

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
(повне найменування вищого навчального закладу)
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
(повне найменування інституту, назва факультету (відділення))
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

(освітній рівень)

на тему: «Розробка програмних засобів метаевристичної підтримки

автоматизованого тестування програм»

Виконав: студент 4 курсу, групи 4ПР1
спеціальності

121 - «Інженерія програмного забезпечення»

(шифр і назва спеціальності)

Юшко Владислав Олексійович

(прізвище та ініціали)

Керівник д.т.н., проф. Шерстюк В.Г.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Херсон - 2021

Херсонський національний технічний університет

(повне найменування вищого навчального закладу)

Факультет, відділення Інформаційних технологій та дизайну
Кафедра Програмних засобів і технологій
Освітній рівень бакалавр
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри
Програмних засобів та технологій
д.т.н. проф. В.Г. Шерстюк
“ ____ ” _____ 2021 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Юшко Владиславу Олексійовичу

(прізвище, ім'я, по батькові)

Тема роботи «Розробка програмних засобів метаевристичної підтримки
автоматизованого тестування програм»

керівник роботи д.т.н. професор Шерстюк В.Г.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ____ . 2021 р. № -

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи літературні та періодичні джерела,
матеріали переддипломної практики

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

1) проектні рішення з розробки системи з підтримки процесу верифікації
програмного продукту

2) моделі та методи побудови планів тестування

3) опис програмного та технічного забезпечення

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

1. Схема структурна контекстної моделі системи. 2. Діаграма діяльності
побудови планів тестування. 3. Діаграма послідовності роботи
розробленого фреймворку. 4. Схема структурна класів програмного
забезпечення. 5. Блок-схема алгоритму генерації популяції. 6. Екранні
форми. 7. Результати експериментів.

6. Консультанти розділів роботи

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання роботи	Термін виконання етапів роботи	Примітки
1.	Отримання завдання	08.02.2021	Виконано
2.	Підбір літератури	09.02.2021-21.02.2021	Виконано
3.	Аналіз предметної області	22.02.2021-28.02.2021	Виконано
4.	Розробка та обґрунтування завдання	01.03.2021-14.03.2021	Виконано
5.	Розробка концептуальної моделі	15.03.2021-21.03.2021	Виконано
6.	Розробка алгоритму	22.03.2021-04.04.2021	Виконано
7.	Проектування програної системи	05.04.2021-11.04.2021	Виконано
8.	Розробка інтерфейсу програмної системи	12.04.2021-18.03.2021	Виконано
9.	Тестування програмної системи	19.04.2021-25.04.2021	Виконано
10.	Оформлення пояснювальної записки	26.04.2021-09.05.2021	Виконано
11.	Захист кваліфікаційної роботи	15.06.2021	Виконано

Студент _____ Юшко В.О.

(підпис)

(прізвище та ініціали)

Керівник роботи _____ Щерстюк В.Г.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 91 сторінка, 25 рисунків, 5 таблиць, 1 додаток, 37 джерел.

Мета роботи – підвищення якості та надійності програмного продукту за рахунок автоматизації процесу тестування, що дозволяє скоротити час та витрати на виконання процесів тестування.

Об'єкт дослідження – процес тестування програмних продуктів.

Предмет дослідження – методи автоматизації верифікації програмних продуктів на основі побудови планів тестування програмних продуктів.

Методи дослідження – генетичний алгоритм, жадібний алгоритм, метаевристичний алгоритм, алгоритм мурашиних колоній.

Результати роботи:

- проведено аналіз існуючих методів автоматизації тестування програмного забезпечення;
- розроблено метаевристичний алгоритм покриття множини для вирішення задачі побудови плану тестування;
- розроблено метод побудови плану тестування в залежності від обраної стратегії;
- виконано програмну реалізацію розробленого алгоритму з інтеграцією в TeamCity.

Новизна роботи: полягає у впровадженні системи для підтримки автоматизованого тестування, яка потребує мінімального втручання людини для знаходження помилок та невідповідностей у роботі програмних продуктів.

Ключові слова: *автоматизація, тестування програмних продуктів, побудова планів тестування, покриття множини, метаевристичні алгоритми, генетичний алгоритм, алгоритм мурашиних колоній.*

АНОТАЦІЯ

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновку, списку використаних джерел та додатку.

У роботі досліджено процес тестування програмних продуктів за допомогою побудови планів тестування на основі діаграми станів та переходів.

У першому розділі розглянуто основні види тестування, описано структуру бізнес процесів у вигляді контекстної діаграми та наведено схеми структурні варіантів використання та діаграми послідовності різних процесів.

