

|  |
| --- |
| 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»**

**Студенты группы** ИКБО-20-21

Хитров Н.С.

Емельянов И.В.

Квашнин Ю.В.

Мухаметшин А.Р.

(подпись студента)

**Принял руководитель работы** Овчинникова М.А.

(подпись руководителя)

Практические работы выполнены « » 2023 г.

Зачтено « » 2023 г.

Содержание

[1. Введение 3](#_Toc149128737)

[1.1 Задание для выполнения 3](#_Toc149128738)

[1.2 Материальная часть 3](#_Toc149128739)

[1.2.1 Функциональное тестирование 3](#_Toc149128740)

[2. Инструменты 3](#_Toc149128743)

[2.1. Используемое ПО 3](#_Toc149128744)

[3. Выполнение задания 4](#_Toc149128745)

[3.1. Тестирование десктопного приложения 4](#_Toc149128746)

[3.1.1. Описание программного продукта 4](#_Toc149128747)

[3.1.2. Тестирование программного продукта 5](#_Toc149128753)

[3.2. Тестирование веб-приложения 9](#_Toc149128754)

[3.2.1. Описание программного продукта 9](#_Toc149128755)

[3.2.2. Тестирование программного продукта 10](#_Toc149128762)

[4. Выводы 16](#_Toc149128763)

[4. Список использованных источников 17](#_Toc149128764)

# Введение

## 1.1 Задание для выполнения

Реализовать автоматическое тестирование десктопного приложения при помощи инструмента TestComplete.

Реализовать автоматическое тестирование веб-приложения при помощи инструмента Selenium IDE.

## 1.2 Материальная часть

## 1.2.1 Функциональное тестирование

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



# Инструменты

## Используемое ПО

Для проведения тестирования программных продуктов использовались инструменты TestComplete и Selenium IDE.

# Выполнение задания

## Тестирование десктопного приложения

## Описание программного продукта

В качестве десктопного приложения для тестирование было выбрано «Games XO», предназначенное для игры в «крестики-нолики». Интерфейс и функционал данного приложения представлены на рисунках 1-3.

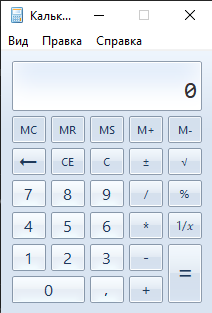


Рисунок 1 – Стартовое меню

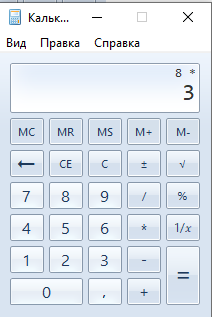


Рисунок 2 – Ввод операции

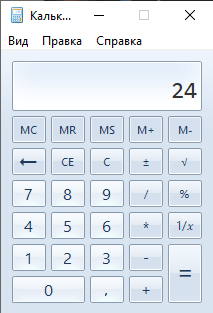


Рисунок 3 – Результат работы



## Тестирование программного продукта

Код тестирования выполнения операции суммы (последовательность операций: 7 + 1 = ; + 4 = ) представлен на рисунке 4.

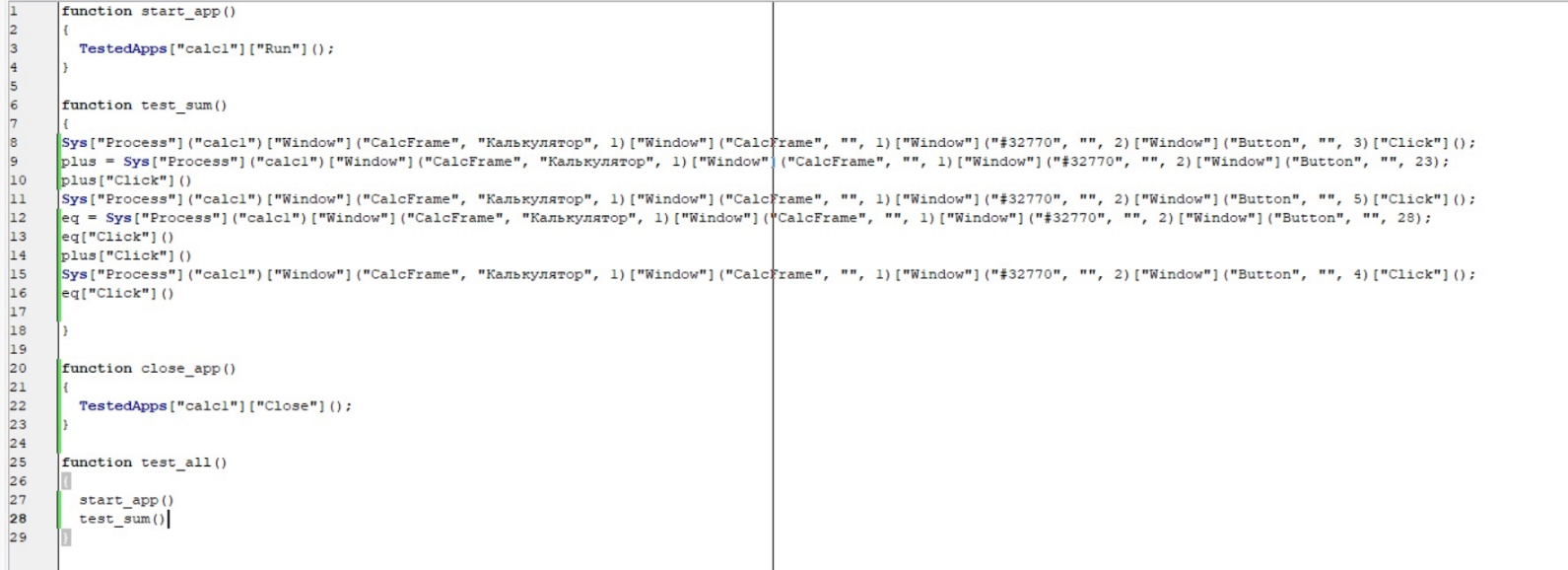


Рисунок 4 – Тестирование выполнения операции суммы

Результат выполнения теста представлен на рисунке 5.

