

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

(повне найменування вищого навчального закладу)

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

(повне найменування інституту, назва факультету (відділення))

КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

(освітньо-кваліфікаційний рівень)

на тему: «Розроблення програмних засобів для модульного тестування

із застосуванням Java-технологій»

Виконав: студент 4 курсу, групи 4ПР1

напряму підготовки (спеціальності)

121 - «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки, спеціальності)

Голубєв Борислав Костянтинович

(прізвище та ініціали)

Керівник к.т.н., доцент Ляшенко О. М.

(прізвище та ініціали)

Рецензент к.т.н., доцент Григорова А. А.

(прізвище та ініціали)

Хмельницький - 2024

Херсонський національний технічний університет

(повне найменування вищого навчального закладу)

Інститут, факультет, відділення Факультет інформаційних технологій та дизайну

Кафедра Програмних засобів і технологій

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки _____

(шифр і назва)

Спеціальність 121 – Інженерія програмного забезпечення

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗіТ

к.т.н. доцент Огнева О.Є.

“ ” _____ 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Голубєва Борислава Костянтинівича

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розроблення програмних засобів для модульного тестування із застосуванням Java-технологій»

керівник роботи к.т.н., доцент Ляшенко О.М.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу №217-с «09» лютого 2024 р.

2. Строк подання студентом проекту (роботи) 05.06.2024 р.

3. Вихідні дані до проекту (роботи) Постановка завдання, ДСТУ, літературні та періодичні джерела, матеріали практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної області та постановка завдання дипломного проектування;

2. Проектування програмного продукту;

3. Програмна реалізація програмного продукту.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Пояснювальна записка;

2. Прикладна програма;

3. Модульні тести (прикладні застосування);

4. Презентація доповіді.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз літературних джерел.	1 тиждень	Виконано
2	Аналіз стану завдання на сучасному етапі розвитку ПЗ. Розроблення першого розділу пояснювальної записки.	2-4 тижні	Виконано
3	Розроблення тестів. Розроблення другого розділу пояснювальної записки.	5-7 тижні	Виконано
4	Розроблення архітектури програмної системи.	8-9 тижні	Виконано
5	Побудова основних класів, методів та об'єктів.	10 тиждень	Виконано
6	Розроблення параметризованих тестів.	11-13 тижні	Виконано
7	Налагодження програмного продукту. Розроблення третього розділу пояснювальної записки.	14-15 тижні	Виконано
8	Розроблення презентації та захист кваліфікаційної роботи бакалавра.	16 тиждень	Виконано

Студент _____ Голубєв Б.К.
 (підпис) (прізвище та ініціали)

Керівник роботи _____ Ляшенко О.М.
 (підпис) (прізвище та ініціали)

Анотація

Кваліфікаційну роботу бакалавра присвячено розробленню програмних засобів для модульного тестування із застосуванням Java-технологій. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків.

В першому розділі проведено аналіз видів тестування програмного забезпечення, проведено аналіз модульного тестування мовою Java, описано класи та методи, що використовуються JUnit фреймворком, описано анотації JUnit, розглянуто основні концепції JUnit Platform та JUnit Jupiter.

В другому розділі описано процес створення проекту в Visual Studio Code, описано процес налаштування системи тестування JUnit у VS Code, розроблено тести для перевірки метода Fft на лінійній послідовності, розроблено тести для перевірки модуля Fft на наявність функціональної залежності максимальної частоти, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Дірака, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Хевисайда.

В третьому розділі описано порядок виконання модульних тестів, описано реалізації класів MethodOrderer та ClassOrderer, описано методи створення параметризованих тестів, описано методи роботи з параметризованими методами тестування, описано методи перетворення аргументів, описано пріоритети визначення ім'я для параметризованого тесту.

Abstract

The qualification work of the bachelor is devoted to the development of software for modular testing using Java technologies. The qualification work consists of an introduction, three sections, conclusions..

The first section analyzes the types of software testing, analyzes unit testing in Java, describes the classes and methods used by the JUnit framework, describes the annotations of JUnit, discusses the main concepts of JUnit Platform and JUnit Jupiter.

The second section describes the process of creating a project in Visual Studio Code, describes the process of configuring the JUnit test system in VS Code, tests have been developed to test the Fft method on a linear sequence, tests are developed to check the Fft module for the presence of a functional dependence of the maximum frequency, tests are developed to test the Fft module with Dft on the Dirac function, tests were developed to check the Fft module using Dft for the Heaviside function..

