|  |  |  |
| --- | --- | --- |
|  | Изображение выглядит как текст, вычерчивание линий  Автоматически созданное описание |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ПРАКТИЧЕСКАЯ РАБОТА №4**

по дисциплине

«Тестирование и верификация программного обеспечения»

**Тема: «Функциональное тестирование ПО»**

**Студенты группы** ИКБО-03-20 Голиков А.Л., Каменцева О.И., Хмизюк С.И., Юрзина С.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студентов)

**Руководитель практической работы**  Трушин С.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Практические работы выполнены «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

Содержание

[1. Цель и описание работы 3](#_Toc117627678)

[2. Тестирование десктопной программы 5](#_Toc117627679)

[2.1. Функциональная спецификация и ТЗ 5](#_Toc117627680)

[2.2. План тестирования 5](#_Toc117627681)

[2.3. Тестирование продукта 6](#_Toc117627682)

[3. Тестирование Web-приложения 22](#_Toc117627683)

[3.1. Функциональная спецификация и ТЗ 22](#_Toc117627684)

[3.2. План тестирования 22](#_Toc117627685)

[3.3. Тестирование продукта 23](#_Toc117627686)

[4. Выводы 28](#_Toc117627687)

[5. Список использованной литературы 28](#_Toc117627688)

# Цель и описание работы

Функциональное тестирование – это один из видов тестирования ПО, который направлен на проверку реализуемости функциональных требований, которые были разработаны на этапе проектирования, а также определяют, какие задачи решаются данным программным обеспечением.

Другими словами, благодаря функциональному тестированию выявляется, соответствуют ли разработанные функциональные требования реальным характеристикам. Таким образом, основной задачей функционального тестирования является подтверждение того, что разрабатываемый программный продукт обладает всем функционалом, требуемым заказчиком.

В зависимости от цели, функциональное тестирование может проводиться:

На основании требований, указанных в функциональной спецификации программного продукта.

При этом создаются тестовые случаи (test cases), составление которых учитывает приоритетность функций ПО, которые необходимо покрыть тестами. Так можно убедиться в том, что все функции разрабатываемого продукта работают корректно при различных типах входных данных, их комбинаций и количества.

На основе процессов, выполнение которых должно быть обеспечено приложением.

При этом выделяются варианты использования (use cases) системы, составление которых позволяет проверить не работоспособность отдельных функций программного продукта, а корректность выполняемых операций с точки зрения сценариев использования системы.

Основным преимуществом функционального тестирования является то, что оно способно сымитировать фактическое использование системы с минимальными затратами.

Однако, в этом и состоит основной его недостаток – возможность упущения логических ошибок в программном обеспечении, поскольку непосредственная работа с системой не исключает существования в логике работы программного продукта значительных ошибок.

На основе изучения материала лекций по дисциплине «Тестирование и верификация программного обеспечения» требуется выполнить следующее.

1. Разделиться на команды из 2-4 человек.

2. Скачать и установить программное обеспечение Test Complete. (В случае возникновения трудности при работе с предложенным программным продуктом для тестирования, допускается выбрать аналогичное программное обеспечение сходное по функционалу и предоставленным возможностям).

Выбрать для тестирования десктопную программу для тестирования. Используя функционал предоставленного программного обеспечения протестировать выбранную программу.

Пользуясь знаниями и навыками, полученными из предыдущих практических занятий написать функциональную спецификацию на программу если ее нет. Записать несколько скриптов поведения программы согласно спецификации.

Убедиться, что программа корректно записывает скрипты.

Записать несколько скриптов по десктопной программе. В отчет свести проверку данной программой для тестирования, найденные ошибки, выполненные условия.

Убедиться, что разработанные скрипты обеспечивают покрытие проверки использованной выбранной десктопной программы.

1. Аналогично выбрать и проверить Web-приложение с использованием Selenium IDE и Selenium WebDriver или аналогов.

# Тестирование десктопной программы

* 1. Функциональная спецификация и ТЗ

Приложение "Калькулятор" для Windows 10 представляет собой версию калькулятора для настольных компьютеров со следующими режимами вычислений: "Обычный", "Программист" и "Вычисление даты". В данной работе для тестирования был выбран режим «Обычный»

В интерфейсе приложения представлены: 10 кнопок с цифрами от 0 до 9, кнопка запятой, которая позволяет делать десятичные числа, кнопка +/-, позволяющая менять знак введенного числа, 4 стандартные арифметические функции (сложение, вычитание, умножение и деление), кнопка «=», позволяющая найти значение введенного выражения, кнопки с различными функциями, такие как 1/х, возведение числа в квадрат, квадратный корень их числа, и процент от введенного числа. Также есть три кнопки очищения: С – очищение выражения полностью, СЕ – очищение последнего (текущего) числа и стрелка с крестиком – очищение последнего символа. Еще имеются кнопки для работы с памятью: МС – очищает ячейку памяти, то есть делает её пустой, MR - читает из памяти сохранённое значение и вставляет его в поле ввода (или же заменяет всю строку значением из памяти), M+ - прибавляет к значению в ячейке памяти результат текущего вычисления, M- - вычитает из значения в ячейке памяти результат текущего вычисления, MS - сохраняет результат вычисления в память.

Техническое задание «Калькулятора» - выполнение арифметических действий с числами, вводимыми пользователем в специальное поле ввода, с использованием кнопок, расположенных внизу этого поля (описанных выше).

* 1. План тестирования

Функциональное тестирование подразумевает под собой проверку функциональной спецификации, которая была описана пунктом выше. В ходе работы должны созданы тестовые случаи, которые проверяют корректность работы тестируемого продукта. Чтобы проверить полноту вычислений, требуется разработать тесты для сложения, вычитания, умножения, деления, взятия квадратичного корня, возведения числа в квадрат, кнопок +/-, кнопки «запятая», и кнопки «стереть».

* 1. Тестирование продукта

Для тестирования выбранного программного продукта было выбрано приложение «TestComplete». На рис. 1-4 приведено создание нового проекта.

