



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №4

по дисциплине «Тестирование и верификация ПО»

Выполнили:

Студент группы ИКБО-13-22

Пашкин В.П.

Проверил:

Преподаватель

Запорожских А.И.

2024 г.

ВВЕДЕНИЕ

Цель работы: освоить основные принципы разработки плана тестирования программного обеспечения и применить их на практике, а также научиться проводить автоматизированное тестирование Web-приложений.

Задача с 3-ей практической работы: Игра угадай мелодию (3 игрока). По очереди проигрывается мелодия и спрашивается, что это за мелодия.

ЧАСТЬ 1. РАЗРАБОТКА ПЛАНА ТЕСТИРОВАНИЯ

1. Идентификатор тестового плана

TP-GM-001 (Testing Plan for "Guess the Melody" Game)

2. Ссылки на используемые документы

1. IEEE Std 829-2019;
2. Документация на библиотеку NUnit;
3. Документация на библиотеку specflow.

3. Введение

Данный план описывает процесс тестирования модуля "Guess the Melody" – программы, в которой три игрока по очереди пытаются угадать название проигрываемой мелодии. Тестирование будет выполнено с использованием методологий BDD для обеспечения соответствия функциональности требованиям.

4. Тестируемые элементы

Метод MusicStart(int index) в классе MusicPlayer;

Метод MusicStop() в классе MusicPlayer;

Метод MusicChecker(string ans) в классе MusicPlayer;

Метод GetScore() в классе Score.

5. Проблема риска тестирования ПП

1. Неправильный вывод результата игры при угадывании мелодии;
2. Ошибки в инициализации и остановке музыкального проигрывателя;
3. Неправильная проверка введённого пользователем ответа;
4. Проблемы с подсчетом очков игроков.

6. Особенности или свойства, подлежащее тестированию

1. Корректность проигрывания мелодий для всех доступных индексов;
2. Точность проверки совпадения ответа игрока с названием мелодии;
3. Корректное завершение игры и вывод итогового счёта.

7. Особенности (свойства), не подлежащее тестированию

1. Качество звучания аудио;
2. Сложность угадывания мелодий.

8. Подход

Используются методологии TDD и BDD. Использование методологий будет применено через юнит-тесты и сценарии поведения соответственно.

9. Критерии смоук-тестирования

Проверка успешного запуска программы, воспроизведения музыки и проверки ответа на соответствие..

10. Критерии прохождения тестов

1. Все тесты должны выполняться без ошибок и исключений;
2. Результаты тестирования должны соответствовать ожидаемым результатам.

11. Критерии приостановки и возобновления работ

1. Критические ошибки во время тестирования;
2. Внести изменения в функционал;
3. Несоответствия результатов с ожидаемыми.

12. Тестовая документация

1. Скрипты юнит-тестов (файл: UnitTest1.cs, UnitTest2.cs, UnitTest3.cs, UnitTest4.cs);
2. Файл с тестами поведения (файл: GameFalse.cs, GameInitialization.cs, Step2.cs);
3. Отчёт о результатах тестирования.

13. Основные задачи тестирования

1. Проверка правильности проигрывания и остановки мелодий;
2. Проверка точности метода в оценке ответов игроков;
3. Проверка корректности подсчёта и отображения очков.

14. Необходимый персонал и обучение

1. Тестировщик с опытом работы с C#, библиотеками NUnit и SpecFlow;;
2. Обучение методологиям TDD и BDD для членов команды.

15. Требование среды

1. .NET Core SDK версии 5.0 или выше;
2. Установленные библиотеки NUnit и SpecFlow.

16. Распределение ответственности

1. Тестировщик отвечает за написание тестов и обнаружение ошибок в продукте;
2. Разработчик отвечает за разработку и исправление найденных ошибок.

17. График работ (календарный план)

Этап	Дата начала	Дата окончания
Анализ требований	12.10.2024	13.10.2024
Разработка тестов	13.10.2024	14.10.2024
Выполнение тестов	14.10.2024	18.10.2024
Анализ результатов	18.10.2024	19.10.2024

18. Риски и непредвиденные обстоятельства

1. Автоматизированное тестирование без игрока;
2. Изменения функционала, требующее дописывание тестов.

19. Утверждение плана тестирования

План тестирования должен быть утверждён руководителем проекта и командой разработки.

