2.1 Вёрстка сайта	4
1.2.2 HTML	5
1.2.3 CSS	5
1.2.4 Bootstrap 4	6
1.3 Инструментарий тестирования ПО Applitools	7
1.3.1 Как работает Applitools?	7
1.3.2 Уровни соответствия	8
1.3.3 Области просмотра	8
1.3.4 Работа с кросс-тестами	9
1.3.5 Сравнение изображений	9
1.3.6 Предоставление отчетов о тестировании	10
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Выпускная квалификационная работа посвящена визуальному регрессионному тестированию программного обеспечения при помощи инструментария Applitools Eyes. Актуальность работы, в первую очередь, обусловлена тем, что Applitools Eyes выявляет баги в пользовательском интерфейсе, используя машинное обучение и искусственный интеллект. Так же, возможно гибко настраивать тесты под разные задачи и форматы изображений. Данный инструмент соответствует множеству требований и подходит для реализации множества разных тестов. Он очень удобен в использовании и достаточно прост. В ходе работы были рассмотрены такие аспекты как верстка сайта, написание программы по тестированию и предоставление отчетов о выполненных тестах.

Целью выпускной работы является исследование возможностей тестирования программного обеспечения с использованием искусственного интеллекта.

Задачи, реализованные в ВКР:

- исследование возможностей инструмента Applitools Eyes;
- создание сайта на языках HTML и CSS с использованием фреймворка BOOTSTRAP 4, как объекта тестирования;
- написание программы на языке Java, проводящей визуальное регрессионное тестирование;
- тестирование сайта;
- предоставление отчётов о тестировании.

Бакалаврская работа включает в себя введение, две главы, заключение, список из 22 источников и пять приложений. Глава «Применение искусственного интеллекта в тестировании» описывает теоретическую часть работы. Рассматривается тестирование программного обеспечения, применение искусственного интеллекта в тестировании. Так же были рассмотрены средства разработки сайта и описаны технологии разработки. Далее был описан инструментарий Applitools Eyes и принцип его работы. В главе «Реализация задания» была описана практическая часть диплома. Объясняются принципы разработки сайтов, наглядно иллюстрируется получившийся результат разработки. Далее описывается работа с инструментом Applitools. Рассматривается разработка программы для тестирования программного обеспечения, а так же проиллюстрированы полные отчеты о проведенных тестах.

1 Применение искусственного интеллекта в тестировании

1.1 Тестирование ПО

Тестирование программного обеспечения – процесс анализа соответствия ожидаемого результата и реального поведения программы с целью выявления дефектов (багов) и обеспечения качества тестируемого продукта. [1]

Тестирование программного обеспечения включает в себя:

- полное исследование программного обеспечения (это делается для того, чтобы иметь полную информацию о качестве продукта);
- проверка того, насколько соответствует конечный продукт заявленным изначально требованиям, какие доработки в связи с этим требуются;
- анализ проведенных тестов и составление отчётов о тестировании. [2]

1.1.1 Искусственный интеллект и тестирование

Что касается искусственного интеллекта в тестировании программного обеспечения, представим, что мы смогли обучить систему на каком-то приложении с огромным количеством ошибок и дадим на сравнение правильную работу того же самого приложения. В таком случае, мы сможем обучить машину предугадывать те места, в которых заведомо может возникнуть ошибка и предотвратить ее появление. А если будет возможность обучаться на сотнях таких приложений? Это может означать, что искусственный интеллект сможет абсолютно точно проводить тесты и проводить поиск дефектов, ведь как бы не считали разработчики, что «их приложение абсолютно уникально и не имеет аналогов», имеет. Какие-то части, блоки однозначно будут похожи на другие. [3]

1.2 Средства разработки сайта

1.2.1 Вёрстка сайта

Сайт будет создаваться на языках HTML, с использованием Bootstrap 4, и CSS . Для начала работы с Bootstrap необходимо установить этот фреймворк на компьютер [4]. Сайт будет состоять из главной страницы и страницы с формой авторизации. Но так как наша главная задача – это визуальное регрессионное тестирование, то будет написано два аналогичных друг другу сайта, с небольшими различиями, чтобы наглядно показать работу визуальных тестов. Так же это сделано для упрощения демонстрации работы. [5]

В первой версии сайта будут в определенной последовательности загружены PDF-файлы и определенным образом будет выглядеть форма поиска по сайту. Во второй версии изменен порядок расположения PDF-файлов и вид формы поиска. Так же, во внутренней странице сайта, на странице авторизации, тоже будут характерные черты, которые помогут явно определять различия между двумя страницами.