У другому розділі наведено змістовну та математичну постановку задачі функції побудови планів тестування, яка зводиться до класичної задачі мінімального покриття множини. Проведено огляд існуючих методів розв'язання задачі та на основі цього виділені критерії оцінки покриттів. В залежності від розмірності задачі та наявних обчислювальних ресурсів для розв'язання задачі запропоновані та розроблені метаевристичні та жадібний алгоритми. Проведені обчислювальні експерименти для визначення параметрів алгоритмів, за якими алгоритм видає кращі результати.

У третьому розділі наведено основні засоби розробки системи з підтримки процесу верифікації програмного продукту. Представлено архітектуру програмного забезпечення за допомогою діаграм компонентів, класів та послідовності. Наведено інструкцію користувача для системи з підтримки процесу верифікації програмного продукту. Результати виконання показали, що впровадження такої системи у процес розробки значно зекономить часові та фінансові ресурси.

ABSTRACT

The bachelor's thesis consists of an introduction, three sections, a conclusion, a list of used sources and an appendix.

The paper investigates the process of software testing by building test plans based on the state and transition diagram.

The first section discusses the main types of testing, describes the structure of business processes in the form of a context diagram and provides diagrams of structural use cases and sequence diagrams of different processes.

The second section presents a meaningful and mathematical formulation of the problem of the function of constructing test plans, which is reduced to the classical problem of the minimum coverage of the set. An overview of the existing methods of solving the problem is carried out and on this basis the criteria for evaluation of coatings are highlighted. Depending on the dimension of the problem and the available computing resources to solve the problem, metaheuristic and greedy algorithms are proposed and developed. Computational experiments were performed to determine the parameters of the algorithms by which the algorithm gives the best results.

The third section presents the main tools for developing a system to support the verification process of the software product. The software architecture is presented using component, class and sequence diagrams. The user manual for the system to support the software product verification process is given. The results showed that the introduction of such a system in the development process will significantly save time and financial resources.

ЗМІСТ

РЕФЕРАТ	Ошибка! Закладка не определена.
ABSTRACT	О
шибка! Закладка не определена.	
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
1. ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ З ПІДТРИМКИ ПРОЦЕСУ ВЕРИФІКАЦІЇ ПРОГРАМНОГО ПРОДУКТУ	13
1.1. Загальні положення	13
1.2. Опис бізнес-процесів	21
1.2.1. Структура бізнес-процесів	21
1.2.2. Актори і функції	22
1.2.3. Опис процесу діяльності	23
1.3. Опис постановки задачі	24
1.4. Рішення з інформаційного забезпечення	27
1.4.1. Вхідні дані	27
1.4.2. Вихідні дані	28
Висновки до розділу	28
2. МОДЕЛІ ТА МЕТОДИ ПОБУДОВИ ПЛАНІВ ТЕСТУВАННЯ	29
2.1. Змістовна постановка задачі	29
2.2. Математична постановка задачі	29
2.3. Огляд методів розв'язання задачі	31
2.3.1. Властивості задачі про покриття	32
2.3.2. Точні методи розв'язання задачі про покриття	33
2.3.3. Наближені методи розв'язання задачі про покриття	35
2.3.4. Евристичні методи розв'язання задачі про покриття	37
2.3.5. Критерії оцінки покриття	37
2.4. Розробка методів розв'язання задачі	38
2.4.1. Оптимізація системи рівнянь	39
2.4.2. Отримання повної множини покриттів методом бектрекінгу	39
2.4.3. Розв'язання задачі із застосуванням жадібного алгоритму	41

2.4.4. Розв’язання задачі із застосуванням алгоритму мурашиних колоній	41
2.5. Результати досліджень ефективності методу	54
2.5.1. Визначення параметрів генетичного алгоритму.....	55
2.5.2. Визначення параметрів АСО	56
Висновки до розділу	61
3. ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ.....	62
3.1. Засоби розробки	62
3.2. Вимоги до технічного забезпечення	64
3.2.1. Вимоги до серверу	64
3.2.2. Вимоги до клієнта	64
3.3. Архітектура програмного забезпечення	64
3.3.1. Діаграма компонентів.....	64
3.3.2. Діаграма послідовності.....	65
3.3.3. Діаграма діяльності.....	66
3.3.4. Діаграма класів.....	66
3.4. Інструкція користувача	70
Висновки до розділу	75
ВИСНОВКИ.....	76
ПЕРЕЛІК ПОСИЛАНЬ.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПП – програмний продукт