The third section describes the procedure for performing unit tests, describes the implementations of the MethodOrderer and ClassOrderer classes, describes the methods for creating parameterized tests, describes the methods for working with parameterized test methods, describes the methods for converting arguments, describes the priorities of determining the name for a parameterized test.

РЕФЕРАТ

Пояснювальна записка до випускної роботи: 80 с., 33 рис., 3 табл., 1 додаток, 31 джерело.

Об'єкт проєктування – процес модульного тестування програмного забезпечення.

Предмет проєктування – технології розроблення модульних тестів.

Мета проєктування – розроблення програмних засобів для модульного тестування із застосуванням Java-технологій.

Методи проєктування – фреймворк JUnit, середовище розроблення Visual Studio Code, мова Java

В межах кваліфікаційної роботи розроблено тести для перевірки метода Fft на лінійній послідовності, розроблено тести для перевірки модуля Fft на наявність функціональної залежності максимальної частоти, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Дірака, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Хевисайда.

Модульне тестування дозволило перевірити на коректність окремі модулі вихідного коду програми, набори з одного або більше програмних модулів разом із відповідними керуючими даними, процедурами використання та обробки.

Ключові слова: МОДУЛЬНЕ ТЕСТУВАННЯ, ТЕХНОЛОГІЇ JAVA, ФРЕЙМВОРК JUNIT, VISUAL STUDIO CODE, MOBA JAVA.

ЗМІСТ

ВСТУП	9
1.ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1. Аналіз видів тестування програмного забезпечення	10
1.2. Аналіз модульного тестування мовою Java. Використання JUnit фреймворку	17
1.3. Огляд тестових класів та методів JUnit фреймворку	24
1.4. Висновки до розділу 1	27
2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	28
2.1. Створення проекту в Visual Studio Code	28
2.2. Процес тестування модуля	33
2.3. Висновок до розділу 2	51
3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ПРОДУКТУ	52
3.1. Порядок виконання модульних тестів	52
3.2. Параметризовані тести	56
3.3. Перетворення аргументів	59
3.4. Висновки до розділу 3	63
ВИСНОВОК	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
Додаток А	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПЗ	Програмне забезпечення
ОС	Операційна система
JUnit	JUnit - фреймворк для модульного тестування програмного забезпечення мовою Java.
VS Code	Visual Studio Code – редактор вихідного коду, розроблений Microsoft для Windows, Linux та MacOS.
Unit testing	Модульне тестування
JUnit	Фреймворк для автоматичного юніт-тестування затосунків
ISTQB	International Software Testing Qualifications Board
FFT	Швидке перетворення Фур'є - алгоритм прискореного обчислення дискретного перетворення Фур'є, що дозволяє отримати результат за час, менший за $O(N^2)$.

ВСТУП

Метою кваліфікаційної роботи є розроблення програмних засобів для модульного тестування із застосуванням Java-технологій

В першому розділі проведено аналіз видів тестування програмного забезпечення, проведено аналіз модульного тестування мовою Java, описано класи та методи, що використовуються JUnit фреймворком, описано анотації JUnit, розглянуто основні концепції JUnit Platform та JUnit Jupiter.

В другому розділі описано процес створення проекту в Visual Studio Code, описано процес налаштування системи тестування JUnit у VS Code, розроблено тести для перевірки метода Fft на лінійній послідовності, розроблено тести для перевірки модуля Fft на наявність функціональної залежності максимальної частоти, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Дірака, розроблено тести для перевірки модуля Fft за допомогою Dft на функцію Хевисайда.

В третьому розділі описано порядок виконання модульних тестів, описано реалізації класів MethodOrderer та ClassOrderer, описано методи створення параметризованих тестів. Параметризовані тести дозволяють запускати тест кілька разів з різними аргументами. Вони оголошуються так само, як звичайні методи @Test, але замість них використовується анотація @ParameterizedTest.

В третьому розділі також описано методи роботи з параметризованими методами тестування, описано методи перетворення аргументів. Встановлено, що JUnit підтримує примітивне перетворення Widening для аргументів, що надаються до @ParameterizedTest. Описано пріоритети визначення ім'я для параметризованого тесту. Встановлено, що ім'я для параметризованого тесту визначається відповідно до таких правил пріоритету: атрибут name в @ParameterizedTest, якщо він присутній; значення параметра конфігурації junit. params.displayname.default, якщо він присутній; константа DEFAULT_DISPLAY_NAME, визначена в @ParameterizedTest.