Во второй версии сайта, будут изменены порядок следования PDF-файлов, форма поиска и форма авторизации. Все изменения будут наглядными и «заметны глазу», так как это демонстрационный вариант работы тестов. [6]

1.2.2 HTML

Язык HTML представляет собой стандартизированный язык разметки, интерпретируемый браузерами. То есть, на вход подается код, написанный на этом языке, и в результате интерпретации в окне браузера компьютера/мобильного устройства отображается написанная нами страница. По сути, HTML только структурирует сайт, «расставляет по местам» компоненты, форматирует их. Грубо говоря, как каркас здания. Мы просто видим, где что расположено и в каком виде. Но без отделки. Отделкой «постройки» занимается CSS и Bootstrap, но это будет рассмотрено позднее. [7]

Внешний вид каждой Web-страницы отличается от других. Но структура, в целом, у всех схожая. Шапка или заголовок, где отображено как правило либо общая информация о сайте, либо логотип. Далее идёт nav-bar или меню навигации. Основная часть страницы обычно содержит главную информацию о сайте/продукте. Затем идет side-bar или боковая панель навигации. И нижняя часть страницы (футер), в котором содержится часто контактная информация, информация об авторских правах и так далее. [8]

1.2.3 CSS

В то время, когда HTML занимается структурированием нашей веб-страницы, язык CSS занимается ее стилизацией. Все, что необходимо сделать со страницей, начиная от межстрочного интервала, заканчивая добавлением декоративных элементов, всем этим занимается CSS. В чем преимущество отдельного прописания стиля сайта? Плюсов очень много, рассмотрим некоторые из них:

- освобождение HTML-кода от оформления. То есть, можно сказать, разграничение структуры сайта и его стилизации;
- при помощи CSS возможно прописать стили для каждого устройства отдельно, например, задана определенная ширина экрана, и для этой ширины используется другой стиль оформления, отличающийся от оформления, который был прописан для полноэкранного режима;
- так же, в отличие от HTML, CSS имеет намного больше возможностей по оформлению Web-страницы и при этом, он очень прост;
- благодаря тому, что стилизация страницы находится в отдельном документе, уменьшается код веб-страницы, что значительно ускоряет ее загрузку в браузере;
- так как множество сайтов имеют примерно схожую структуру, то слегка подредактировав стили, один можно использовать для нескольких версток. [9]

1.2.4 Bootstrap 4

Bootstrap — это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений. При изучении данного инструмента стоит начать с изучения системы сеток Bootstrap. Это основополагающая часть при вёрстке сайта. Весь сайт делится на блоки или контейнеры. Каждый элемент сайта занимает свой контейнер. Сетка так же адаптирована к изменениям размера экрана, но размер нужно прописывать в коде. [10]

Основные компоненты Bootstrap:

- сетки;
- шаблоны;
- типографика;
- медиа;
- таблицы;
- формы;
- навигация;
- алерты.

1.3 Инструментарий тестирования ПО Applitools

1.3.1 Как работает Applitools?

Applitools Eyes способен обеспечить визуальное регрессионное тестирование пользовательского интерфейса с помощью машинного обучения для Web-приложений и мобильных приложений. Принцип работы этого инструментария следующий: Applitools проводит тестирование на основе искусственного интеллекта, то есть сканирует экран как будто наши глаза и обрабатывает информацию как мозг, но с мощностью и точностью машины. Это позволяет Applitools сохранять точность теста до 99%, что является очень хорошим показателем в визуальном тестировании. [11]

Рассмотрим работу Applitools на практике. В ходе работы программы происходит следующее (далее будет описан самый первый запуск программы, в котором создается первое контрольное изображение, то есть baseline):

- открывается страница тестируемого сайта;
- производится скриншот страницы;
- на данном скриншоте страница происходит создание последовательности контрольных точек;
- скриншот, с выделенными контрольными точками, сохраняется и демонстрируется в Web-приложении Applitools Eyes.

Первый запуск создает «эталонное» изображение. В дальнейшем, при следующих запусках, в случаях, если будут какие-то отличия, и Applitools Eyes это продемонстрирует в своём отчете, то можно будет следующее изображение отметить как «успешно пройденный тест». Это очень удобная функция, так как визуальные изменения бывают не только в случаях каких-либо багов, но и просто при изменении пользовательского интерфейса. [12]

В ходе повторного запуска происходит следующее:

- первые четыре пункта повторяют первый запуск;
- далее, скриншот последнего протестируемого приложения сравнивается с эталонным по контрольным точкам;
- результат теста, если требуется, можно изменить, или оставить с пометкой «непройденный», если обнаружен баг.