ПЗ – програмне забезпечення

ГА – генетичний алгоритм

АСО – мурашиний алгоритм

ТС – TeamCity

ТЗ – технічне завдання

ПК – персональний комп'ютер

ІТ – інформаційні технології

ВСТУП

Тестування – це процес дослідження технології, який виконується на вимогу замовників, та призначений для розкриття інформації про якість продукту відносно середовища його використання. Цей процес передбачає виконання програми з метою знайдення невідповідностей та помилок [1]. Методи тестування ПЗ, що існують на сьогоднішній день, не дозволяють у повній мірі виявити всі невідповідності та встановити правильність функціонування аналізованої програми, всі існуючі методи тестування діють в рамках формального процесу перевірки досліджуваного ПЗ, що розробляється. Такий процес формальної перевірки може довести, що з погляду використовуваного методу дефекти відсутні. Іншими словами, враховуючи людські фактори, що існують на кожному етапі життєвого циклу програмного забезпечення, неможливо точно визначити або гарантувати відсутність дефектів програмного продукту. Існує багато підходів до рішення завдання тестування і перевірки програмного забезпечення, але ефективне тестування складних ПЗ – це дослідницький процес, такий, що не зводиться до проходження строгим і чітким процедурам або створенню таких.

За стандартом ISO 9126, якість програмних засобів визначається як сукупність характеристик досліджуваного ПЗ з урахуванням таких складових, як надійність, ефективність, практичність, супровід, функціональність, мобільність.

Перші програмні системи були розроблені в рамках програм наукових досліджень або програм, які задовольняють потреби міністерств оборони. Тестування таких програмних продуктів проводилося строго формалізовано із записом всіх тестових даних, тестових процедур, отриманих результатів. Тестування виділялося як окремий процес, який починався після завершення кодування, та, як правило, виконувалося тим же персоналом [2].

Вже у 1980-х роках тестування було розширено, включаючи такі поняття, як попередження дефектів. Проектування тестів – це найбільш ефективний з усіх відомих методів попередження помилок. В той же час стали висловлюватися думки, що необхідна методологія тестування, зокрема, що тестування повинне включати перевірки протягом циклу розробки, та це повинен бути контрольований процес. В ході тестування треба перевірити не тільки розроблену та зібрану програму, але й вимоги, архітектуру, код, самі тести. «Традиційне» тестування, яке існувало до початку 1980-х років, відносилось тільки до скомпільованої, готової системи (наразі це зазвичай називається системне тестування), але надалі тестери стали залучатися до всіх аспектів життєвого циклу розробки. Це дозволяло раніше знаходити проблеми у вимогах і архітектурі і тим самим скорочувати терміни і бюджет розробки. В середині 1980-х років з'явилися перші інструменти для автоматизованого тестування. Передбачалося, що комп'ютер зможе проводити більше тестів, ніж людина, і зробить це більш надійнішим. Спочатку ці інструменти були надзвичайно простими, без можливості написання сценаріїв на скриптових мовах.

На початку 1990-х років в поняття «тестування» почали включати проектування, планування, створення, підтримку і виконання тестів, та це означало перехід від звичайного тестування до забезпечення якості, яке охоплює важливий цикл розробки ПЗ. В той час починають з'являтися різні програмні засоби та інструменти для підтримки процесу тестування - просунуті середовища для автоматизації з можливістю створення скриптів і генерації звітів, системи управління тестами, ПЗ для проведення тестування навантаження [2]. В середині 1990-х років з розвитком Інтернету та розробкою великої кількості web-додатків популярність стало отримувати «гнучке тестування» (по аналогії з гнучкими методологіями програмування).

У 2000-х роках з'явилося ще ширше визначення терміну тестування, в нього було додано поняття «Оптимізація бізнес-технологій», що направляє розвиток ІТ відповідно до цілей бізнесу. Основний метод полягає в оцінці та максимізації важливості кожного етапу життєвого циклу розробки ПЗ для досягнення необхідного рівня доступності, якості, продуктивності.

Сучасна наука забезпечує багато способів вирішення завдань тестування та верифікації ПЗ, проте ефективне тестування нових та складних програмних продуктів – це завжди креативний процес, який не обмежується чіткими та одноманітними процедурами. Він вимагає від тестувальника розгляду кожного випадку в індивідуальному порядку.

Для розробки ПЗ витрачається достатньо велика кількість коштів, адже кожен замовник хоче отримати якісний та надійний результат, покриття певної кількості функціональності, тому впровадження тестування як однієї з основних частин розробки ПЗ є особливо важливою та існує саме для забезпечення якості та ефективності програмного продукту, підтримки робочого стану системи. Тому завдання забезпечення якості розроблюємого ПЗ є одним із найактуальніших в індустрії ІТ.