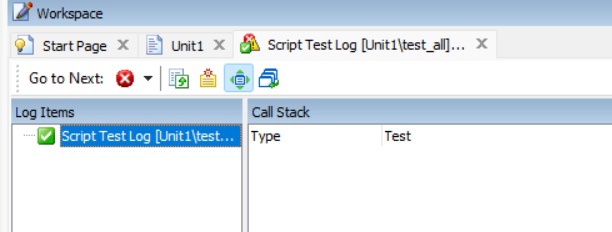


Рисунок 5 – Результат выполнения теста

Тестирование вывода произведения чисел представлено на рисунке 6.

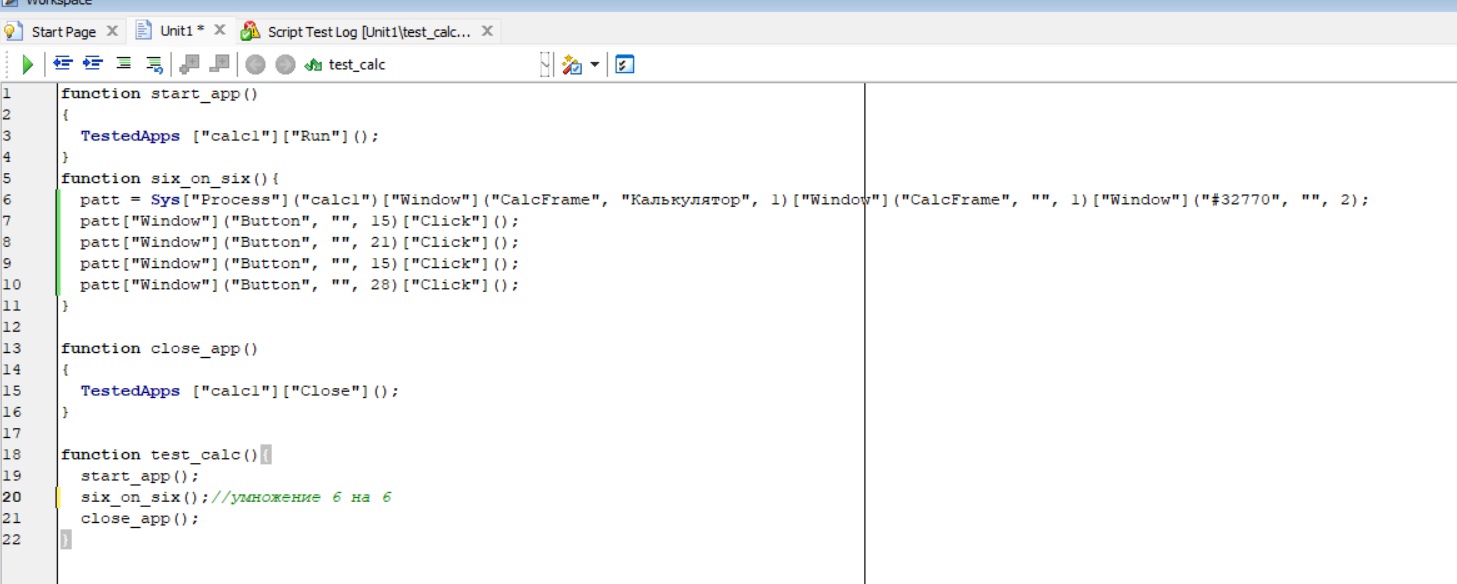


Рисунок 6 – Тестирование вывода произведения чисел

Результат выполнения теста представлен на рисунке 7.

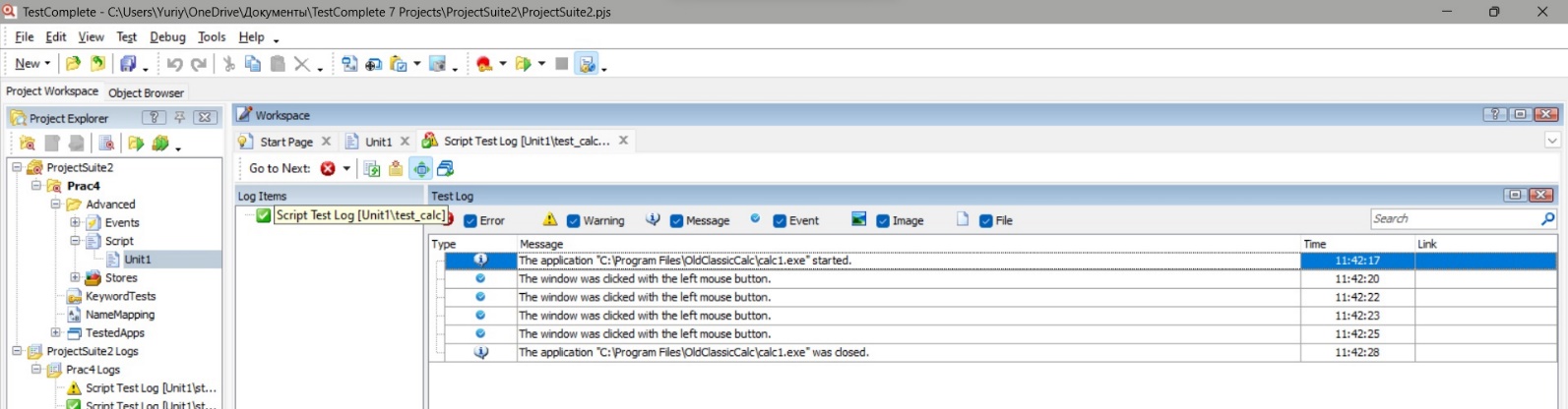


Рисунок 7 – Результат выполнения теста

Тестирование функции инженерного режима представлено на рисунке 8.

