

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»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы ИКБО-11-22 | Гришин Андрей Валерьевич |
| **Проверил:**  Преподаватель | Мельников Денис Александрович |

2024 г.

**СОДЕРЖАНИЕ**

[Разработка плана тестирования 3](#_Toc182318848)

[Автоматизация тестирования Web-приложения 6](#_Toc182318849)

# Разработка плана тестирования

1. **Идентификатор тестового плана:**

TP-TicTacToe-001.

1. **Ссылки на используемые документы**:

TDD:

<https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D1%87%D0%B5%D1%80%D0%B5%D0%B7_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5>

BDD:

<https://ru.wikipedia.org/wiki/BDD_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)>

1. **Введение**:

Этот план тестирования описывает подход к тестированию консольной игры "Крестики-нолики", которая позволяет двум игрокам поочередно делать ходы, пока один из них не победит или игра не закончится вничью. Основные цели тестирования – проверить корректность логики игры и убедиться, что она работает без сбоев и ошибок.

1. **Тестируемые элементы**:

* Основной модуль игры, реализующий логику игры (класс TicTacToe)
* Основные функции:
  + make\_move: совершение хода
  + check\_winner: проверка победителя
  + is\_draw: проверка на ничью
  + switch\_player: смена текущего игрока
* Пользовательский интерфейс в консоли для взаимодействия с игроками через функцию main

1. **Проблемы риска тестирования ПП**:

* Логические ошибки, приводящие к некорректному определению победителя или ничьей.
* Сбои при попытке сделать ход в уже занятую клетку.
* Неверные или некорректные вводные данные со стороны пользователя.
* Непредвиденные завершения работы программы при некорректном использовании.

1. **Особенности или свойства, подлежащие тестированию**:

* Корректность выполнения основных функций (make\_move, check\_winner, is\_draw, switch\_player).
* Проверка правильности смены игрока после каждого хода.
* Обработка сценария победы одного из игроков.
* Обработка сценария ничьей, когда все клетки заполнены.
* Проверка ввода данных и обработка ошибок ввода.

1. **Особенности (свойства), не подлежащие тестированию**:

* Графический интерфейс, так как игра реализована в консоли и графический интерфейс отсутствует.
* Производительность, так как данное приложение не имеет высоких требований по скорости или ресурсоемкости.

1. **Подход**:

Тестирование будет проходить посредством TDD и BDD тестирования

1. **Критерии смоук-тестирования**:

Приложение должно загружаться без ошибок.

Основные функции должны работать.

1. **Критерии прохождения тестов**:

* Все функциональные и логические тесты должны проходить успешно.
* Программа должна корректно определять победителя и ничью.
* Все сценарии, описанные в BDD-тестах, должны проходить успешно.

1. **Критерии приостановки и возобновления работ**:

Тестирование приостанавливается при наличии критических ошибок, препятствующих дальнейшей работе.

Тестирование возобновляется после исправления всех блокирующих багов.

1. **Тестовая документация**:

Тест-кейсы

Отчёт о тестировании

Скрипты для автоматизированного тестирования

1. **Основные задачи тестирования**:

* Проверить корректность логики игры (победа, ничья, смена игроков).
* Убедиться в корректной обработке ошибок ввода.
* Выполнить BDD-сценарии, охватывающие все возможные игровые исходы.

1. **Необходимый персонал и обучение**:

* Разработчик для написания тестов и исправления ошибок.
* Тестировщик для выполнения тестов и составления отчетов. Обучение не требуется, так как игра имеет базовую логику и реализована в консольном интерфейсе.

1. **Требования среды**:

* Python 3.7 или новее
* Библиотека behave для BDD-тестирования
* Среда разработки PyCharm

1. **Распределение ответственности**:

Вся работа была выполнена мной.

1. **График работ (календарный план)**:

|  |  |  |  |
| --- | --- | --- | --- |
| Этап | Задача | Дата начала | Дата окончания |
| 1 | Разработка тест-плана | 02.11.2024 | 03.11.2024 |
| 2 | Подготовка тестовой среды | 03.11.2024 | 04.11.2024 |
| 3 | Ручное тестирование | 05.11.2024 | 06.11.2024 |
| 4 | Автоматизация тестов | 07.11.2024 | 08.11.2024 |
| 5 | Подготовка отчета | 11.11.2024 | 12.11.2024 |

1. **Риски и непредвиденные обстоятельства**:

 Задержка в исправлении ошибок, если будет обнаружен сложный логический дефект.

 Проблемы с настройкой тестового окружения или конфликт библиотек.

1. **Утверждение плана тестирования**:

План тестирования утверждён:

Преподаватель: Мельников Д.А.