20. Глоссарий

TDD (Test-Driven Development) – методология разработки, при которой сначала создаются тесты, а потом на их основе пишутся модули, которые должны их проходить.

BDD (Behavior Driven Development) – методология разработки, при которой сначала описываются сценарии поведения системы, а потом на их основе пишутся сами модули.

Юнит-тест – один из видов тестов, который подразумевает тестирование отдельных модулей на работоспособность и выдачу правильных результатов.

ЧАСТЬ 2. АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЯ

Выбор программного продукта для разработки скриптов для тестирования Web-приложения

В качестве выбора продукта для разработки скриптов для тестирования была выбрана среда Selenium IDE. Она была установлена как расширение для браузера Google Chrome, где и проводилось дальнейшее тестирование web-приложений.

Выбор ресурса (или ресурсов), для написания скриптов

В качестве тестируемых web-ресурсов были выбраны три сайта:

1. Демонстрационный сайт, который является интернет магазином: <https://store.steampowered.com/>.

Разработка скриптов

Были разработаны три скрипта, которые тестируют выбранные ресурсы: авторизация на сайте, открытие комментариев под постом, добавление товара в корзину.

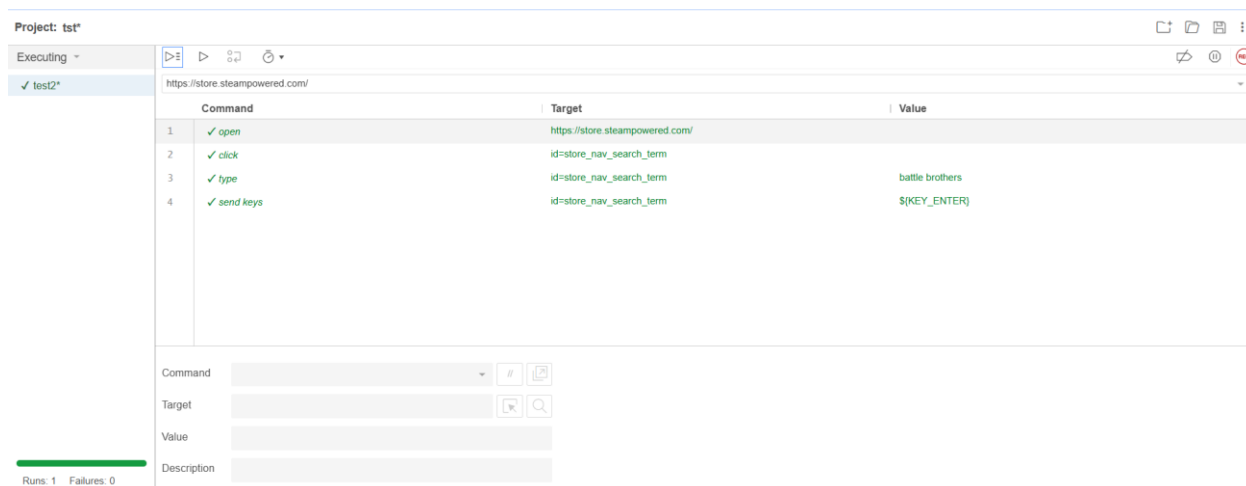


Рисунок 1 – Скрипт поиска товара через поисковик

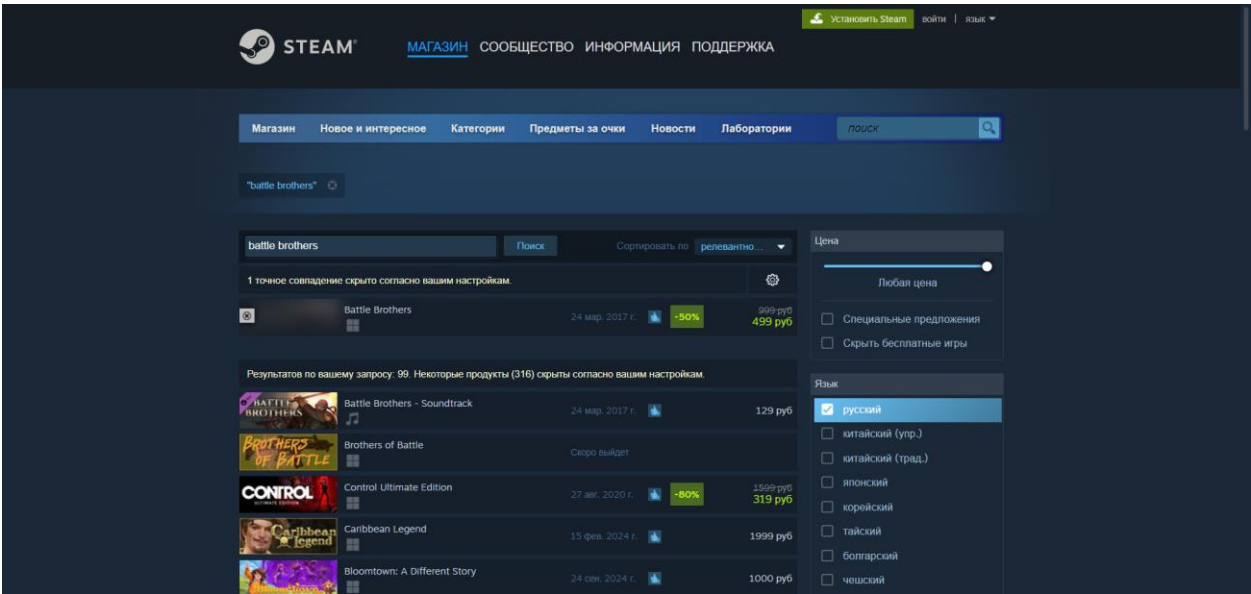


Рисунок 2 – Результат скрипта поиска товара

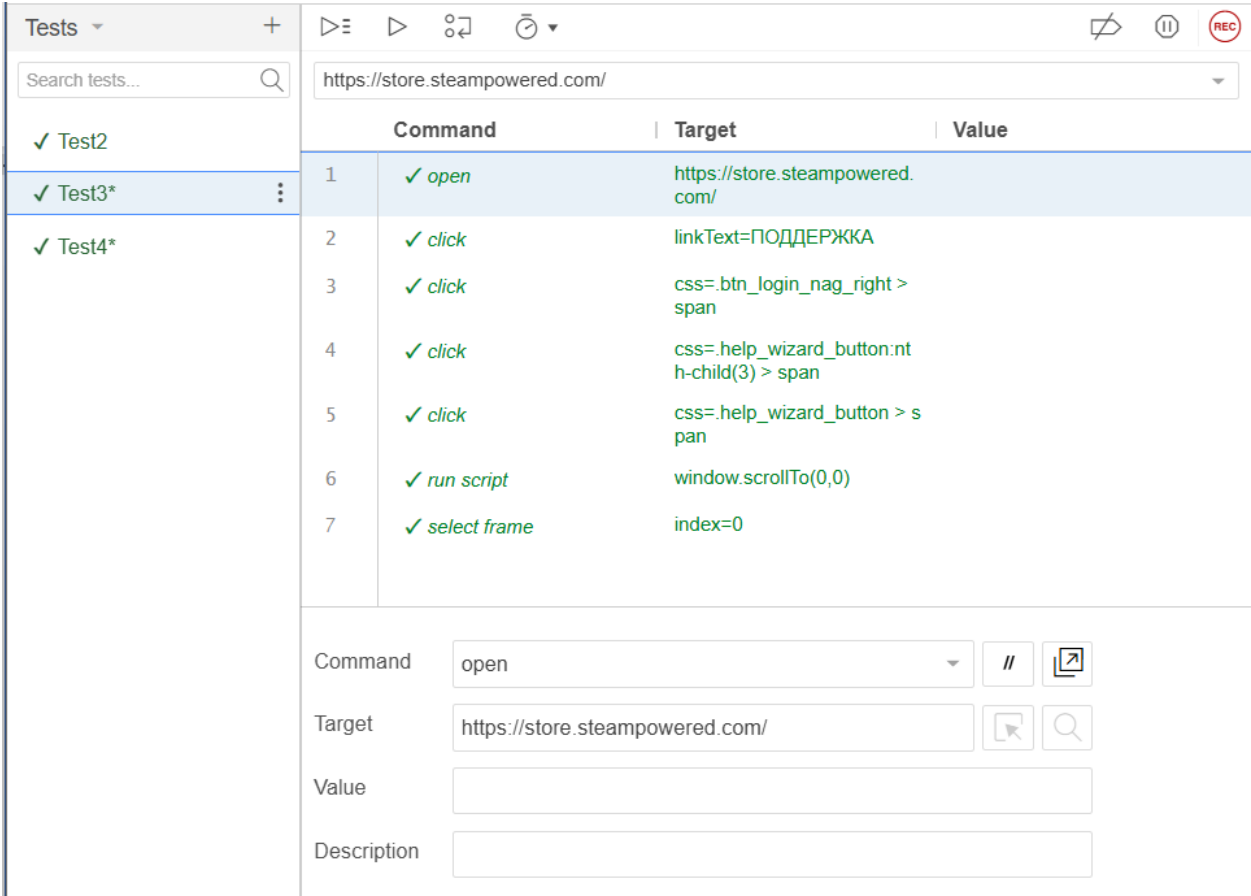


Рисунок 3 – Скрипт перехода на страницу поддержки и выбора опций

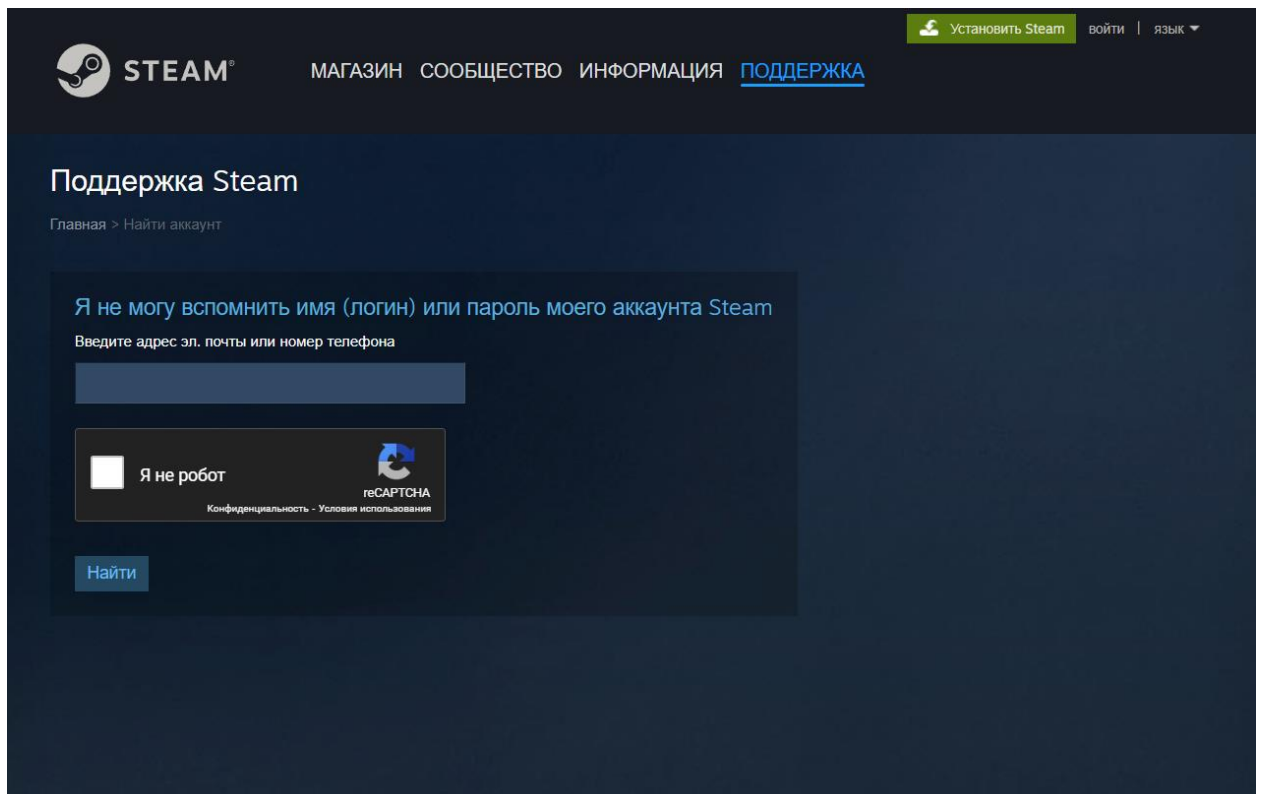


Рисунок 4 – Результат перехода на страницу поддержки и выбора опций

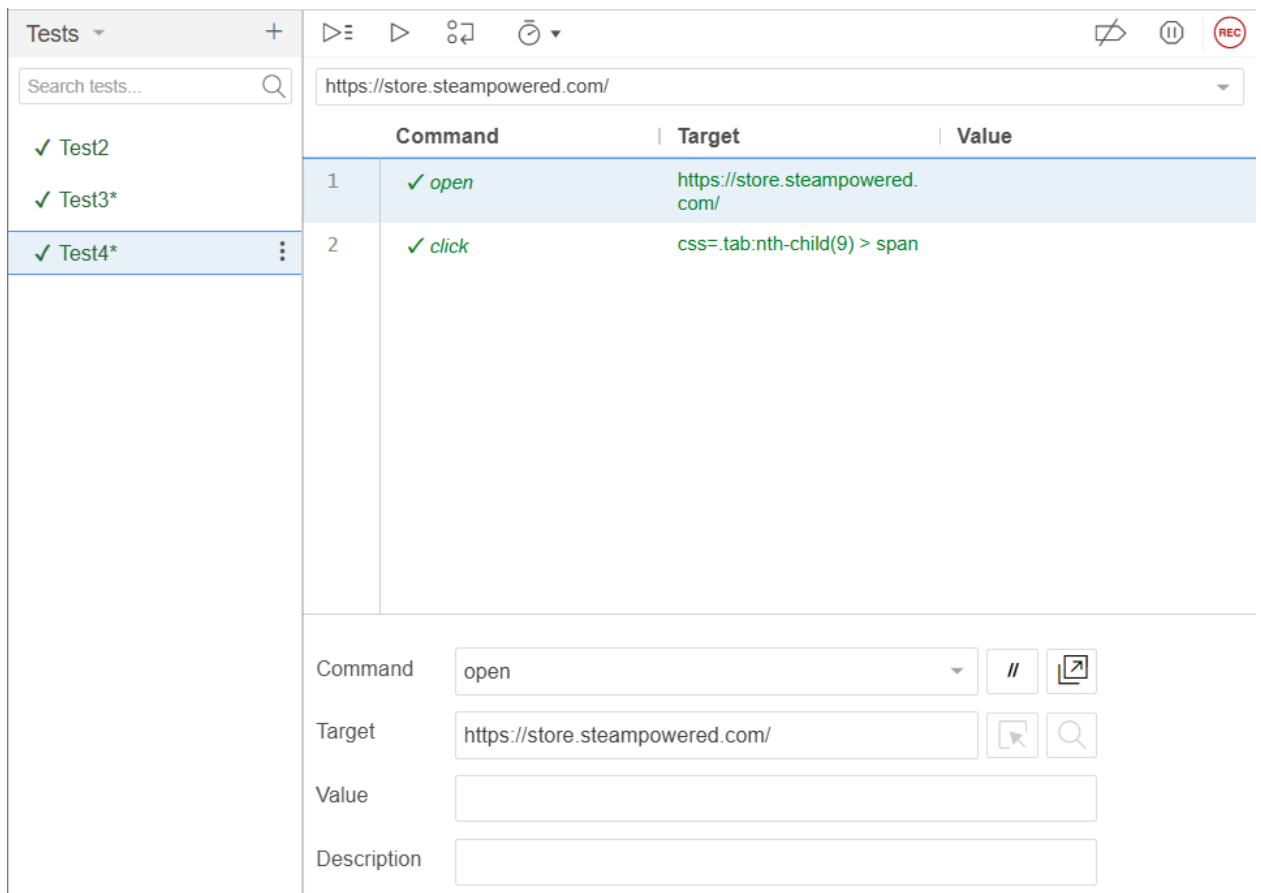


Рисунок 5 – Скрипт перехода в новинки

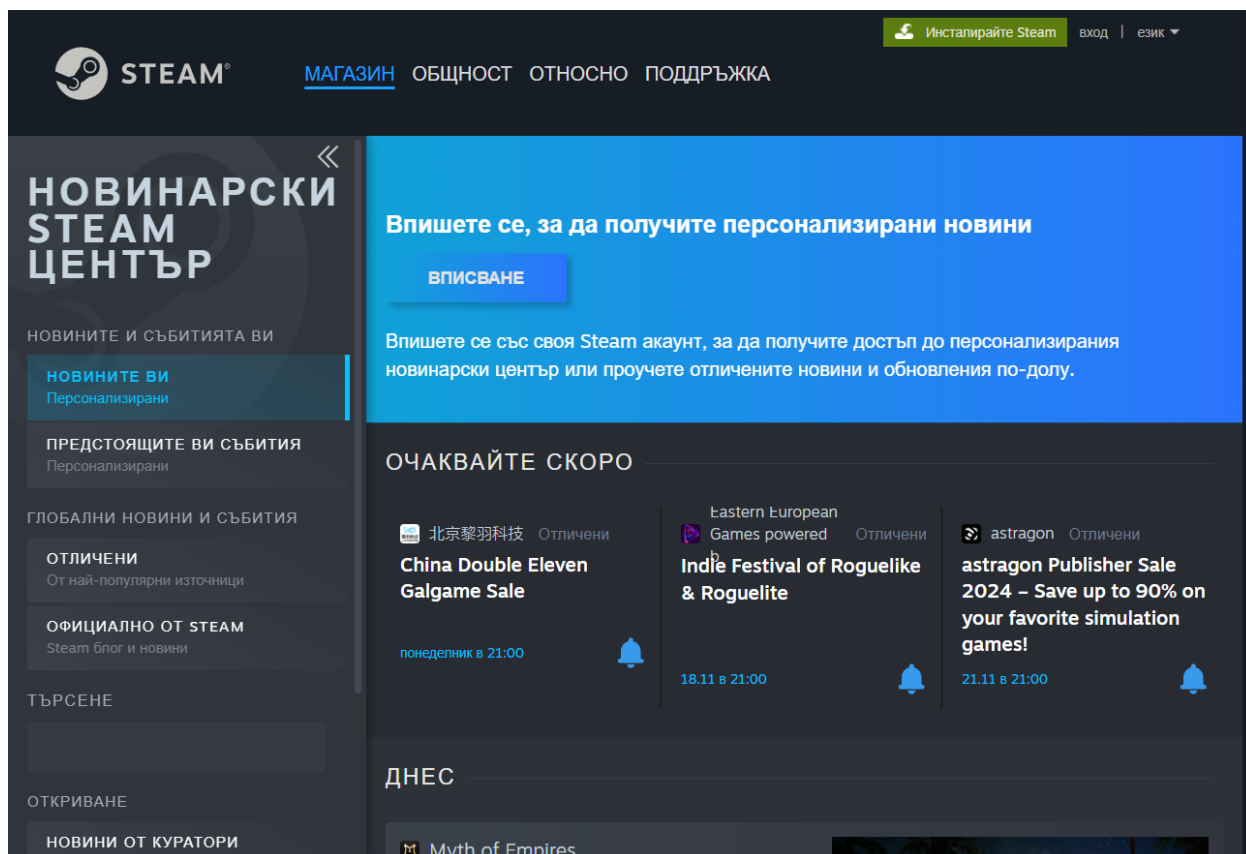