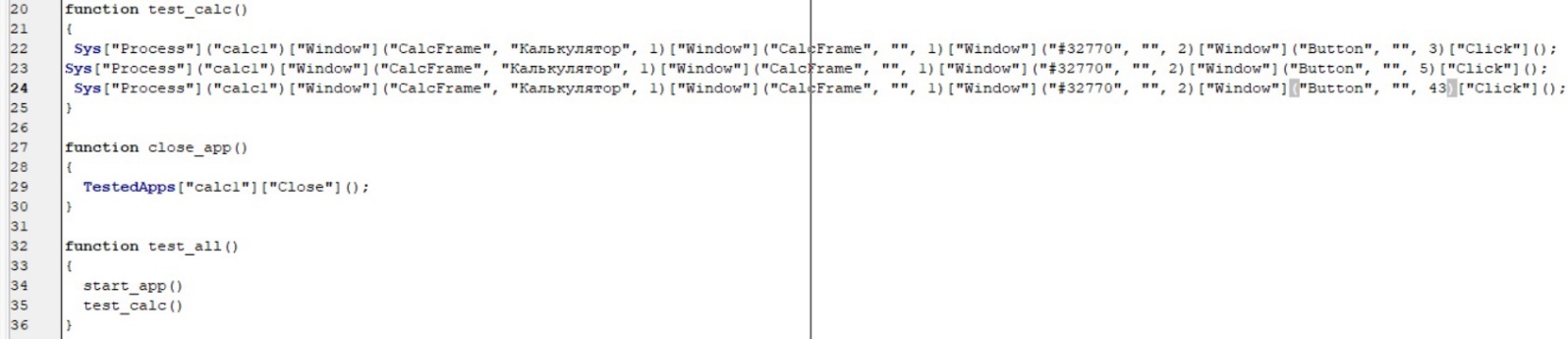


Рисунок 8 – Тестирование функции инженерного режима

Результат выполнения теста представлен на рисунке 9.

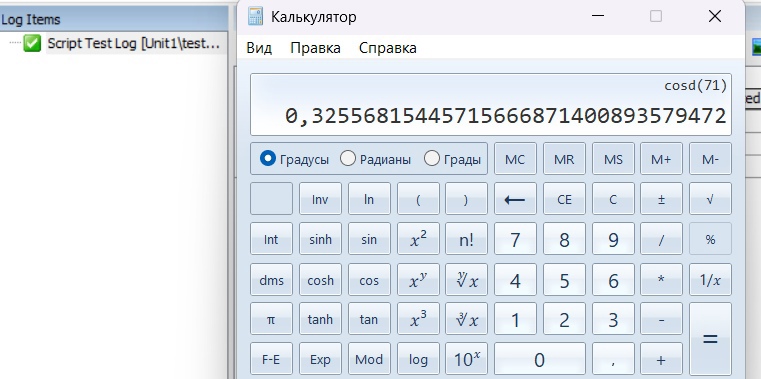


Рисунок 9 – Результат выполнения теста

Тестирование функции функций инженерного режима, функция возведения в квадрат представлено на рисунке 10.