Дата утверждения: 29.10.2024

1. **Глоссарий**:

* **TDD (Test-Driven Development)** – подход к разработке, при котором тесты пишутся до написания основной функциональности.
* **BDD (Behavior-Driven Development)** – подход к разработке, при котором создаются сценарии, описывающие поведение системы с точки зрения пользователя.
* **Смоук-тестирование** – начальное тестирование, чтобы убедиться, что основные функции работают корректно.

# Автоматизация тестирования Web-приложения

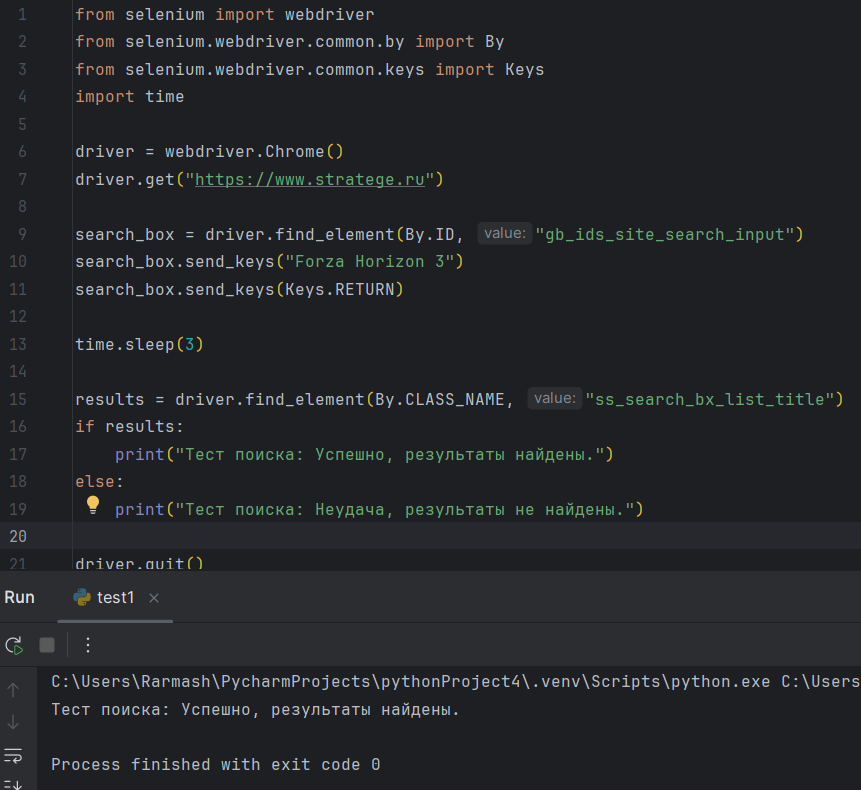


Рисунок 1 – Результат 1 теста

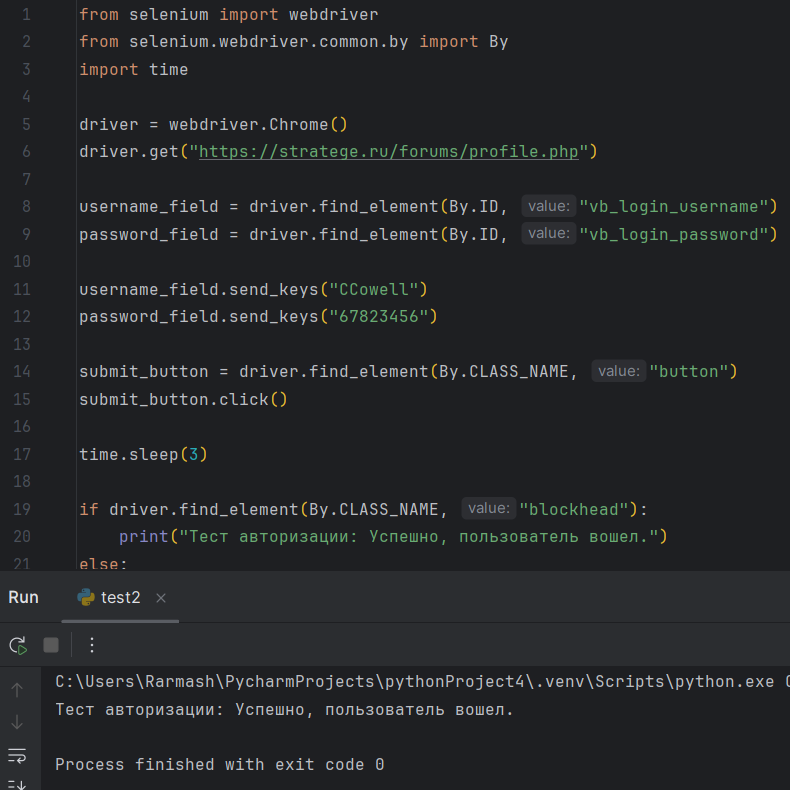


Рисунок 2 – Результат 2 теста

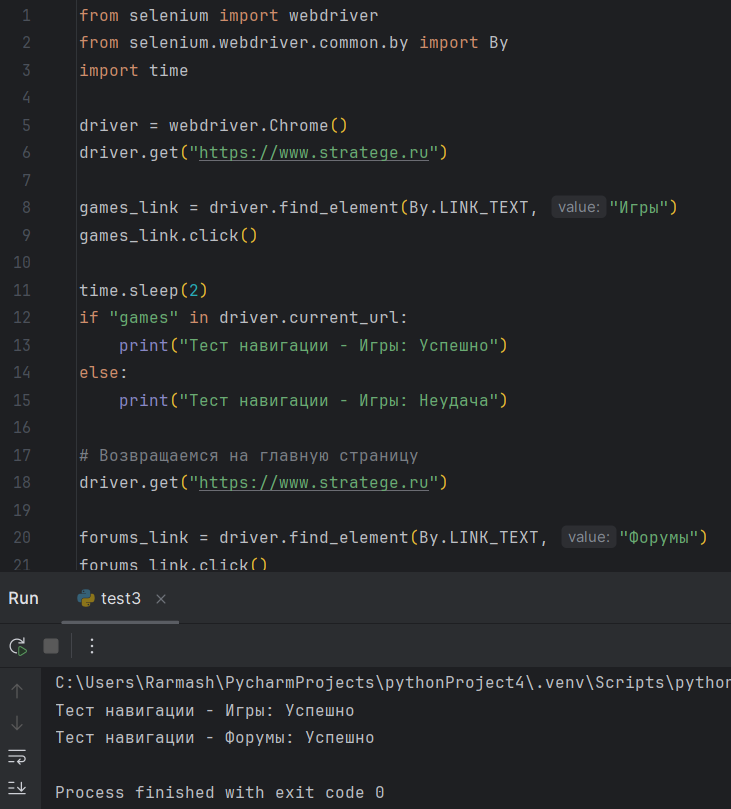


Рисунок 3 – Результат 3 теста

*Листинг 1 – Исходный код теста 1*

|  |
| --- |
| from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.keys import Keys import time  driver = webdriver.Chrome() driver.get("https://www.stratege.ru")  search\_box = driver.find\_element(By.ID, "gb\_ids\_site\_search\_input") search\_box.send\_keys("Forza Horizon 3") search\_box.send\_keys(Keys.RETURN)  time.sleep(3)  results = driver.find\_element(By.CLASS\_NAME, "ss\_search\_bx\_list\_title") if results:  print("Тест поиска: Успешно, результаты найдены.") else:  print("Тест поиска: Неудача, результаты не найдены.")  driver.quit() |

*Листинг 2 – Исходный код теста 2*

|  |
| --- |
| from selenium import webdriver from selenium.webdriver.common.by import By import time  driver = webdriver.Chrome() driver.get("https://stratege.ru/forums/profile.php")  username\_field = driver.find\_element(By.ID, "vb\_login\_username") password\_field = driver.find\_element(By.ID, "vb\_login\_password")  