Рисунок 6 – Резултат скрипта перехода в новинки

ЗАКЛЮЧЕНИЕ

Было изучено составление плана тестирования приложения. Был написан план тестирования приложения по стандарту IEEE Std 829-2019. Было изучено тестирование веб-приложений через Selenium IDE. Были написаны скрипты, которые тестируют веб-приложения.

ПРИЛОЖЕНИЕ

Листинг 1 – Скрипт поиска товара через поисковик

```
Command: open
Target: https://store.steampowered.com/
Value:
Command: click
Target: id=store_nav_search_tem
Value:
Command: type
Target: id=store_nav_search_tem
Value: battle brothers
Command: send keys
Target: id=store_nav_search_tem
Value: ${KEY_ENTER}
```

Листинг 2 – Скрипт перехода на страницу поддержки

```
Command: open
Target: https://store.steampowered.com/
Value:
Command: click
Target: linkText=ПОДДЕРЖКА
Value:
Command: click
Target: css=.btn_login_nag_right > span
Value:
Command: click
Target: css=.help_wizard_button:nth-child(3) > span
Value:
Command: click
Target: css=.help_wizard_button > span
Value:
Command: select frame
Target: index 0
Value:
```

Листинг 3 – Скрипт перехода в новинки

```
Command: open
Target: https://store.steampowered.com/
Value:
Command: click
Target: css=.tab:nth-child(9) > span
Value:
```