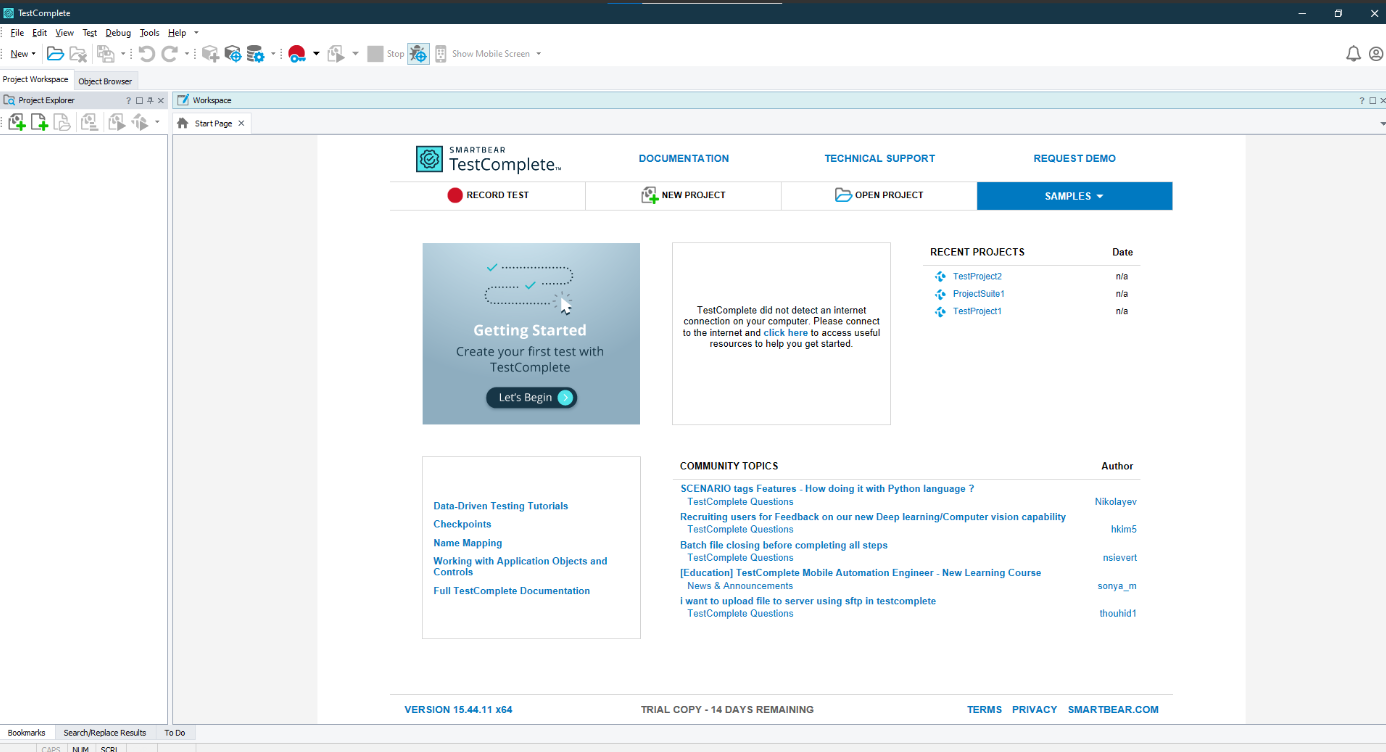


Рис. 1 – Стартовый интерфейс программы

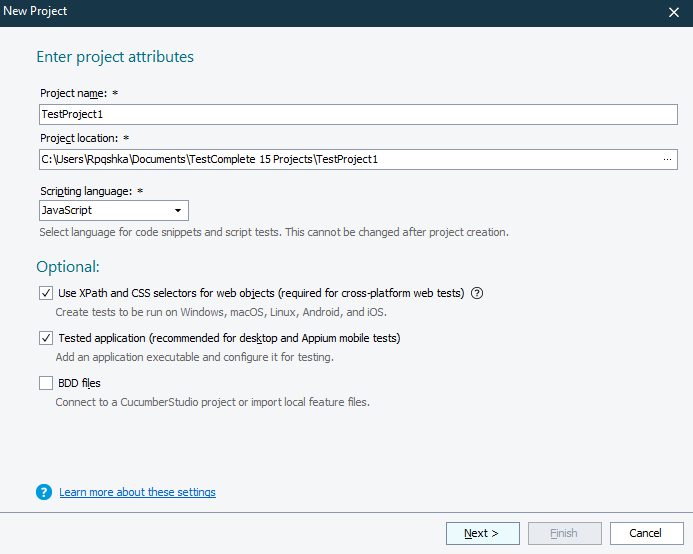


Рис.2 – Задание имени нового проекта

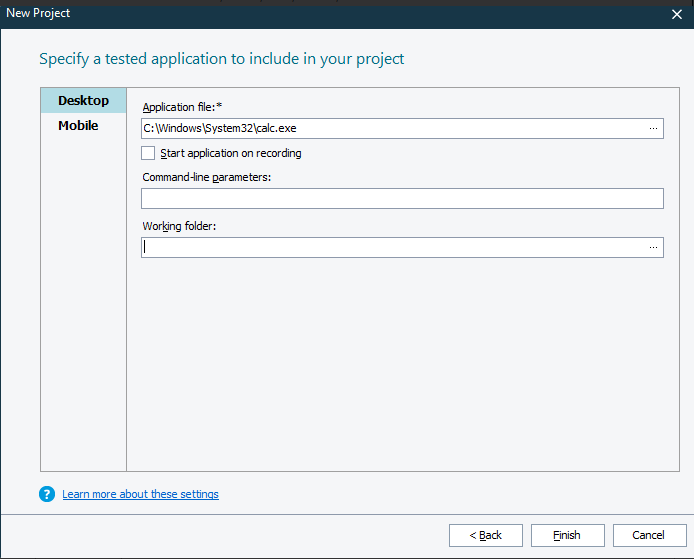


Рис. 3 – Выбор тестируемого приложения

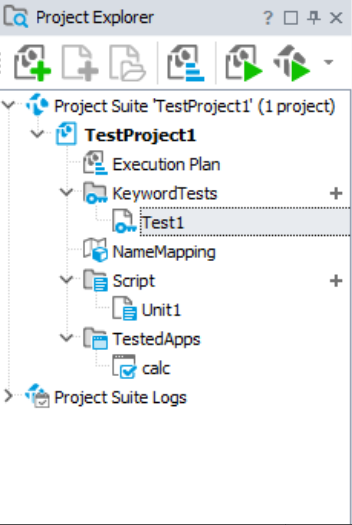


Рис. 4 – Боковое меню с новым проектом

Далее прописываем функцию для старта приложения и запускаем ее.