При сравнении результатов двух пройденных тестов, может увидеть, в чем заключается разница. Во первых, была изменена форма поиска по сайту, кнопка «поиск», вид поля для ввода текста, во-вторых, я поменяла местами

PDF-файлы, что так же отображено в отчёте о прохождении теста. [13] [14]

1.3.2 Уровни соответствия

Рассмотрим уровни соответствия Applitoools. Уровень соответствия определяет способ, которым Eyes сравнивает изображение контрольной точки с базовым изображением. Уровни соответствия бывают следующих видов:

- STRICT. Это рекомендуемый уровень, когда нужно убедиться, что содержимое страницы соответствует базовому изображению;
- LAYOUT. Этот уровень полезен для проверки соответствия макета страницы;
- CONTENT. Этот уровень соответствия похож на уровень строгого соответствия (первый уровень), но игнорирует различия в цветах и стиле;
- EXACT. Этот уровень совпадения заставляет механизм делать попиксельное сравнение контрольной точки и базовых изображений. [15] [16]

Для того, чтобы поменять уровни соответствия, имеется несколько вариантов:

1. установка уровня соответствия для нескольких тестов;
2. установка уровня соответствия для ряда контрольных точек;
3. установка уровня соответствия для конкретной контрольной точки;
4. установка уровня соответствия в области checkpoint.

1.3.3 Области просмотра

Окно просмотра - это видимая область приложения в окне приложения. Размер области просмотра может повлиять на макет страницы. Например, изменение ширины браузера может изменить способ отображения абзаца текста. На платформах, которые позволяют контролировать размер окна, таких как браузеры компьютера, если размер области просмотра определен явно в тесте, то когда тест вызывает метод `open()`, Eyes вычисляет и корректирует размер операционного окна так, чтобы размер области просмотра соответствовал заданным размерам. На платформах, где пользователь не может контролировать размер области просмотра, например, на мобильных телефонах, не следует явно устанавливать размер области просмотра. [17]

При первом запуске теста Eyes сохраняет размер области просмотра вместе с операционной системой и платформой приложения в качестве среды выполнения базовой линии. Установка размера области просмотра не является

обязательной. Если вы не установите размер области просмотра, то Eyes будет работать с любым размером области просмотра, установленным приложением или базовой инфраструктурой. [18]

1.3.4 Работа с кросс-тестами

Функция межсредового тестирования Eyes позволяет тестировать приложение на нескольких платформах, используя единую общую основу. Чтобы понять, как проводится межсредовое тестирование в Eyes, мы должны сначала определить два понятия: базовая подпись и название среды.

Базовая подпись. Базовая линия визуального тестирования состоит из набора изображений, которые используются в качестве основы для сравнения изображений, снятых во время теста. Конкретный базовый уровень отличается от других базовых уровней двумя типами информации: его имя и его среда. Когда определенный тест выполняется на нескольких платформах, все платформы будут иметь одинаковое базовое имя, но по умолчанию среда будет индивидуальной для каждой платформы. В результате для каждой тестируемой среды существует отдельный базовый уровень. Название среды. Eyes предоставляет способ присвоения имени определенной среде. [19]

Визуальное тестирование с использованием межсредового сценария состоит из следующих этапов:

1. планирование;
2. настройка теста в SDK и менеджере тестов;
3. рассмотрение и устранение несоответствий.

1.3.5 Сравнение изображений

Вместо статической связи между каждой контрольной точкой и соответствующим базовым изображением, Eyes сравнивает последовательность изображений, снятых контрольными точками, с последовательностью базовых изображений, учитывая тот факт, что контрольные точки могут быть добавлены или удалены. Затем, Eyes находит изображения, которые были добавлены или удалены, а также изображения, которые имеют соответствующее базовое изображение и проверяет, совпадают они или не совпадают. Далее, Eyes позволяет проверить существующие контрольные точки и убедиться, что набор базовых изображений обновлен. [20] Каждый шаг представляет собой один из четырех возможных результатов:

1. Match;
2. Diff;
3. Missing;
4. New.

Хотя можно свободно добавлять и удалять контрольные точки, Eyes предполагает, что порядок снятых изображений, которые не являются новыми, не меняется. Если изменить порядок двух контрольных точек, то Eyes может сообщить о двух пропущенных шагах и двух новых шагах или о двух шагах с несоответствиями. [21]

1.3.6 Предоставление отчетов о тестировании

Помимо всех преимуществ в использовании Applitools Eyes, так же одним из огромных плюсов у данного инструмента является то, в каком виде могут быть представлены отчёты о проведенном тестировании. В отчете, в явном виде, представлена следующая информация: [22]

- статус пройденного теста;
- вид проведенного теста (Smoke Test, Critical Path Test, Extended Test);
- используемый браузер и его версия;
- операционная система;
- время, когда было проведено тестирование и многое другое;

В ходе работы проводились тесты, выявляющие дефекты (в данном случае искусственно созданные), которые успешно были пройдены, используемый инструмент для тестирования программного обеспечения Applitools, безошибочно нашел все внесенные в созданный сайт изменения, наглядно предоставил отчёты по проведенным тестам и доказал, что использование данного инструмента в тестировании ПО значительно упрощает и ускоряет работу тестировщика. Минусы данного инструмента заключаются в том, что нужно постоянно отслеживать версии всех необходимых дополнений, например, чтобы не было расхождений в версиях между браузером Google Chrome и драйвером ChromeDriver, а так как данный браузер обновляется автоматически, то под новую версию приходится скачивать и устанавливать новый драйвер, что крайне неудобно и отнимает время.

