

ДИСЦИПЛИНА

«Автоматизация тестирования (Java, Python)»

Задание

В рамках финала необходимо дополнить и усовершенствовать тестовый фреймворк, разработанный на этапе полуфинала.

Ваша задача — не писать всё с нуля, а расширить существующую архитектуру, внедрив новые возможности и улучшив качество автоматизации.

Вы можете реализовать задание на Java или Python — выбор остаётся за вами.

Для Java: требуется использовать Java 11+, Selenium, систему сборки (Maven или Gradle) и фреймворк тестирования (JUnit 5 или TestNG).

Для Python: используйте Python 3.12+, Selenium, Pytest

Поддерживаемые браузеры: Chrome или Firefox (с актуальными версиями драйверов).

Оба варианта оцениваются одинаково. Главное — корректная работа тестов, чистота кода и возможность запуска без дополнительных сложностей.

Что требуется сделать

1. Зайти на страницу с регистрационной формой https://way2automation.com/way2auto_jquery/registration.php#load_box
2. Проанализировать формы и составить тесты для страницы
3. Реализовать автотесты
4. В поле About Yourself ввести решения для задач

Решения должны быть вычислены программно в коде (не подставлены вручную),

Ответы вводятся в одно поле — через запятую, без пробелов

Используйте Selenium для поиска поля и ввода строки с ответами.

Уровень 1:

Вывести все возможные перестановки массива без повторяющихся элементов, массив [2025, 3, 11]

Уровень 2:

Найти длину самой длинной строго возрастающей подпоследовательности в массиве [1, 3, 6, 7, 9, 4, 10, 5, 6, 12, 2, 7, 11]

Уровень 3:

Найти самую длинную подпоследовательность, которая встречается в двух строках в одинаковом порядке, но не обязательно подряд

s1 = "aabbdsaacc"

s2 = "abadc"

Задачи в п4. являются опциональными. Вы можете решить одну, две или все три.

Чем больше задач вы решите правильно, тем выше будет итоговая оценка.

Даже одно корректно вычисленное и автоматически введенное решение принесёт вам дополнительные баллы.

Требования к реализации:

1. Тестовые сценарии должны быть описаны в текстовом файле (README);
2. Для обеспечения читаемости, переиспользования и поддержки кода требуется применить структурированный подход к автоматизации (рекомендуется паттерн Page Object Model, так же допускаются другие подходы)
3. Использовать один из тестовых фреймворков (Java — TestNG, JUnit 4/5, Python — PyTest)
4. Использовать один из сборщиков (для Java) — Maven, Gradle.

Дополнительные требования:

1. Настроена сборка отчетов Allure:
Отчёт должен:
 - Чётко отражать, какие тесты были запущены и с каким результатом
 - Содержать осмысленные названия тестов и шагов
 - Включать в себя скриншоты при ошибках
2. Реализовано параллельное выполнение тестов для сокращения общего времени прогона тестового набора

Отправка задания:

1. Разместите Ваш проект на одной из платформ Github, Bitbucket, GitVerse
2. Отправьте ссылку на проект через личный кабинет для проверки