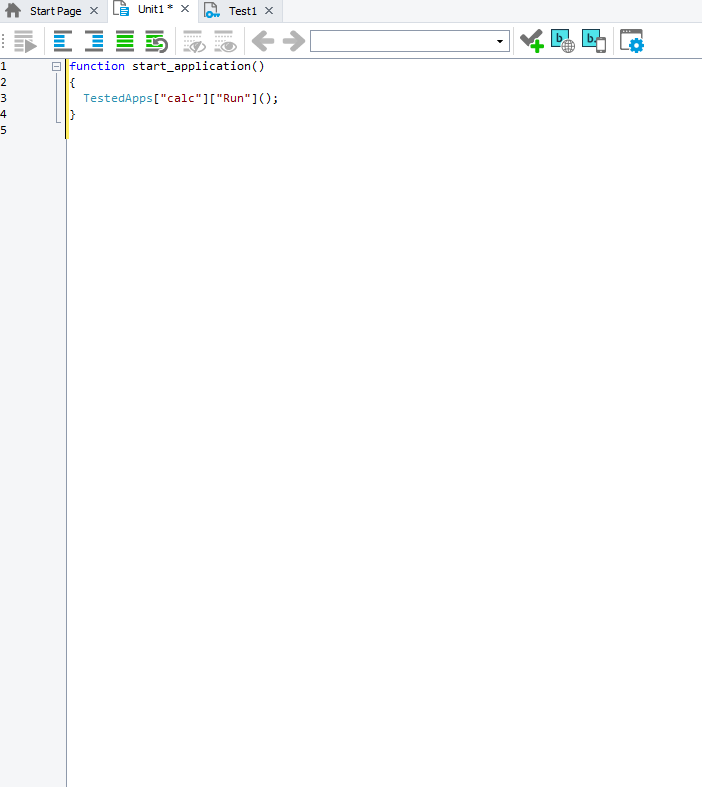


Рис. 5 – Функция старта приложения

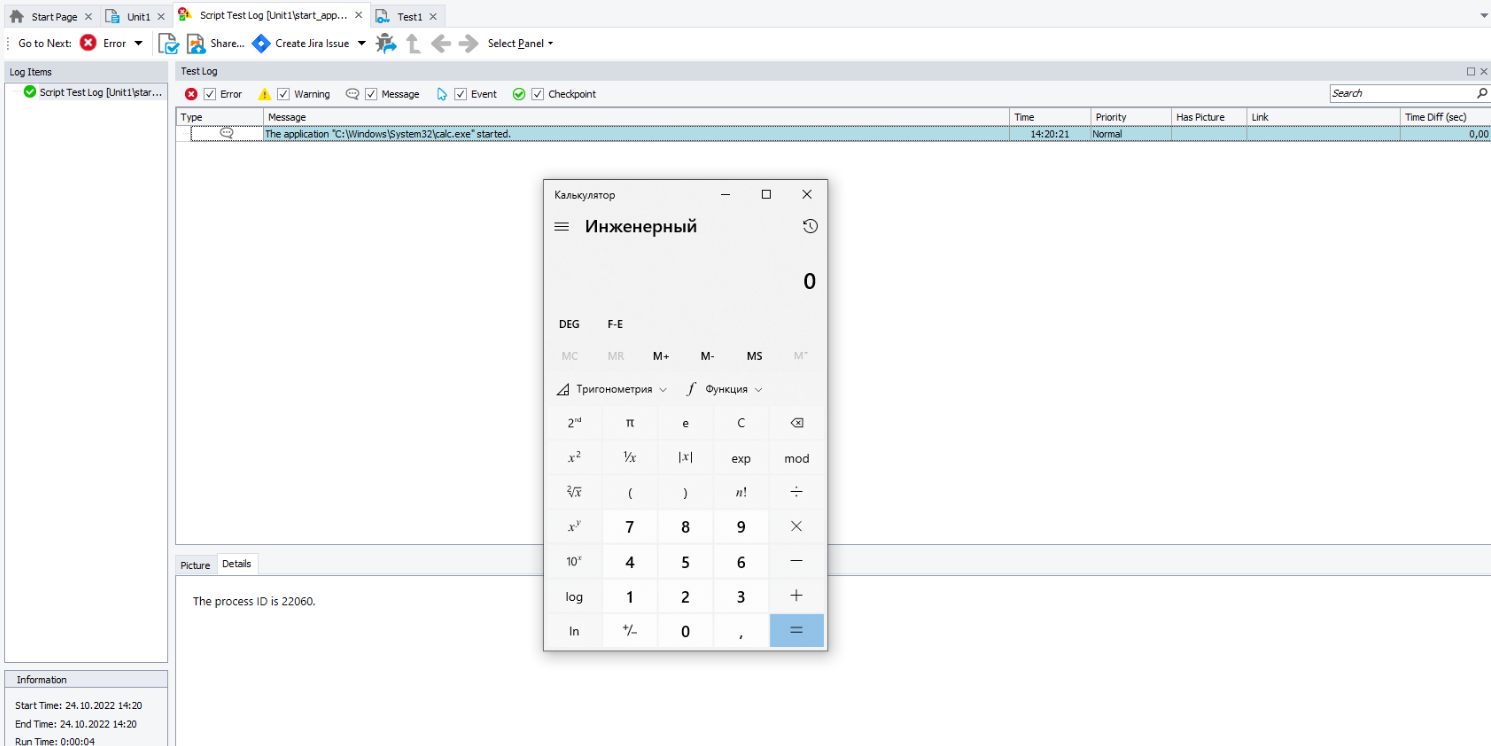


Рис. 6 – Запуск функции старта

Далее, переходим непосредственно к кликам по кнопкам.

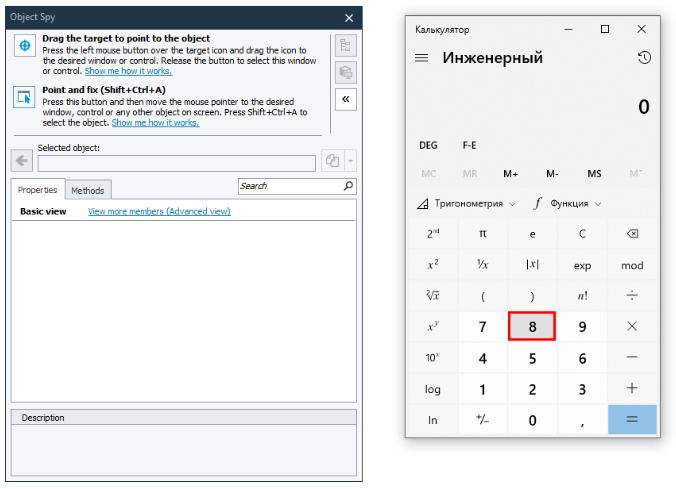


Рис. 7 – Выбор «Point and Fix» как способ захвата

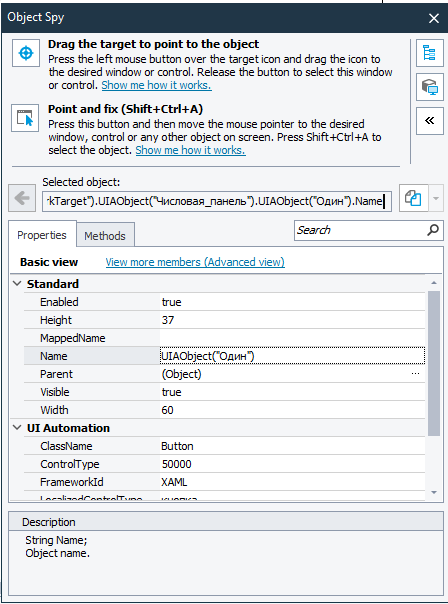


Рис. 8 – Properties кнопки «Один»

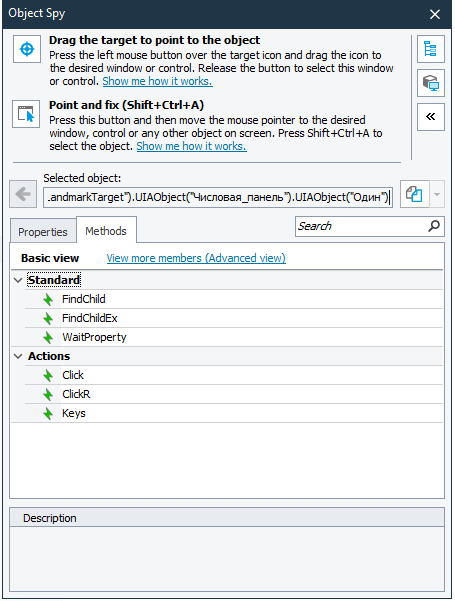


Рис. 9 - Methods кнопки “Один”

Далее приведен пример кода одного из всех разработанных методов тестирования.