username\_field.send\_keys("CCowell") password\_field.send\_keys("67823456")  submit\_button = driver.find\_element(By.CLASS\_NAME, "button") submit\_button.click()  time.sleep(3)  if driver.find\_element(By.CLASS\_NAME, "blockhead"):  print("Тест авторизации: Успешно, пользователь вошел.") else:  print("Тест авторизации: Неудача, пользователь не вошел.")  driver.quit() |

*Листинг 3 – Исходный код теста 3*

|  |
| --- |
| from selenium import webdriver from selenium.webdriver.common.by import By import time  driver = webdriver.Chrome() driver.get("https://www.stratege.ru")  games\_link = driver.find\_element(By.LINK\_TEXT, "Игры") games\_link.click()  time.sleep(2) if "games" in driver.current\_url:  print("Тест навигации - Игры: Успешно") else:  print("Тест навигации - Игры: Неудача")  # Возвращаемся на главную страницу driver.get("https://www.stratege.ru")  forums\_link = driver.find\_element(By.LINK\_TEXT, "Форумы") forums\_link.click()  time.sleep(2) if "forums" in driver.current\_url:  print("Тест навигации - Форумы: Успешно") else:  print("Тест навигации - Форумы: Неудача")  driver.quit() |

1. **Вывод**

В ходе данной работы мы изучили основы автоматизированного тестирования веб-приложений с использованием Selenium WebDriver, а также применили подходы для создания функциональных тестов. Мы разработали тестовые сценарии для различных элементов веб-сайта, написали тесты для проверки корректности авторизации, загрузки файлов и управления элементами страницы. В процессе выполнения работы были выявлены и исправлены ошибки, связанные с путями к файлам, что позволило повысить стабильность тестов. Результатом работы стало улучшение качества тестируемого продукта и повышение надежности тестовых сценариев.