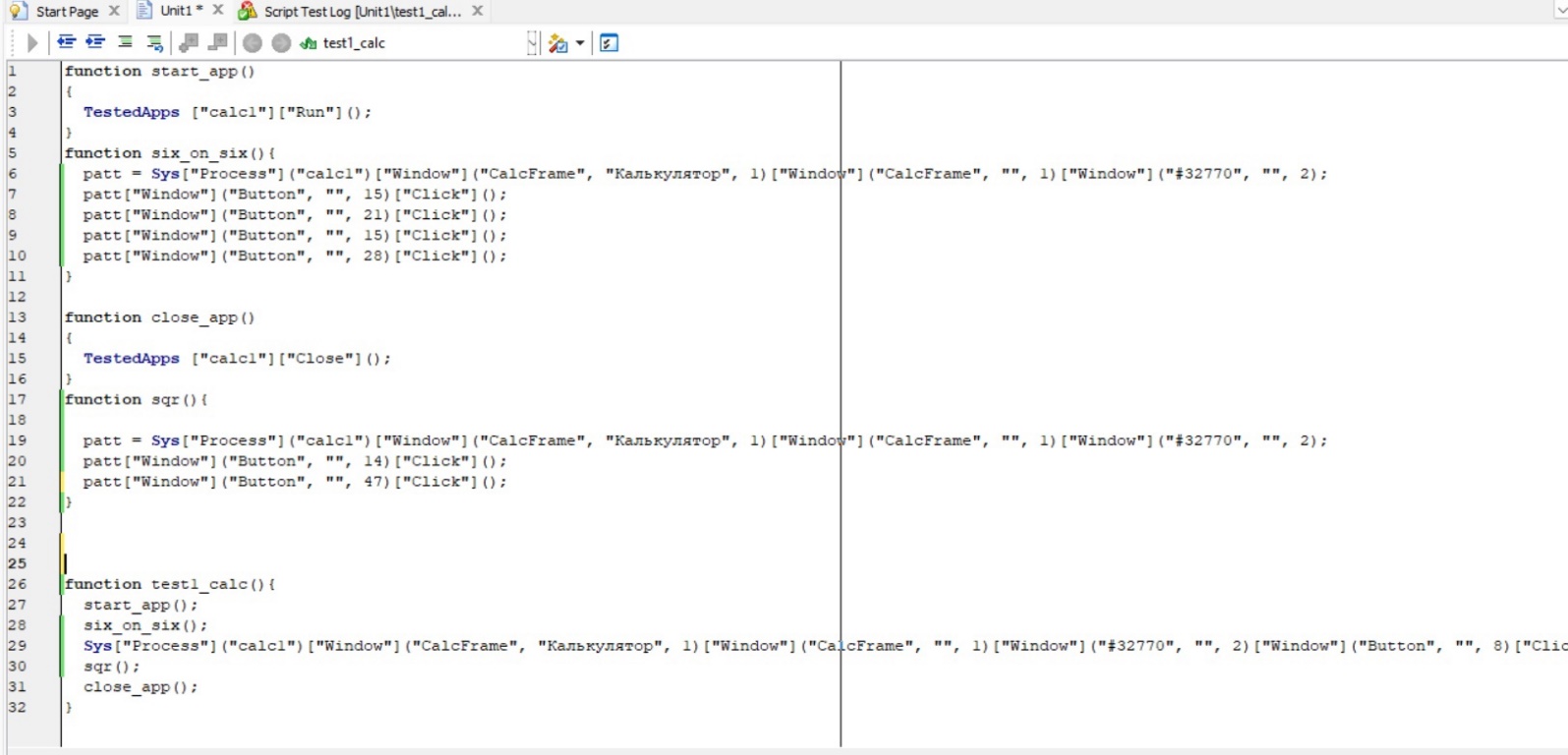


Рисунок 10 – Тестирование функций инженерного режима, функция возведения в квадрат

Результат выполнения теста представлен на рисунке 11.

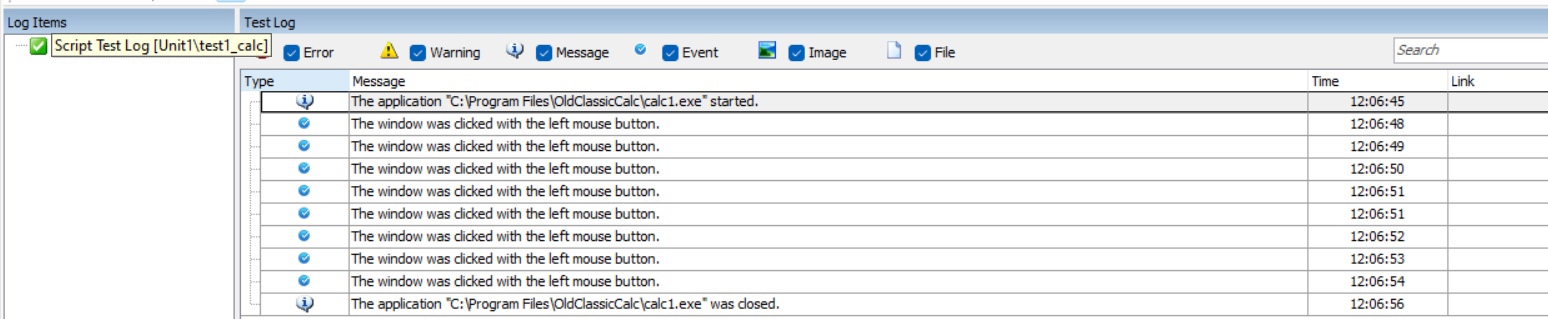


Рисунок 11 – Результат выполнения теста

## Тестирование веб-приложения

## Описание программного продукта

В качестве веб-приложения для тестирования был выбран веб-сервис «Яндекс.Музыка», один из крупнейших, самых многофункциональных и популярных веб-сервисов для прослушивания музыки. Интерфейс данного приложения представлен на рисунках 12, 13.