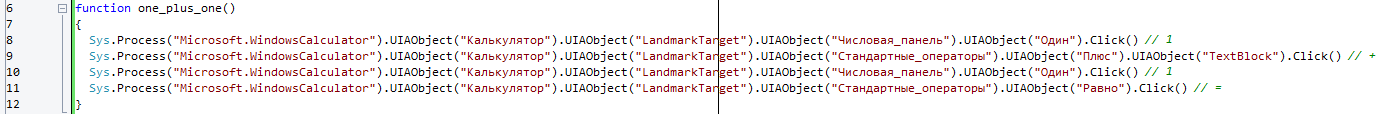


Рис. 10 - Метод one\_plus\_one

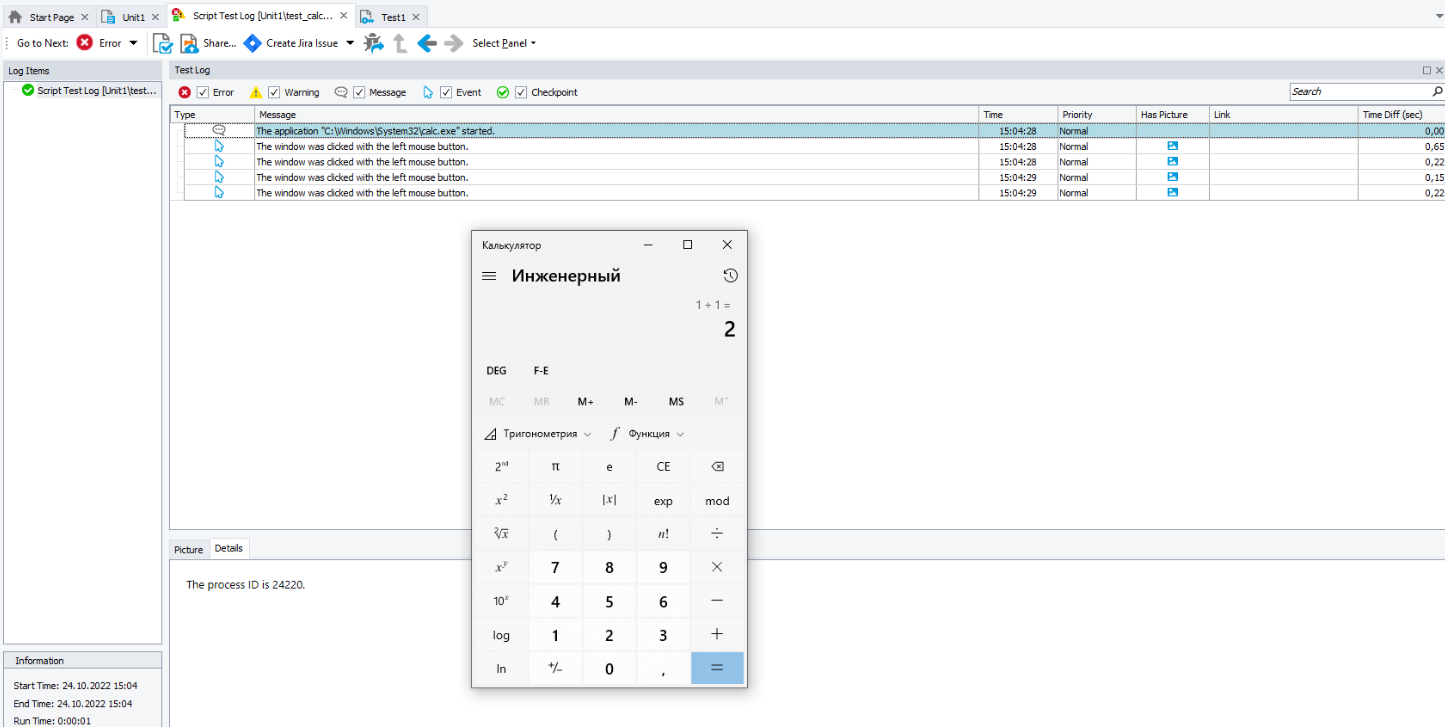


Рис. 11 - Тест метода one\_plus\_one

Теперь объединим функции запуска приложения, тестирования метода one\_plus\_one и закрытия приложения.