ЗАКЛЮЧЕНИЕ

Тестирование – очень важная часть в создании ПО. Именно тестирование обеспечивает и отвечает за качество представленного заказчику продукта. В ходе работы была создана программа на языке Java, которая проводит визуальное регрессионное тестирование написанного нами сайта на языках HTML и CSS. Так же были рассмотрены особенности работы с инструментом Applitools. Выявлены достоинства и недостатки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Как и зачем тестировать верстку[Электронный ресурс]. — URL: <https://cutt.ly/KyGa3i0> (Дата обращения 20.05.2020). Загл. с экр. Яз. рус.
- 2 Applitools[Электронный ресурс]. — URL: <https://applitools.com/> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 3 *Ясницкий, Л. Н.* Введение в искусственный интеллект / Л. Н. Ясницкий. — Издат. центр «Академия», 2005.
- 4 Bootstrap-4[Электронный ресурс]. — URL: <https://bootstrap-4.ru/> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 5 HTML5CSS.ru[Электронный ресурс]. — URL: <https://html5css.ru/bootstrap4/default.php> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 6 ChromeDriver - WebDriver for Chrome [Электронный ресурс]. — URL: <https://chromedriver.chromium.org/> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 7 Справочник по HTML[Электронный ресурс]. — URL: <http://htmlbook.ru/html> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 8 Setting the JAVA-HOME Variable in Windows [Электронный ресурс]. — URL: <https://confluence.atlassian.com/doc/setting-the-java-home-variable-in-windows-8895.html> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 9 ИТ КОНСАЛТИНГ [Электронный ресурс]. — URL: <https://8d9.ru/program/applitools-eyes> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 10 Features[Электронный ресурс]. — URL: https://applitools.com/features/#visual-ai?utm_source=tutorials&utm_medium=tutorials (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 11 Visual-locators[Электронный ресурс]. — URL: <https://applitools.com/docs/features/visual-locators.html> (Дата обращения 11.05.2020). Загл. с экр. Яз. рус.

- 12 *Тей, А.* Логический подход к искусственному интеллекту / А. Тей, П. Грибомон, Ж. Луи. — Мир, 1990.
- 13 Applitools Tutorial [Электронный ресурс]. — URL: <https://applitools.com/tutorials/> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 14 How Eyes compares checkpoints and baselines images [Электронный ресурс]. — URL: <https://applitools.com/docs/topics/general-concepts/how-eyes-compares-checkpoints.html> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 15 *Куликов, С.* Тестирование программного обеспечения / С. Куликов. — Четыре четверти, 2017.
- 16 test automation university [Электронный ресурс]. — URL: <https://testautomationu.applitools.com/automated-visual-testing-a-fast-path-to-test-automation-success/> (Дата обращения 02.05.2020). Загл. с экр. Яз. англ.
- 17 *Калацкая, Л. В.* Организация и обучение искусственных нейронных сетей: Экспериментальное учеб. пособие / Л. В. Калацкая, В. А. Новиков, В. С. Садов. — Минск: Изд-во БГУ, 2003.
- 18 HTML-book[Электронный ресурс]. — URL: <http://htmlbook.ru/html> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 19 ПОДХОДЫ К ТЕСТИРОВАНИЮ ПО НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА[Электронный ресурс]. — URL: <https://testmatick.com/ru/podhody-k-testirovaniyu-po-na-osnove-iskusstvennogo-intellekta/> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 20 Автоматизированное тестирование и искусственный интеллект [Электронный ресурс]. — URL: <https://qaat.ru/avtomatizirovannoe-testirovanie-i-iskusstvennyj-intellekt/> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.
- 21 Когда использовать Applitools [Электронный ресурс]. — URL: <https://software-testing.ru/forum/index.php?/topic/34086-kogda-primeniat-applitools/> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.

- 22 AppliTools for Selenium IDE Tutorial [Электронный ресурс]. — URL: <https://applitools.com/tutorials/selenium-ide.html#selenium-ide-videos> (Дата обращения 02.05.2020). Загл. с экр. Яз. рус.