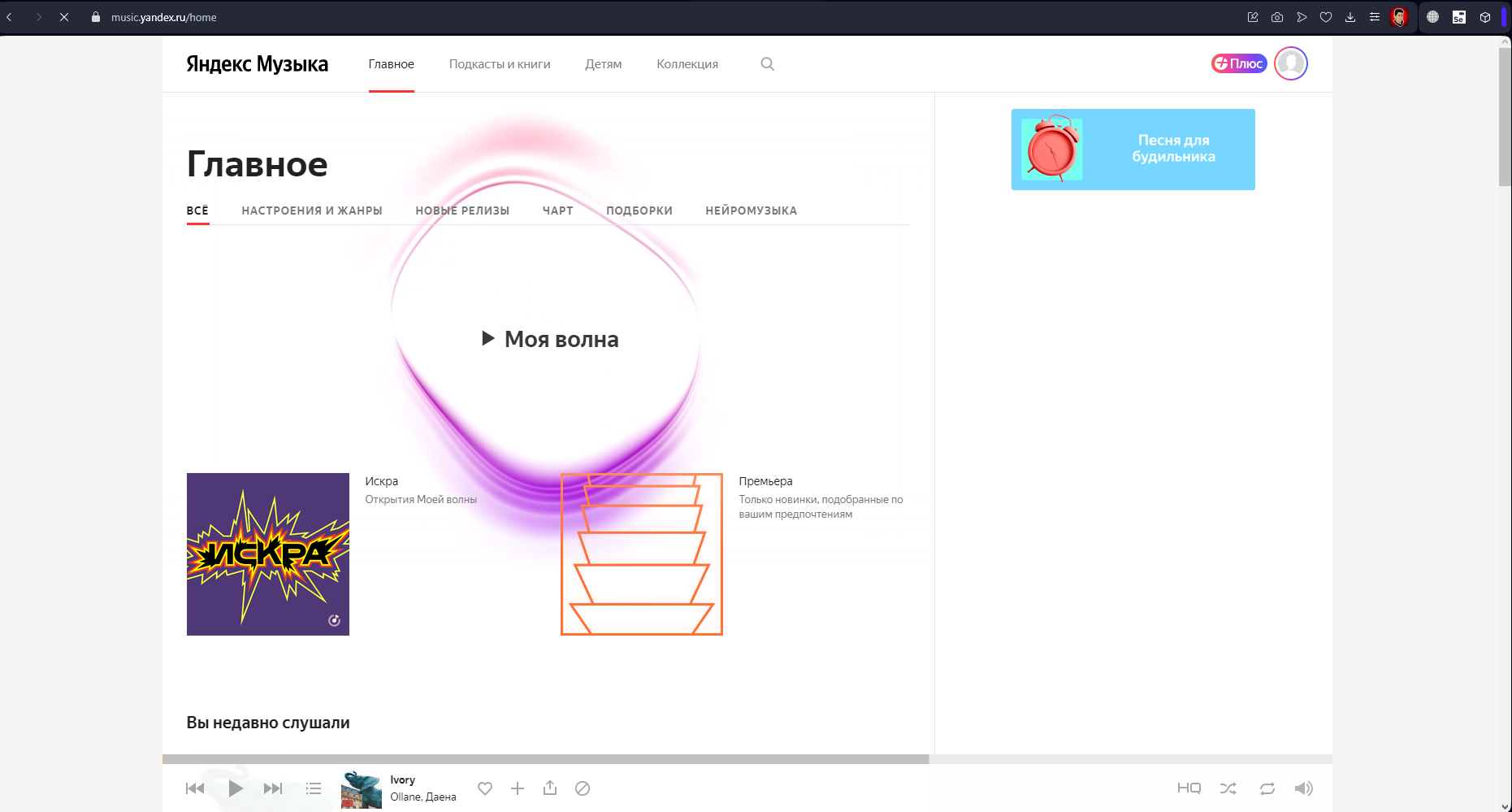


Рисунок 12 – Интерфейс веб-приложения

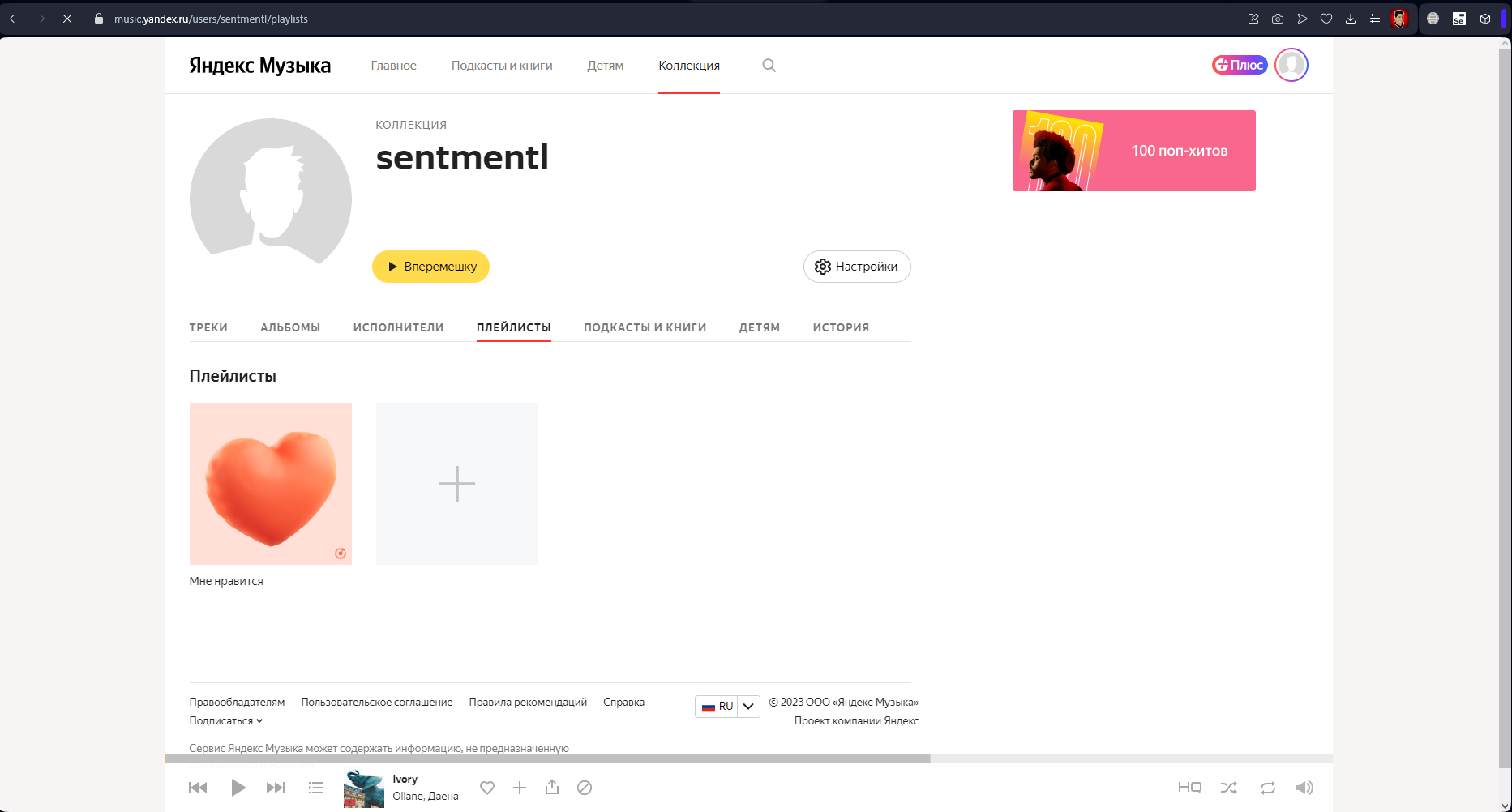


Рисунок 13 – Интерфейс веб-приложения



## Тестирование программного продукта

Тестирование добавления песни в коллекцию «Мне нравится» представлено на рисунке 14.