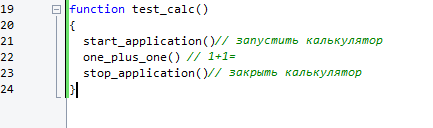


Рис. 12 – Общая функция тестирования

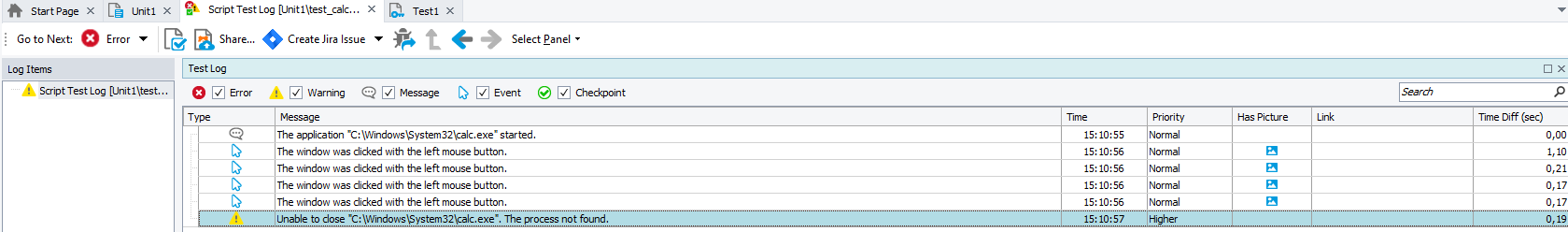


Рис. 13 – Тестирование общей функции

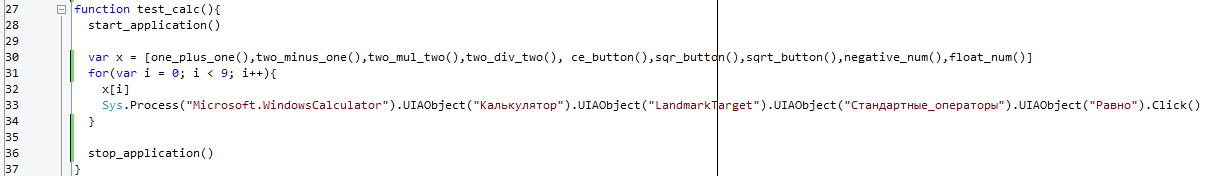


Рис. 14 – Добавление и автоматизация тестов

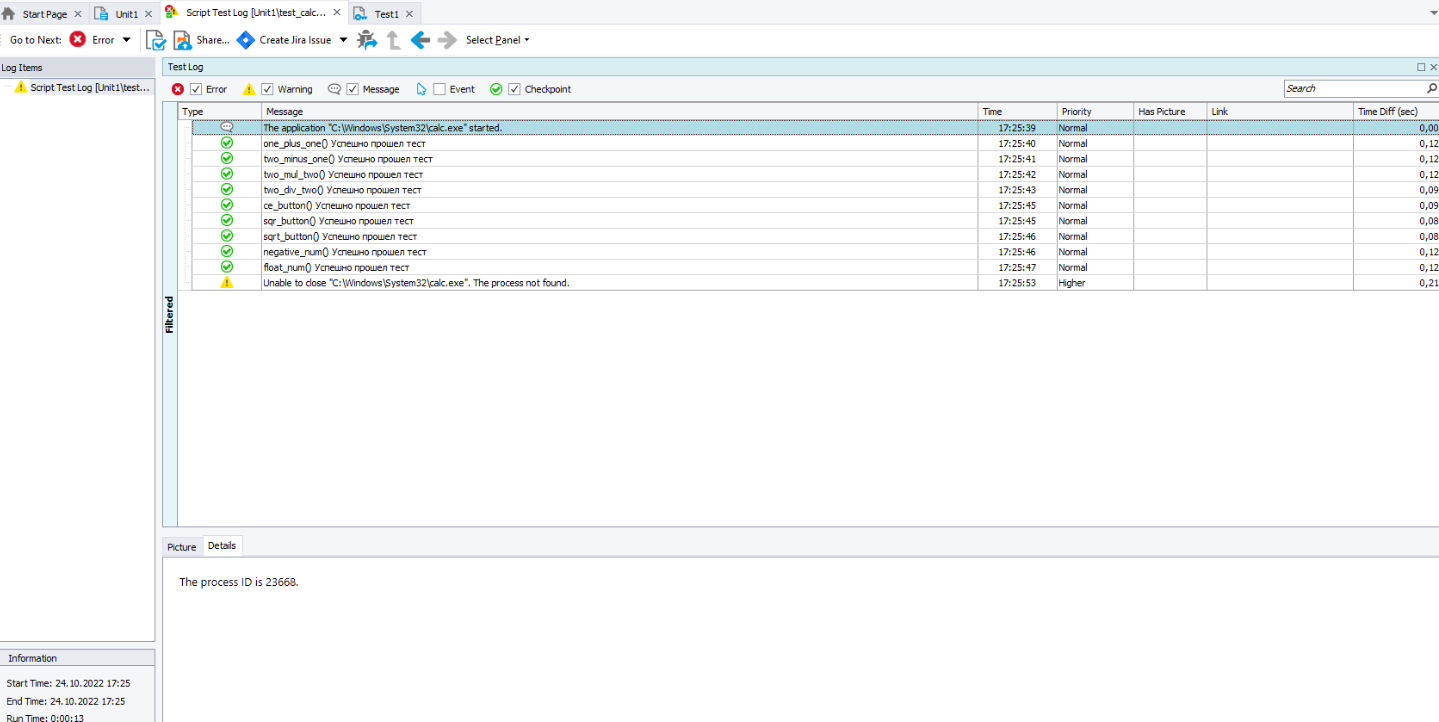


Рис. 15 – Успешное прохождение всех тестов

Все тесты были пройдены успешно, ошибки не найдены.

Листнг 1. Код всех тестов

function start\_application()

{

TestedApps["calc"]["Run"]();

}

function stop\_application()

{

TestedApps["calc"]["Close"]();

}

function one\_plus\_one()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Один").Click() // 1

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Плюс").UIAObject("TextBlock").Click() // +

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Один").Click() // 1

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click() // =

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_2").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "2"){

Log.Checkpoint('one\_plus\_one() Успешно прошел тест')

}

else{

Log.Error('one\_plus\_one() Провалил тест')

}

}

function test\_calc(){

start\_application()

var x = [one\_plus\_one(),two\_minus\_one(),two\_mul\_two(),two\_div\_two(), ce\_button(),sqr\_button(),sqrt\_button(),negative\_num(),float\_num()]

for(var i = 0; i < 9; i++){

x[i]

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

}

stop\_application()

}

function two\_minus\_one(){

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Минус").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Один").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_1").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "1"){

Log.Checkpoint('two\_minus\_one() Успешно прошел тест')

}

else{

Log.Error('two\_minus\_one() Провалил тест')

}

}

function two\_mul\_two()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Умножить\_на").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_4").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "4"){

Log.Checkpoint('two\_mul\_two() Успешно прошел тест')

}

else{

Log.Error('two\_mul\_two() Провалил тест')

}

}

function two\_div\_two()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Разделить\_на").UIAObject("TextBlock").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_4").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "1"){

Log.Checkpoint('two\_div\_two() Успешно прошел тест')

}

else{

Log.Error('two\_div\_two() Провалил тест')

}

}

function ce\_button()

{

var x = ["Один", "Два", "Три", "Четыре", "Пять", "Шесть", "Семь", "Восемь", "Девять", "Нуль"]

for(var i = 0; i< 10; i++){

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject(x[i]).Click()

}

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Элементы\_управления\_отображением").UIAObject("Очистить\_запись").UIAObject("CE").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_4").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "0"){

Log.Checkpoint('ce\_button() Успешно прошел тест')

}

else{

Log.Error('ce\_button() Провалил тест')

}

}

function sqr\_button()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Пять").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_функции").UIAObject("Квадрат").UIAObject("TextBlock").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_4").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "25"){

Log.Checkpoint('sqr\_button() Успешно прошел тест')

}

else{

Log.Error('sqr\_button() Провалил тест')

}

}

function sqrt\_button()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Пять").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_функции").UIAObject("Квадратный\_корень").UIAObject("TextBlock").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_4").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "5"){

Log.Checkpoint('sqrt\_button() Успешно прошел тест')

}

else{

Log.Error('sqrt\_button() Провалил тест')

}

}

function negative\_num()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Два").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Положительное\_отрицательное").UIAObject("TextBlock").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_2").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "-2"){

Log.Checkpoint('negative\_num() Успешно прошел тест')

}

else{

Log.Error('negative\_num() Провалил тест')

}

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Стандартные\_операторы").UIAObject("Равно").Click()

}

function float\_num()

{

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Девять").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Десятичный\_разделитель").Click()

Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Числовая\_панель").UIAObject("Семь").Click()

if(Sys.Process("Microsoft.WindowsCalculator").UIAObject("Калькулятор").UIAObject("LandmarkTarget").UIAObject("Отображать\_как\_9\_7").UIAObject("TextContainer").UIAObject("NormalOutput").Text == "9,7"){

Log.Checkpoint('float\_num() Успешно прошел тест')

}

else{

Log.Error('float\_num() Провалил тест')

}

}

# Тестирование Web-приложения

* 1. Функциональная спецификация и ТЗ

Web-приложение "Cambridge Dictionary" представляет собой словарь английских слов. Сайт позволяет узнать точное значение любого английского слова, приведя его толкование простыми словами, а также перевод на русский отдельных слов и целых текстов, озвучить введенное слово, узнать грамматику по поиску определенной темы, тезаурус (находит похожие слова по смыслу, синонимы и антонимы к введенному слову).

В интерфейсе приложения представлены:

в хедере сайта: выбор страницы со словарем, где можно вводить отдельные слова или фразы, страница с переводчиком, страница с грамматикой, и страница с тезаурусом. Также есть логотип, на который можно нажать, при нажатии он перенесет пользователя на главную страницу со словарем. Есть вход в личный кабинет и регистрация;

в основной части сайта на страницах расположены строки поиска, на каждой странице она соответственно выполняет свою функцию;

в футере сайта в основном расположены страницы с помощью, лицензиями и правилами сайта.

