Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Керівник роботи

\_\_\_\_\_\_\_\_ Євгеній Вовк

“\_13\_” \_\_\_\_грудня\_\_\_\_ 2022 р.

“Веб-сервіс автоматизованої перевірки програм лабораторного практикуму на мові програмування Java”

**Програма та методика тестування**

КПІ.ІП-0110.045440.04.51

“ПОГОДЖЕНО”

Керівник роботи:

\_\_\_\_\_\_\_\_\_\_\_\_ Євгеній ВОВК

|  |  |
| --- | --- |
| Консультант: | Виконавець: |
| \_\_\_\_\_\_\_\_\_\_\_ Максим ГОЛОВЧЕНКО | \_\_\_\_\_\_\_\_\_\_ Богдан ЗАРАНІК |

Київ – 2022

зміст

[1 ОБ’ЄКТ ВИПРОБУВАНЬ 3](#_Toc118330135)

[2 МЕТА ТЕСТУВАННЯ 4](#_Toc118330136)

[3 МЕТОДИ ТЕСТУВАННЯ 5](#_Toc118330137)

[4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ 6](#_Toc118330138)

# ОБ’ЄКТ ВИПРОБУВАНЬ

Об’єктом випробування є система онлайн-перевірки рішень на мові Java. Основною функцією програмного забезпечення є створення задачі і перевірка рішення на коректність та відповідність певному набору метрик.

Головною метою роботи було розробити веб-застосунок систему онлайн-перевірки рішень на мові програмування Java, метою якого буде допомога користувачеві перевірити правильність виконання завдання у автоматичному режимі.

Для початку, користувач повинен зареєструватися та увійти систему. Клієнт має можливість перейти на персональну сторінку, де може змінити пароль, переглянути або видалити завдання(для викладача). Щоб створити завдання викладач має натиснути на кнопку “Створити завдання” на панелі управління, потім заповнити форму створення завдання, завантажити вихідні дані та натиснути “Створити”. Далі завдання з’явиться у списку завдань і будь-хто може його виконати. На сторінці завдання будь-який авторизований користувач може завантажити шаблон завдання, прочитати пояснення, подати код на перевірку у вигляді zip-архіву коду рішення та переглянути статистику виконання даного завдання всіма учасниками. Також викладач має змогу видалили власні задачі та змінювати їх. Також можуть бути встановлені обмеження на кількість  спроб відправки рішень від одного учасника по задачі. Головним завданням адміністратора є підтверження та відмова у наданні прав викладача учасникам, які відправили такий запит.

# МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

* перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
* перевірка збереження даних;
* знаходження проблем, помилок і недоліків з метою їх усунення;
* перевірка зручності графічного інтерфейсу;
* перевірка коректності поведінки при некоректних вхідних даних;
* перевірка стійкості до виконання користувачем шкідливих дій;

# МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

* статичне тестування: перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;
* динамічне тестування: застосовується в процесі виконання програми. Коректність програмного засобу перевіряється на певній кількості тестів. При прогоні кожного з них збираються та аналізуються дані про проблеми та помилки в роботі програми;
* функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікуваній;
* системне тестування – перевіряється усе програмне забезпечення в цілому;
* мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення;
* тестування «чорної скриньки»: об’єктом тестування тут є функції присутні у програмі. Перевіряється коректність вихідних даних при заданих вхідних;
* тестування «білої скриньки»: об’єктом тестування тут є внутрішня поведінка програми. Перевіряється коректність побудови всіх елементів програми та правильність їхньої взаємодії один з одним;
* тестування «сірої скриньки»: об’єктом тестування тут є деякі особливості внутрішньої поведінки програми. Перевіряється коректність вихідних даних при заданих вхідних, застосовується для тестування окремих алгоритмів (функцій).

# ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально з використанням наскрізного тестування (End-to-end, E2E, Chain testing) та написанням unit-тестів, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування. Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

* динамічне тестування на відповідність функціональним вимогам;
* тестування на виведення повідомлень про помилку, коли це необхідно;
* тестування виведення помилок програми у випадку відсутності з’єднання до мережі;
* тестування інтерфейсу користувача;
* тестування зручності використання;
* тестування коректності поведінки при введенні неочікуваних даних;
* тестування коректності поведінки при намаганні користувача нашкодити системі;
* тестування коректності поведінки системи при “зависанні” рішення користувача;