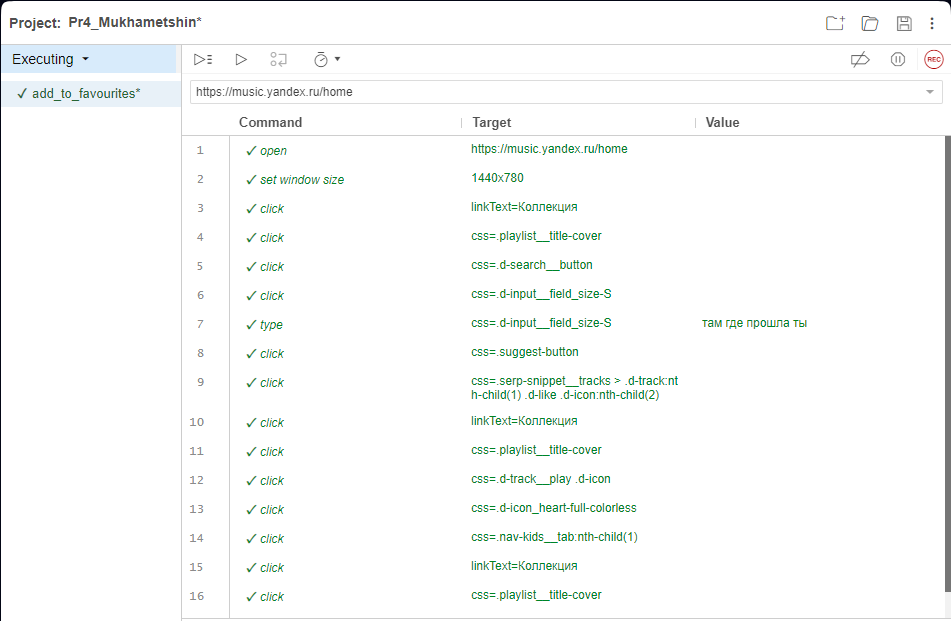


Рисунок 14 – Тестирование добавления песни в коллекцию «Мне нравится»

Результат тестирования представлен на рисунке 15.

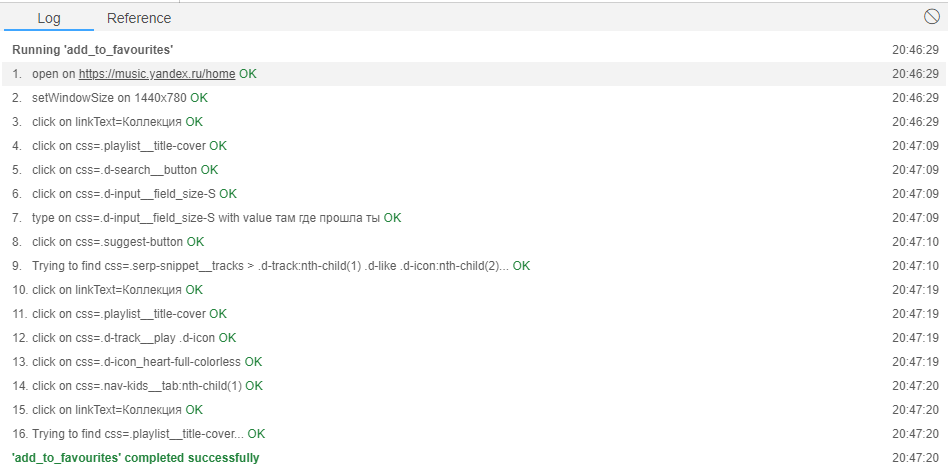


Рисунок 15 – Результат тестирования

Тестирование поиска по всем трекам и внутри коллекции «Мне нравится» представлено на рисунке 16.

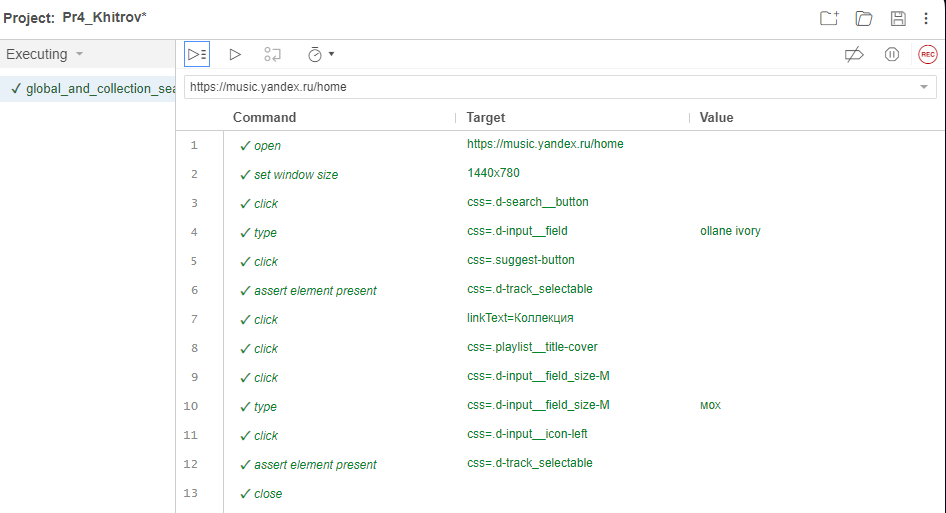


Рисунок 16 – Тестирование поиска по всем трекам и внутри коллекции «Мне нравится»

Результат тестирования представлен на рисунке 17.