Техническое задание "Cambridge Dictionary" – перевод слов и текстов, выяснения их смысла и объяснение грамматических правил.

* 1. План тестирования

Функциональное тестирование подразумевает под собой проверку функциональной спецификации, которая была описана пунктом выше. В ходе работы должны созданы тестовые случаи, которые проверяют корректность работы тестируемого продукта. Чтобы проверить работу выбранного сайта, нужно разработать тесты, которые проверяют рабочую способность web-приложения.

* 1. Тестирование продукта

Тестирование продукта производилось с помощью программы Selenium IDE. Все тесты приведены на рисунках 16-20, внизу к каждой строчке дано пояснение. Все тесты дали положительный результат, ошибок найдено не было.

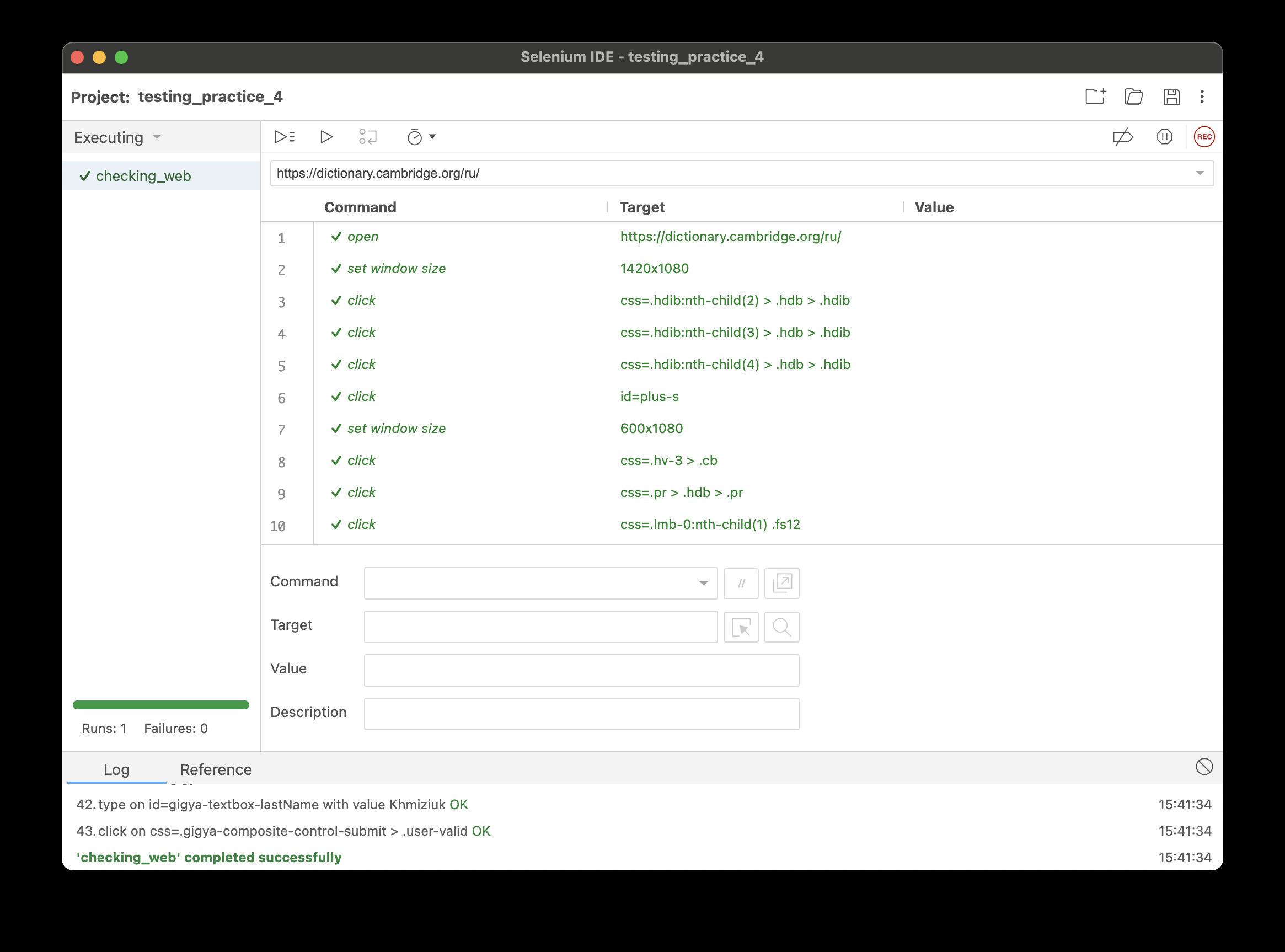