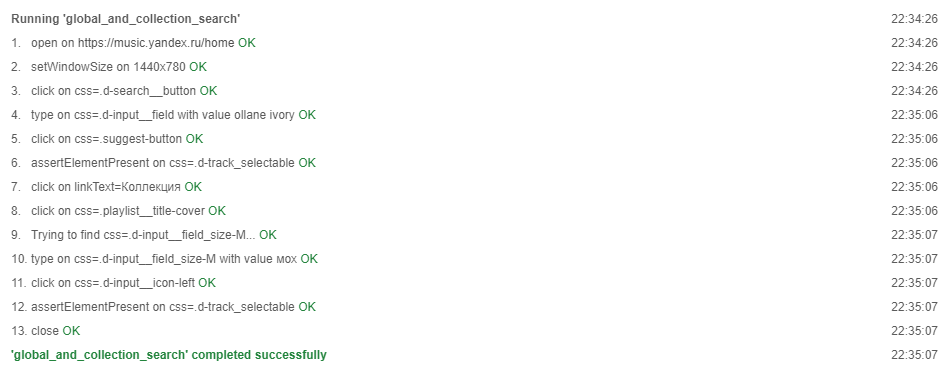


Рисунок 17 – Результат тестирования

Тестирование переключения музыки, смены режима перемешивания треков, перемещения вперед - назад, остановки представлено на рисунках 18-20.

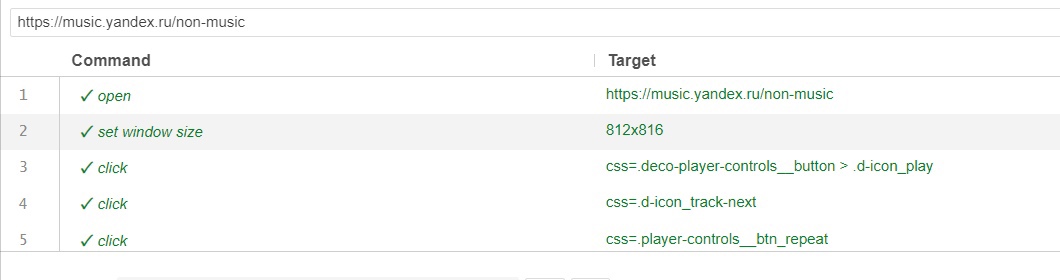


Рисунок 18 – Тестирование переключения музыки

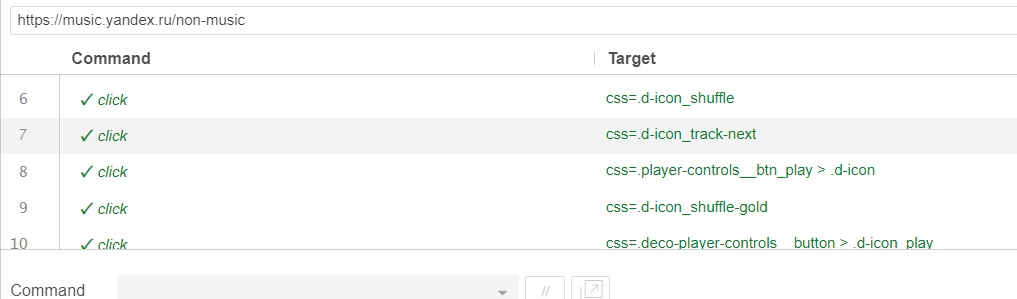


Рисунок 19 – Тестирование смены режима перемешивания треков

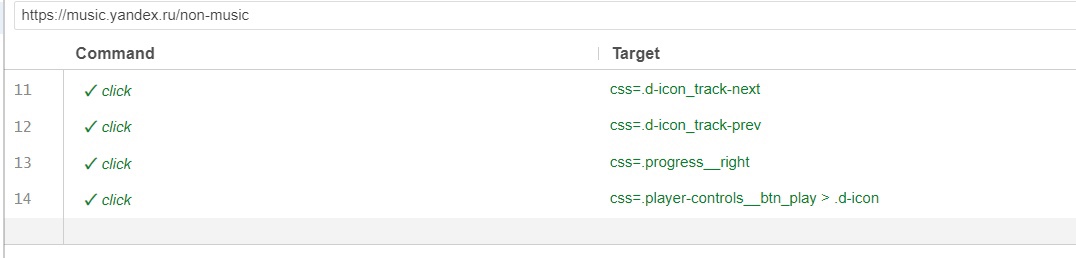


Рисунок 20 – Тестирование перемещения вперед – назад по таймлайну трека, остановки

Результат тестирования представлен на рисунках 21-22.

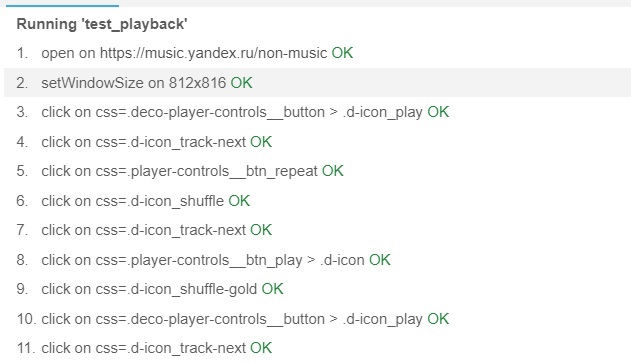


Рисунок 21 – Результат тестирования (часть 1)

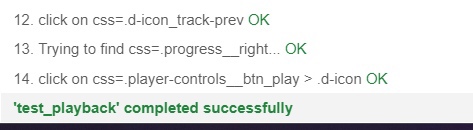


Рисунок 22 – Результат тестирования (часть 2)

Тестирование громкости, эквалайзера, перемещения ползунков, выключения звука, выключения эквалайзера представлено на рисунках 23-26.

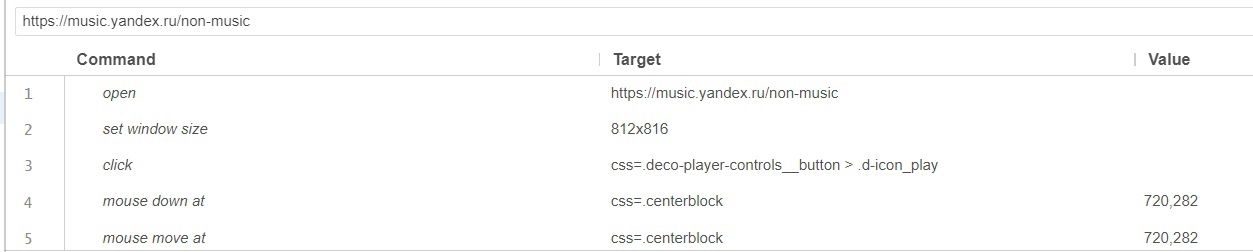


Рисунок 23 – Тестирование громкости громкости

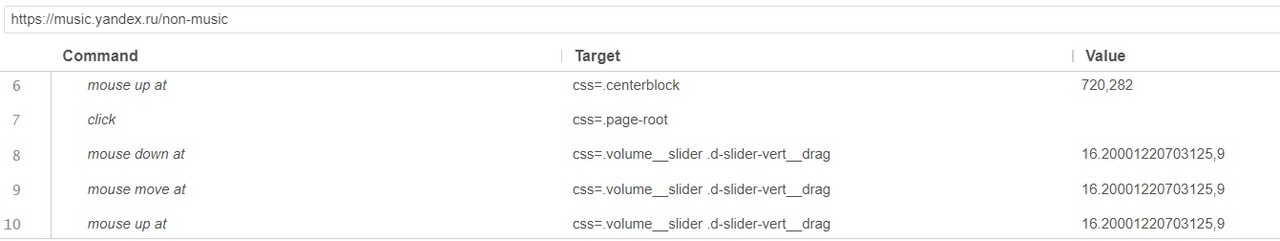


Рисунок 24 – Тестирование громкости



Рисунок 25 – Тестирование просмотра информации



Рисунок 26 – Тестирование просмотра информации

Результат тестирования представлен на рисунках 27-28.

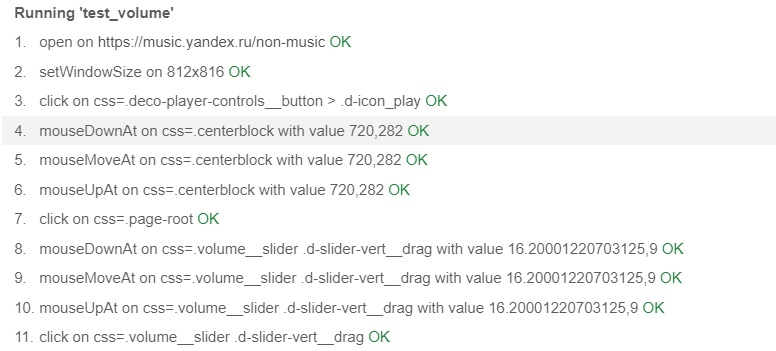


Рисунок 27 – Результат тестирования (часть 1)

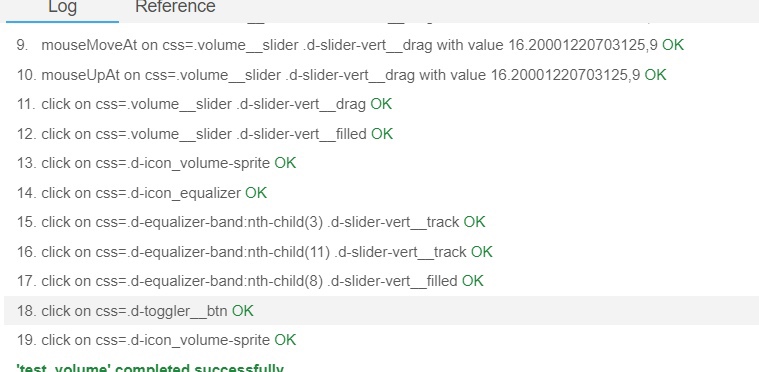


Рисунок 28 – Результат тестирования (часть 2)

# Выводы

Тестирование десктопного приложения и веб-приложения успешно реализовано с использованием инструментов TestComplete и Selenium IDE. Все ключевые аспекты функциональности покрыты тестами, что обеспечивает устойчивость и надежность кода.

# Список использованных источников

1. Статический анализ кода [Электронный ресурс] – <https://www.jetbrains.com/ru-ru/resharper/features/code_analysis.html>
2. Динамический анализ кода [Электронный ресурс] – <https://habr.com/ru/companies/pvs-studio/articles/580196/>
3. WebStrom – анализ кода [Электронный ресурс] – <https://www.jetbrains.com/ru-ru/resharper/features/code_analysis.html>
4. Использование статического и динамического анализа для повышения качества продукции и эффективности разработки [Электронный ресурс] – <https://www.swd.ru/print.php3?pid=828>