Рис. 16 – Первая часть выполненных тестов

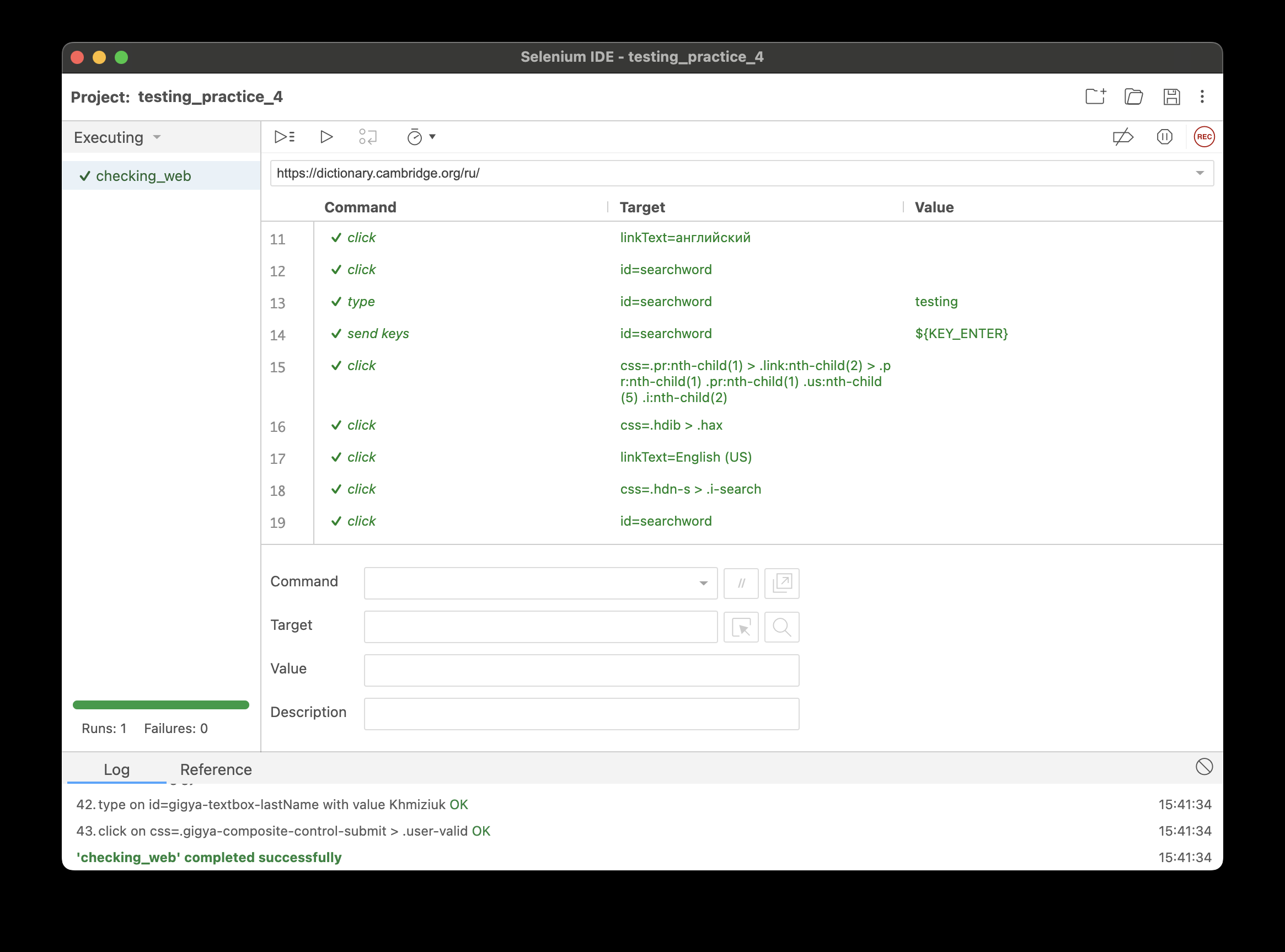


Рис. 17 – Вторая часть выполненных тестов

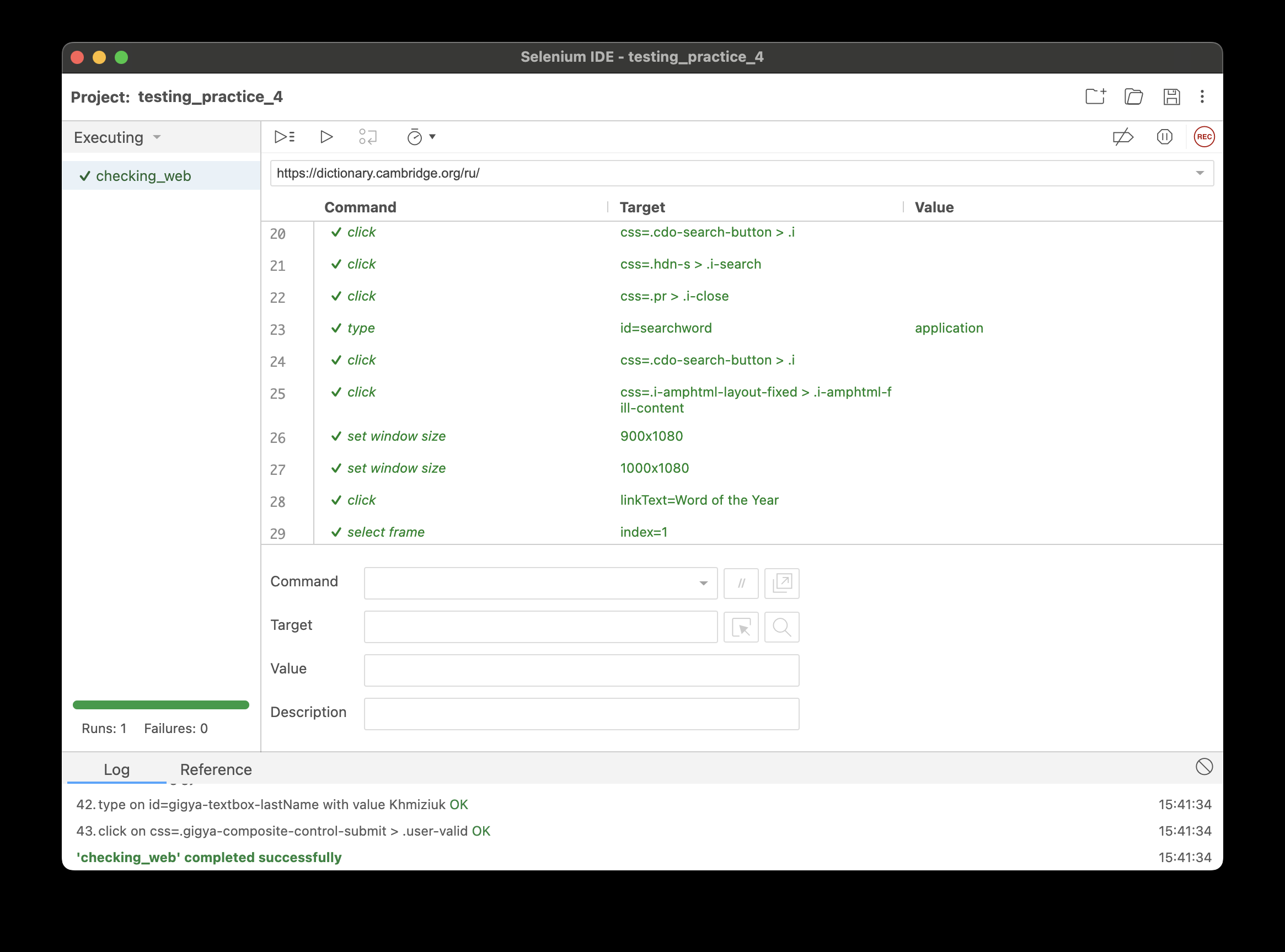


Рис. 18 – Третья часть выполненных тестов

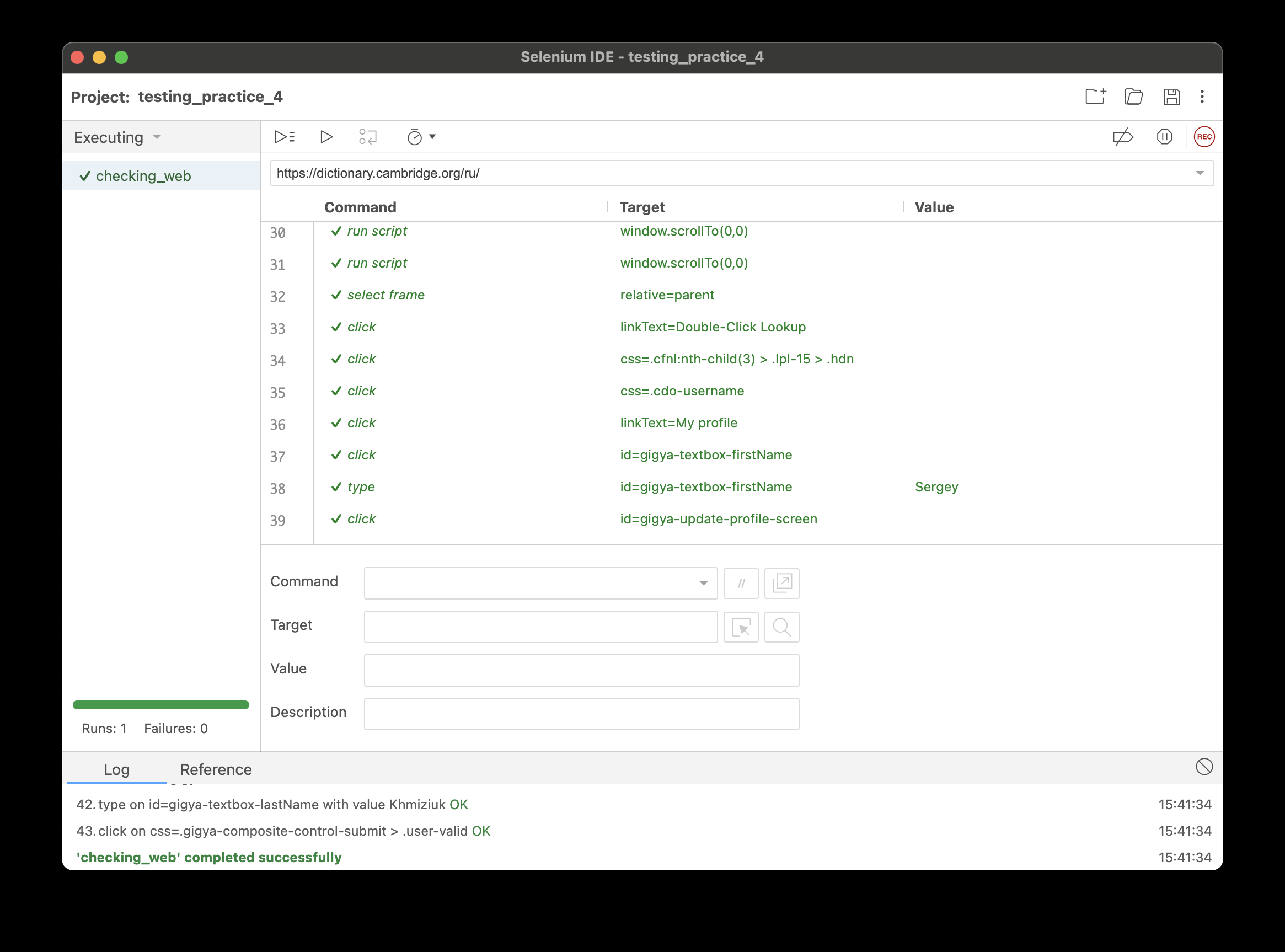


Рис. 19 – Четвертая часть выполненных тестов

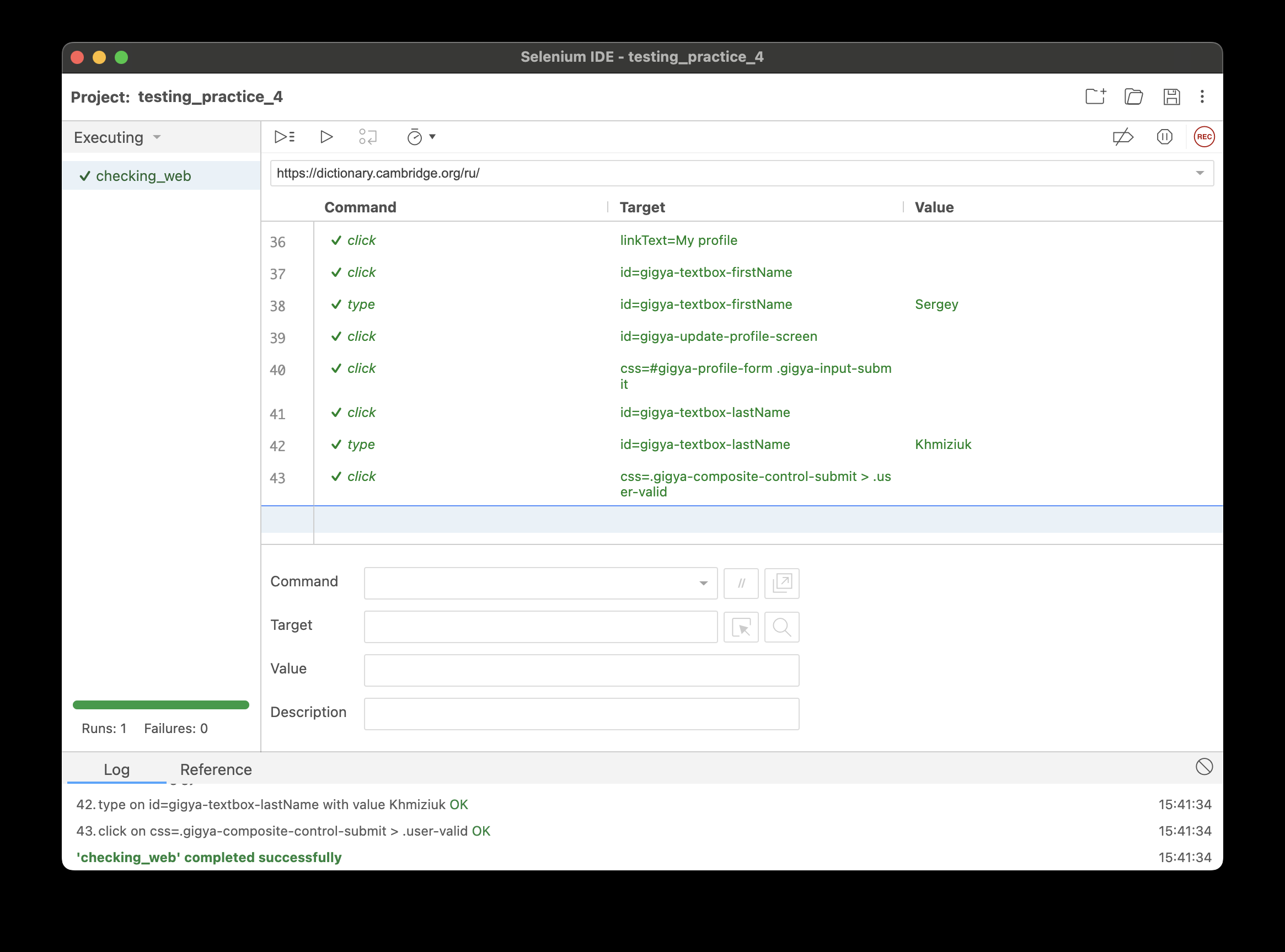


Рис. 20 – Пятая часть выполненных тестов

Пояснения к тестам:

1. Открытие сайта

2. Изменения размера окна

3-6. Тестирование хедера

7. Изменение размера окна

8-11. Тестирование хедера, свернутого в сэндвич

12-14. Введение слова для определения его значения

15. Нажатие на кнопку озвучивания произношения слова

16-17. Смена языка на английский

18-24. Поиск значения второго слова

25. Переход на главный экран по нажатию на логотип сайта в хедере

26-27. Изменение размера окна

28-34. Тестирование хедера путем выборки каждой страницы и перехода на нее

35-36. Переход в профиль

37-39. Изменение и сохранение имени

40-43. Изменение и сохранение фамилии

# Выводы

В ходе выполнения данной практической работы нами был изучен метод функционального программирования, протестированы десктопное и web- приложения. В ходе тестирования ошибок найдено не было.

# Список использованной литературы

1. Лекции Петренко А.А. по предмету «Тестирование и верификация программного обеспечения»
2. Cambrige Dictionary - <https://dictionary.cambridge.org/>
3. Расширение Selenium IDE - <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd>
4. Знакомство с TestComplete - <https://habr.com/ru/sandbox/73760/>